

Lecture Notes
in Geoinformation and Cartography

LNG&C

Thomas H. Kolbe
Andreas Donaubaue
Christof Beil *Editors*

Recent Advances in 3D Geoinformation Science

Proceedings of the 18th 3D GeoInfo
Conference

 Springer

Lecture Notes in Geoinformation and Cartography

Series Editors

William Cartwright, Department of Land Information, RMIT University,
Melbourne, VIC, Australia

Georg Gartner, Department of Geodesy and Geoinformation, Vienna
University of Technology, Vienna, Austria

Liqiu Meng, Lehrstuhl für Kartographie, TU München, München, Bayern,
Germany

Michael P. Peterson, Department of Geography and Geology,
University of Nebraska at Omaha, Omaha, NE, USA

The Lecture Notes in Geoinformation and Cartography series provides a contemporary view of current research and development in Geoinformation and Cartography, including GIS and Geographic Information Science. Publications with associated electronic media examine areas of development and current technology. Editors from multiple continents, in association with national and international organizations and societies bring together the most comprehensive forum for Geoinformation and Cartography.

The scope of Lecture Notes in Geoinformation and Cartography spans the range of interdisciplinary topics in a variety of research and application fields. The type of material published traditionally includes:

- proceedings that are peer-reviewed and published in association with a conference;
- post-proceedings consisting of thoroughly revised final papers; and
- research monographs that may be based on individual research projects.

The Lecture Notes in Geoinformation and Cartography series also includes various other publications, including:

- tutorials or collections of lectures for advanced courses;
- contemporary surveys that offer an objective summary of a current topic of interest; and
- emerging areas of research directed at a broad community of practitioners.

Thomas H. Kolbe · Andreas Donaubauer ·
Christof Beil
Editors

Recent Advances in 3D Geoinformation Science

Proceedings of the 18th 3D GeoInfo
Conference

 Springer

Editors

Thomas H. Kolbe
Chair of Geoinformatics
Department of Aerospace and Geodesy
Technical University of Munich
Munich, Germany

Andreas Donaubaue
Chair of Geoinformatics
Department of Aerospace and Geodesy
Technical University of Munich
Munich, Germany

Christof Beil
Chair of Geoinformatics
Department of Aerospace and Geodesy
Technical University of Munich
Munich, Germany

ISSN 1863-2246

ISSN 1863-2351 (electronic)

Lecture Notes in Geoinformation and Cartography

ISBN 978-3-031-43698-7

ISBN 978-3-031-43699-4 (eBook)

<https://doi.org/10.1007/978-3-031-43699-4>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

Organization

Conference Chair

Thomas H. Kolbe, Technical University of Munich, Germany

Scientific Committee

Alias Abdul-Rahman, Universiti Teknologi Malaysia (UTM), Malaysia
Mulhim Al-Doori, Roads and Transport Authority, Dubai, UAE
Giorgio Agugiario, Delft University of Technology, Netherlands
Filip Biljecki, National University of Singapore, Singapore
Roland Billen, University of Liege, Belgium
Jörg Blankenbach, RWTH Aachen, Germany
Lars Bodum, Aalborg University, Denmark
Pawel Boguslawski, Wroclaw University of Environmental and Life Sciences, Poland
Martin Breunig, Karlsruhe Institute of Technology, Germany
Eliseo Clementini, University of L'Aquila, Italy
Volker Coors, HFT Stuttgart, Germany
Youness Dehbi, HafenCity University Hamburg, Germany
Lucia Díaz, Vilariño Universidade de Vigo, Spain
Efi Dimopoulou, National Technical University of Athens, Greece
Andreas Donaubauer, Technical University of Munich, Germany
Claire Ellul, University College London, UK
Gilles Falquet, University of Geneva, Switzerland
Gilles Gesquière, Université Lumière Lyon 2, France
Lars Harrie, Lund University, Sweden
Ihab Hijazi, Technical University of Munich, Germany
Mike Horhammer, Oracle, USA
Ümit Işıkdağ, Mimar Sinan Fine Arts University, Turkey

Martin Kada, Technical University of Berlin, Germany
Mohsen Kalantari, The University of New South Wales, Australia
Mila Koeva, University of Twente, Netherlands
Thomas H. Kolbe, Technical University of Munich, Germany
Hugo Ledoux, Delft University of Technology, Netherlands
Jiyeong Lee, University of Seoul, South Korea
Ki-Joune Li, Pusan National University, South Korea
Francesca Noardo, Open Geospatial Consortium, USA
Sander Oude Elberink, University of Twente, Netherlands
Jacynthe Pouliot, Université Laval, Canada
Jantien Stoter, Delft University of Technology, Netherlands
Rudi Stouffs, National University of Singapore, Singapore
Peter van Oosterom, Delft University of Technology, Netherlands
Qing Zhu, Southwest Jiaotong University, China
Sisi Zlatanova, University of New South Wales, Sydney, Australia

Local Organizing Committee

Andreas Donaubaueer, Technical University of Munich, Germany
Christof Beil, Technical University of Munich, Germany
Roland Dietrich, Technical University of Munich, Germany
Tanja Nyc, Technical University of Munich, Germany
Medhini Heeramaglore, Technical University of Munich, Germany
Ihab Hijazi, Technical University of Munich, Germany
Kasimir Forth, Technical University of Munich, Germany
Gabriele Aumann, Technical University of Munich, Germany

Preface

In order to discuss the latest advances in 3D and 4D geoinformation and to bring together researchers, practitioners, and professionals from academia, industry, and government, the 18th International 3D GeoInfo Conference was held in Munich, Germany, from 12 to 14 September 2023. The conference was organized by the Chair of Geoinformatics at the Technical University of Munich and chaired by Thomas H. Kolbe.

The 3D GeoInfo Conference has been hosted annually since 2006 on four continents and attracts international experts in fields such as 3D/4D data acquisition, processing, management and modelling, urban digital twins, GIS-BIM integration, GIS, AR/VR-visualization, AI and related topics.

Contained within this book are carefully selected papers showing the highest quality research that was presented at the conference. In total, 51 papers were accepted for inclusion in the Springer book based on double-blind reviews by members of an international scientific committee. Twenty-three papers were accepted based on the reviews of full paper submissions and another 28 papers on extended abstract review. All accepted papers have been revised by the authors to reflect the comments and suggestions for improvement of the reviewers.

Organizing and successfully conducting such an event would not be possible without the collaborative effort of many people. We thank the organizing committee chaired by Andreas Donaubaauer (in alphabetical order): Gabriele Aumann, Christof Beil, Roland Dietrich, Kasimir Forth, Medhini Heeramaglore, Ihab Hijazi and Tanja Nyc.

Additionally, we thank the following institutions for their support of the conference: Leonhard Obermeyer Center (LOC), Deutsche Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation (DGPF) and Runder Tisch GIS e.V.

Furthermore, we thank the members of the scientific committee for their valuable work, including the review of all scientific contributions. We are grateful to the sponsors of the conference whose contributions allowed us to keep conference fees at a reasonable level.

Lastly, we thank the esteemed speakers and all colleagues who contributed by submitting their research papers, with special appreciation to those who actively participated in the conference by presenting their work.

Munich, Germany
September 2023

Thomas H. Kolbe
Andreas Donaubaueer
Christof Beil

Contents

| | |
|---|-----|
| Applications of 3D City Models and Digital Twins | |
| Recommendation for Vegetation Information in Semantic 3D City Models Used in Urban Planning Applications | 3 |
| Karolina Pantazatou, Jouri Kanters, Kristoffer Mattisson, Per-Ola Olsson, and Lars Harrie | |
| Shadowing Calculation on Urban Areas from Semantic 3D City Models | 31 |
| Longxiang Xu, Camilo León-Sánchez, Giorgio Agugiaro, and Jantien Stoter | |
| Supporting Teleoperated Humanitarian Aid Missions with 3D Visualization Using Remote Sensing Data | 49 |
| Lucas Dominik Angermann, Magdalena Felicitas Halbgewachs, and Konstanze Lechner | |
| Solid Waste in the Virtual World: A Digital Twinning Approach for Waste Collection Planning | 61 |
| Iván Cárdenas, Mila Koeva, Calayde Davey, and Pirouz Nourian | |
| Automatically Evaluating the Service Quality of Bicycle Paths Based on Semantic 3D City Models | 75 |
| Christof Beil, Mario Ilic, Andreas Keler, and Thomas H. Kolbe | |
| Virtual and Augmented Reality and Visualization | |
| Virtual Reality Experience Analysis from Point Cloud Data | 95 |
| Diego Aneiros-Egido, Jesús Balado, Ha Tran, and Lucía Díaz-Vilariño | |
| Visualisation of 3D Uncertainties for Subsurface Infrastructure Using Augmented Reality | 111 |
| Simon Quaade Vinther, Frida Dalbjerg Kunnerup, Lars Bodum, Lasse Hedegaard Hansen, and Simon Wyke | |

Immersive Virtual Reality to Verify the As-built State of Electric Line Networks in Buildings 129
Julius Knechtel, Weilian Li, Yannick Orgeig, Jan-Henrik Haurert, and Youness Dehbi

3D Data Mapping with Augmented Reality 145
Ming-Chun Lee

Creating a 3D Multi-Dataset Bubble in Support of OGC Testbed-19 and Metaverse Standards Prototypes 155
James Clarke, Steve Smyth, Rob Smith, and Jeremy Morley

GIS/BIM Integration

Assessment of the LoD Specification for the Integration of BIM-Derived Building Models in 3D City Models 171
Jasper van der Vaart, Jantien Stoter, Abdoulaye Diakité, Filip Biljecki, Ken Arroyo Ohori, and Amir Hakim

IFC Georeferencing for OSM 193
Helga Tauscher, Dominik Heigener, Subhashini Krishnakumar, Thomas Graichen, Rebecca Schmidt, and Julia Richter

Merging BIM, Land Use and 2D Cadastral Maps into a Digital Twin Fit—For—Purpose Geospatial Infrastructure 211
Dimitra Andritsou, Sofia Soile, and Chryssy Potsiou

Artificial Intelligence for the Automated Creation of Multi-scale Digital Twins of the Built World—AI4TWINNING 233
André Borrmann, Manoj Biswanath, Alex Braun, Zhaiyu Chen, Daniel Cremers, Medhini Heeramaglore, Ludwig Hoegner, Mansour Mehranfar, Thomas H. Kolbe, Frank Petzold, Alejandro Rueda, Sergei Solonets, and Xiao Xiang Zhu

Development of a Geo to BIM Converter: CityJSON Importer Plugin for Autodesk Revit 249
Amir Hakim, Jasper van der Vaart, Ken Arroyo Ohori, and Jantien Stoter

3D Point Cloud Processing and Analysis

Efficient In-Memory Point Cloud Query Processing 267
Balthasar Teuscher, Oliver Geißendörfer, Xuanshu Luo, Hao Li, Katharina Anders, Christoph Holst, and Martin Werner

Transferring Façade Labels Between Point Clouds with Semantic Octrees While Considering Change Detection 287
Sophia Schwarz, Tanja Pilz, Olaf Wysocki, Ludwig Hoegner, and Uwe Stilla

Investigating Data Fusion from Three Different Point Cloud Datasets Using Iterative Closest Point (ICP) Registration 299
 Wahyu Marta Mutiarasari, Muhammad Zulkarnain Abd Rahman, and Alias Abdul Rahman

Sensing Heathland Vegetation Structure from Unmanned Aircraft System Laser Scanner: Comparing Sensors and Flying Heights 309
 Nina Homainejad, Lukas Winiwarter, Markus Hollaus, Sisi Zlatanova, and Norbert Pfeifer

Comparison of Cloud-to-Cloud Distance Calculation Methods - Is the Most Complex Always the Most Suitable? 329
 Vitali Diaz, Peter van Oosterom, Martijn Meijers, Edward Verbree, Nauman Ahmed, and Thijs van Lankveld

Deriving 3D Models from Point Clouds

Reconstructing Façade Details Using MLS Point Clouds and Bag-of-Words Approach 337
 Thomas Froech, Olaf Wysocki, Ludwig Hoegner, and Uwe Stilla

Generating 3D Roof Models from ALS Point Clouds Using Roof Line Topologies 357
 Gefei Kong and Hongchao Fan

MLS2LoD3: Refining Low LoDs Building Models with MLS Point Clouds to Reconstruct Semantic LoD3 Building Models 367
 Olaf Wysocki, Ludwig Hoegner, and Uwe Stilla

Semantic Segmentation of Buildings Using Multisource ALS Data 381
 Agata Walicka and Norbert Pfeifer

Classifying Point Clouds at the Facade-Level Using Geometric Features and Deep Learning Networks 391
 Yue Tan, Olaf Wysocki, Ludwig Hoegner, and Uwe Stilla

3D Data Acquisition, Analysis and Simulation for Urban Digital Twins

Enriched Semantic 3D Point Clouds: An Alternative to 3D City Models for Digital Twin for Cities? 407
 Imane Jeddoub, Zouhair Ballouch, Rafika Hajji, and Roland Billen

Unsupervised Roofline Extraction from True Orthophotos for LoD2 Building Model Reconstruction 425
 Weixiao Gao, Ravi Peters, and Jantien Stoter

Enhancing Realism in Urban Simulations: A Mapping Framework for the German National Standard XPlanung and CityGML 437
 Hamza Zahid, Ihab Hijazi, Andreas Donaubaauer, and Thomas H. Kolbe

An Alternative Raw Data Acquisition Approach for Reconstruction of LOD3 Models 459
 Florian Frank, Ludwig Hoegner, Peter Buckel, and Kris Dalm

Identification and Interpretation of Change Patterns in Semantic 3D City Models 479
 Son H. Nguyen and Thomas H. Kolbe

3D Data Modelling and Topology

A Level of As-Is Detail Concept for Digital Twins of Roads—A Case Study 499
 David Crampen, Marcel Hein, and Jörg Blankenbach

Digital geoTwin: A CityGML-Based Data Model for the Virtual Replica of the City of Vienna 517
 Hubert Lehner, Sara Lena Kordasch, Charlotte Glatz, and Giorgio Agugiaro

A Hierarchy of Levels of Detail for 3D Utility Network Models 543
 Zihan Chen, Jacynthe Pouliot, and Frédéric Hubert

Topological Representation of a 4D Cell Complex and Its Dual-Feasibility Study 563
 Pawel Boguslawski

3D Topology Rules Implementation in Spatial Database 573
 Syahiirah Salleh, Uznir Ujang, and Suhaibah Azri

Indoor/Outdoor Modelling and Navigation

RGB-D Semantic Segmentation for Indoor Modeling Using Deep Learning: A Review 587
 Ishraq Rached, Rafika Hajji, and Tania Landes

A Framework for Generating IndoorGML Data from Omnidirectional Images 605
 Misun Kim, Jeongwon Lee, and Jiyeong Lee

Deep Adaptive Network for WiFi-Based Indoor Localization 617
 Afnan Ahmad and Gunho Sohn

MoLi-PoseGAN: Model-based Indoor Relocalization Using GAN and Deep Pose Regression from Synthetic LiDAR Scans 633
 Hang Zhao, Martin Tomko, and Kourosh Khoshelham

Digital Twins: Simulating Robot-Human Sidewalk Interactions 645
 Ali Hassan, Muhammad Usman, Melissa Kremer, Seungho Yang, Michael Luubert, Petros Faloutsos, G. Brent Hall, and Gunho Sohn

Sensors and Dynamic Data in Urban Digital Twins

Dynamic Digital Twins: Challenges, Perspectives and Practical Implementation from a City’s Perspective 671

Rico Richter, Frank Knospe, Matthias Trapp, and Jürgen Döllner

Humans As Sensors in Urban Digital Twins 693

Binyu Lei, Yunlei Su, and Filip Biljecki

Requirements for Web-Based 4D Visualisation of Integrated 3D City Models and Sensor Data in Urban Digital Twins 707

Joseph Gitahi and Thomas H. Kolbe

Investigation of CityGML 3.0 for Modelling Temporal Aspects in Underground Land Administration 727

Bahram Saeidian, Abbas Rajabifard, Behnam Atazadeh, and Mohsen Kalantari

Integrating Dynamic Data with 3D City Models via CityJSON Extension 745

Khawla Boumhidi, Gilles-Antoine Nys, and Rafika Hajji

Software and Tools for Spatial Data

OGC Data Exchange Toolkit: Interoperable and Reusable 3D Data at the End of the OGC Rainbow 761

Francesca Noardo, Rob Atkinson, Ingo Simonis, Alejandro Villar, and Piotr Zaborowski

cjdb: A Simple, Fast, and Lean Database Solution for the CityGML Data Model 781

Leon Powalka, Chris Poon, Yitong Xia, Siebren Meines, Lan Yan, Yudian Cai, Gina Stavropoulou, Balázs Dukai, and Hugo Ledoux

Introducing the 3DCityDB-Tools Plug-In for QGIS 797

Giorgio Agugiaro, Konstantinos Pantelios, Camilo León-Sánchez, Zhihang Yao, and Claus Nagel

Challenges and Steps Toward Implementing 3D Cadastral Database—Physical Data Model of LADM 823

Javad Shahidinejad, Mohsen Kalantari, and Abbas Rajabifard

Optim3D: Efficient and Scalable Generation of Large-Scale 3D Building Models 835

Anass Yarroudh, Abderrazzaq Kharroubi, and Roland Billen

An Overview of the 3D GeoInfo Conference Series

The 3D GeoInfo Journey from 2006 to 2022: A Reflection 853

Alias Abdul Rahman

Applications of 3D City Models and Digital Twins

Recommendation for Vegetation Information in Semantic 3D City Models Used in Urban Planning Applications



Karolina Pantazatou , Jouri Kanters , Kristoffer Mattisson ,
Per-Ola Olsson , and Lars Harrie 

Abstract Cities are growing in size and becoming increasingly dense. This situation calls for strategic planning of green infrastructure in the urban planning process. Safeguarding the green infrastructure is important for maintaining urban ecosystem services and increasing the well-being of urban populations. To facilitate appropriate urban planning that enables cities to grow sustainably, it is important that the geospatial community provides adequate vegetation information. In this study, we investigate the need for vegetation information in urban planning applications such as modelling ecosystem services and noise, as well as performing case studies of using vegetation information in daylight and solar energy simulations. Based on these investigations, we formulate a recommendation of how vegetation information should be included in 3D city models. The study is focused on the development of a Swedish national profile of CityGML, but many of the conclusions are general and universally applicable. In short, the recommendations are that: (1) the vegetation theme should follow CityGML 3.0 with some additional attributes (e.g., popular name of tree species) added as an application domain extension, (2) no LOD division is required for the vegetation information stored (but rather derived if necessary), (3) the vegetation theme should only contain 3D vegetation objects while the 2D vegetation is part of the land cover theme, and (4) the building specification (and city furniture specification) must include the possibility to store information on whether building roofs or facades (and walls) are covered with vegetation.

Keywords 3D city models · Vegetation · Land cover · Solar energy simulations · Daylight simulations · Sustainable urban planning

This article was selected based on the results of a double-blind review of the full paper.

K. Pantazatou (✉) · P.-O. Olsson · L. Harrie
Department of Physical Geography and Ecosystem Science, Lund University, Lund, Sweden
e-mail: karolina.pantazatou@nateko.lu.se

J. Kanters
Division of Energy and Building Design, Department of Building and Environmental Technology,
Lund University, Lund, Sweden

K. Mattisson
Division of Occupational and Environmental Medicine, Lund University, Lund, Sweden

1 Introduction

1.1 Background

Cities are expanding and becoming denser. Urban densifications may significantly impact access to direct or scattered sunlight, which may result in increased usage of electric light and limited potential of installing photovoltaic solar panels on building roofs and facades. Additionally, urban densifications often lead to less green areas which, in turn, can affect public health (World Health Organization 2023) and cause environmental problems such as increased exposure to noise (Skovbro 2002). This situation calls for strategic planning of the remaining green infrastructure in the urban planning process. To facilitate appropriate urban planning and enable cities to grow sustainably, it is important that the geospatial community provides adequate vegetation information, e.g., as part of a 3D city model.

1.2 Aim

The general aim of this paper is to discuss the need for storing vegetation information in 3D city models from an urban planning perspective. The focus is set on examining if and how vegetation should be included in the new Swedish specifications for 3D city models. The following research questions are addressed:

- What urban vegetation information is collected today in Sweden for urban planning applications? What is the main use of this vegetation data?
- What possibilities does CityGML provide for storing vegetation data? And what studies have been conducted to extend the CityGML specifications for vegetation?
- Which urban planning applications do require vegetation information? Which type of vegetation information is needed for these applications?
- How much does vegetation information affect daylight and solar energy simulations?

In this study, as well as in the abovementioned specifications, we do not consider vegetation information required for e.g.: management of the green infrastructure (such as operational systems for park areas), tree inventories, and more advanced ecosystem modelling (e.g., carbon uptake and evapotranspiration modelling).

2 Vegetation Data in 3D City Models

Most cities in Europe have 3D city models. Earlier these models were primarily presentation models, but 3D city models facilitating analyses and simulations are becoming increasingly more common (Gröger and Plümer 2012; Biljecki et al. 2018).

The most common open standard for 3D city models is CityGML (Kolbe et al. 2021). CityGML 3.0 has ten thematic modules, including the *vegetation* module.

One central part of CityGML is the concept of *level of detail* (LOD) which supports multi-scale modelling. The LOD concept has mainly been studied for the *building* theme (Biljecki et al. 2016; Löwner et al. 2016), but the same concept is also valid for other themes. In CityGML 3.0, the LOD concept only refers to the geometric resolution of objects, and not the thematic (meaning that all attributes and code lists could be the same for all LODs). Furthermore, CityGML supports *Application Domain Extensions* (ADEs) to cover potential application-specific needs by introducing user-defined objects and attributes (Biljecki et al. 2018). ADEs in the context of vegetation information are discussed in e.g., Guerrero Iñiguez (Guerrero Iñiguez 2017). Another feature of CityGML is that it allows objects of similar shape (trees, traffic lights, etc.) to be represented as prototypic objects (implicit geometries) which are instantiated multiple times at different locations. The CityGML specifications do not specify in detail what information should be stored in the datasets. Therefore, they must be complemented with measuring guidelines that provide detailed rules.

2.1 Vegetation Theme in CityGML

The vegetation module in CityGML 3.0 includes two object types: *SolitaryVegetationObject* (SVO) and *PlantCover* (PC), where the latter can be divided based on geometric representation to *MultiSolid* objects (PC-Solid) and *MultiSurface* objects (PC-Surface) (see Fig. 1). SVOs are appropriate for representing objects like trees and may have the attributes *class*, *function*, *usage*, *species*, *height*, *trunkDiameter*, *crownDiameter*, *rootBallDiameter* and *maxRootBallDepth* (Kolbe et al. 2021). PC-Solid is appropriate for vegetation areas characterized by high vegetation (e.g., forests) and PC-Surface for low vegetation (e.g., grassland, shrubland, etc.); they have the attributes *class*, *function*, *usage*, *averageHeight*, *minHeight* and *maxHeight*.

There are, to the authors' knowledge, no defined LODs for the vegetation theme in CityGML 3.0. In the CityGML 2.0 specifications (Gröger et al. 2012), a table is included that describes a proposal of LOD definitions for several themes (based on (Albert et al. 2003)). In short, it is stated that SVO and PC objects of a certain size should be included in a specific LOD and that the more detailed LOD could use prototypical objects. There is even the possibility to use individual object forms in the highest LOD.

One proposal of a CityGML LOD definition for vegetation is provided by Ortega-Córdova (Ortega-Córdova 2018). Based on a study of use cases she proposes 14 LOD definitions (see Fig. 2). Also, Zhang et al. (2022) propose several LOD definitions (Fig. 2). Common for both these studies is that they start with the general definition of LOD (LOD0 is for regional scale information, etc.), and how the LOD concept is formalized for the building theme and then extrapolate this information for defining the vegetation LOD. Both models also contain detailed lists of attributes required for each LOD.

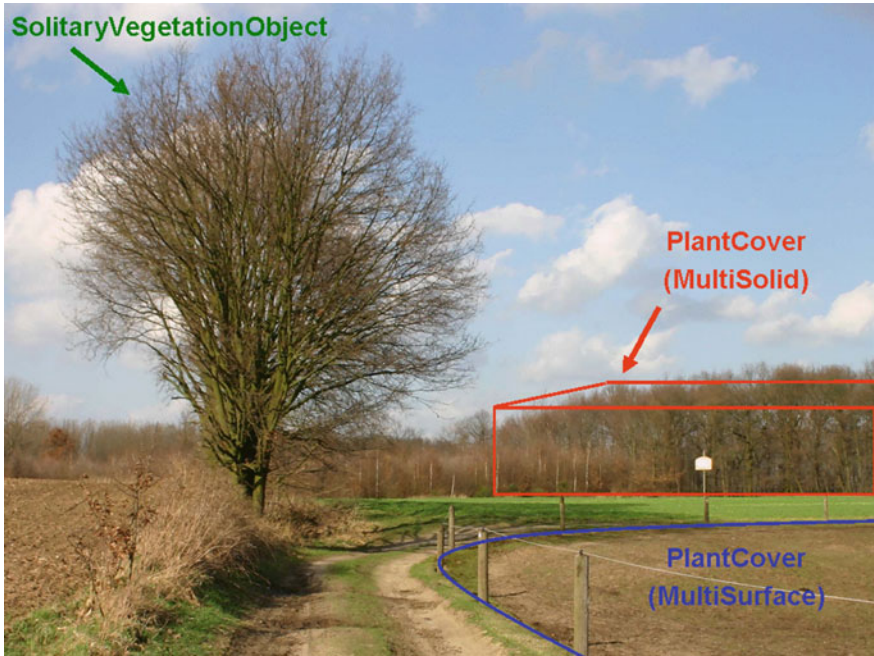


Fig. 1 Example of CityGML vegetation objects for the classes *SolitaryVegetationObject* and *PlantCover*. (Source p. 85 in Gröger et al. (2012))

| | LODx.A | | | | LODx.B | | | | LODx.C | | | | LODx.D | | | | | | | |
|--|---|--|---|--|---|--|--|--|---|--|---|--|---|--|---|--|--|--|---|--|
| | LOD0.x | | LOD1.x | | LOD2.x | | LOD3.x | | LOD4.x | | LOD0.x | | LOD1.x | | LOD2.x | | LOD3.x | | LOD4.x | |
| | 2D plane | | 2.5D extruded solid | | Simple 3D solid | | Complicated 3D solid | | 3D structured components | | 2D plane | | 2.5D extruded solid | | Simple 3D solid | | Complicated 3D solid | | 3D structured components | |
| | 2D plane presenting the position and size | | 2.5D extruded solid presenting height/ root depth of buildings and vegetation | | Simple 3D solid, with coarse morphology | | Complicated 3D solid presenting morphological structure of the model | | 3D structured components and semantic components presenting model details | | 2D plane presenting the position and size | | 2.5D extruded solid presenting height/ root depth of buildings and vegetation | | Simple 3D solid, with coarse morphology | | Complicated 3D solid presenting morphological structure of the model | | 3D structured components and semantic components presenting model details | |

Fig. 2 Vegetation LOD definitions by Ortega-Córdoba [p. 29 in Ortega-Córdoba (2018)] (CC-BY license; some images are based on ESRI library models) and by Zhang et al. (p. 204 in Zhang et al. (2022))

2.2 Vegetation Data in a Swedish Context

Currently there is no harmonized vegetation information in the Swedish municipalities. Most municipalities have some vegetation information in their 2D base maps,

but not necessarily with any attribute information connected to e.g., trees. Sometimes, there is more detailed vegetation information for trees that are growing on municipality owned land, but more seldom for privately owned land.

In the older national specifications (SGP), trees were included in the city furniture theme. However, the SGP specifications have not been used much in practice and were replaced by a new set of specifications denoted *National Specifications* (NS). Currently, the NS do not include any specification for vegetation, but hopefully the work for creating such specifications could commence soon. The cities of Stockholm, Gothenburg, and Malmö have, together with Lund University, created a Swedish profile of CityGML 2.0 (Uggla et al. 2023). This profile, which is implemented as a CityGML ADE, adds some attributes to the SVO and PC classes (see Fig. 3).

One important aspect of 3CIM is that it is a semantically thin ADE, including few attributes. The basic idea is to create linkages between 3CIM and operational systems on object level to obtain up-to-date information on demand [see (Uggla et al. 2023) for the motivation behind this]. The aforementioned linkages are implemented using the CityGML *External References* class. E.g., a 3CIM SVO includes attributes for *height*, *trunk diameter*, and *crown diameter* (in accordance with CityGML). It also supports the use of implicit geometries (i.e., tree prototypes) in the representation of SVOs. Only LODs 1–2 are allowed. Unlike CityGML, in 3CIM the *Function* attribute is utilized to decide the LOD in a hierarchical way. For instance, the *Function* attribute value for a solitary coniferous tree in a park is 10500 in LOD1 (where the 1st digit “1” denotes a park and 2nd–3rd digits “05” denote a tree), and 10,501 in LOD2 (where the last two digits “01” denote a coniferous tree). The popular species name of a 3CIM SVO is stored in the *speciesPopularName* attribute of the ADE while the

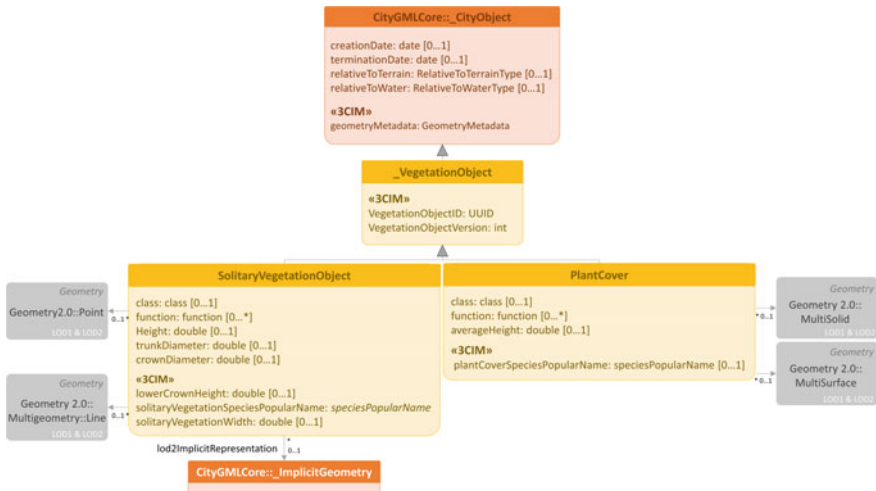


Fig. 3 The vegetation theme in 3CIM. The terms marked 3CIM are part of the ADE. Adapted from “Future Swedish 3D City Models—Specifications, Test Data, and Evaluation.”, by Uggla et al. (2023), ISPRS International Journal of Geo-Information, 12, p. 47

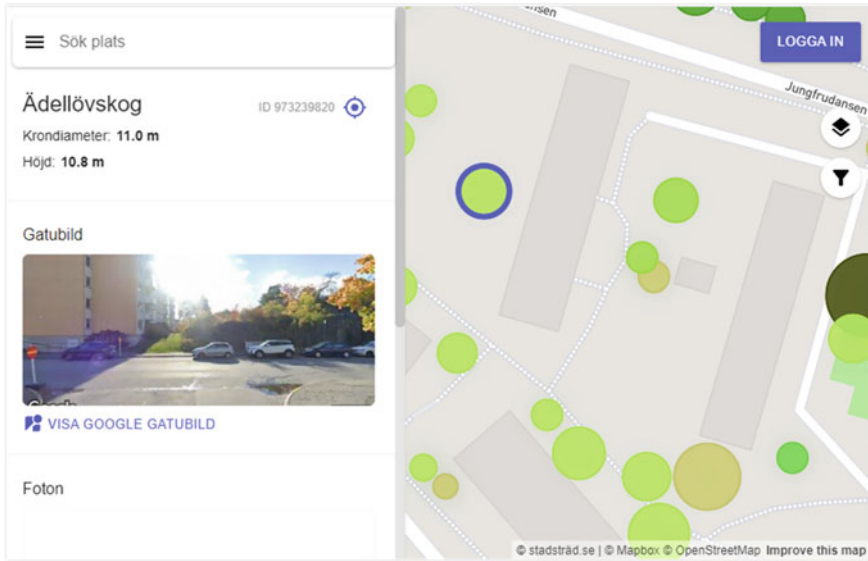


Fig. 4 A map service showing type of tree, crown width, height of (group of) trees including map view and street view images (Stadsträd 2023)

Latin name is retrieved from an external database that is linked to 3CIM based on the *vegetationObjectID*.

Several municipalities are connected to a service for urban trees (Stadsträd 2023) (see Fig. 4). This service collects tree information from laser-scanning and provides an application software for adding trees along with attribute information manually. The software can be used both by professionals and volunteers.

3 Urban Vegetation in Analyses, Simulations, and Visualisations

This section describes some of the most important applications utilising vegetation information in an urban planning context.

3.1 *Ecosystem Services*

One important role for trees in urban settings is to provide urban ecosystem services (UES). On a regional level, the UES can be evaluated based on e.g., CORINE Land Cover data and remote sensing data (Maes et al. 2020). There is also a need for UES indicators on a city level. A study initiated by the European Commission

(Commission and Directorate-General for Environment, Mapping and assessment of ecosystems and their services 2016) included ecosystem system inventories in ten European cities. The UES indicators used were e.g., connected to urban forest patterns (canopy, coverage, fragmentation, etc.), tree health and damage (e.g., foliage damage and crown dieback; measurements based on visual inspection of trees), and connectivity of urban green infrastructure. Such indicators have also frequently been used in research. One example is provided by Kourdounouli and Jönsson (Kourdounouli and Jönsson 2020), who created an index based on eight UES indicators, none of which required detailed data down to the level of individual trees or group of trees as traditional land cover and land use data was deemed sufficient.

Two tools have recently been developed in Sweden for incorporating ecosystem services in the urban planning: QGYF (C/O City 2019) and ESTER (Boverket: ESTER 2023). The tools consider the vegetation (and blue infrastructure) contribution to biodiversity, noise reduction, microclimate improvement, recreation, and other UES. In the context of this study, the tools only require 2D data on vegetation along with information about vegetation on building surfaces (walls, roofs, and facades).

3.2 *Human Well-Being*

Many studies have been conducted examining the positive effects of vegetation and green areas on urban populations; it is for example important to be able to view the landscape (Velarde et al. 2007; Ko et al. 2022), and vegetation plays an important role in decreasing heat stress and wind (Wang and Akbari 2016; Duncan et al. 2019). Much of this research has led to the development of guidelines for urban planners. One such guideline, which is frequently applied in Sweden [e.g., (Skåne 2023)], is the 3-30-300 rule. This rule sets the thresholds of “*having at least 3 well-established trees in view from every home, school, and place of work; within a neighbourhood with no less than a 30% tree canopy coverage; and no more than 300 m distance to the nearest public green space from every residence*” (Konijnendijk 2023).

3.3 *Noise*

The potential effect of vegetation on noise propagation has for many years been considered as limited, even so, there is increasing evidence of that there can be a significant effect (Peng et al. 2014). Vegetation can have a reducing effect on noise levels in three main ways (Renterghem et al. 2012). Firstly, through reflection and scattering by stems, branches, and leaves. Secondly, through sound absorption via mechanical vibrations and conversion to heat. Thirdly, through the acoustical ground effect, which is attributed to the fact that vegetation contributes to acoustically very soft ground (the vegetation’s root system makes the soil more porous). Previous

studies investigating the effect of trees on noise propagation near roads, have shown varying results (Renterghem et al. 2012). Regarding tree belts, important aspects identified in these previous studies include leaf thickness, visibility through the vegetation, height of trees, belt width, stem diameter, and spacing between trees (Renterghem et al. 2012). E.g., in an Australian study, vegetation close to the road was found most efficient in damping noise while an attenuation of noise through 10–20 m wide tree belts with approximately 0.5 m spacing could reduce the noise up to 2–3 dB (Peng et al. 2014). In urban settings, low vegetated surfaces have been shown to reduce noise levels with 5 dB, while the same barrier without vegetation only reduced noise with 1 dB (Hornikx 2016). Another recent study showed that covering facades and roofs with vegetation could have a significant impact on reducing noise levels, especially in closed inner yards (Forssen et al. 2022).

In sound propagation models, like the Nordic model Nord2000 (SP: Nord 2000), it is possible to consider the effect of vegetation. The characteristics of the ground surface have an impact on how noise disperses and is in Nord2000 assessed by eight acoustic impedance classes (Gustafsson et al. 2023). In sound simulations, geographical data on land cover can be classified into these acoustic classes so that the effect of vegetation on ground absorption of sound is represented. In addition, it is possible to consider the screening and scattering effect of the stems and the tree trunks (Tarrero et al. 2008).

3.4 Daylight and Solar Energy

As vegetation is not considered a permanent obstruction, it is frequently overlooked when executing daylight and sunlight simulations. This is attributed to the seasonal cycle of deciduous trees and to the fact that trees can be cut. The national Swedish legislation and international recommendations regulating access to sunlight/daylight make therefore no reference to vegetation. In practice however, the presence of tall trees can have a significant impact on the daylight access of rooms located near the ground floor as well as on the efficiency of photovoltaic solar panels installed on buildings facades or on the roofs of lower buildings. Indicatory, Sadeghi and Mistrick (2021) compared empirical daylight measurements with their corresponding daylight simulation results and found that ignoring exterior surrounding objects (e.g., trees) could yield on average 67.9% erroneous estimations while including high-detailed vegetation information produced an average error of 11.3%. As computing systems are getting more powerful, daylight/sunlight metrics are also getting more complex. An increasing amount of them have started to consider the influence of vegetation. As these changes are beginning to occur, it is important to ensure access to good quality vegetation information.

Limited access to daylight may lead to increased energy demands for lighting (Lechner 2014) and have negative consequences on people's health (Osterhaus 2005; Lee et al. 2022), productivity in workplace as well as academic achievements (Vischer 2008). Though this is known, few studies have been conducted examining the impact

of vegetation (Villalba et al. 2014). Yet, we are on the verge of change, and increasingly more advanced daylight metrics are being created, some of which also consider vegetation. Vegetation has also been studied as a possible solution for reducing glare (Coder 1996). Pan and Du (2021) used i.a., the *Sky View Factor* and *daytime average horizontal illuminance* to study outdoor daylight conditions in conjunction with various urban objects (including vegetation). A recent technical report, by Littlefair et al. (2022), advises architects and urban planners to compute hours of direct sunlight during a single day (March 21st) for new dwellings that are obstructed by trees one time by considering trees as solid objects and one without trees and then check for compliance with the European daylight recommendation as a way of compensating for the worst-case scenario. The same report also provides indicatory transmittance values per tree type category for the calculation of the *Daylight Factor* (one of the most widely implemented daylight metrics).

While Littlefair et al. (2022) describe the simulation procedure and transmittance values, the representation of the vegetation geometry is not well-defined. Villalba et al. (2014) study trees as sun control elements in oasis cities and highlight problems related to the representation of how sunlight is reflected and transmitted through trees. Balakrishnan and Jakubiec (2022) propose a solution to this issue in the form of a novel, open-source, low-cost method that utilizes photography and image processing to measure the on-site transmittance of a tree crown with the purpose of producing a tree prototype that interacts more realistically with light.

From a solar energy perspective, the efficiency of photovoltaic panels, affected by the amount of incident solar irradiance on building roofs and facades, can be greatly affected by tall trees that grow close to buildings (Berry et al. 2013). This is especially important to consider in the urban planning of countries located at higher latitudes (Konarska et al. 2014). For this purpose, Pan and Jakubiec (Pan et al. 2022) created realistic models of 12 tree species most commonly growing in Canada and studied the impact of dynamically changing leaf foliage in daylight and solar energy simulations. They found that differing tree canopy dimensions, gap fractions, leaf drop, and regrowth schedules can influence simulation output. Another recent study by Matusiak and Arbab (2022) examined the effect of increasing vegetation in inner yards of buildings in Oslo and found that only a slight increase of vegetation led to an average decrease of 11% of solar irradiation in the yard and up to 13% on lower parts of the building façades. There is currently no standard for performing simulations examining the energy potential of active solar energy production, but future investors in these systems would like to have the highest amount of detail in the simulations to make more well-informed decisions.

3.5 *Underground Applications*

The underground surface is becoming increasingly more important in urban planning. What is most interesting, from a vegetation information perspective, is mainly

the effect of roots on the utility networks, buildings, and hydrological modelling (Guerrero Iñiguez 2017).

3.6 Visualizations

Vegetation is a central part of urban planning visualizations. These visualizations often use tree models already included in software packages (see Fig. 2, 5-left and 6), point clouds (see Fig. 5-right) or photographs (see Fig. 4).

4 Case Studies

Our paper includes three case studies focusing on examining if and to what degree vegetation influences daylight and sunlight simulations results. The first two use cases are dedicated to daylight simulation metrics. Metrics are commonly used in the urban planning process for determining whether a building complies to the daylight- and sunlight-regulating national legislation and international (e.g., European) recommendations. The final case study investigates the impact of vegetation on the potential of installing photovoltaic panels on building roofs and facades.

Daylight and sunlight simulations should be performed at an early stage in the planning of urban densifications. But what is the role of vegetation in these simulations? To answer this question, we execute four scenarios for every case study, using the following data as input to the simulations:

- existing buildings (EBNV)
- existing & planned buildings (PBNV)
- existing buildings with vegetation (EBV)
- existing & planned buildings with vegetation (PBV).

The four scenarios were created to highlight the impact of vegetation on daylight/sunlight simulations on existing buildings (important when testing the potential of



Fig. 5 (left) Visualization of 3CIM data using Unreal Engine and their built-in tree models [cf. (Uggla et al. 2023)]. (right) Visualization based on point clouds (Nebiker et al. 2010)



Fig. 6 Study area: New Bellevue neighbourhood, Malmö municipality, Southern Sweden. The grey buildings are existing buildings, and the orange buildings are planned

installing PVC on roofs/facades of existing buildings) as well as on a combination of existing and planned buildings (important in the planning of urban densifications to ensure that all planned buildings as well as the existing ones continue to fulfil the national and international daylight-regulating requirements). To examine the impact of vegetation on this question, we included one set of scenarios where the vegetation is absent (EBNV and PBNV) and one where it is present (EBV and PBV).

4.1 Study Area

All use cases are performed in the same study area which is located in the New Bellevue neighbourhood of Malmö municipality in southern Sweden, Europe (see Fig. 6). This is an originally sparsely built neighbourhood, that has been deemed suitable for urban densification. The neighbourhood is dominated by residential buildings but does also include offices, commercial buildings, and schools. The dimensions of the study area are approximately 450 m × 500 m.

4.2 Data

The 3D city model (including existing and planned buildings, vegetation, etc.) used in the case studies is a 3CIM test data set provided by the city of Malmö [for details,

see (Ugglå et al. 2023)]. The vegetation information in this data set originates from the city of Malmö, including 2D vegetation information on trees growing on public as well as on private land. The 3CIM CityGML data (buildings, vegetation) were converted to ESRI Multipatch Shapefiles using FME. Additional vegetation information (e.g., species, Latin name, popular name) was provided for trees located on municipality-owned land. An inventory of the height and crown width of every tree was created, based on Google Earth images. This inventory was linked with Malmö municipality's 2D point tree layer using ESRI ArcGIS Pro and inserted into an FME-script to produce CityGML SVOs. The FME-script was developed by con Terra GmbH and Virtual City Systems (2023), and included two SketchUp prototype models of deciduous and coniferous trees. It required the geographic coordinates, (latitude, longitude, and elevation), height, crown width, and type of every tree in the study area for adjusting the prototypes to those measures and producing a corresponding 3D representation of the tree.

In addition to the 3CIM data we also used 3D window information. This information was produced as part of a collaboration between researchers at the Norwegian University of Science and Technology and Lund University (Hartman 2021). Distortion-corrected photographs of building facades (right angle) along with the 3CIM buildings were inserted to Blender,¹ where the windows were created based on interactive modelling. This was the case as the size and complexity of the buildings did not allow the implementation of the software developed by Fan et al. (2021) that uses a convolutional neural network to produce CityGML LOD3 building models.

Finally, a DTM with 1 m spatial resolution was downloaded from the Swedish National Mapping Agency (Lantmäteriet).

Figure 7 shows the existing, along with the planned buildings, as well as some of the vegetation in the study area. The red lines are used to mark the building facades for which the simulations were executed.

4.3 Daylight Modelling Case Study

In Sweden, access to daylight is regulated by law. The regulations are checked late in the urban planning process and based on daylight metrics that are used to estimate the magnitude and quality of daylight entering a room: The *daylight factor* (DF) is expressed as the percentage ratio of scattered light that enters the room to the light received from an unobstructed surface under an overcast sky (see Fig. 8) (Dubois et al. 2019) and the *obstruction angle* (OA) is measured from the middle of the window (centroid) to the mean top point of the opposing obstruction (Fig. 8).

According to Swedish legislation, for a building to receive a building permit it is compulsory that at least one window exists in every room that is visited more than occasionally (Boverkett 2011). From this window, it should also be possible for the occupants to follow the course of the day and seasonal changes. These requirements

¹ <https://www.blender.org/>.

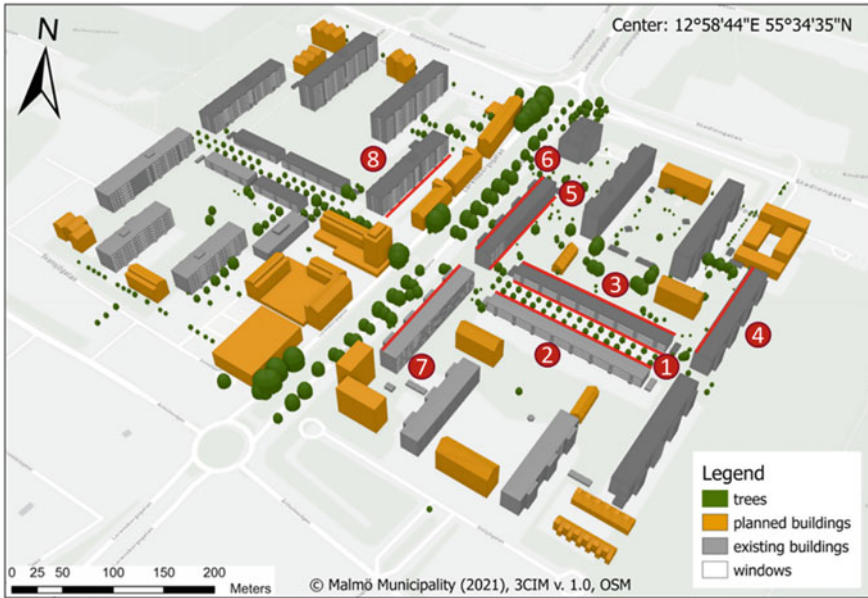


Fig. 7 Existing and planned buildings along with vegetation presented as part of the 3CIM 3D city model over the New Bellevue neighbourhood in Malmö, Southern Sweden. The facades for which the simulations are executed for are highlighted in red and numbered

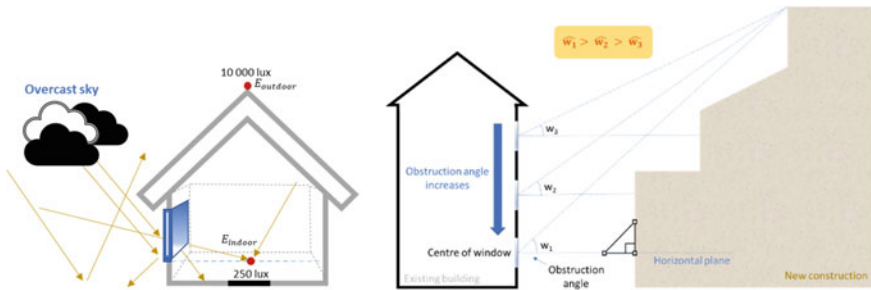


Fig. 8 Definition examples of the daylight factor (left) and the obstruction angle (right)

are considered fulfilled if the room reaches a $DF \geq 1\%$, which is commonly achieved if the $OA \leq 30^\circ$.

Additionally, the European Standard (EN 17037:2018 for Daylight in buildings) includes recommendations to achieve comfortable levels of indoor light by utilizing natural light (CEN 2018). One such example is the minimum requirement of 1.5-h *direct sunlight* in a frequently visited room during a day between February 1st and March 21st. The second daylight case study is dedicated to calculating the direct sunlight hours received by a window centroid on September 21st. The date is chosen to simulate the worst-case scenario, as the sun will be at the same position then as

on March 21st, with the difference that in September the trees will still have their leaves.

Methods. A tool was developed (in ArcPy²) for the purpose of executing the OA simulation of this case study. The tool requires 3 input layers (3D window layer, Digital Surface Model—DSM, and 2D building footprint layer) and returns the OA for every window. Note that the tool doesn't return the OA for the highest obstructing point, but the OA for the obstructing point that generates the highest OA. The DSM was created by adding the 3D building and vegetation information to the DTM. To achieve this the CityGML building, and vegetation layers were first converted to ESRI Multipatch Shapefiles in FME and then rasterized in ArcGIS Pro with the *Multipatch to Raster* conversion tool. Four different DSMs were created, one for every implementation scenario. The spatial resolution of the DSM was set to 0.2 m, to capture all details of the building exterior as well as the vegetation.

For the second simulation of this case study, we investigate compliance to the European recommendation of 1.5 direct sunlight hours on the 21st of September. The process involves the computation of *Direct Sun Hour* (DSH) values calculated for one day (0.5 min temporal resolution) over a grid of points (0.2 m spatial resolution) placed over every window of the selected facades (see Fig. 7). Finally, this implementation also takes into consideration how including or excluding more detailed objects of the urban fabric such as vegetation and planned buildings affects daylight access (i.e., total hours of direct sunlight per grid point on September 21st). The position of the Sun during that time of the year is estimated based on the latitude of our study area. A sun path is created over our 3D scene and direct sun rays are simulated. This simulation is performed using the CAD-based software environments Rhino3D³ and Grasshopper⁴ executing the Ladybug *DirectSunHours tool*. Ladybug is based on Radiance-engine, which has been proven to produce accurate results (Giannelli et al. 2022). To import the building and vegetation information in Rhino the corresponding Multipatch shapefiles were converted to OBJ files in FME. The OBJ file format was chosen as it preserves the coordinate system information (here: SWEREF99 13 30). This way the building and vegetation information was displayed in actual SWEREF99 13 30 coordinates in Rhino. The Grasshopper script used to execute the simulation is made available on GitHub.⁵

Results. Table 1 presents some cumulative statistics over the OA simulation output from the four different scenarios. The impact of vegetation is greater when comparing the EBNV and EBV scenarios rather than the ones implicating the planned buildings. This is, to an extent, attributed to the fact that the planned buildings require some of the vegetation to be cut down. Not only trees overlapping with the planned buildings are removed but also trees that are in closer proximity than 2 m from them.

The plots in Fig. 9 highlight how daylight is restricted for windows depending on what floor they belong to. As expected, the lower building floors are more affected,

² Obstruction Angle Tool: https://github.com/Sustainable3DCities/Obstruction_Angle_Tool.

³ <https://www.rhino3d.com/>.

⁴ <https://www.grasshopper3d.com/>.

⁵ https://github.com/Sustainable3DCities/Direct_Sun_Hours.

Table 1 OA statistics for the different scenarios and their corresponding OA-value differences

| Scenarios | Min OA (degrees) | Max OA (degrees) | Mean OA (degrees) | Median OA (degrees) | St dev OA (degrees) |
|-----------|------------------|------------------|-------------------|---------------------|---------------------|
| EBNV | 0.00 | 34.04 | 7.93 | 5.98 | 6.88 |
| EBV | 0.00 | 84.34 | 16.94 | 14.93 | 12.38 |
| PBNV | 0.00 | 50.59 | 12.65 | 11.37 | 8.71 |
| PBV | 0.00 | 85.23 | 18.04 | 16.69 | 12.04 |
| PBNV—EBNV | 0.00 | 45.32 | 4.71 | 1.21 | 7.23 |
| EBV—EBNV | 0.00 | 73.72 | 9.01 | 4.98 | 11.15 |
| PBV—EBNV | 0.00 | 74.58 | 10.10 | 7.01 | 11.14 |
| PBV—PBNV | 0.00 | 70.58 | 5.39 | 0.00 | 9.70 |

with windows located on the ground and first floor showing a difference in OA value that exceeds 10°.

Figure 10 clearly shows that including vegetation would lead to less windows satisfying the Swedish daylight criterion. Especially, if we consider that even in a not-so-densely-built urban environment, such as our study area, the likelihood of a

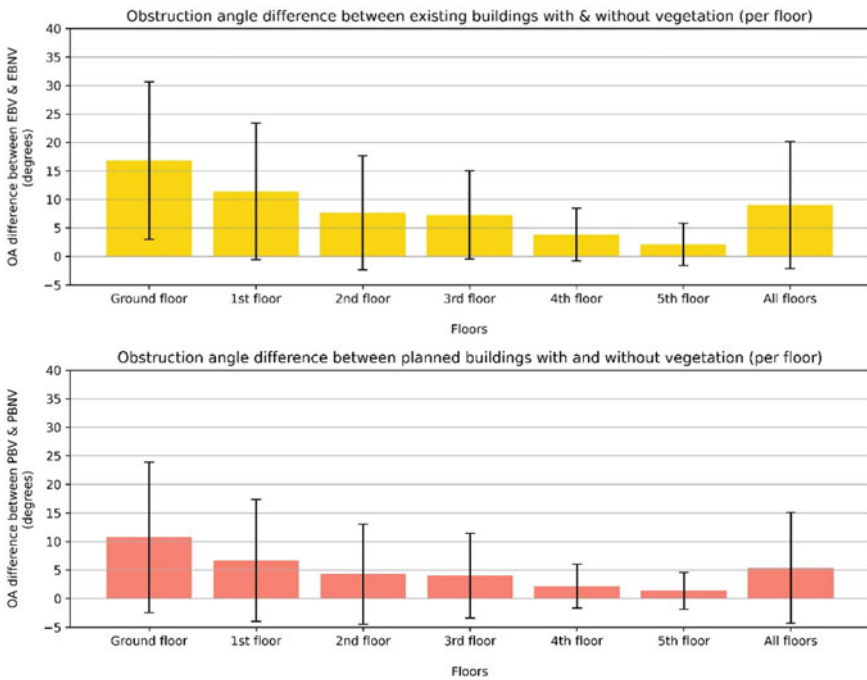


Fig. 9 OA differences between EBNV and EBV (top) as well as PBNV and PBV (bottom) per floor

window on the ground floor meeting the criterion is about 50%. Though there are trees that can block daylight quite effectively, arguing that a tree is completely solid in any case is not totally realistic.

Table 2 shows us that vegetation does not seem to have any significant impact on access to direct sunlight.

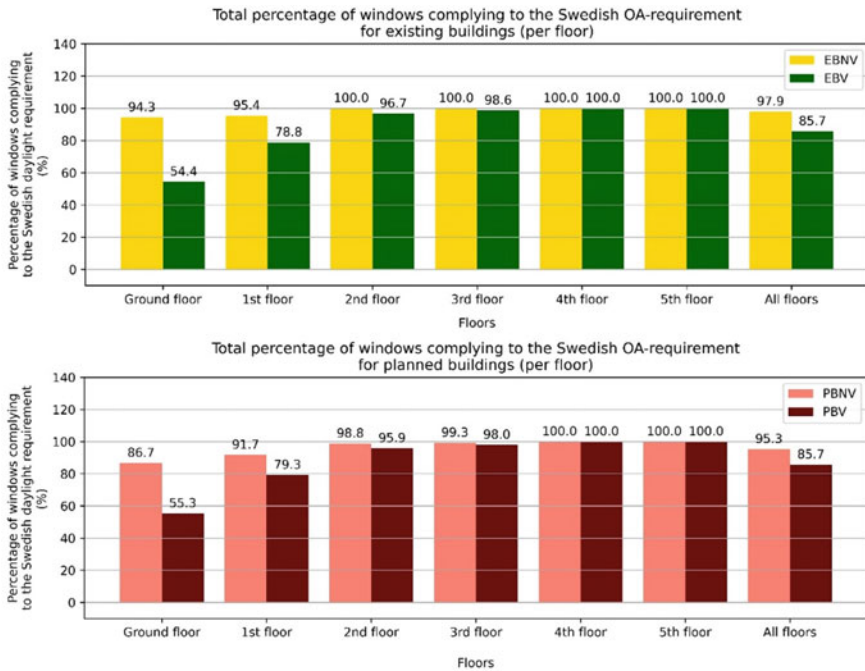


Fig. 10 Percentage of windows complying with the Swedish daylight legislation per scenario; EBNV and EBV (top) as well as PBNV and PBV (bottom)

Table 2 Direct sunlight hours (DSH) statistics for the different scenarios

| Scenarios | Min DSH (h) | Max DSH (h) | Mean DSH (h) | Median DSH (h) | St dev DSH (h) |
|-----------|-------------|-------------|--------------|----------------|----------------|
| EBNV | 2.5 | 12.5 | 10.0 | 10.5 | 1.6 |
| EBV | 0.0 | 12.0 | 8.3 | 8.0 | 2.3 |
| PBNV | 2.0 | 11.5 | 8.9 | 9.5 | 1.8 |
| PBV | 0.0 | 12.0 | 7.9 | 8.0 | 2.2 |
| PBNV—EBNV | - 4.5 | 0.0 | - 1.1 | - 1.0 | 1.0 |
| EBV—EBNV | - 11.5 | 1.0 | - 1.6 | - 1.5 | 1.8 |
| PBV—EBNV | - 11.5 | 1.0 | - 2.1 | - 2.0 | 1.8 |
| PBV—PBNV | - 11.5 | 1.0 | - 1.0 | - 0.5 | 1.7 |

Almost all windows regardless of what floor they belong to seem to comply to the European daylight recommendation (see Fig. 11).

However, this should not be perceived as that vegetation is irrelevant in daylight studies. The study area may have been densified, but it is still a quite sparsely built urban area and vegetation in densely built urban areas will impact daylight access. An indicatory example of this is shown in Fig. 12. This is the only relatively densely built part within our study area. Here 47% (almost half) of the windows fail to meet the European daylight requirement when vegetation is included in the simulation.

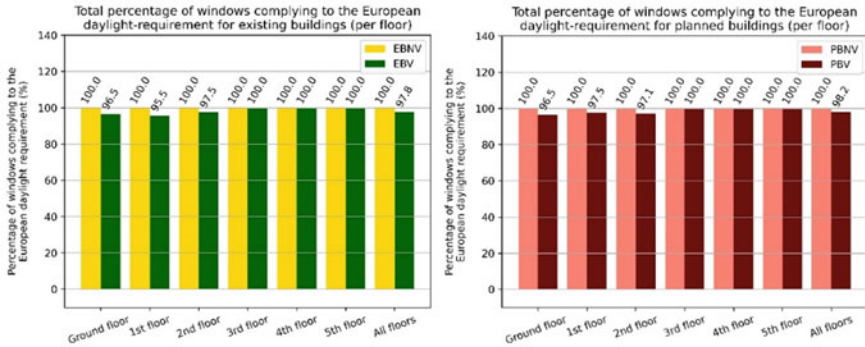


Fig. 11 Percentage of windows complying with the European daylight recommendation per scenario and floor

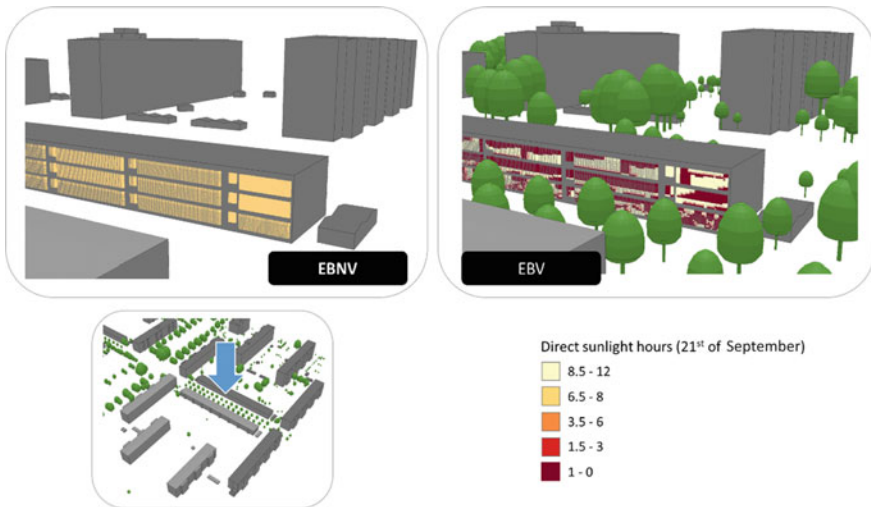


Fig. 12 Example of neighbourhood where windows fail to meet the European daylight recommendation (1.5-h direct sunlight) once vegetation is included in the simulation

4.4 Sunlight Access Modelling Case Study

For this case study, we examine the potential of building facades supporting photovoltaic solar panels and whether vegetation should be included in such simulations.

Methods. This study simulates the annual irradiation received by the building facades highlighted in Fig. 6. The simulation is executed in a CAD-environment and its output is used to estimate the potential of installing photovoltaic solar panels along a façade while also taking into consideration the effect of surrounding vegetation. The annual irradiation was simulated using Rhino3D, Grasshopper, combined with ClimateStudio.⁶ ClimateStudio is based on Radiance and Daysim which have been tested extensively and validated to ensure the production of high-quality simulation output (Reinhart and Herkel 2000; Jakubiec and Reinhart 2013; Santos et al. 2021). The used weather data (EPW⁷) was from Copenhagen. For the simulations, the following reflectance values were used: buildings (20%), ground (10%), and vegetation (15%). The chosen reflectance values for buildings adhere to the corresponding recommended values in EN17037 (CEN 2018) and BREEAM (Jakubiec 2022) while the ground reflectance value was chosen to represent asphalt. The reflectance value for vegetation was selected based on the ClimateStudio Radiance material panel (Solemma 2023) which is in accordance with the values mentioned in recent studies utilising measured data (Jakubiec 2022; Balakrishnan and Jakubiec 2016). The used Radiance parameters where: number of ambient samples per sensor per pass (1024), ambient bounces – ab^8 (6) and weight limit – lw^9 (0.01).

Results. Table 3 shows how excluding vegetation from the annual irradiance simulation can lead to overestimation of the insolation and potential energy production of solar panels installed on building facades. Especially, in the case of South-facing facades (see façades F1, F5 and F8; scenario EBNV and EBV) who are considered most optimal for the installation of solar panels in countries of higher latitudes, the overestimation is prominent. Some North-West-facing facades (F6 & F7) exhibit a similar pattern. Generally, when comparing the mean total annual irradiance of the EBV and PBNV scenarios, it is found that they differ by 2.9%, while the difference between the EBV and PBV scenarios is 5.1%. Fig. 13 presents an example of a North-West-facing facade (F7) where the inclusion of vegetation significantly affects the suitability for installing solar panels. If no vegetation information is included when computing the annual total irradiance before (EBNV) and after (PBNV) the densification, then the sunlight is expected to decrease by 11% (from 405.8 kW/m² to 359.7 kW/m²). When vegetation information is used, the corresponding difference between the pre-densification scenario (EBV) and the post densification scenario (PBV) is 28.8% (from 405.8 kW/m² to 288.8 kW/m²). This means that an urban

⁶ <https://www.solemma.com/climatestudio>.

⁷ <https://climate.onebuilding.org>.

⁸ Maximum number of ambient bounces before a light path terminates.

⁹ Weight below which a light path switches to Russian roulette (as opposed to continuing automatically). Weights are evaluated per path, not for an entire ray tree as in classical Radiance.

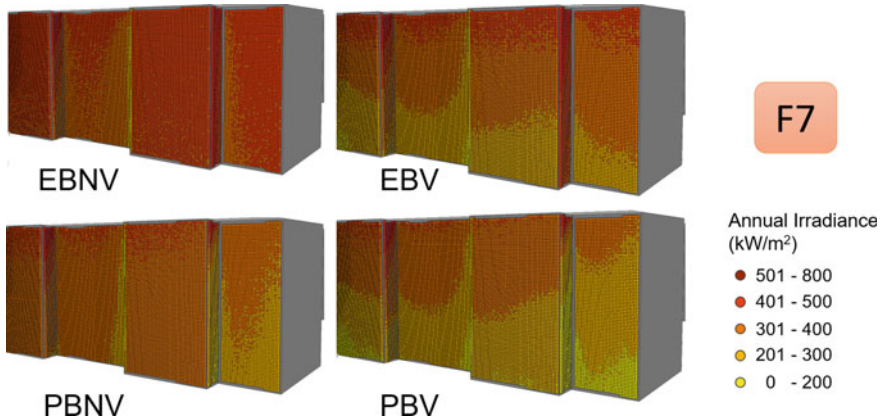


Fig. 13 Example of a building facade (F7) whose suitability for installing a solar panel is affected by including or excluding vegetation from the annual irradiance simulation

planner who omits vegetation from the annual irradiance simulations, will underestimate the actual decrease of sunlight access by 28.8%, which, in this case, puts the facade in the category of less optimal facades for installing solar panels.

5 Recommendation for Inclusion of Vegetation Information in 3D City Models

In Sect. 5.1 we state our recommendations for including vegetation information in the 3D city model. Then in Sect. 5.2 we evaluate this proposal in the urban planning context.

5.1 Recommendation for Vegetation Specifications and Vegetation Related Information in Other Specifications

The recommendations in this section are primarily to improve the current specifications of the Swedish 3D city model (Uggla et al. 2023), but many of the recommendations are valid also for other countries. It should be noted that the Swedish specifications are a CityGML extension (ADE) and that all features in CityGML are supported. The restrictions noted below are recommendations stated in the measuring guidelines to improve the interoperability. We recommend that:

- (1) The vegetation theme should follow CityGML 3.0. Besides the CityGML attributes, there should be some additional attributes (e.g., popular name of

Table 3 Cumulative statistics over the annual irradiation simulation output per scenario and facade

| Scenarios | Min AIrr (kW/m ²) | Max AIrr (kW/m ²) | Mean AIrr (kW/m ²) | Median AIrr (kW/m ²) | St dev AIrr (kW/m ²) | Orientation |
|-----------------|----------------------------------|----------------------------------|-----------------------------------|--|--|-------------|
| F1_EBNV | 586.0 | 793.9 | 697.1 | 702.7 | 43.2 | SW |
| F1_EBV | 424.1 | 786.1 | 644.7 | 661.1 | 86.5 | SW |
| F1_PBNV | 583.2 | 781.5 | 689.6 | 695.2 | 40.4 | SW |
| F1_PBV | 420.8 | 786.4 | 639.1 | 656.9 | 83.7 | SW |
| F1_ EBV-EBNV | - 201.8 | 52.6 | - 52.3 | - 43.1 | 47.4 | SW |
| F1_ PBV-PBNV | - 204.1 | 52.7 | - 50.5 | - 40.5 | 47.4 | SW |
| F2_EBNV | 202.7 | 314.8 | 262.2 | 264.2 | 18.5 | NE |
| F2_EBV | 128.8 | 305.4 | 225.2 | 230.0 | 43.5 | NE |
| F2_PBNV | 196.7 | 307.7 | 258.7 | 260.5 | 17.9 | NE |
| F2_PBV | 125.5 | 300.0 | 223.1 | 227.9 | 43.0 | NE |
| F2_ EBV-EBNV | - 111.8 | 32.8 | - 37.0 | - 34.2 | 28.6 | NE |
| F2_ PBV-PBNV | - 11.04 | 29.1 | - 35.6 | - 32.1 | 28.8 | NE |
| F3_EBNV | 182.3 | 296.3 | 258.2 | 260.7 | 15.7 | NE |
| F3_EBV | 161.2 | 285.4 | 230.2 | 232.3 | 20.8 | NE |
| F3_PBNV | 152.3 | 291.8 | 240.4 | 243.0 | 21.3 | NE |
| F3_PBV | 139.3 | 281.4 | 220.0 | 221.6 | 23.7 | NE |
| F3_ EBV-EBNV | - 98.1 | 17.5 | - 28.0 | - 26.8 | 16.1 | NE |
| F3_ PBV-PBNV | - 79.2 | 24.4 | - 20.5 | - 20.2 | 13.0 | NE |
| F4_EBNV | 261.3 | 445.6 | 378.3 | 382.3 | 27.9 | NW |
| F4_EBV | 119.5 | 446.3 | 352.9 | 368.6 | 55.5 | NW |
| F4_PBNV | 211.1 | 438.6 | 344.7 | 351.0 | 46.7 | NW |
| F4_PBV | 121.8 | 440.5 | 324.1 | 335.3 | 62.9 | NW |
| F4_ EBV-EBNV | 41.4 | - 251.3 | - 25.4 | - 14.9 | 40.0 | NW |
| F4_ PBV-PBNV | - 253.1 | 41.2 | - 20.5 | - 10.8 | 39.3 | NW |
| F5_EBNV | 395.5 | 710.2 | 603.8 | 615.0 | 57.9 | SE |
| F5_EBV | 107.1 | 212.1 | 558.7 | 584.3 | 98.2 | SE |
| F5_PBNV | 386.3 | 713.8 | 598.6 | 610.2 | 58.5 | SE |
| F5_PBV | 204.1 | 711.8 | 556.0 | 581.8 | 98.8 | SE |
| F5_ EBV-EBNV | - 401.4 | 56.9 | - 45.1 | - 21.7 | 68.4 | SE |

(continued)

Table 3 (continued)

| Scenarios | Min AIrr (kW/m ²) | Max AIrr (kW/m ²) | Mean AIrr (kW/m ²) | Median AIrr (kW/m ²) | St dev AIrr (kW/m ²) | Orientation |
|-----------------|----------------------------------|----------------------------------|-----------------------------------|--|--|-------------|
| F5_ PBV-PBNV | - 399.4 | 56.5 | - 42.6 | - 19.7 | 67.3 | SE |
| F6_EBNV | 0.068 | 800.0 | 365.2 | 393.3 | 90.0 | NW |
| F6_EBV | 0.0 | 780.9 | 285.3 | 305.4 | 106.4 | NW |
| F6_PBNV | 0.15 | 785.2 | 347.5 | 370.1 | 87.2 | NW |
| F6_PBV | 0.0 | 786.7 | 278.9 | 296.1 | 105.3 | NW |
| F6_ EBV-EBNV | - 428.2 | 36.5 | - 79.9 | - 60.4 | 74.1 | NW |
| F6_ PBV-PBNV | - 408.3 | 39.8 | - 68.6 | - 50.7 | 67.9 | NW |
| F7_EBNV | 97.3 | 803.3 | 382.8 | 405.8 | 80.2 | NW |
| F7_EBV | 76.2 | 795.1 | 306.1 | 314.3 | 92.7 | NW |
| F7_PBNV | 90.0 | 775.1 | 340.2 | 359.7 | 75.5 | NW |
| F7_PBV | 779.7 | 64.6 | 283.7 | 288.8 | 88.4 | NW |
| F7_ EBV-EBNV | 35.9 | - 251.9 | - 76.6 | - 62.0 | 58.8 | NW |
| F7_ PBV-PBNV | - 218.2 | 32.9 | - 56.5 | - 43.5 | 47.3 | NW |
| F8_EBNV | 297.8 | 830.1 | 636.8 | 650.7 | 69.6 | SE |
| F8_EBV | 165.5 | 810.2 | 532.3 | 554.8 | 110.3 | SE |
| F8_PBNV | 171.2 | 810.4 | 543.4 | 560.3 | 103.7 | SE |
| F8_PBV | 156.2 | 811.5 | 532.5 | 547.1 | 101.9 | SE |
| F8_ EBV-EBNV | - 332.6 | 41.3 | - 104.5 | - 85.5 | 79.7 | SE |
| F8_ PBV-PBNV | - 78.1 | 50.8 | - 10.8 | - 10.3 | 16.4 | SE |

tree species) as added in 3CIM ver. 1.0 (in accordance with Fig. 3). In the 3CIM measuring guidelines it should be stated that the vegetation theme should be used for SVO and PC 3D objects, that is, the 2D (multi-) surface PC object should not be used in the vegetation theme. Furthermore, the recommendation is not to use any prototypic object for SVO objects.

- (2) The Land cover theme must include 2D vegetation information. Together with other land cover classes (and 2D information from the transportation and building theme) the land cover features should constitute a partition.
- (3) The building specification (and city furniture specification) must include the possibility to store information on whether building roofs and facades (and walls) are covered with vegetation.

- (4) A classified point cloud should be available in conjunction to the 3D city model (we do not here provide any recommendation whether this point cloud should be part of the city model or not, that is a separate issue outside the scope of this study).

5.2 *Evaluation of the Recommended Specifications for Urban Planning Applications*

In this section we evaluate the need of vegetation data for the applications described in Sect. 3.3 from the perspective of the recommendations in 5.1.

Ecosystem services. From an urban planning perspective, it is important to compute the common urban ecosystem services (UES) indicators and use tools developed for the evaluation of UES. From a geospatial perspective, the data demand for these indicators and tools are mainly 2D land cover maps (used for computing the total green area, connectivity/fragmentation indicators, etc.) and tree information (for e.g., canopy coverage). For this purpose, the recommended land cover theme and vegetation theme would be sufficient. However, for computing indicators concerning the health of the vegetation (e.g., foliage damage) more attribute information would be required. But we argue here that assessing the tree health is not a common urban planning application, it is more related to e.g., park management and hence this type of information is not mandatory in the 3D city model (in line with our limitation above that the 3D city model is primarily designed for urban planning purposes).

Human well-being. Human well-being is of course a wide concept, that includes many other factors treated separately here such as noise and daylight. Therefore, we restrict the discussion to a common tool for evaluating the well-being in connection to vegetation in Sweden, the 3-30-300 method [cf 3.2 in Konijnendijk (2023)]. What this method requires is information about (1) position of individual trees, (2) the crown diameters of the trees and (3) extension of public green spaces. Information about (1) and (2) are found in the vegetation theme, while (3) is part of the land cover theme (possibly in combination with administrative and/or cadastre information that describes if the space is public).

Noise. Vegetation information is an important aspect in noise modelling for defining the impedance classes of surfaces (softness of the surfaces). In our recommendation this information is mainly taken from the land cover classes. What is required here is that it is made possible to map the CityGML land use classes to the impedance classes in the noise propagation models, e.g., CNOSSOS (Kephelopoulous et al. 2012) and Nord2000 (SP: Nord 2000 2001). Vegetation on facades and roofs (as well as walls) are also important for setting the correct impedance classes. In our recommendation, this information should be included in the building (and city furniture) classes as attributes to the building (wall) classes. Furthermore, trees are important for the acoustical ground effect since the vegetation's root system make the soil porous. This information could be fetched from the recommended vegetation theme.

Also, information required for the modelling of the scattering by the vegetation (stem, branches, etc.) should be possible to retrieve from this theme. To the authors' knowledge, these vegetation scattering effects are not included in the most common noise propagation models, but in a future scenario they might be included and then it is likely that the attribute that describes the vegetation (e.g., species, height, trunkDiameter, crownDiameter) should be sufficient.

Daylight and solar energy. Currently, vegetation is seldom included in daylight and solar energy simulations. But, as discussed in 3.4, there is now a trend in research to model the effect of vegetation which possibly will change the legal framework and practice in this area. Our case studies can be regarded as typical examples of daylight and solar energy simulations with inclusion of vegetation information. To conduct the daylight simulations, we used information corresponding to our recommended SVO objects in the vegetation theme. Also, for other daylight metrics we cannot foresee a need for additional vegetation information that is not included in our recommendation. Also, for the solar energy modelling, as in the last case study, we do not anticipate the need for any additional vegetation information.

In the case studies we simulated the effect on daylight and solar energy by using the two extremes: no vegetation and solid vegetation. The true value is of course somewhere in between these extremes and depends on the amount of transmitted light through the vegetation. We would argue that the transmittance values should not be included in the 3CIM specifications, but that such a value should be estimated based on the stored attribute. The species (e.g., cherry trees allow more sun to pass through their foliage compared to oak trees) and crown diameter, in combination with season of the year, would provide reasonable estimates of the amount of light passing through the vegetation (Littlefair et al. 2022). In addition to this, we also need simulation tools that could handle the semitransparency of the vegetation. Or, for metrics such as direct sunlight hours you could follow the recommendation by Littlefair et al. (2022) and compute two values; one without vegetation and one with vegetation modelled as solid, as extreme cases for evaluating whether the legal framework is fulfilled (cf. Fig. 11).

One other aspect to consider is whether the definitions of the daylight metrics are suitable to include vegetation information (Pantazatou et al. 2023). As can be seen in our case study, a simple measure such as OA (used in e.g., the Swedish legislation) is not capable of handling information about vegetation in a good way. The OA value simulated based on solid vegetation objects is not really meaningful, since it underestimates daylight access in rooms. But for the daylight factor, annual irradiance and direct sunlight hours the simulated values (e.g., using no and solid modelling of vegetation as in the case studies above) are useful for including the effect of the vegetation. One could (hopefully) anticipate that the future legal framework could be adapted to use daylight metrics that can include effect of the vegetation in a satisfying way.

Underground applications. For the applications we have considered here (root effect on utility and buildings) it is sufficient to use the information for the attribute *rootBallDiameter* and *maxRootBallDepth* stored in the SVO and PC classes in the vegetation theme.

Visualizations. The information stored in the proposed vegetation theme (in the SVO and PC classes) are sufficient for visualisation of trees in Fig. 2 on species level (but not on individual level), e.g., by using vegetation models included in software packages. This has been done for e.g., the examples shown in Fig. 5 (left) and Figs. 6 and 7. If information of individual trees is important, then that must be fetched from point clouds as in Fig. 5 (right).

6 Discussion

The focus of this paper has been to investigate the need for adding vegetation to a 3D city model in the context of urban planning. One might argue that this is a too narrow scope since the 3D city models have other application areas. Our limitation stems from the current need for cities to have a comparatively simple model that supports the most important internal needs.

Three things are important to note in the recommendations for vegetation information in Sect. 5.1. Firstly, the recommendation states that the vegetation theme is solely for 3D objects and not for a 2D description of green areas. One could have included the information of 2D green areas in the vegetation theme, and then have holes for the vegetation in the land cover theme. Our choice is based on the fact that it is more common for a person to use 2D vegetation information together with other land cover information than combining 2D and 3D vegetation information. Furthermore, the recommendations make a stronger connection between the city models and the upcoming national specifications of 2D land cover data (not based on CityGML).

A second thing to note is that the recommendation does not introduce the LOD-concept for the vegetation theme. As noted above (cf. Fig. 2) there are some proposals to use various LODs for vegetation. From our perspective, we should not store vegetation objects in several LODs, but the geometries and attributes stored enable the generation of several LODs. What our recommendation is not capable of is describing the trees on an individual level. On the other hand, our model allows linkages between 3CIM and operational systems for e.g., park maintenance, which could store more individual information about the trees. Another aspect of the decision to not define LOD levels for the vegetation theme is that for CityGML 3.0 the LOD concept is only used for the geometric representations (Kolbe et al. 2021).

A third thing to note is the proposed recommendation (in the measuring guideline) to not use prototypic objects for the vegetation theme. One reason for this is that the 3CIM data should be regarded as authority data that should be easy to import to other systems. Inclusion of prototypic objects for vegetation objects would likely decrease the interoperability to many systems. Another important reason is that several tools, especially for visualization, have their own built-in vegetation models (cf. Fig. 5) which facilitate the creation of good vegetation objects based on the recommended SVO information.

7 Conclusions

The motivation for urban vegetation information collection (in Sweden) is mainly for urban planning with a focus on ecosystem monitoring. We foresee that urban vegetation information will be increasingly important in e.g., noise, daylight, and solar energy modelling. To fulfil these current and near future requirements the cities need a comparatively simple model for including the vegetation information in 3D city models. In this paper we recommended that: (1) the vegetation theme should follow CityGML 3.0 with some additional attributes (e.g., popular name of tree species) added as an ADE, (2) no LOD division should be done for the vegetation information stored (but rather derived if necessary), (3) the vegetation theme should only contain 3D vegetation objects while the 2D vegetation is part of the land cover theme, and (4) the building specification (and city furniture specification) must include the possibility to store information on whether building roofs and facades (and walls) are covered with vegetation. These recommendations are based on a literature study of the most common urban planning applications as well as on case studies for including vegetation information in daylight and sunlight access modelling. These case studies indicate that vegetation needs to be considered in the modelling and that it is important that the geospatial community could provide this information, e.g., as part of a 3D city model.

Acknowledgements The study is financed by Formas grant 2020-01460.

We would like to thank Malmö municipality as well as Lantmäteriet for providing geospatial data. We would also like to extend our gratitude to people in the 3CIM project and Lantmäteriet for discussions about city model specifications. Many thanks to Stephan Nebiker and Wei Zhang who have kindly granted us permission to use their figures.

References

- Albert J, Bachmann M, Hellmeier A (2003) Zielgruppen und Anwendungen für Digitale Stadtmodelle und Digitale Geländemodelle. Erhebungen im Rahmen der SIG 3D der GDI NRW (in German only)
- Balakrishnan P, Jakubiec JA (2016) Measuring light through trees for daylight simulations: A photographic and photometric method. International Building Performance Simulation Association. In: Proceedings Conference: building simulation and optimization. IBPSA Publications, Newcastle
- Balakrishnan P, Jakubiec JA (2022) Trees in daylight simulation—measuring and modelling realistic light transmittance through trees. *Leukos* 19(3):241–268
- Berry R, Livesley SJ, Aye L (2013) Tree canopy shade impacts on solar irradiance received by building walls and their surface temperature. *Build Environ* 69:91–100
- Biljecki F, Ledoux H, Stoter J (2016) An improved LOD specification for 3D building models. *Comput Environ Urban Syst* 59:25–37
- Biljecki F, Kumar K, Nagel C (2018) CityGML application domain extension (ADE): overview of developments. *Open Geospat. Data Softw Stand* 3:13

- Boverket: Boverket's mandatory provisions and general recommendations, BBR: BFS 2011:6 with amendments up to BFS 2018:4. <https://www.boverket.se/globalassets/publikationer/publikationer/dokument/2019/bbr-2011-6-tom-2018-4-english-2.pdf>. Last accessed 06 Feb 2023
- Boverket: ESTER—verktyg för kartläggning av ekosystemtjänster. Boverket Homepage, <https://www.boverket.se/sv/PBL-kunskapsbanken/teman/ekosystemtjanster/verktyg/ester/>. Last accessed 08 May 2023
- C/O City 2019: QGYF—ett open source-verktyg för grönytefaktor i QGIS (in Swedish). c/o City Homepage, <https://www.cocity.se/verktyg/qgyf/>. Last accessed 08 May 2023
- CEN 2018. European Standard EN17037 Daylight in buildings. Brussels: Comité Européen de Normalisation, Dec 2018
- Coder K (1996) Identified benefits of community trees and forests. University of Georgia School of Forest Resources, Athens (USA)
- con terra GmbH—Geo-IT solutions and virtual city systems: CityGML Create SolitaryVegetationObject with implicit representation (Version 1.1.0) [FME-script]. <https://hub.safe.com/publishers/con-terra/templates/citygml-create-solitaryvegetationobject-with-implicit-representation#description>. Last accessed 02 May 2023
- Dubois M-C, Gentile N, Laike T, Bournas I, Alenius M (2019) Daylighting and lighting under a Nordic sky. (First ed) Studentlitteratur AB, Lund
- Duncan JMA, Boruff B, Saunders A, Sun Q, Hurley J, Amati M (2019) Turning down the heat: an enhanced understanding of the relationship between urban vegetation and surface temperature at the city scale. *Sci Total Environ* 656:118–128
- European Commission, Directorate-General for Environment, Mapping and assessment of ecosystems and their services: mapping and assessing the condition of Europe's ecosystems: progress and challenges: 3rd report—final, March 2016. Publications Office. <https://data.europa.eu/doi/10.2779/351581>
- Fan H, Kong G, Zhang C (2021) An interactive platform for low-cost 3D building modelling from VGI data using convolutional neural network. *Big Earth Data* 5(1):49–65
- Forssen J, Gustafsson A, Berghauser Pont M, Haeger-Eugensson M, Achberger C, Rosholm N (2022) Effects of urban morphology on traffic noise: a parameter study including indirect noise exposure and estimated health impact. *Appl Acoust* 186:108436
- Giannelli D, León-Sánchez C, Agugiaro G (2022) Comparison and evaluation of different GIS software tools to estimate solar irradiation. *ISPRS Ann Photogrammetry Remote Sens Spat Inf Sci* 4:275–282
- Gröger G, Kolbe TH, Nagel C, Häfele KH (eds) OGC city geography markup language (CityGML) encoding standard, Ver. 2.0. OGC Homepage, <http://www.opengeospatial.org/standards/citygml>. Last accessed 03 May 2023
- Gröger G, Plümer L (2012) CityGML—Interoperable semantic 3D city models. *ISPRS J Photogramm Remote Sens* 71:12–33
- Guerrero Iñiguez JI (2017) Geometric modelling of tree roots with different levels of detail. *ISPRS Ann Photogrammetry Remote Sens Spat Inf Sci* 4:29–35
- Gustafsson A, Genell A, Ögren M (2023) NORD2000—Beräkning av buller från väg och spårtrafik för svenskt bruk – en användarhandledning (in Swedish), Kunskapscentrum inom buller, report
- Hartman K (2021) Modelling solar access at the window level of buildings—a case study for urban densification simulations in Sweden. LUP Student Papers. <http://lup.lub.lu.se/student-papers/record/9100070>. Last accessed 28 Apr 2023
- Hornikx M (2016) Ten questions concerning computational urban acoustics. *Build Environ* 106:409–421
- Jakubiec JA (2022) Data-driven selection of typical opaque material reflectances for lighting simulation. *Leukos* 19(2):176–189
- Jakubiec JA, Reinhart C (2013) A method for predicting city-wide electricity gains from photovoltaic panels based on LiDAR and GIS data combined with hourly Daysim simulations. *Sol Energy* 93:127–143

- Kephalopoulos S, Paviotti M, Anfosso-Lédée F (2012) Common noise assessment methods in Europe (CNOSSOS-EU), 180-p
- Ko WH, Kent MG, Schiavon S, Levitt B, Betti G (2022) A window view quality assessment framework. *Leukos* 18(3):268–293
- Kolbe TH, Kutzner T, Smyth CS, Nagel C, Roensdorf C, Heazel C (eds) OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard, Ver. 3.0.0. (2021). OGC Homepage, <http://www.opengeospatial.org/standards/citygml>. Last accessed on 03 May 2023
- Konarska J, Lindberg F, Larsson AT, Thorsson S, Holmer B (2014) Transmissivity of solar radiation through crowns of single urban trees—application for outdoor thermal comfort modelling. *Theoret Appl Climatol* 117(3–4):363–376
- Konijnendijk C (2023) Evidence-based guidelines for greener, healthier, more resilient neighbourhoods: introducing the 3-30-300 rule. *J for Res* 34:821–830
- Kourdounouli C, Jönsson AM (2020) Urban ecosystem conditions and ecosystem services—a comparison between large urban zones and city cores in the EU. *J Environ Planning Manage* 63(5):798–817
- Lechner R (2014) Heating, cooling, lighting: sustainable design methods for architects, 4th edn. Wiley, New Jersey
- Lee ES, Matusiak BS, Geisler-Moroder D, Selkowitz SE, Hescong L (2022) Advocating for view and daylight in buildings: next steps. *Energy Build* 265
- Littlefair P, King S, Howlett G, Ticleanu C, Longfield A (2022) Site layout planning for daylight and sunlight: a guide to good practice (No. 978-1-84806-483-6). Build Res Establishment
- Löwner MO, Gröger G, Benner J, Biljecki F, Nagel C (2016) Proposal for a new LOD and multi-representation concept for CityGml. *ISPRS Ann Photogrammetry Remote Sens Spat Inf Sci* 4:3
- Maes J et al (2020) Mapping and assessment of ecosystems and their services: an EU ecosystem assessment, EUR 30161 EN. Publications Office of the European Union, Ispra
- Matusiak B, Arbab S (2022) The impact of greenery and surface reflectances on solar radiation in perimeter blocks. *Archit Sci Rev* 65(5):359–369
- Nebiker S, Bleisch S, Christen M (2010) Rich point clouds in virtual globes—A new paradigm in city modelling? *Comput Environ Urban Syst* 34(6):508–517
- Ortega-Córdova L (2018) Urban vegetation modeling 3D levels of detail. Master thesis Delft University. <https://repository.tudelft.nl/islandora/object/uuid%3A8b8967a8-0a0f-498f-9d37-71c6c3e532af>. Last accessed 04 May 2023
- Osterhaus W (2005) Discomfort glare assessment and prevention for daylight applications in office environments. *Sol Energy* 79(2):140–158
- Pan W, Du J (2021) Effects of neighbourhood morphological characteristics on outdoor daylight and insights for sustainable urban design. *J Asian Archit Build Eng* 21(2):342–367
- Pan J, Jakubiec JA (2022) Simulating the impact of deciduous trees on energy, daylight, and visual comfort: impact analysis and a practical framework for implementation. In: *Proceedings of ESIM 2022: 12th conference of IBPSA-Canada*, IBPSA Publications, Toronto
- Pantazatou K, Kanters J, Olsson PO, Lande Nyborg J, Harrie L (2023) Input data requirements for daylight simulations in urban densifications. *Urban Info* 2(2)
- Peng J, Bullen R, Keam S (2014) The effect of vegetation on road traffic noise. Conference proceeding, *Inter.noise*
- Region Skåne: 3-30-300 i Skåne—Analysmodell för grönnare och hälsosammare städer (in Swedish). Region Skåne Homepage. <https://utveckling.skane.se/publikationer/regional-utveckling/3-30-300-i-skane-analysmodell-for-gronnare-och-halsosammare-stader/>. Last accessed 05 May 2023
- Reinhart C, Herkel S (2000) The simulation of annual daylight illuminance distributions—a state-of-the-art comparison of six RADIANCE-based methods. *Energy Build* 32(2):167–187
- Sadeghi R, Mistrick R (2021) The impact of exterior surround detail on daylighting simulation results. *Leukos* 18(3):341–356

- Santos L, Caetano I, Leitão AE, Pereira IC (2021) Uncertainty in daylight simulations of algorithmically generated complex shading screens. In Proceedings of building simulation 2021: 17th conference of IBPSA, pp 2261–2268. IBPSA Publications, Bruges
- Skovbro A (2002) Urban densification—a sustainable urban policy? WIT Transa Ecol Environ 54
- Solemma LLC (2023) Materials—ClimateStudio latest documentation. ClimateStudio. <https://climatestudiodocs.com/docs/materials.html>. Last accessed 04 July 2023
- SP: Nord 2000. New Nordic prediction method for road traffic noise, SP Rapport 2001:10 (2001). <https://www.diva-portal.org/smash/get/diva2:1190134/FULLTEXT01.pdf>. Last accessed 05 May 2023
- Stadsträd: Stadsträd Homepage. <https://info.stadstrad.se/>. Last accessed 05 May 2023
- Tarrero A, Martin MA, Gonzalez J, Machimbarrens M, Jacobsen F (2008) Sound propagation in forests: a comparison of experimental results and values predicted by the Nord 2000 model. *Appl Acoust* 69(7):662–671
- Ugglå M, Olsson P, Abdi B, Axelsson B, Calvert M, Christensen U, Gardevärm D, Hirsch G, Jeansson E, Kadric Z, Lord J, Loreman A, Persson A, Setterby O, Sjöberger M, Stewart P, Rudenå A, Ahlström A, Bauner M, Hartman K, Pantazatou K, Liu W, Fan H, Kong G, Li H, Harrie L (2023) Future Swedish 3D city models—specifications, test data, and evaluation. *ISPRS Int J Geo Inf* 12(2):47
- Van Renterghem T, Botteldooren D, Verheyen K (2012) Road traffic noise shielding by vegetation belts of limited depth. *J Sound Vibr* 331(10):2404–2425
- Velarde MD, Fry G, Tveit M (2007) Health effects of viewing landscapes—landscape types in environmental psychology. *Urban for Urban Greening* 6(4):199–212
- Villalba AM, Pattini AE, Correa ÉN (2014) An approach to urban tree daylight permeability simulation using models based on louvers. *Build Environ* 73:75–87
- Vischer JC (2008) Towards an environmental psychology of workspace: how people are affected by environments for work. *Archit Sci Rev* 51(2):97–108
- Wang Y, Akbari H (2016) The effects of street tree planting on urban heat island mitigation in Montreal. *Sustain Cities Soc* 27:122–128
- World Health Organization. Regional Office for Europe: Urban green spaces and health. <https://apps.who.int/iris/handle/10665/345751>. Last accessed 08 May 2023
- Zhang W, Li X, He Z (2022) Semantic urban vegetation modelling based on an extended CityGML description. *J Digital Landscape Archit* 200–212

Shadowing Calculation on Urban Areas from Semantic 3D City Models



Longxiang Xu, Camilo León-Sánchez , Giorgio Agugiaro ,
and Jantien Stoter 

Abstract Nowadays, our society is in the transit to adopt more sustainable energy sources to reduce our impact on the environment; one alternative is solar energy. However, this is highly affected by the surroundings, which might cause shadowing effects. In this paper, we present our method to perform shadowing calculations in urban areas using semantic 3D city models, which is split into five sections: Point Grid Generation, Sun-Ray Generation, Nightside Filtering, Bounding Volume Hierarchy and the intersection between the sun rays and the BVH to identify which locations are shadowed at a given moment (epoch). Our tests are performed in Rotterdam's city center, a dense urban area in The Netherlands. Our initial results indicate that the computational time per 100k grid points fluctuates within 0.2–0.7s.

Keywords Solar potential · 3DCM · Shadows analysis · Raytracing

1 Introduction

Currently, our society faces several energy challenges. Most of the produced energy comes from fossil fuels with the consequence of greenhouse gas emissions. On the other side, there are still hundreds of millions of people that do not have access to electricity.

This article was selected based on the results of a double-blind review of an extended abstract.

L. Xu
Delft University of Technology, Delft, The Netherlands
e-mail: L.Xu-18@student.tudelft.nl

C. León-Sánchez (✉) · G. Agugiaro · J. Stoter
Faculty of Architecture and the Built Environment, Department of Urbanism, 3D Geoinformation group, Delft University of Technology, Julianalaan 134, 2628BL Delft, The Netherlands
e-mail: c.a.leonsanchez@tudelft.nl

G. Agugiaro
e-mail: g.agugiaro@tudelft.nl

J. Stoter
e-mail: j.e.stoter@tudelft.nl

The United Nations states that by 2050, at least 66% of the world's population will be living in urban areas UN. Population Division (2018). Incorrect quantification of the energy requirements of buildings can result in flawed judgments and misguided strategies for energy supply.

Seeking for clean energy sources is a consequence of the current time Shukla et al. (2022). According to the International Energy Agency (IEA), solar energy and wind power are expected to contribute to 43% of global electricity generation by 2030, an increase from the current 28% IEA (2022). Additionally, the Council of the European Union has reached a political agreement that demands renewable energy sources to comprise 42.5% of Europe's total energy consumption by 2030, with a target of at least 49% renewable energy consumption specifically for buildings, heating, and cooling purposes European Council (2023). In this respect, solar energy plays a major role as an alternative to meet the growing demand for energy in many countries. However, this energy source is not without challenges such as shadowing, which rapidly decreases the performance of any solar panel, and it is always changing due to the sun movements over the sky.

When solar panels are placed in remote areas with abundant sunlight and generate electricity, significant losses occur during the transmission process. According to the Energy Information Administration (EIA) U.S. Energy Information Administration (2023), losses in power transmission and distribution can amount to 5%, whereas energy losses within urban areas can increase to 50% due to urban morphology Perera et al. (2019). To mitigate energy losses, a decentralized approach of installing solar panels in residential homes can be adopted. Installing photovoltaic systems in residential buildings not only reduces transmission losses and improves energy efficiency but also allows for excess energy generation to be fed back into the urban grid Perera et al. (2019), thereby reducing reliance on fossil fuel-based power generation. However, the decentralized installation of solar panels on rooftops in urban residences may result in fluctuating solar energy output due to dynamic shading caused by cloud cover and surrounding structures.

Shadowing analysis is therefore important in urban planning Palme et al. (2020), especially for solar potential by helping to identify suitable locations in urban areas considering existing structures and future developments; this analysis is relevant for policymakers by encouraging of the population to adopt solar energy while ensuring an efficient utilization of the available resources for a given location.

To perform a shadowing analysis, one needs to consider the geographical location and its surroundings, that is the case for near and far topography, constructions, and vegetation obstacles de Sá et al. (2022). These types of urban objects can be represented by means of semantic 3D city models (3DCM) Aguiaro et al. (2020), which are datasets that allow for a coherent geometrical and semantic representation of urban features in a well-defined data structure.

In this paper, we present the development of a shadowing analysis tool based on 3DCM. Our work consumes the 3DBAG Peters et al. (2022), which is an open dataset containing 3D models of all building in The Netherlands. It is available as downloads in several formats and multiple levels of detail—LoD—(0, 1.2, 1.3, 2.2) Biljecki et al. (2016). Current version consumes the OBJ format.

2 Related Work

In urban contexts, accurately calculating the potential solar irradiance relies on the estimation of shadows during specific times. By integrating solar irradiance and shadowing effect over time, accurate estimates of former can be obtained. Currently, there are three main categories of shadow calculation methods that aims at facilitating solar irradiance estimation: raster-based, viewport, and vector-tracing.

2.1 Raster-Based Methods

Raster-based methods utilize digital elevation models (DEM) or digital surface models (DSM) to represent the urban environment. This approach involves performing shadow calculations in 2.5D, where each location is assigned a single elevation value. The illumination is determined by rasterizing the sun's rays into pixels with elevation and testing for intersections between the rasterized sun rays and the terrain. The method can be also called line-of-sight query Ledoux et al. (2022). Fu and Rich (2002) developed one of the earliest software applications using this method, which has been adopted by ArcGIS Fu and Rich (2002); Hurkmans (2022). Huang et al. (2015) employed this raster method to estimate solar potential in Shanghai, incorporating Graphic Processing Unit (GPU) acceleration Huang et al. (2015). Lukač and Žalik (2013) used a raster DEM to calculate shadows, integrating a variable-resolution grid to capture more details in the DEM for complex urban environments Lukač and Žalik (2013). Raster-based methods are relatively fast and straightforward to implement. However, they are limited to processing 2.5D data, which is a simplified representation of the complex urban environment. These methods therefore struggle to account for intricate vertical structures, and the resolution and accuracy of the DSM/DEM significantly impacts the accuracy and performance.

2.2 Viewport Method

Viewport methods are visualization-based approaches that simulate the view from a prospective shadow-receiving position. The shadowing effect is estimated by the intersection of the sun rays and the surrounding objects within that view.

Tabik et al. (2013) implemented a method to calculate the horizon from the DEM of a shadow-receiving solar panel Tabik et al. (2013). This technique involves determining both a high-resolution near-end horizon and a low-resolution far-end horizon to represent shadows. The method calculates the horizon with an angle resolution of 1°. Calcabrini et al (2019) utilized the sky view factor (SVF) and sun coverage factor (SCF) to estimate shadows Calcabrini et al (2019). SVF represents the portion of the sky visible from the shadow-receiving point, while SCF quantifies the ratio

between shadowed and illuminated times at that location. These factors are derived from skyline profiles obtained by projecting the 3D environment onto 2D shapes at the shadow-receiving position. Stendardo et al. (2020) also employed SVF and SCF calculations using a raster DSM and accelerated the computation with a GPU Stendardo et al. (2020). Zhou et al. (2022) scanned a raster DSM to calculate SVF and SCF. Lan et al. (2021) utilized fisheye street view images to calculate SVF and SCF. Liang et al. (2020) proposed a method that generates cube maps for each shadow-receiving point, where the surrounding environment is projected onto the faces of the cube maps. The shadowing effect is achieved by projecting the sun rays onto the cube maps and testing for intersections. Cube map generation is realized through graphics rendering engines. This straightforward method can be easily implemented using graphics engines and offers high accuracy due to the inherently higher horizontal and vertical angle resolution compared to scanning DSM at intervals.

Viewport methods simulate the view from a shadow-receiving point in a 3D environment. They typically exhibit high accuracy and can incorporate reflective and diffuse irradiance in solar irradiance calculations. However, generating views for all surfaces, whether through DSM scanning or graphic rendering engines, is computationally expensive, especially at a city scale. High-resolution views are necessary for accuracy, but this further intensifies the computational demands. Consequently, viewport methods are more suitable for small-scale simulations and are not applicable for city-scale simulations.

2.3 *Vector-Tracing Method*

The term “vector tracing method” refers to a technique used to determine shadows by casting sun beams from a surface towards the sun. If the sun’s rays are obstructed, the surface is considered to be in shadow. This method relies on testing the intersection between rays and objects. The process of testing whether rays intersect with objects is also known as ray-object intersection test.

Hofierka and Zlocha (2012) developed a method that builds upon raster shadow calculation. They transformed 3D city models into voxels and performed shadow calculations by testing the intersection between sun rays and voxels. This approach involved creating a voxel grid with dynamic resolution to represent different parts of buildings. However, the voxel data model inherently led to redundancy due to a large portion of unoccupied spaces.

The vector tracing method is particularly accurate in complex urban environments since the accuracy of the intersection test remains consistent regardless of scene complexity. However, the computational requirements can be enormous when exhaustively testing the intersection for each ray and each geometric primitive in the scene, such as triangles. Therefore, significant efforts have been made to reduce the number of intersection tests.

Liang et al. (2014) introduced shadow-radius-based culling and view-frustum-based culling, which significantly decreased the number of intersection tests. Wang

et al. (2023) introduced the “neighbourhood shading building surfaces” determination criterion, which includes distance filtering, solar azimuth quadrant filtering, and nightside filtering Wang et al. (2023). These techniques decrease the number of surrounding objects that need to be considered in the intersection test, thereby speeding up the calculation. Hurkmans (2022) utilized an R-tree to partition the scene and accelerate the ray-object intersection. However, the R-tree used in this method is two-dimensional. Similar radius culling methods were also employed.

Although current vector tracing methods have significantly reduced computational requirements, they are implemented with assumptions that sacrifice the accuracy of the vector tracing methods. For example, radius culling methods may result in inaccurate shadow calculations in cases involving high-rise buildings or mountainous terrains. Therefore, the objective of this article is to develop a vector tracing method that does not rely on culling assumptions while efficiently applying shadow casting.

2.4 Other Methods

There are also other methods for shadow calculation available, such as polygon clipping, which involves projecting the polygon surfaces of the surroundings onto the shadow receiving surfaces in the direction of the sun ray Viana-Fons et al. (2020); Zhu et al. (2020). Another method is pixel counting, where a graphics rendering engine is used to render an image at the vantage point representing the sun location to observe the shadow receiving surface. The visible surfaces are illuminated, while the others are considered shadowed Jones and Greenberg (2011). Robledo et al. (2019) utilized graphics engine and determines if a surface is shadowed based on its image z-coordinate. Objects that are closer to the point of view will have a lower z-coordinate. If the shadow receiving surface does not have the smallest z-value in the pixel, it is considered shadowed.

The polygon clipping method is not capable of effectively handling complex city environments due to its limitations in modelling small geometric variations. On the other hand, while pixel counting methods offer high accuracy, they suffer from the same drawback as the viewport method, which requires the graphic engine to exhaustively render different views to determine the shading effect on all surfaces. In the context of a city-scale application, setting up the camera position, rendering fineness, depth of field, and other render options necessitate careful fine-tuning to strike a balance between efficiency and accuracy. Consequently, these two methods are unsuitable for accurately calculating shadows at the city scale.

3 Method

The workflow diagram depicting our process of shadow analysis is presented in Fig. 1. The computation of shadowing effects for a given region involves five key

steps. First, the generation of a point grid on the surfaces of buildings, serving as location samples of the surfaces on which the shadow effect will be calculated. Second, the lines that represent the sun rays are generated with the grid points as starting point and the direction of the sunlight as the ray direction after that we calculate the nightside filtering to exclude from the analysis those points located in surfaces that are opposite to the sun location since this already indicate that they are shadowed. Third, Bounding Volume Hierarchy (BVH) Meister et al. (2021) are reconstructed to efficiently store the city’s geometries. Finally, an intersection test is performed between the sun rays and the BVH. The outcome of the intersection tests determines whether the corresponding point’s surface is in the shadow at a given epoch.

3.1 Point Grid Generation

Our implementation uses the triangular meshes stored in obj files. We produce a point grid over the building surfaces to locate the samples for shadowing calculation. We use the barycentric coordinates of the input triangles to generate its point grid to

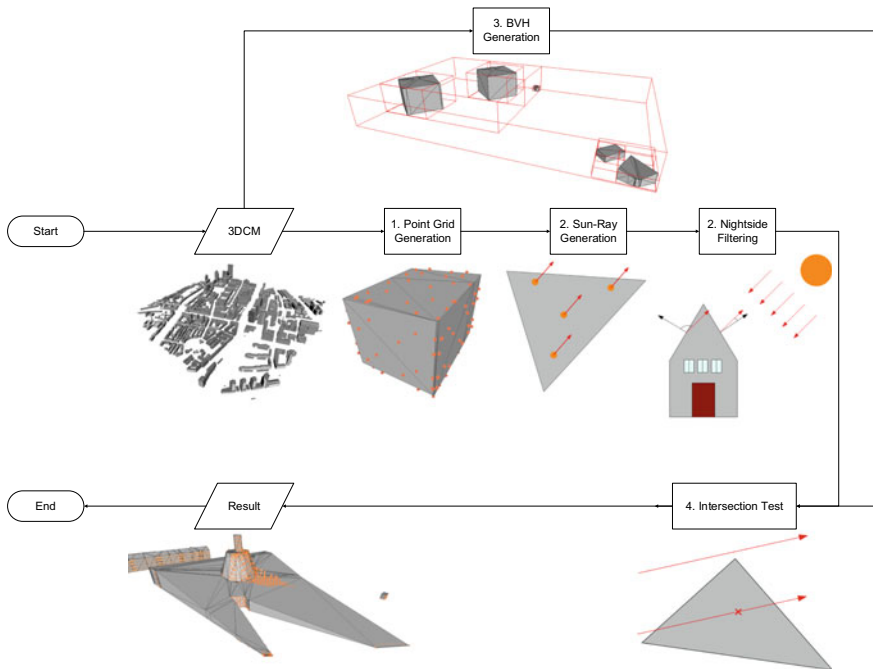


Fig. 1 General workflow of the implemented method

guarantee that point(seconds) are on the triangle's plane. We also add the constraint that the sum of all coefficients of the vector that define the point location is equal to 1 ($w_0 + w_1 + w_2 = 1$) to ensure that each point lies inside the triangle. The adoption of barycentric coordinates and recursive splitting can also handle the situation of sliver triangles and small triangles while uniform spacing will result in undersampling.

The calculation of the point grid for a given triangle and the subsequence division of big triangles into smaller ones consists of the following steps: Figure 2 shows a sketch of the process.

1. Calculate the area of the triangle.
2. Calculate the number of grid's points needed based on 1. (One point per given a defined threshold by the user).
3. Calculate the number of splits needed based on 2.
4. Split the triangle recursively by linking mid points of the bigger triangle edges.
5. Calculate the barycentre of the split triangles.

To ensure uniform samples and full coverage of sliver triangles, an additional constraint is added: When the triangle is sliver, one more split will occur in the corner of acute angle. This constraint mitigates the degenerate case of the method that point grid sampled is uniform in one axis while cluttered in another axis. Figure 4 shows the result of the implementation of this constraint.

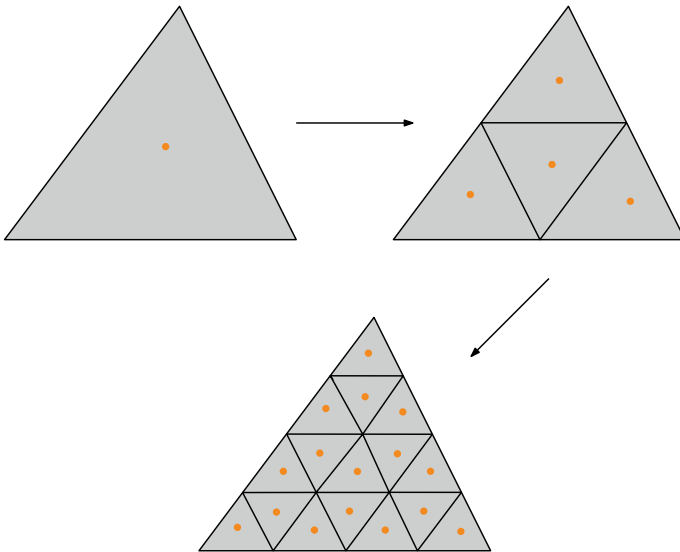


Fig. 2 Illustration of point grid generation

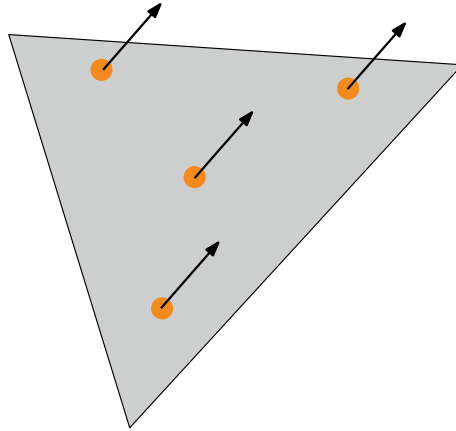


Fig. 3 The illustration of Sun rays generated from point grid on a surface

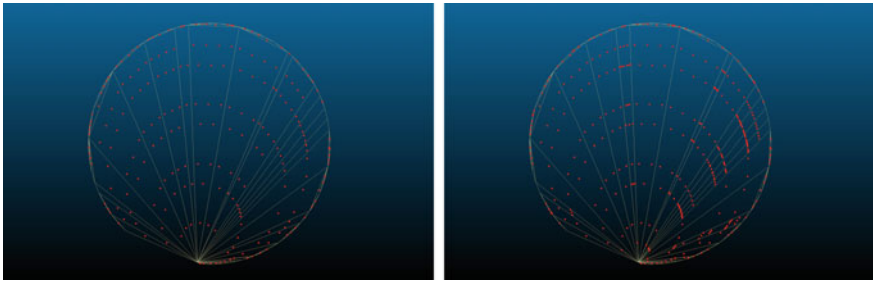


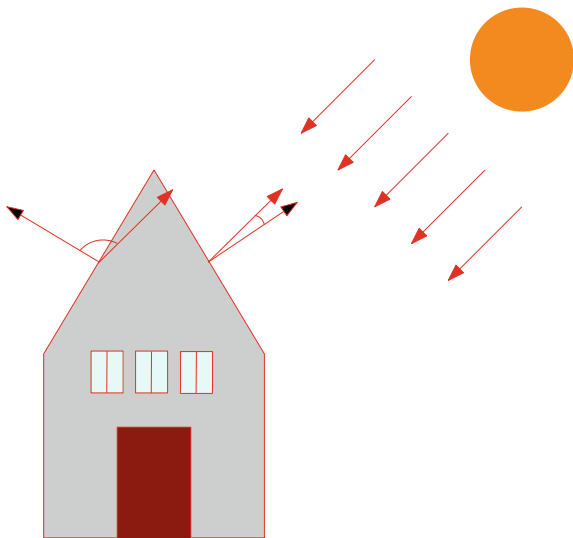
Fig. 4 Comparison of point grid on sliver triangles with (right) and without (left) constraint. Notice there are closer samples covering the acute angle with the constraint. And grid points are spread more evenly across the long axis of the triangles

3.2 Sun Ray Generation with Nightside Filtering

Sun rays are an essential element in shadowing calculations. A ray can be defined by its origin and direction. The location of each point in the previously created grid is used as the starting point to reconstruct the ray and its direction is set to be the vector pointing to the sun as shown in Fig. 3. By this approach, we can test, if the ray originating from the surface will hit any of the surrounding building surfaces. The rays that hit any surface will indicate that its origin (the corresponding grid point) is shadowed.

In this phase of the process, the Nightside Filtering methodology proposed by Wang et al. (2023) is applied to eliminate sun rays that are inevitably incident on surfaces due to self-shadowing. This strategy significantly minimizes the requirement for additional intersection tests. The phenomenon of self-shadowing can be identified by examining the angle between the orientation of the surface and the direction of the

Fig. 5 The illustration of nightside filtering. The black arrow represents the surface normal, and the red arrow represents the sun ray direction



sun ray. When the angle exceeds 90 degrees, self-shadowing is present, eliminating the need for further intersection tests. The respective conditions can be mathematically formulated as:

$$\begin{cases} \mathbf{d} \cdot \mathbf{n} < 0, & \text{self-shadowing occurs (angle} \geq 90^\circ) \\ \mathbf{d} \cdot \mathbf{n} \geq 0, & \text{no self-shadowing (angle} < 90^\circ) \end{cases} \quad (1)$$

In these expressions, \mathbf{d} and \mathbf{n} denote the direction of the sun ray and the surface normal, respectively. It is crucial to note that, in the context of this study, the sun ray’s direction originates from the grid point, which is the surface. Consequently, the direction of the sun ray is the inverse of the actual direction. This process is illustrated in Fig. 5.

3.3 Constructing the Bounding Volume Hierarchy

We use the Bounding Volume Hierarchy tree-like acceleration structure to speed-up the sun ray traced rendering Meister et al. (2021). The root node of the BVH tree represents the Axis Aligned Bounding Box (AABB) of the whole scene, then triangles are recursively partitioned into two different groups that contain the same number of triangles. An example of BVH is shown in Fig. 6.

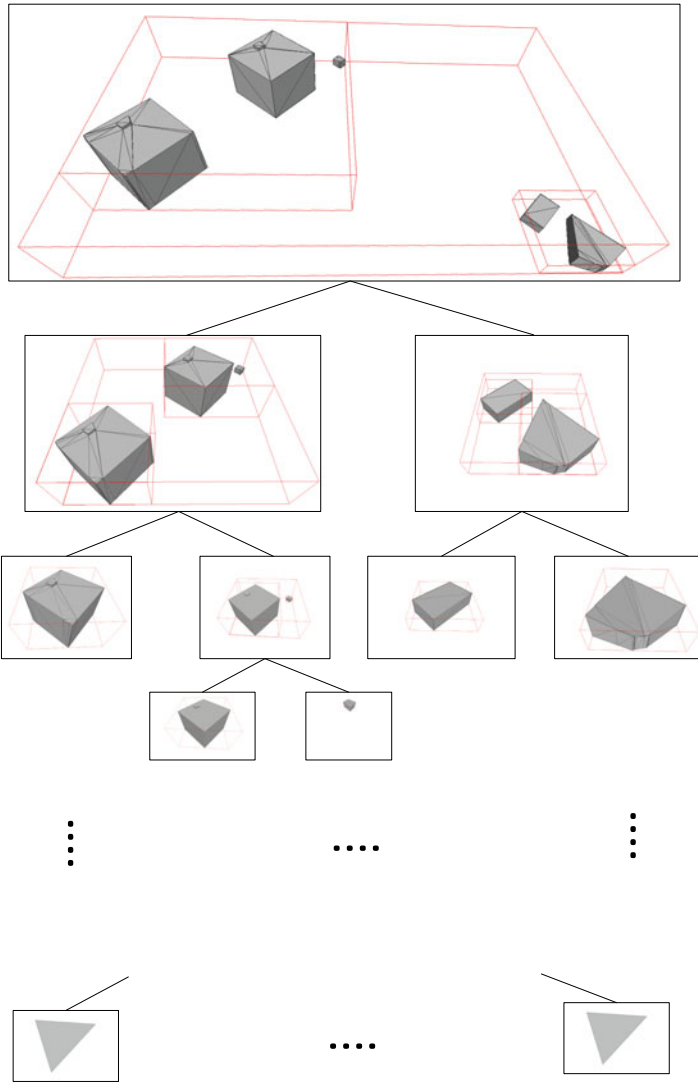


Fig. 6 Illustration of the BVH. Note that the leaf nodes are triangles

3.4 Intersection Test

The tests are applied to each ray and the BVH. Before the intersection test, night-side filtering will be carried out. During this step, the orientation of the sun on the surface, whether it is facing towards or away from the surface, will be determined. Surfaces that are oriented away from the sun (associated sun rays) will be identi-

fied as shadowed and will not undergo any further intersection testing. In the case of parent nodes, further intersection tests are applied on its children's nodes. Since only leaf-nodes are triangles, most of the tests are of the ray-box intersection type, which is computationally less complex compared to ray-triangle intersection tests. Additionally, due to the binary tree structure, the time complexity of the intersection tests decreases from $O(n \cdot m)$ to $O(n \cdot \log_2 m)$, where n represents the number of rays and m represents the number of triangles in the scene.

This hierarchical approach yields significant performance acceleration, particularly for complex scenes covering large areas. When a ray intersects with a triangle, the corresponding origin (grid point) is marked as shadowed.

4 Results

Our method has been implemented in c++. The BVH construction is adopted from the open book *Ray Tracing: The Next Week Ray Tracing in One Weekend Book Series* (2020). We use an Ubuntu 20.04 distribution over a WSL for Windows 11. The hardware specifications of the computer include a 12th Gen Intel(R) Core (TM) i7-12700H CPU and 32GB RAM.

The experiment on shadow calculation was conducted using the 3D BAG data Peters et al. (2022) of Rotterdam Centrum (see Fig. 7), which encompasses an area of 3.7 km by 3.7 km and includes over 20,000 buildings. The selection of this study area was motivated by its complex urban structure, characterized by numerous tall and densely packed modern buildings, which provided a better reflection of the effectiveness of the algorithm. The tiles corresponding to this area, along with their details, are listed in Table 1.

In this experiment, shadow conditions are computed for the year of 2023. The experiment dates are set on the 1st and 15th of each month, and the corresponding positions of the sun were obtained for these dates. Shadows are calculated at hourly intervals, excluding cases corresponding to the solar altitude below the horizon, which in total, lead to 301 epochs. The setup is intended to evaluate the performance of the model in a real-world scenario.

The experiment involved calculating the shadow for each tile's LoD 2.2 data. For each tile, the shadow analysis is performed for all the surfaces of the buildings contained are computed. Future work will focus on specific surfaces, such as rooftops and building facades, for targeted calculations, leveraging semantic information of 3D city models. Table 2 shows the results of the shadow calculation for the tiles in the study area.

We perform an additional experiment on tile 3340. This tile was designated as the target shadow calculation area, while all surrounding tiles were utilized solely for intersection tests, employing less detailed models (their LoD 1.2 data), thus further reducing the number of potential intersection tests. Specifically, only tile 3340 was sampled (using point grid generation), and all the buildings of tile and the surrounding tiles were constructed into a bounding volume hierarchy. This setup

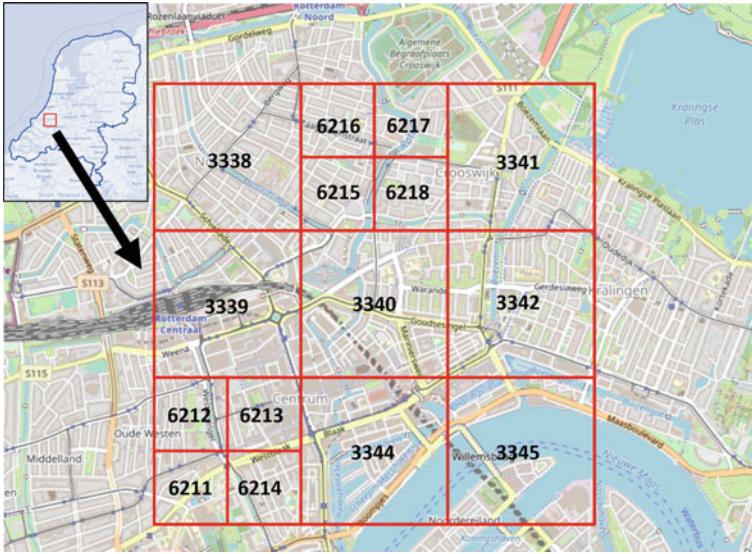


Fig. 7 Study area in Rotterdam, The Netherlands

Table 1 Tile information

| Tile | Number of buildings | Number of faces | | Extent (m) |
|--------------|---------------------|------------------|------------------|------------------|
| | | LoD 1.2 | LoD 2.2 | |
| 3338 | 1490 | 111,874 | 153,834 | 1251×1253 |
| 3339 | 3347 | 269,560 | 391,854 | 1270×1302 |
| 3340 | 3288 | 331,432 | 567,140 | 1286×1321 |
| 3341 | 568 | 57,654 | 91,798 | 1189×1275 |
| 3342 | 2597 | 241,951 | 363,118 | 1283×1264 |
| 3344 | 1401 | 200,229 | 502,830 | 1339×1354 |
| 3345 | 983 | 117,122 | 270,514 | 1701×1273 |
| 6211 | 1151 | 117,200 | 201,096 | 634×639 |
| 6212 | 825 | 80,748 | 128,794 | 635×576 |
| 6213 | 708 | 79,964 | 156,222 | 669×619 |
| 6214 | 1056 | 114,448 | 200,296 | 644×650 |
| 6215 | 1079 | 98,149 | 151,314 | 644×599 |
| 6216 | 1379 | 146,891 | 216,914 | 638×621 |
| 6217 | 902 | 103,856 | 166,366 | 659×624 |
| 6218 | 1065 | 96,312 | 152,625 | 659×602 |
| Total | 21,893 | 2,167,390 | 3,714,715 | 3759×3794 |

Number of faces correspond to the triangles

Table 2 Experimental results

| Tile | PGN (Points) | BVH time (seconds) | Duration (seconds) | D/M (seconds) | D/M/100k Pts (seconds) |
|------|--------------|--------------------|--------------------|---------------|------------------------|
| 3338 | 836,540 | 0.311 | 619.091 | 2.056 | 0.246 |
| 3339 | 2,449,172 | 0.987 | 5189.92 | 17.242 | 0.704 |
| 3340 | 3,534,632 | 1.662 | 5845.02 | 19.419 | 0.549 |
| 3341 | 603253 | 0.235 | 448.67 | 1.491 | 0.247 |
| 3342 | 2,367,691 | 0.946 | 4446.05 | 14.771 | 0.624 |
| 3344 | 3,904,597 | 1.292 | 5800.20 | 19.536 | 0.500 |
| 3345 | 2,022,331 | 0.629 | 1748.39 | 5.809 | 0.287 |
| 6211 | 1,084,487 | 0.436 | 2143.96 | 7.123 | 0.657 |
| 6212 | 780,436 | 0.249 | 1030.72 | 3.424 | 0.439 |
| 6213 | 925,722 | 0.315 | 916.196 | 3.044 | 0.329 |
| 6214 | 1,086,824 | 0.439 | 2091.39 | 6.948 | 0.639 |
| 6215 | 847,248 | 0.296 | 1482.22 | 4.924 | 0.581 |
| 6216 | 948,222 | 0.504 | 2042.10 | 6.784 | 0.715 |
| 6217 | 709,075 | 0.346 | 675.816 | 2.245 | 0.317 |
| 6218 | 904,768 | 0.306 | 1098.04 | 3.648 | 0.403 |

Note PGN represent point grid number, Duration represent the total simulation time, D/M represent simulation time for a single moment, D/M/100k Pts represent simulation time for a single moment per 100k grid points

Table 3 Experiment results for tile 3340

| Tile | PGN (Points) | BVH time (seconds) | Duration (seconds) | D/M (seconds) | D/M/100k Pts (seconds) |
|--------------------------|--------------|--------------------|--------------------|---------------|------------------------|
| 3340 | 3,534,632 | 1.662 | 5845.02 | 19.419 | 0.549 |
| 3340 (with buffer tiles) | 3,534,632 | 9.036 | 7401.05 | 24.588 | 0.630 |
| 3340 (whole year) | 3,534,632 | 1.519 | 84812.10 | 18.931 | 0.536 |

Columns and values definitions are the same as the ones in Table 2

aimed to test the scalability of the BVH method. Although the results already indicate that the higher the number of buildings, the longer the duration of the calculation. We have not tested the use of surrounding tiles as obstacles for the location of interest and based on this to analyse whether an increased number of buildings would significantly affect computational costs for the same set of rays. Additionally, this configuration facilitated more accurate shadow calculations for the edge buildings of tile 3340. Table 3 shows the preliminary results of the calculations, the first insights are promising and cover an interesting aspect to deal with adjacent tiles.

Additionally, we performed hourly shadow calculations for each day throughout the entire year 2023, to address the accuracy loss resulting from the bi-monthly time sampling. The hourly calculation facilitates the evaluation of the method's stability at a higher temporal scale.

5 Discussion

The experimental results demonstrate that the proposed method is scalable for computing shadows in urban areas of varying sizes. Without any parallel processing, the average computation time for calculating shadows in a city region, with a spatial extent of $1.3 \text{ km} \times 1.3 \text{ km}$ and thousands of buildings with complex structures at a specific moment, ranges from 1 to 20 s. Based on the theory of $O(\log n)$ for calculating the time complexity of a single ray, adding surrounding shadow-casting surfaces does not significantly increase the computation time for a single ray. Table 3 shows the results for tile 3340 with the inclusion of surrounding tiles in a lower LoD leading to an increase in the total computational time in $\sim 26\%$.

Furthermore, referring to Table 2, the computation time for a single ray does not exhibit a clear positive correlation with the number of shadow-casting surfaces in different tiles. The computation time per 100k grid points fluctuates within the range of 0.2–0.7s. This behaviour might be attributed to the overlapping bounding volume hierarchies. In BVH, the bounding boxes of different BVH nodes may overlap, causing rays to intersect with both child nodes separately. This overlapping of bounding boxes is scene-specific, and thus, the observed results do not violate the $O(\log n)$ theoretical time complexity. In conclusion, employing BVH for partitioning shadow-casting surfaces is an efficient and scalable method for shadow computation.

As a vector tracing method, this approach avoids making assumptions about shadow projections, such as filtering surrounding buildings based on a fixed ratio of shadow projection distance to building height. Through the use of Bounding Volume Hierarchy, this method circumvents the need for such assumptions and accurately handles extreme scenarios in urban environments, including extremely tall buildings, without incurring additional computational overhead. It also implies that the method can also effectively addresses shadow computations in mountainous urban areas by incorporating DTM (Digital Terrain Model) into the BVH. In addition, our sampling strategy ensures the shadowing calculation for small surfaces such as corners and is therefore also potentially suitable for indoor environments.

Furthermore, the current implementation does not utilize parallel processing, indicating significant room for improvement in computational speed. The proposed shadow computation method for urban-scale scenarios exhibits high parallelizability since intersection tests for each ray are independent. With efficient utilization of modern GPU cores (can be as many as several thousands cores in consumer level devices), taking into account the relatively lower single-core performance of GPUs compared to CPUs, the method has the potential to achieve a several-hundred-fold speedup. Moreover, leveraging the specialized RTX core in Nvidia GPUs, specifically designed to accelerate ray-triangle intersection tests, could further enhance the performance of the proposed method after adapting to the RTX core.

Additionally, the current experiment does not utilize semantic information of buildings, it is easily achievable in future code improvements. First use is the distinction between the input obstacles and target elements, the former correspond to all features in the input data-set and the latter correspond to building features and their

corresponding roof and wall surfaces. By recording ray-triangle intersection information, it becomes possible to analyze which buildings occlude specific structures the most severely. Semantics can also facilitate user selection of target computation areas or specific urban objects of interest for calculations i.e., buildings, trees, city furniture, etc.

One limitation of this method is its lack of consideration for diffuse and reflection of light rays. This may restrict its application in accurately calculating solar irradiance.

We expect our implementation to be accurate in complex urban environments; BVH accelerates the intersection tests without relying on any assumption (like buffer distance).

6 Conclusions and Future Work

This work is part of an on-going research about the development of a solar irradiance simulation tool based on semantic 3D city models. Our proposed method works for the calculation of shadows using semantic 3D city models and it is an improvement of existing ones since we use the vector tracing method without relying on culling assumptions. We include the Nightside filtering to optimise the computation to remove all surfaces that their orientation points to the opposite to the sun location.

Future versions of the tool will support CityGML encoding formats (CityGML, CityJSON and the 3DCityDB) to expand its usability. We plan to include parallel computing to speed up the computational time since the steps our method are independent so they can be execute simultaneously.

References

- Agugiario G, González FGG, Cavallo R (2020) The city of tomorrow from ...the data of today. *ISPRS Int J Geo-Inf* 9(9). ISSN: 22209964. <https://doi.org/10.3390/ijgi9090554>
- Biljecki F, Ledoux H, Stoter J (2016) An improved LOD specification for 3D building models. *Comput, Environ Urban Syst* 59:25–37. ISSN: 01989715. <https://doi.org/10.1016/j.compenvurbsys.2016.04.005>
- Calcabrini A, Ziar H, Isabella O, Zeman M (2019) A simplified skyline based method for estimating the annual solar energy potential in urban environments. *Nat Energy* 4(3):206–215. ISSN:2058-7546. <https://doi.org/10.1038/s41560-018-0318-6>. [Online]. Available: <https://www.nature.com/articles/s41560-018-0318-6> (visited on 21 Feb 2023)
- de Sá BA, Dezuo T, Ohf D (2022) Shadow modelling algorithm for photovoltaic systems: extended analysis and simulation. *J Control, Autom Electr Syst* 33(5):1507–1518. ISSN: 2195-3899. <https://doi.org/10.1007/s40313-022-00905-2>. [Online]. Available: <https://doi.org/10.1007/s40313-022-00905-2>
- European Council (2023) Council and parliament reach provisional deal on renewable energy directive. [Online]. Available: www.consilium.europa.eu/en/press/press-releases/2023/03/30/council-andparliament-reach-provisional-deal-on-renewable-energy-directive/ (visited on 29 May 2023)

- Fu P, Rich PM (1 Dec 2002) A geometric solar radiation model with applications in agriculture and forestry. *Comput Electr Agric* 37(1):25–35. ISSN: 0168-1699. [https://doi.org/10.1016/S0168-1699\(02\)00115-1](https://doi.org/10.1016/S0168-1699(02)00115-1). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169902001151> (visited on 1 June 2023)
- Hofierka J, Zlocha M (2012) A new 3-D solar radiation model for 3-D city models. *Trans GIS* 16(5):681–690. ISSN: 1467-9671. <https://doi.org/10.1111/j.1467-9671.2012.01337.x>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9671.2012.01337.x> (visited on 1 June 2023)
- Huang Y, Chen Z, Wu B et al (2015) Estimating roof solar energy potential in the downtown area using a GPU-accelerated solar radiation model and airborne LiDAR data. *Remote Sens* 7(12):17212–17233. Number: 12, Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2072-4292. <https://doi.org/10.3390/rs71215877>. [Online]. Available: <https://mdpi.com/2072-4292/7/12/15877> (visited on 1 June 2023)
- Hurkmans R (2022) Efficient solar potential estimation of 3D buildings: 3D BAG as use case. MSc Thesis, Delft University of Technology, Delft, NL. [Online]. Available: <http://resolver.tudelft.nl/uuid:0d6fe201-5676-48c0-850f-942d1ee9a353>
- iea (2022) Outlook for energy demand—world energy outlook 2022—analysis. Int Energy Agency. [Online]. Available: <https://iea.org/reports/world-energy-outlook-2022/outlook-for-energydemand> (visited on 29 May 2023)
- Jones NL, Greenberg DP (2011) Fast computation of incident solar radiation from preliminary to final building design. Presented at the building simulation 2011, series. Building simulation, vol 12. IBPSA, pp 577–584. <https://doi.org/10.26868/25222708.2011.1271>. [Online]. Available: <https://publications.ibpsa.org/conference/paper/?id=bs20111271> (visited on 02 June 2023)
- Lan H, Gou Z, Xie X (2021) A simplified evaluation method of rooftop solar energy potential based on image semantic segmentation of urban streetscapes. *Solar Energy* 230:912–924. ISSN: 0038-092X. <https://doi.org/10.1016/j.solener.2021.10.085>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0038092X2100949X> (visited on 21 Feb 2023)
- Ledoux H, Arroyo Ohori K, Peters R, Pronk M (2022) Computational modelling of terrains. v0.9.2.1. [Online]. Available: <https://tudelft3d.github.io/terrainbook/> (visited on 18 May 2023)
- Liang J, Gong J, Li W, Ibrahim AN (3 April 2014) A visualization-oriented 3D method for efficient computation of urban solar radiation based on 3D-2D surface mapping. *Int J Geogr Inf Sci* 28(4):780–798. ISSN: 1365-8816, 1362-3087. <https://doi.org/10.1080/13658816.2014.880168>. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/13658816.2014.880168> (visited on 21 Feb 2023)
- Liang J, Gong J, Xie X, Sun J (1 Sept 2020) Solar3D: an open-source tool for estimating solar radiation in urban environments. *ISPRS Int J Geo-Inf* 9(9):524. ISSN: 2220-9964. <https://doi.org/10.3390/ijgi9090524>. [Online]. Available: <https://mdpi.com/2220-9964/9/9/524> (visited on 21 Feb 2023)
- Lukač N, Žalik B (2013) GPU-based roofs solar potential estimation using LiDAR data. *Comput Geosci* 52:34–41. ISSN: 0098-3004. <https://doi.org/10.1016/j.cageo.2012.10.010>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0098300412003500> (visited on 1 June 2023)
- Meister D, Ogaki S, Benthin C, Doyle M, Guthe M, Bittner J (2021) A survey on bounding volume hierarchies for ray tracing. *Comput Graph Forum* 40:683–712. <https://doi.org/10.1111/cgf.142662>
- Meister D, Ogaki S, Benthin C, Doyle MJ, Guthe M, Bittner J (2021) A survey on bounding volume hierarchies for ray tracing. *Comput Graph Forum* 40(2):683–712. ISSN: 1467-8659. <https://doi.org/10.1111/cgf.142662>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.142662> (visited on 15 May 2023)
- Palme M, Privitera R, Rosa DL (2020) The shading effects of green infrastructure in private residential areas: building performance simulation to support urban planning. *Energy Build* 229:110–531. ISSN: 0378-7788. <https://doi.org/10.1016/j.enbuild.2020.110531>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0378778820318697>

- Perera ATD, Coccolo S, Scartezzini J-L (28 Nov 2019) The influence of urban form on the grid integration of renewable energy technologies and distributed energy systems. *Sci Rep* 9(1):17–756. Number: 1, Publisher: Nature Publishing Group, ISSN: 2045-2322. <https://doi.org/10.1038/s41598-019-53653-w>. [Online]. Available: <https://nature.com/articles/s41598-019-53653-w> (visited on 29 May 2023)
- Peters R, Dukai B, Vitalis S, van Liempt J, Stoter J (2022) Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of The Netherlands, English. <https://doi.org/10.14358/PERS.21-00032R2>
- Ray Tracing in One Weekend Book Series (2020) [Online]. Available: <https://github.com/RayTracing/raytracing.github.io> (visited on 16 May 2023)
- Robledo J, Leloux J, Lorenzo E, Gueymard CA (2019) From video games to solar energy: 3D shading simulation for PV using GPU. *Solar Energy* 193:962–980. ISSN: 0038-092X. <https://doi.org/10.1016/j.solener.2019.09.041>. [Online]. Available: <https://sciencedirect.com/science/article/pii/S0038092X19309168> (visited on 02 June 2023)
- Shukla PR, Skea J, Reisinger A, Slade R, Fradera R, Pathak M (eds) (2022) Climate change 2022 mitigation of climate change, ISBN: 978-92-9169-160-9. [Online]. Available: <https://www.ipcc.ch/report/ar6/wg3/>
- Stendardo N, Desthieux G, Abdennadher N, Gallinelli P (2020) GPUenabled shadow casting for solar potential estimation in large urban areas. application to the solar cadaster of greater Geneva. *Appl Sci* 10(15):5361. ISSN: 2076-3417. <https://doi.org/10.3390/app10155361>. [Online]. Available: <https://www.mdpi.com/2076-3417/10/15/5361> (visited on 21 Feb 2023)
- Tabik S, Villegas A, Zapata EL, Romero LF (2013) Optimal tilt and orientation maps: a multi-algorithm approach for heterogeneous multicore-GPU systems. *J Supercomputing* 66(1):135–147. ISSN: 1573-0484. <https://doi.org/10.1007/s11227-013-0891-1>. [Online]. Available: <https://doi.org/10.1007/s11227-013-0891-1> (visited on 1 June 2023)
- UN. Population Division (2018) The world's cities in 2018, United Nations, New York, USA, Technology Report. ISBN: 978-92-1-151549-7, p 34. [On-line]. Available: <https://digitallibrary.un.org/record/3799524>
- US Energy Information Administration (2023) Frequently asked questions (FAQs)—U.S. energy information administration (EIA). [Online]. Available: <https://eia.gov/tools/faqs/faq.php> (visited on 29 May 2023)
- Viana-Fons J, González-Maciá J, Payá J (2020) Development and validation in a 2D-GIS environment of a 3D shadow cast vector-based model on arbitrarily orientated and tilted surfaces. *Energy Build* 224:110–258. ISSN: 03787788. <https://doi.org/10.1016/j.enbuild.2020.110258>. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0378778820303777> (visited on 21 Feb 2023)
- Wang X, Zhang X, Zhu S et al (2023) A novel and efficient method for calculating beam shadows on exterior surfaces of buildings in dense urban contexts. *Build Environ* 229:109–937. ISSN: 0360-1323. <https://doi.org/10.1016/j.buildenv.2022.109937>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360132322011672> (visited on 26 Feb 2023)
- Zhou Y, Verkou M, Zeman M, Ziar H, Isabella O (2022) A comprehensive workflow for high resolution 3D solar photovoltaic potential mapping in dense urban environment: a case study on campus of Delft University of technology. *Solar RRL* 6(5):2100–478. ISSN: 2367-198X, 2367-198X. <https://doi.org/10.1002/solr.202100478>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1002/solr.202100478> (visited on 21 Feb 2023)
- Zhu R, Wong MS, You L et al (2020) The effect of urban morphology on the solar capacity of three-dimensional cities. *Renew Energy* 153:1111–1126. ISSN: 0960-1481. <https://doi.org/10.1016/j.renene.2020.02.050>. [Online]. Available: <https://sciencedirect.com/science/article/pii/S0960148120302378> (visited on 21 Feb 2023)

Supporting Teleoperated Humanitarian Aid Missions with 3D Visualization Using Remote Sensing Data



Lucas Dominik Angermann, Magdalena Felicitas Halbgewachs,
and Konstanze Lechner

Abstract Natural disasters, conflicts and vulnerable supply chains pose challenges for humanitarian aid efforts. The AHEAD project, a collaboration between the German Aerospace Center (DLR) and the World Food Programme (WFP), aims to develop a comprehensive system for delivering relief supplies through remote-controlled trucks. This study explores the potential of a multimodal 3D situational awareness map that integrates remote sensing and other geo data. This advanced map could support teleoperated truck missions in challenging environments, e.g. for route planning, terrain analysis or evaluation purposes. A key focus of this work is the additional value of a 3D visualization compared to established 2D mapping applications. Structured interviews were conducted with end users, scientists and engineers in the project to identify their needs and requirements for a 3D situational awareness map. Based on the outcomes an exemplary 3D web application integrating geodata and crisis information from remote sensing, governmental and open sources was designed and implemented for test sites in Bavaria. In situ drone imagery was captured and added to increase the local situational awareness. During the operation, the live position of the truck was continuously transmitted to the application and displayed as 3D model. The created web-based application was very well received by the end users. Especially the integration of drone imagery draped on the derived surface model in combination with the available satellite data provided a high additional value by highlighting steep slopes or other blockages in the truck's path.

Keywords Humanitarian aid · Remote sensing · 3D visualization · Web application · Teleoperation

This article was selected based on the results of a double-blind review of an extended abstract.

L. D. Angermann (✉) · M. F. Halbgewachs · K. Lechner
German Remote Sensing Data Center (DFD), German Aerospace Center (DLR), 82234 Weßling,
Germany
e-mail: lucas.angermann@dlr.de

M. F. Halbgewachs
e-mail: Magdalena.Halbgewachs@dlr.de

K. Lechner
e-mail: Konstanze.Lechner@dlr.de

1 Introduction

Delivering much needed supplies in remote and harsh environments is a difficult and demanding task WFP (2020). At the same time, robust amphibious trucks make it possible to reach such hard to access locations and distribute goods, as can be seen in Fig. 1 WFP (2019).

For the World Food Programme (WFP), the delivery on land with trucks is much more cost effective, than the delivery with planes. In this way the delivery costs in South Sudan in 2019 could be reduced by 79 million US\$ WFP (2020). In 2022 the WFP reached a number of 5.6 million people in South Sudan with insufficient access to food WFP (2023). At the same time, drivers of such humanitarian aid trucks are risking their lives due to diseases like malaria or conflicts Prothom (2023); WFP (2020); WHO (2022). Therefore, remotely operated vehicles are a promising option for humanitarian organizations, in significantly reducing the risk for drivers in the field GAC (2020). Recent progress in the domain of semi-autonomous teleoperation has made it possible to conduct first driving experiments in real-world scenarios Kettwich et al. (2021). To help teleoperators in navigating the vehicle, this study analyzes the additional value of a multimodal 3D situational awareness map for path planning, terrain analysis or mission assessment.

Together with the WFP, the German Aerospace Center (DLR) initiated the Autonomous Humanitarian Emergency Aid Devices (AHEAD) project GAC (2021). In this project, the teleoperation mission is split in three phases: preparation,



Fig. 1 Delivery of supplies by a WFP truck WFP (2019)

operation, and evaluation. The proposed mapping application has the capability to support both path planners and truck drivers throughout all phases of the mission. Up until now, a conventional 2D mapping application has been utilized in the project as the standard tool for displaying and editing geodata such as path routes. Similarly, for most spatially related applications of the WFP, the standard visualization technique has been predominantly 2D WFP (2019). In the ongoing extension of the project, MaiSHU, a 3D application for the visualization was implemented.

1.1 Mission Concept

The overall aim of the AHEAD and MaiSHU projects is to be able to deliver humanitarian aid supplies in remote areas in South Sudan by teleoperated trucks to increase the amount of transported supplies and at the same time to make the delivery safer for the drivers. Technology from prior research and development in the fields of space exploration, earth observation, and robotics is used to achieve this goal and support the people on site. To improve the delivery of supplies, the teleoperation is coordinated by a global and local mission operation center. The objective of the global mission operation center (GMOC) is to suggest possible routes to the local team by evaluating the current regional situation, including e.g. up-to-date crisis information such as floods and data from previous missions. The local mission operation center (LMOC) is responsible for the actual teleoperation and therefore establishing and maintaining the remote connections to the trucks in the field. A robust internet connection enables seamless data exchange between the global and local center. In Fig. 2 the main tasks and interaction between the GMOC and LMOC is shown. In the end the developed framework is intended to be integrated in the vast infrastructure network of the WFP.

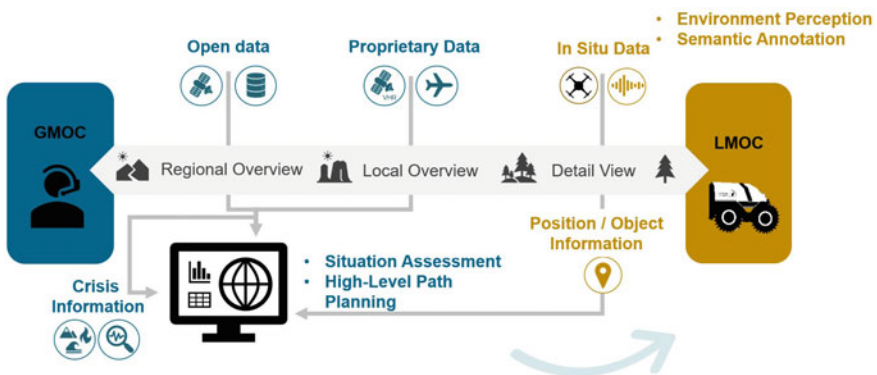


Fig. 2 Interaction between GMOC and LMOC

1.2 Crisis Information Based on Remote Sensing Data

At the Center for Satellite Based Crisis Information (ZKI[®]) of the DLR, multiple datasets for different crisis scenarios are created and processed GAC (2023). This includes constant fire and flood monitoring and event based rapid damage mapping Martinis et al. (2015); Nolde et al. (2020). The generated information is crucial for the successful execution of the teleoperated truck missions in the target area of South Sudan. Flooding information based in satellite data is a valuable asset for mission planning and can help to determine the most suitable route to be taken. Therefore, the ZKI[®] plays a key role in providing crisis information and coordinating the mission within the GMOC. Another important aspect of using remote sensing data in this scenario, is the global availability of most used data products like flood layers and elevation models. Access to global datasets is particularly crucial for remote and challenging-to-reach locations, as it allows for independence from local data acquisition. The regional overview provided by satellite imagery is also vital for path planning e.g. identifying alternative routes in case of unexpected path blockages due to natural disaster or conflicts.

1.3 Study Areas

Two test sites were used in this project. One is located on the premises of DLR in Oberpfaffenhofen, Germany. This location primarily serves for testing the overall technical system and the integration of different components. The second is a training site of the German Armed Forces located near Volkach, Germany. The site allows for field tests inside a controlled area and features multiple waterways and rough terrain. The change from one test site to the other poses a higher difficulty for the teleoperation of the vehicle due to the longer distances, water contact, obstacles in the radio coverage and steeper slopes. This is also quite important for the visualization aspect, as the terrain is more diverse and the LMOC is more dependent on the data depicted in the 3D situational awareness map application. Conversely, the first site offered a good opportunity to evaluate the communication of the truck with the applications inside the LMOC and GMOC without much additional infrastructure effort.

1.4 Prior Works and Studies

As part of the AHEAD project, a web based 2D mapping application was created, as can be seen in Fig. 3. Multiple datasets obtained from remote sensing, authorities and open sources of the test site in Bavaria, Germany were integrated. An orthophoto calculated from drone imagery depicts the test track shortly before a major demonstration of the project. As shown in Fig. 3, the 2D data in the mapping application primarily offers a birds-eye perspective from above. While this is valuable for regional



Fig. 3 AHEAD web based 2D mapping application

planning in the GMOC, teleoperators in the LMOC require more detailed terrain information to navigate the vehicle efficiently.

Leskens et al. (2017) also point out that a 3D view can provide a far better perception of the environment, especially for persons who are not used to interact with GIS software. This is also the case for the teleoperators, who are not used to GIS interfaces but are experts in the field. Additionally, a 3D representation depicts the surroundings of the transport vehicles more naturally Leskens et al. (2017). Nevertheless, Shepherd (2008) is pointing out, one must carefully choose when to use 3D or when a 2D representation may be better suited. The experience gained from using the 2D AHED application indicated that both mission centers have distinct sets of expectations and requirements for the mapping application. To analyse this further and to assess the need for a 3D visualisation, interviews with the project partners involved have been scheduled as part of the MaiSHU project extension. The insights gained from these interviews, along with the findings from previous 3D projects, will be incorporated in the design of the 3D multimodal awareness map application for the MaiSHU project. This implementation process ensures that the final result will be optimized to meet the specific requirements and preferences of the target group.

2 Methods

2.1 Structured Interviews

Direct conversations with users and experts are an effective approach in gaining valuable insights into their requirements and expectations for interacting with the application Abbott and McKinney (2012). In a first step, interview questions were created concentrating on the past user experience with the 2D AHEAD web map application. These questions were designed to gather insights and feedback

regarding the usability, effectiveness, and overall satisfaction of users with the previous application. By exploring their experiences and perspectives, valuable information can be obtained to identify strengths, weaknesses, and areas for improvement in the design and functionality of the new 3D multimodal awareness map application. The structured interviews were conducted with engineers, scientists and end users participating in the AHEAD and MaiSHU projects. The experts were selected by their different field of work in the project to ensure that all aspects during the mission cycle would be covered. The interviews were carried out online via video calls and summarized afterwards. During the interviews, various examples of 2D and 3D web applications created by the ZKI[®] were showcased to familiarize the participants with the current state of the art in web mapping applications. These demonstrations aimed to provide the interviewees with a comprehensive understanding of the capabilities and functionalities offered by modern web mapping applications. Following the presentations and discussions, the interview partners were asked to identify the essential tools or features that were vital for their respective roles in the ongoing project. They were also encouraged to suggest additional datasets or tools that could be helpful for the teleoperation mission, independent of their current feasibility for implementation. After the interviews were completed, the written protocols served as valuable resources for extracting and refining the user requirements based on the statements provided by the participants for the implementation of the new 3D multimodal situational awareness map.

2.2 Selection and Implementation of Geodata

In the beginning of the project, geo basis data was obtained for the defined test sites in Bavaria, Germany. The expert interviews later identified the specific datasets that were additionally required. Sources for the data are aerial, drone and satellite imagery, governmental providers and open source geo information portals. As the Bavarian Surveying and Mapping Authority (LDBV) made some of their high-quality three-dimensional data openly publicly available BSMFRI (2022), detailed terrain and Level of Detail (LoD) 2 buildings could be included in the project. Further, current near real-time crisis information derived by existing process chains from satellite data could also added as valuable information layers. In this case data from the ZKI[®] Fire Monitoring System was integrated GAC (2023); Nolde et al. (2020).

During the data selection it was kept in mind, that not all the datasets available in Germany would be also available in the target area of South Sudan. Therefore, globally available data products were also integrated, even if their quality or resolution were not as good as already collected local data sets. For example, a very detailed digital surface model (DSM) with a spatial resolution of 1m from the LDBV of the Volkach test site and a globally available DSM with 12m spatial resolution derived from the satellite mission TerraSAR TanDEM-X were both integrated to allow for a direct comparison. In this way the users can imagine, what the visualization experience would be for the South Sudan use case.

For the implementation of the demonstrator application commercial software from Esri was used, as this platform is also used by WFP which makes it later easier to transfer results and data to their systems WFP (2019). In an existing web framework, the geodata was integrated and accordingly visualized in a web map and 3D scene. During the process the performance of the data services and the restricted access to the application were also addressed since some of the datasets were quite large or contained restricted information. Furthermore, a 3D model of the truck was integrated in the application as a reference for its actual size and to serve as a visual guide. Moreover, the live position of the real truck was connected to the position of the feature marker. This position is determined by GNSS antennas on the truck, transferred to the LMOC, and then updated regularly in the GIS environment when the truck is moving.

2.3 Availability in Remote Locations

As presented here, the current application is implemented with high detailed local geodata and for easily accessible test sites in Germany, where additional data acquisition is unproblematic. There are sufficient sources for high quality geodata in high spatial and temporal resolution. However, the aim of the project is the implementation of a teleoperated truck mission in a rough environment without the data infrastructure which is present in Germany. As mentioned, we also collected and integrated not only the datasets with the highest resolution and quality for the test sites, but also data which is globally available and can therefore be used in the target area of South Sudan. This helps mission planners and truck operators to decide with data sets are crucial for their work and the intended use. As the LMOC is the local data hub for the teleoperation of the vehicles and the communication with the GMOC, it is planned to have a sufficient data bandwidth which is needed for the fluent web-based 3D visualization of the environment. In case of low bandwidth or no internet connection, the LMOC could still operate offline and the situational awareness map application could be accessed through ArcGIS Pro and locally stored data, though there may be limitations in the visualization of up to date data like satellite derived crisis information.

3 Results

During the interviews it became clear, that there is a high demand for a 3D visualization in the context of teleoperation. In some situations, a planar 2D representation of the environment is not sufficient enough for the live teleoperation use case. As example where the benefits of 3D visualization are evident is in the navigation across a river bank with challenging terrain. Here the teleoperators, who are not used to work

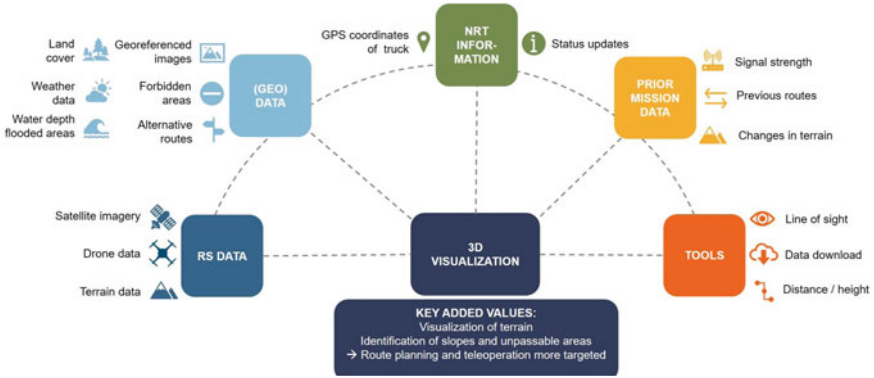


Fig. 4 User requirements derived from interviews



Fig. 5 Screenshot of the demo application. The terrain consists of a digital surface model with a draped orthophoto derived from aerial images together with LoD2 buildings

with GIS interfaces, can more intuitively navigate through the 3D scene and estimate more quickly the areas which should be avoided.

In Fig. 4 an overview of the identified requirements is shown. Especially tools, which allow the measurement of distances and areas in 3D were considered as helpful. The measurement tools inside the three-dimensional terrain visualization are also helping to identify areas along the proposed path where the gradient is too steep for the vehicle to pass safely. This kind of information is crucial for the teleoperators during the truck mission and should be emphasized in more detail in the visualization.

Another finding from the interviews was the need for current remote sensing data with both high spatial and temporal resolution. The interviewed experts had varying opinions regarding the necessary data resolution requirements. High spatial resolution satellite coverage is essential for regional planning purposes. Up-to-date



Fig. 6 Detailed view of the demo application and the 3D SHERP model

recordings are useful in detecting changes in the terrain surface, such as alterations in the course and extent of a river. However, for the local path planning during a mission, satellite imagery alone is not sufficient. The teleoperators stated, that they needed a higher spatial resolution for the assessment of a path route as could be provided by the shown globally available terrain datasets. As a suggested solution, an additional aerial or drone flight is needed on site directly before the start of the teleoperation to obtain a very high detailed insight in the situation on the ground and the collection of high-resolution data.

Based on the interview findings, a first prototype of the application was implemented, as can be seen in Fig. 5. The yellow dots represent georeferenced photographs, which were taken on site and added to the web scene. A more detailed view of the application is shown in Fig. 6. The selected and processed data could be integrated and displayed. The application was made easily accessible for the project participants by sharing a link to the web application.

Supplementary information, which could be shown in near real time (NRT) on the map, like the current truck position, it's speed, fuel gauge, or tire pressure was also considered as important for the visualization. This is not only important for the operators inside the LMOC, but also for the coordination of multiple operations in the GMOC. For example, the up-to-date fuel consumption could be aggregated from different trucks to estimate the overall demand, which could then be used to secure the fuel supply for the truck service camps in remote locations. Currently, only the truck's position has been integrated in the application, as the implementation of sensors for gathering additional truck information is still ongoing work in the research vehicle.

The planned extension of the application involves integrating prior mission data from the environmental sensors of the truck. Currently, these sensors are enclosed

in an own system and not yet available in compatible data format. Further research and development are necessary to ensure an efficient integration of this data source into the situational awareness map and enhance the existing remote sensing data.

4 Discussion and Outlook

This paper presented how a 3D representation of the on-site situation can support the teleoperation of trucks in harsh environments. The requirements for the demonstrator application were collected during conducted interviews. In a large project demonstration, the application was tested and presented to the end users. The feedback was overall positive. As the project is still ongoing, a final conclusion could not be made yet. In the end of the project, additional interviews with the end users are certainly necessary, to evaluate if the implemented solution meets the requirements. As suggested during the interviews, a stronger focus should be made in highlighting detected changes on the ground surface and derived slopes. Additional sensor information obtained by cameras and drones could be added to the 3D visualization. As the truck's onboard system calculates a local point cloud to identify near obstacles for the operators, a next step could be the integration and display of this data together with the already visualized environment in the teleoperator's systems. A proposed further enhancement was to link supplementary information of the truck, such as its current direction or fuel level gauge. When considering the currently available geodata for the target area in South Sudan, it is highly recommended to conduct drone flights directly before the start of a truck mission to improve the quality and spatial resolution of the data within the application. As the WFP is already operating a large drone fleet, this might be feasible in the future WFP (2023).

References

- Abbott M, McKinney J (2012) Understanding and applying research design. Wiley, Incorporated
- Bavarian state ministry of finance and regional identity: zahlreiche geobasisdaten der bayrischen vermessungsverwaltung künftig kostenfrei verfügbar (2022). <https://www.stmfh.bayern.de/internet/stmf/aktuelles/pressemitteilungen/24966/>. Accessed 18 July 2023
- German aerospace center: center for satellite based crisis information (ZKI) (2023). <https://www.dlr.de/eoc/en/desktopdefault.aspx/tabid-12797/#gallery/32034>. Accessed 18 July 2023
- German aerospace center: DLR robotic vehicles will support deliveries in difficult areas for the world food programme (2020). https://www.dlr.de/en/latest/news/2020/04/20201021_remote-controlled-access-in-crisis-regions. Accessed 18 July 2023
- German Aerospace Center: Projekt A.H.E.A.D. (2021). <https://projekt-ahead.de/projekt-a-h-e-a-d/>. Accessed 18 July 2023
- German aerospace center: ZKI fire monitoring system (2023). <https://services.zki.dlr.de/fire/>. Accessed 18 July 2023

- Kettwich C, Schrank A, Oehl M (2021) Teleoperation of highly automated vehicles in public transport: user-centered design of a human-machine interface for remote-operation and its expert usability evaluation. *Multimodal Technol Interact*. <https://doi.org/10.3390/mti5050026>
- Leskens J, Kehl C, Tutenel T, Kol T, Haan G, Stelling G, Eisemann E (2017) An interactive simulation and visualization tool for flood analysis usable for practitioners. *Mitig Adapt Strat Glob Change* 22(2):307–324. <https://doi.org/10.1007/s11027-015-9651-2>
- Martinis S, Kersten J, Twele A (2015) A fully automated TerraSAR-X based flood service. *ISPRS J Photogrammetry Remote Sens* 104:203–212. <https://doi.org/10.1016/j.isprsjprs.2014.07.014>
- Nolde M, Plank S, Riedlinger T (2020) An adaptive and extensible system for satellite-based, large scale burnt area monitoring in near-real time. *Remote Sens* 12:2162. <https://doi.org/10.3390/rs12132162>
- Prothom alo: world food programme halts operations in Sudan after 3 employees killed (2023). <https://en.prothomalo.com/international/africa/2ksxzca75q>. Accessed 18 July 2023
- Shepherd I (2008) *Travails in the third dimension: a critical evaluation of 3D geographical visualization*. Chichester England, Hoboken NJ, Wiley. <https://doi.org/10.1002/9780470987643.ch10>
- WFP drones broadening humanitarian horizons (2023). <https://drones.wfp.org/>. Accessed 18 July 2023
- World food programme: summary—2022 annual country report South Sudan (2023). <https://www.wfp.org/publications/annual-country-reports-south-sudan>. Accessed 18 July 2023
- World food programme: the humanitarian monster truck (2019). <https://www.wfp.org/stories/humanitarian-monster-truck>. Accessed 18 July 2023
- World food programme: WFP emergencies: GIS catalogue (2019). <https://docs.wfp.org/api/documents/31929a7ead2b4d5d92ad0a72fd496179/download/>. Accessed 18 July 2023
- World food programme: WFP supply chain annual report 2019 (2020). <https://www.wfp.org/publications/wfp-supply-chain-annual-report-2019>. Accessed 18 July 2023
- World health organization: world malaria report 2022 (2022). <https://www.who.int/teams/global-malaria-programme/reports/world-malaria-report-2022>. Accessed 18 July 2023

Solid Waste in the Virtual World: A Digital Twinning Approach for Waste Collection Planning



Iván Cárdenas , Mila Koeva , Calayde Davey , and Pirouz Nourian 

Abstract Solid waste (SW) management is a crucial challenge for urban sustainability. With global waste generation exceeding two billion metric tons every year, it poses various negative impacts, including health risks and environmental contamination. Despite not being a primary SDG, addressing SW management is essential for achieving several goals, as it relates to 12 out of 17 SDGs. South Africa, in particular, struggles with issues as large generation, and inadequate waste collection services. To address these challenges, a dynamic model incorporating real-time monitoring, optimized collection routes, and citizen participation is proposed. This study case involves a real-time citizen engagement method to locate, identify, and visualize SW containers and littering sites via open-source tool called Epicollect5 based on geospatial information. This can be considered as a first step towards a Digital Twin for SW management. In a three-day data collection campaign, a total of 1270 containers and 820 littering sites were identified. The result show inadequate distribution of containers in public spaces, highlighting the need for citizen involvement in reporting container information to achieve a comprehensive understanding of their distribution. Additionally, the concentration of litter reports in peripheral park areas emphasizes the importance of providing well-distributed containers and prompt maintenance. The inclusion of photographs in reports helps identify areas requiring immediate attention, while citizen participation mitigates challenges associated with location accuracy and resource requirements. Digital twinning, multi-stakeholder engagement, and citizen participation provide insights into waste container distribution and combating illegal dumping. This approach benefits lower-income countries with limited resources, improving SW management practices.

Keywords Solid waste · Digital twins · Citizen participation · South Africa

This article was selected based on the results of a double-blind review of an extended abstract.

I. Cárdenas · M. Koeva (✉) · P. Nourian
Faculty of Geo-Information Science and Earth Observation, University of Twente, Enschede,
Netherlands
e-mail: m.n.koeva@utwente.nl

C. Davey
University of Pretoria, Tshwane, South Africa

1 Introduction

Solid waste management is a crucial challenge for achieving city sustainability (Ismagilova et al. 2019). In 2020, it was estimated that around 2.24 billion metric tons of municipal solid waste were generated worldwide (Kaza et al. 2021). This waste increased later due to medical waste produced during the COVID-19 pandemic (Yousefi et al. 2021; Liang et al. 2021). Approximately 33% of the overall waste generated is not environmentally safe (Kaza et al. 2018). It has several negative impacts, including health risks, sewage system blockages, soil contamination, and potential disease vectors (Ziraba et al. 2016; Pervin et al. 2020; Sharma et al. 2018; USAID 2007). Despite not being included as a primary Sustainable Development Goal (SDG), addressing solid waste management is related to 12 of 17 SDGs, making it essential to achieve city sustainability (Rodić et al. 2017; Wilson et al. 2015).

In South Africa, 30.5 million tons of solid waste were generated in 2017, with only 34.5% being recycled and 11% not having adequate final disposal (Department of Environmental Affairs 2018; Department of Environment Forestry and Fisheries 2020). The country has an estimated generation of 1.48 kg/capita/day of solid waste, which is higher than the Sub-Saharan average and at similar levels to some countries in Europe and Central Asia (Kaza et al. 2018). One of the primary challenges in South Africa is reducing the waste disposed in landfills (Department of Environment Forestry and Fisheries 2020), which is hindered by littering, illegal dumping, lack of regular collection services, incomplete coverage, and historical spatial and service delivery inequalities (Polasi 2018; Polasi et al. 2020).

To improve solid waste management, a dynamic model that incorporates real-time monitoring, frequent collection route optimization, and active citizen participation is needed, as suggested by different authors (Ramson and Moni 2017; Karthik et al. 2021; Joshi et al. 2022; Chaudhari and Bhole 2018). However, this is difficult to be achieved (1) without understanding the local waste landscape and (2) without having a population active and aware of the waste issues.

Researchers have been investigating different methods for monitoring waste management. Some suggestions include designs for sensors to monitor solid waste containers, including ultrasonic sensors on container lids (Ramson and Moni 2017; Karthik et al. 2021; Joshi et al. 2022; Chaudhari and Bhole 2018; Mahajan et al. 2017), weight sensors at the bottom of containers (Rovetta et al. 2009), a combination of both (Vicentini et al. 2009; Ali et al. 2020), as well as infrared sensors (Singh et al. 2016) to detect fullness levels. However, many of these designs have only been tested at a prototype level, not in outdoor or large-scale conditions. While some studies have tested sensors outdoors, they have used operators for the containers or invited citizens to use them, which could create biases in the data. In the Netherlands, the city of Utrecht has successfully incorporated ultrasonic sensors, and daily rerouting based on fullness levels (Gemeente Utrecht 2021) but implementing these technologies in lower-income countries may be challenging.

The current research proposes a digital twinning approach to pilot the deployment of crowd-sourced processes in South Africa to improve solid waste management. It is

aimed to record and monitor the spatial distribution and characteristics of collection points and identify the location of illegal dumping sites in a near-real time manner.

2 Methods

This section details the characteristics of the study area and the methodology employed, including data collection and subsequent data processing and analysis.

2.1 Study Area

The study focused on the Hatfield and Hillcrest neighborhoods of the city of Tshwane, Greater Pretoria, in South Africa. The area comprises 9.45 km² surrounding the University of Pretoria (UP) main campus (see Fig. 1) with different land uses such as residential, institutional (embassies), commercial, agricultural, and educational. This area is part of the ongoing project of African Future Cities from the Department of Architecture in the Faculty of Engineering, Built Environment, and Information Technology of the University of Pretoria.

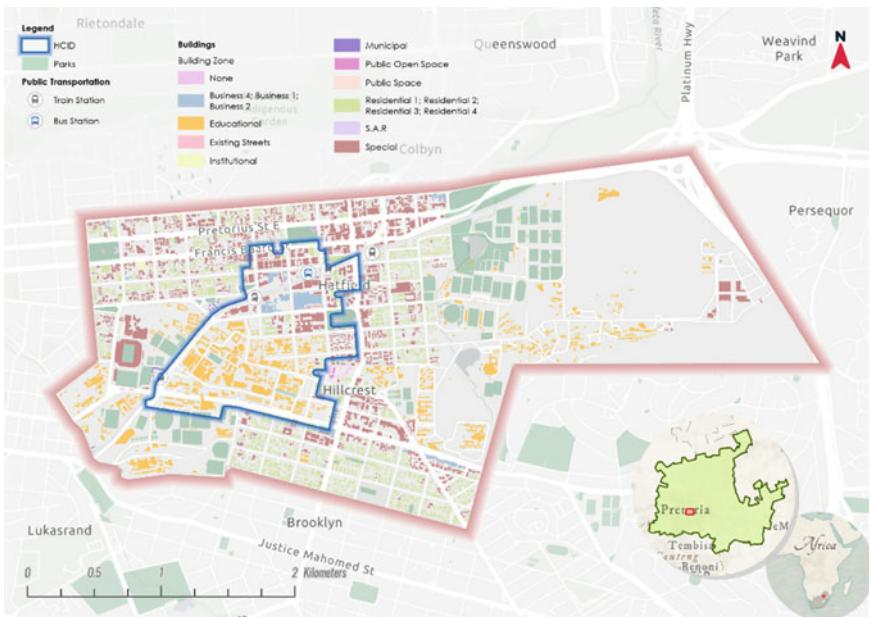


Fig. 1 Hatfield digital twin city study area

The area also includes the Hatfield City Improvement District (HCID). This non-profit and private organization performs corporate governance of the area. It is funded by a taxpayer's property levy collected by the municipality and transferred to the HCID for operation, providing additional services such as cleaning and maintaining public spaces, private security, and urban embellishment.

2.2 Methodology

The primary data collection relied on the open-source Epicollect5 tool (Aanensen et al. 2014), an application that is both freely accessible and highly adaptable. It works on smartphones for iOS and Android and is designed for on-field information gathering. The app does not require a constant network connection and can store records on the device and later be updated to the cloud when a Wi-Fi connection is available. This is a particularly important function for South African citizens since constant internet connection might be a challenge.

On this app, a survey including the questions shown in Table 1 created focusing on container capacity, dimension, and littering reporting in three levels of severity (minimal, moderate, and severe), engaging students, as citizen pilots, in the Hatfield and Brooklyn neighborhoods of the City of Tshwane. By involving students in this pilot initiative, the feasibility and effectiveness of crowd-sourced data collection in addressing solid waste management challenges is assessed.

The data was collected, in a three-day period, between 28 February and 02 March, by 15 bachelor's students in their final year of the Architecture program at the University of Pretoria. The survey was sent to the students with a detailed guide on downloading and using Epicollect5 on a mobile phone (Fig. 2).

Likewise, an introductory session explaining the tool's usage was delivered to students, and they were asked to keep an accuracy of at least 5 m at the moment of collection. The records with an accuracy of less than 20 m were excluded from the final dataset as a larger distance would overlap with other containers, and data quality could not be assured. The obtained data was extracted using Epicollect5 API through the PyEpiCollect library developed by Principe and Masera (2020), transformed using Geopandas, and stored as a GeoJSON file and ESRI's Shapefile.

Data was geolocated and immediately visualized in a 2D GIS environment to monitor the data collection process and constantly communicate with students. After data was collected, a 3D representation of the study area was created in a GIS online environment to visualize the results and have a monitoring tool using 3D buildings as context for navigation. To reach better realism, the solid waste containers were represented according to their geometries and scaled according to their volume to increase the contrast against the built environment. The littering reports were visualized using point 3D objects represented according to the severity of the report: minimal: one can of soda; moderate: a trash bag; severe: a pile of trash bags (see Fig. 3).

Table 1 Survey design for solid waste containers and littering identification

| Category | Question | Data type | Domains | Required | Optional |
|--|---|-----------|----------------------|----------|----------|
| Basic information | Report date | Date | | X | |
| | Report time | Date | | X | |
| | Where is the report located? | Geopoint | | X | |
| Report type | What kind of report do you want to make? | String | Litter report | X | |
| | | | Register a trash bin | | |
| | | | Other report | | |
| Trash bin information | What is the status of the container | Boolean | Broken | X | |
| | | | Non-broken | | |
| | Is the bin movable? | Boolean | Movable | X | |
| | | | Fixed (static) | | |
| | Height (in m) | Double | | | X |
| | Radius in m (if circle) | Double | | | X |
| | Length in m (if rectangular) | Double | | | X |
| | Width in m (if rectangular) | Double | | | X |
| | Can you estimate the capacity (in Cubic meters) | Double | | | X |
| Please include a photograph of the container | Photo | | | X | |
| Comments | String | | | X | |
| Litter report | Severity (How much trash is there) | Integer | Minimal (1) | X | |
| | | | Moderate (3) | | |
| | | | Severe (5) | | |
| | Please include a photograph of the container | Photo | | | X |
| Comments | String | | | X | |

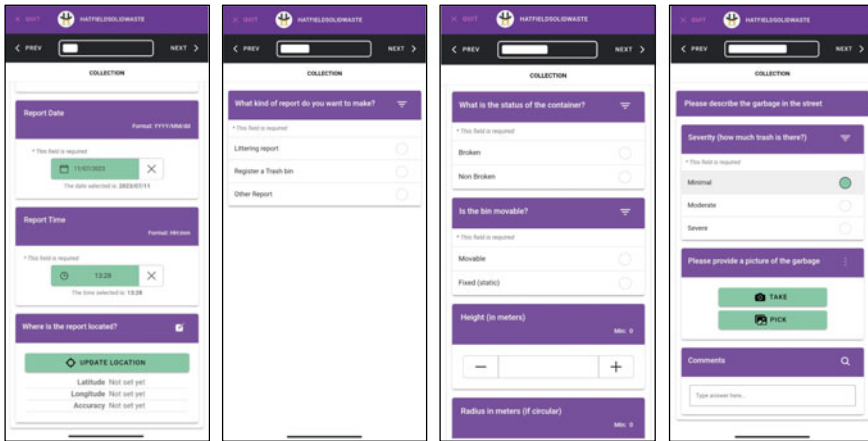


Fig. 2 Epicollect5 Mobile collection app screenshots

The visualization includes various features to help viewers navigate and interact with the data, such as a 240L standard bin 3D model used to visualize the containers and color-coded for broken (red) and good state (green) containers. The survey results were analyzed in their relation to closeness to transport nodes and land use.

3 Results

The data collection results showed 2.236 reports. Of them, 3.04% were invalid, i.e., without coordinates or points without information, and 3.76% were inaccurate. A total of 1.270 containers were identified, with 4.16% reported as broken (see Figs. 4 and 5). Also, 136 ashtrays were identified inside the main campus and only one outside.

Container volumes vary between 28 L and 10.58 m³ on 67 different types of containers (Fig. 6). The ones of 118 L are the most frequent, with 426 (33.54%) of the total recorded. This type of container corresponds to a standardized concrete or metal container (see Fig. 7a and b). Likewise, the containers 240L correspond to standard plastic movable containers (see Fig. 7c), which are distributed to each home by the municipality as they help load waste to the waste collection trucks. The identified containers also include 26 dumpsters known as skips (see Fig. 7d). These skips can be attached to collectors facilitating collection. The skips are loaded one at a time, compared to plastic containers which are collected in several bins in the same vehicle.

Near-real-time monitoring of the reports allowed us to understand that containers were mainly inside the UP campus and the HCID. On the live monitoring of the 02 March, it was possible to observe containers whose distribution obeyed to streets

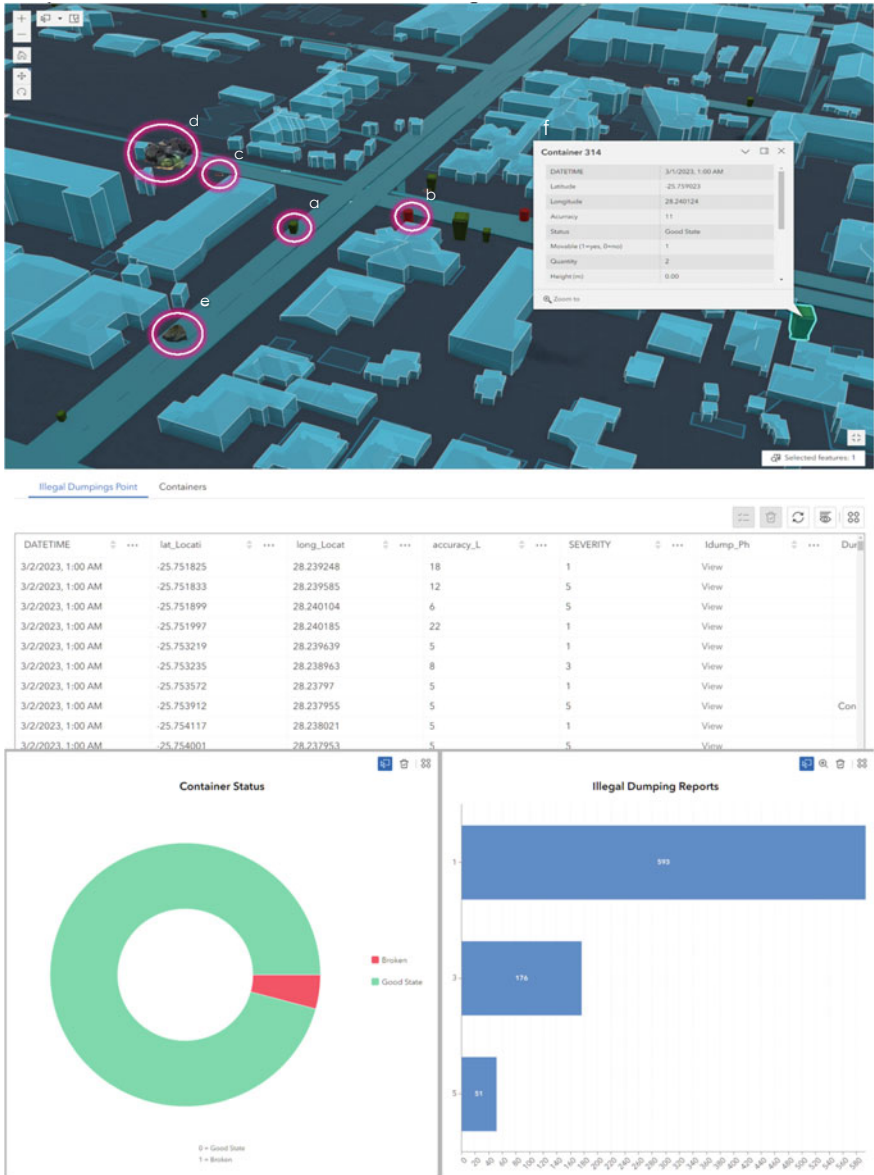


Fig. 3 Visualization of Containers and Illegal dumping sites. **a** Good state container. **b** broken container. **c** Minimal severity illegal dumping report. **d** Moderate severity illegal dumping report. **e** Severe illegal dumping report. **f** Container query attributes

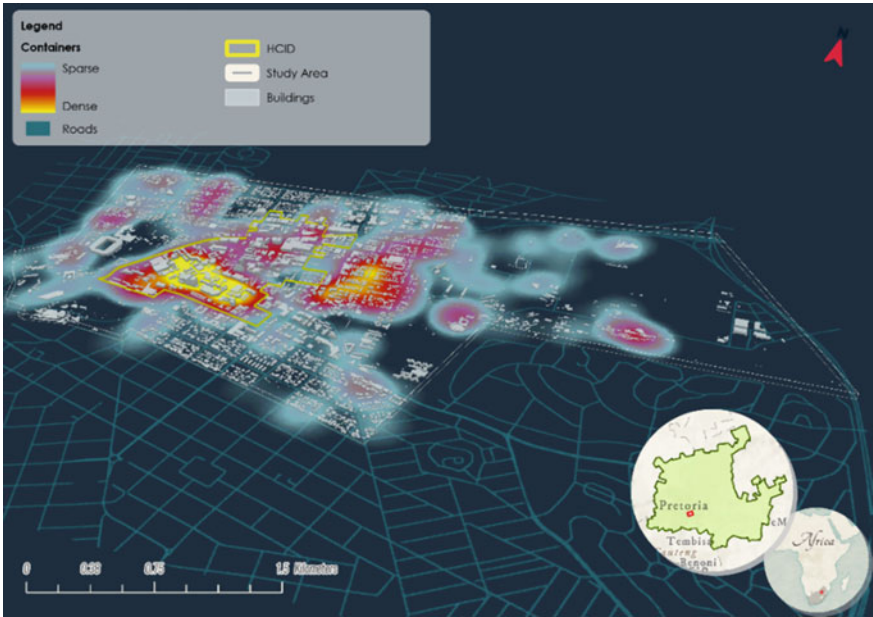


Fig. 4 Containers distribution—heatmap—in study area

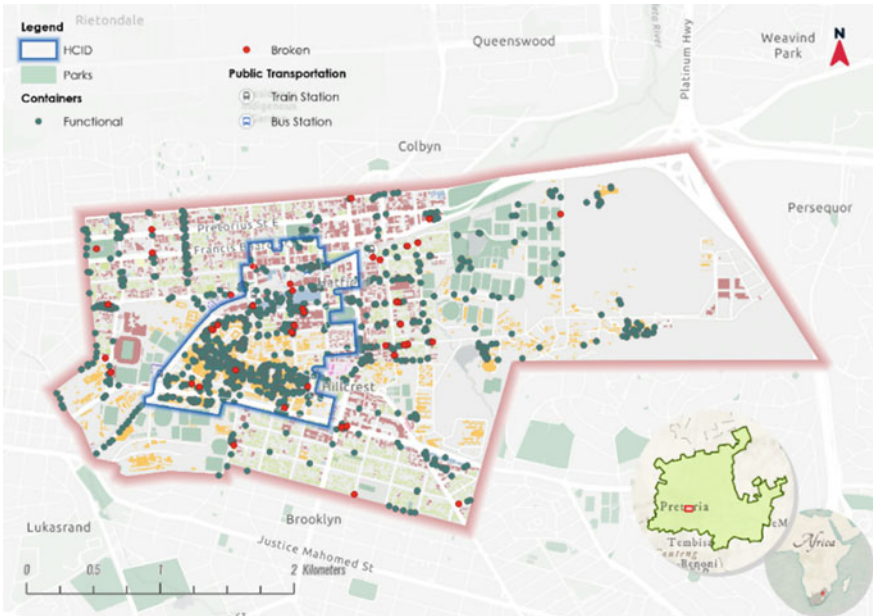


Fig. 5 Container's location and functionality (Broken—functional)

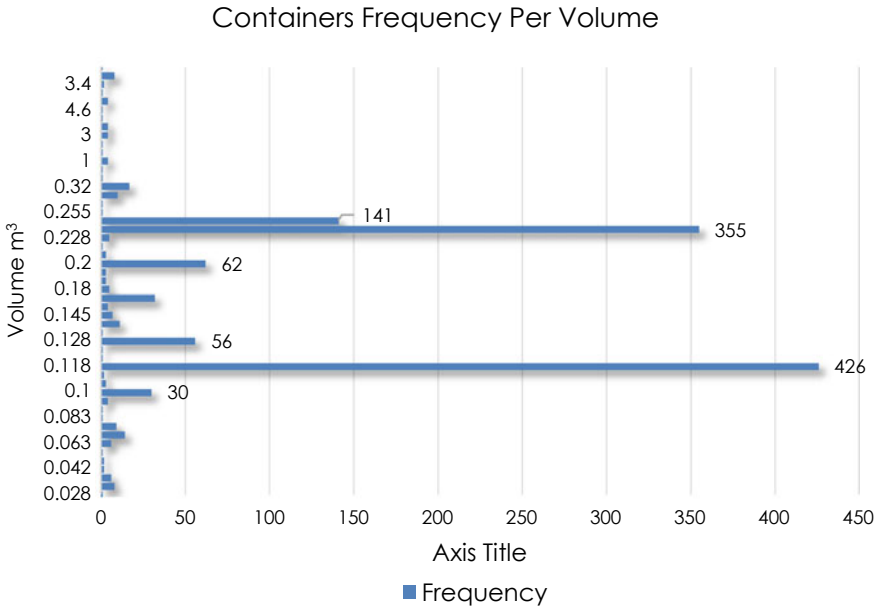


Fig. 6 Containers’ volume frequency distribution

that had their assigned collection on the same day that data was collected, which relates to residential buildings and residents locating their containers on the curbside (Fig. 5).

On the illegal dumping and littering side, 820 reports were made and distributed in 593 minimal (72.3%), 176 moderate (21.5%), and 51 severe reports (6.2%) (see Fig. 8). The photographs taken by students to report on the illegal dumping show that areas underneath the trees are being used as the frequent place for littering, which adds up to the biowaste of tree maintenance and the natural accumulation of bark and leaves (see Fig. 9).



Fig. 7 Main observed type of containers. **a** Concrete 0.118 m³ **b** Metal 0.118 m³ **c** Plastic 0.240 m³ **d** Skip 4 m³

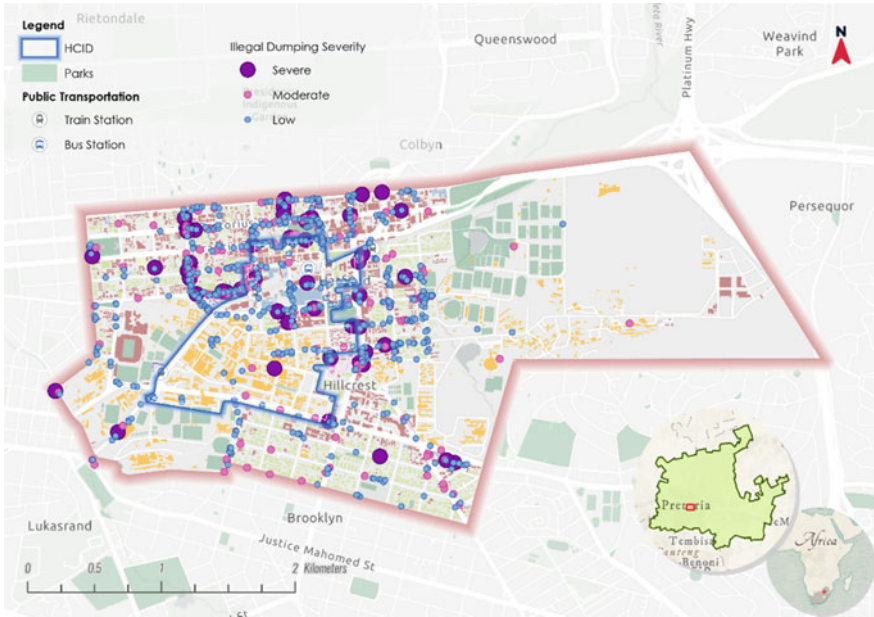


Fig. 8 Illegal dumping severity and distribution



Fig. 9 Illegal dumping reports. **a** Event residues inside UP campus. **b** Car bumper disposed under a tree. **c** Littering under tree. **d** Biowaste, littering, and illegal dumping under trees

4 Discussion

The designed survey on Epicollect5 works as a tool for reporting waste containers, illegal dumping, and littering. It provides a fast method of information collection where citizens can engage rapidly and keep track of the waste system in public and private spaces. This allows anyone with access to a mobile device to inform authorities of waste status and help make decisions in improving Urban solid waste management.

The data showed that public spaces do not have an even distribution of solid waste containers. This is seen in areas where no record of solid waste containers was documented. They correspond to exclusively residential areas in the North and South of the study area, outside the HCID, where containers are located in private areas, not in public spaces. Therefore, its information was not collected, which is a limitation of this research. To have a complete understanding of the distribution of waste containers, it is necessary to include the residents of the area and ask them to report their containers, capacity, and frequency on which they refuse waste. Collecting information on the type and status of containers can help the city and HCID management, where maintenance is required, and progressively standardize the type of containers to ease recognition and collection of waste.

The main concentration of the litter reports was in the periphery of the parks, green areas, and playgrounds, such as Springbok Park and Hatfield Playpark, suggesting that the recreational and resting areas are common places for illegal dumping and require larger containers. Likewise, these parks show containers in the center of them and no containers in the periphery. Areas of the parks surrounding the container do not have reports of litter, which shows the importance of providing sufficient and well-distributed containers. Additionally, reports that include photographs help understand the process of waste accumulation. For instance, the pictures in Fig. 9 indicate that maintenance of green areas requires a rapid response from the collection company to avoid accumulating and attracting other litter and disease vectors.

The research faced some adversities as the proposed data collection method did not consider regional numerical format. Some of the collecting devices could not include decimal separators (iOS), and data was informed in the comments section of the report. This generates anomalies in the monitoring as geometrical properties could not be visualized on the spot.

Contrary to assumptions, students found the study area to be cleaner than expected and recognized the importance of container distribution and governance on their waste flows. The littering reports did not consider the type of waste thrown in the inappropriate location. Nonetheless, students reported biowaste and construction waste on the streets. The live information retrieved from these waste flows can also be used by composting farms and circular economy companies that take advantage of these residues and reduce the amount of waste that goes to landfill.

Although the monitoring was designed to be in a 3D environment to create a closer to the real-life visualisation, it showed no advantages compared to the 2D environment monitoring system. Nonetheless, 3D model can help citizens and non-technical users provide a better context, improve the situation's mental image, and provide an immersive decision-making medium (Herbert and Chen 2015; Bouzguenda et al. 2022).

By involving citizen participation, this method reduces challenges, such as location accuracy, high resource requirements, and disagreement on labeling, identified in Artificial Intelligence computer vision detection research (Moral et al. 2022). It also confirms the importance of citizen testimony in mapping solid waste (Al-Joburi 2018). The real-time monitoring helps address the randomness of minor severity littering for improved solid waste management, including multiple stakeholders.

5 Conclusion

The research concluded that digital twinning, multi-stakeholder engagement, and citizen participation could provide valuable insights into the distribution of solid waste containers and the occurrence of illegal dumping and littering. It can be a hybrid and collective approach for addressing solid waste management challenges in lower-income countries without large financial and technological capacity.

By developing digital counterparts of waste management infrastructure and mapping out their spatial distribution, policymakers and stakeholders comprehensively understand the current state of solid waste container placement. This knowledge serves as a foundation for evidence-based decision-making and targeted interventions. Through collective efforts and integration of technology and community engagement, improved solid waste management can be achieved, even in resource-constrained settings.

6 Further Research

The next steps in developing a Digital Twin for solid waste management include calculating solid waste production per building, considering their total floor area and their relationship with each container. Waste collection managers can also use the frequent report of container saturation status and littering locations to optimize the collection system via Capacitated Vehicle Routing Problem (CVRP). Further research could explore the effectiveness of interventions informed by this data, such as targeted container placement or increased cleaning frequency in areas with high littering.

References

- Aanensen DM, Huntley DM, Menegazzo M, Powell CI, Spratt BG (2014) EpiCollect+: linking smartphones to web applications for complex data collection projects. *F1000Res* 3. <https://doi.org/10.12688/F1000RESEARCH.4702.1>
- Ali T, Irfan M, Alwadie AS, Glowacz A (2020) IoT-Based smart waste bin monitoring and municipal solid waste management system for smart cities. *Arab J Sci Eng* 45:10185–10198. <https://doi.org/10.1007/S13369-020-04637-W/TABLES/7>
- Al-Joburi KI (2018) Mapping Bahrain's subsurface municipal solid waste. *Arab J Geosci* 11:1–14. <https://doi.org/10.1007/S12517-018-3456-Z/TABLES/7>
- Bouzguenda I, Fava N, Alalouch C (2022) Would 3D digital participatory planning improve social sustainability in smart cities? An empirical evaluation study in less-advantaged areas. *J Urban Technol* 29:41–71. <https://doi.org/10.1080/10630732.2021.1900772>
- Chaudhari SS, Bhole VY (2018) Solid waste collection as a service using IoT-solution for smart cities. In: 2018 International conference on smart city and emerging technology (ICSCET). IEEE, pp 1–5

- Department of Environment Forestry and Fisheries (2020) National waste management strategy. Pretoria
- Department of Environmental Affairs (2018) South Africa state of waste report. Pretoria
- Herbert G, Chen X (2015) A comparison of usefulness of 2D and 3D representations of urban planning. *Cartogr Geogr Inf Sci* 42:22–32. <https://doi.org/10.1080/15230406.2014.987694>
- Ismagilova E, Hughes L, Dwivedi YK, Raman KR (2019) Smart cities: advances in research—an information systems perspective. *Int J Inf Manage* 47:88–100. <https://doi.org/10.1016/J.IJINFO MGT.2019.01.004>
- Joshi LM, Bharti RK, Singh R, Malik PK (2022) Real time monitoring of solid waste with customized hardware and Internet of Things. *Comput Electr Eng* 102:108262. <https://doi.org/10.1016/J.COMPELECENG.2022.108262>
- Karthik M, Sreevidya L, Nithya Devi R, Thangaraj M, Hemalatha G, Yamini R (2021) An efficient waste management technique with IoT based smart garbage system. *Mater Today Proc*. <https://doi.org/10.1016/J.MATPR.2021.07.179>
- Kaza S, Yao LC, Bhada-Tata P, van Woerden F (2018) What a Waste 2.0. World Bank, Washington, DC
- Kaza S, Shrikanth S, Chaudhary Kaza S (2021) More growth, less garbage. World Bank, Washington, DC
- Liang Y, Song Q, Wu N, Li J, Zhong Y, Zeng W (2021) Repercussions of COVID-19 pandemic on solid waste generation and management strategies. *Front Environ Sci Eng* 15(6):1–18. <https://doi.org/10.1007/S11783-021-1407-5>
- Mahajan S, Kokane A, Shewale A, Shinde M, Ingale S (2017) Smart waste management system using IoT. *Int J Adv Eng Res Sci (IJAERS)* 4:2456–1908. <https://doi.org/10.22161/ijaers.4.4.12>
- Moral P, García-Martín Á, Escudero-Viñolo M, Martínez JM, Bescós J, Peñuela J, Martínez JC, Alvis G (2022) Towards automatic waste containers management in cities via computer vision: containers localization and geo-positioning in city maps. *Waste Manage* 152:59–68. <https://doi.org/10.1016/J.WASMAN.2022.08.007>
- Pervin IA, Rahman SMM, Nepal M, Haque AKE, Karim H, Dhakal G (2020) Adapting to urban flooding: a case of two cities in South Asia. *Water Policy* 22:162–188. <https://doi.org/10.2166/WP.2019.174>
- Polasi LT (2018) Factors associated with illegal dumping in the Zondi area, City of Johannesburg, South Africa. In: *WasteCon 2018*
- Polasi T, Matinise Sihle, Oelofse S (2020) South African municipal waste management systems: challenges and solutions ii supervision and project management
- Principe RE, Masera P (2020) *PyEpiCollect*
- Ramson SRJ, Moni DJ (2017) Wireless sensor networks based smart bin. *Comput Electr Eng* 64:337–353. <https://doi.org/10.1016/J.COMPELECENG.2016.11.030>
- Rodić L, Wilson DC (2017) Resolving governance issues to achieve priority sustainable development goals related to solid waste management in developing countries. *Sustainability* 9:404. <https://doi.org/10.3390/SU9030404>
- Rovetta A, Xiumin F, Vicentini F, Minghua Z, Giusti A, Qichang H (2009) Early detection and evaluation of waste through sensorized containers for a collection monitoring application. *Waste Manage* 29:2939–2949. <https://doi.org/10.1016/J.WASMAN.2009.08.016>
- Sharma A, Gupta AK, Ganguly R (2018) Impact of open dumping of municipal solid waste on soil properties in mountainous region. *J Rock Mech Geotech Eng* 10:725–739. <https://doi.org/10.1016/J.JRMGE.2017.12.009>
- Singh A, Aggarwal P, Arora R (2016) IoT based waste collection system using infrared sensors. In: *2016 5th International conference on reliability, Infocom technologies and optimization, ICRITO 2016: trends and future directions*, pp 505–509. <https://doi.org/10.1109/ICRITO.2016.7785008>
- USAID (2007) Environmental guidelines for small-scale activities in Africa: environmentally sound design for planning and implementing development activities

- Gemeente Utrecht (2021) Underground containers | Municipality of Utrecht. <https://www.utrecht.nl/wonen-en-leven/afval/ondergrondse-containers/>. Accessed 14 Dec 2022
- Vicentini F, Giusti A, Rovetta A, Fan X, He Q, Zhu M, Liu B (2009) Sensorized waste collection container for content estimation and collection optimization. *Waste Manage* 29:1467–1472. <https://doi.org/10.1016/J.WASMAN.2008.10.017>
- Wilson DC, Rodic L, Modak P, Soos R, Carpinetero A, Velis C, Iyer M, Simonett O (2015) Global waste management outlook. United Nations Environment Programme
- Yousefi M, Oskoei V, Jonidi Jafari A, Farzadkia M, Hasham Firooz M, Abdollahinejad B, Torkashvand J (2021) Municipal solid waste management during COVID-19 pandemic: effects and repercussions. *Environ Sci Pollut Res* 28:32200–32209. <https://doi.org/10.1007/S11356-021-14214-9/TABLES/2>
- Ziraba AK, Haregu TN, Mberu B (2016) A review and framework for understanding the potential impact of poor solid waste management on health in developing countries. *Arch Public Health* 74:1–11. <https://doi.org/10.1186/S13690-016-0166-4/FIGURES/1>

Automatically Evaluating the Service Quality of Bicycle Paths Based on Semantic 3D City Models



Christof Beil , Mario Ilic , Andreas Keler , and Thomas H. Kolbe 

Abstract The growing demand for sustainable mobility has led to an increased focus on the development and improvement of bicycle infrastructure, especially within cities. However, evaluating the quality of existing or planned bicycle paths is a complex task mostly done manually. This paper presents a novel approach for automatically evaluating the service quality of bicycle paths using parameters derived from semantic 3D city and streetspace models compliant with the international OGC standard CityGML version 3.0. These models contain detailed 3D information with lane-level accuracy, including precise outlines of individual surfaces. This allows for accurate and high-resolution evaluations of changing bicycle path widths and slopes, as well as information on adjacent surfaces and local disturbances such as bus stops. Additionally, estimated, measured or simulated bicycle traffic volumes are considered. Based on these parameters a method for calculating the Bicycle Levels of Service (BLOS) described in a national technical regulation is adapted and implemented for a microscopic analysis. Results of this analysis are then transferred back to the original semantic 3D city objects, allowing for the attributive description of BLOS values for bicycle paths. In addition, results are visually represented by coloring corresponding bicycle path segments according to evaluation results and integrating the colored objects within a web-based Cesium visualization of a semantic 3D city model.

Keywords Bicycle infrastructure · Traffic · Transportation · Mobility · 3D city model · CityGML · OpenDRIVE · Bicycle level of service

This article was selected based on the results of a double-blind review of the full paper.

C. Beil (✉) · T. H. Kolbe
Chair of Geoinformatics, Technical University of Munich, 80333 Munich, Germany
e-mail: christof.beil@tum.de

M. Ilic
Chair of Traffic Engineering and Control, Technical University of Munich, 80333 Munich, Germany

A. Keler
Applied Geoinformatics, University of Augsburg, 86159 Augsburg, Germany

1 Introduction

The growing demand for sustainable and efficient transportation and mobility concepts has led to an increased focus on improving and expanding bicycle infrastructure. However, determining the quality of existing or planned bicycle paths is challenging. Kazemzadeh et al. (2020) present an extensive review on methods for evaluating a Bicycle Level of Service (BLOS). While some methods for the quality assessment of bicycle infrastructure focus on specific aspects of the cycling experience such as safety, connectivity and accessibility, other methods focus on the perceived level of comfort, stress or bikeability (Kellstedt et al. 2021). The BLOS method is considering several aspects of the cycling experience and provides an easy to understand metric, often standardized in national technical regulations (FGSV 2015; Transportation Research Board 2022). There are methods for calculating levels of service for bicycle paths based on parameters such as width, slope, adjacent bus stops and bicycle traffic volumes. However, relevant input parameters for these standardized methods are so far mostly gathered manually and on a macroscopic scale. In the context of urban digital twins, several cities are in the process of gathering data for creating a detailed 3D representation of the streetspace, which contains detailed geometric and semantic information on road infrastructure including exact outlines of bicycle paths. Thus, high-resolution information with surface-based lane-level accuracy is increasingly available.

In this paper, a novel approach for automatically evaluating the quality of bicycle paths using parameters derived from semantic 3D city and streetspace models compliant to the international OGC standard CityGML version 3.0 is proposed. These models contain detailed 3D information with lane-level accuracy and thus include precise outlines of individual surfaces such as bicycle paths and its surroundings. This allows accurate and high-resolution evaluations of bicycle path widths and slopes on a microscopic level. Additionally, information on local disturbances of cyclists due to adjacent surfaces such as bus stops, driving or parking lanes can be determined. Information on bicycle traffic volumes is considered by linking estimated, measured or simulated values with corresponding semantic objects. Results of these analyses are then transferred back to original semantic 3D city objects (e.g., individual bicycle paths). In addition to the attributive description of BLOS, results are visually represented by coloring corresponding bicycle paths according to analysis results and integrating the colored objects within a web-based Cesium visualization of a semantic 3D city model.

2 Related Work

2.1 *Semantic 3D Representations of Bicycle Infrastructure*

The representation of semantic 3D road and streetspace models has become increasingly relevant due to the growing demand for accurate, detailed, and up-to-date information on transportation infrastructure (Schwab et al. 2020). Additionally, increased availability of highly detailed data (e.g., from mobile mapping) allows the creation of such models. The integration of semantic information, such as classification and usage attributes, in combination with accurate geometric details and topological information ensures the usability of these models for various applications (Labetski et al. 2018). Wierbos et al. (2019) show the effect of changing bicycle path widths and bottlenecks on traffic flow and capacity. Beil et al. (2020) present a detailed review of different standards and data formats in the field of semantic road modelling as well as their capabilities. While standards and data formats such as OpenStreetMap (OSM), Geographic Data Files (GDF), Industry Foundation Classes (IFC) or OpenDRIVE (ASAM: ASAM OpenDRIVE Version 2021) can contain information on bicycle infrastructure, representations mostly focus on linear (network-based) or parametric representations of geometries. In contrast, the newest version 3.0 of the international OGC standard CityGML provides concepts for the integration of linear, polygonal, and volumetric (explicit) geometries within a consistent semantic 3D city model, including bicycle infrastructure (Kolbe et al. 2021).

2.2 *Methods and Concepts for Determining the Quality of Bicycle Paths*

Several methods have been proposed to assess the quality of bicycle infrastructure that all differ in the type, amount and weighting of considered impact factors. In general, existing methods can be classified into seven overarching categories clustered according to their specific focus of assessment. While each of the categories and methods provides a unique perspective on the quality of bicycle infrastructure, they can all help to guide investments in bicycle infrastructure to promote a safe, convenient and comfortable cycling experience for all types of cyclists. The choice of the specific method applied may depend on the specific context and goals of the assessment, as well as the availability of data and resources. Table 1 summarizes common methods to assess the quality of bicycle infrastructure, clustered into seven overarching categories and selected exemplary methods are referenced.

In this paper, the focus is on a method for calculating the BLOS based on geometric and semantic parameters, explained in detail in Sect. 3.1. The calculation of a BLOS is a commonly used method to assess the quality of bicycle infrastructure and is incorporated in a wide range of national standards and guidelines and thus provides a clear and standardized grading system that can be easily understood by stakeholders.

Table 1 Summary of common methods to assess the quality of bicycle infrastructure

| Category | Description | Exemplary methods |
|---|--|--|
| Bicycle level of service (BLOS) | Evaluation of the quality of cycling facilities based on quantitative factors, similar to the Level of Service (LOS) approach used for motorized traffic | Technical regulation/handbook for road infrastructure design (HBS) (Kellstedt et al. 2021) |
| | | Highway capacity manual (HCM) (Transportation Research Board 2022) |
| Safety analysis or safety audits | Evaluation of the safety of cycling infrastructure by conducting a systematic review of potential hazards and safety issues | Bicycle interaction hazard score (IHS) (Landis 1994) |
| | | Bicycle intersection safety Index (ISI) (Carter et al. 2007) |
| Connectivity analysis | Evaluation of the extent to which cycling networks provide access to key destinations, in terms of the network connectivity | Cyclist routing algorithm for network connectivity (CRANC) (Gehrke et al. 2020) |
| Accessibility analysis | Evaluation of the extent to which cycling networks provide access to different user groups, such as people with disabilities or elderlies | TUM accessibility atlas (Büttner et al. 2018) |
| | | Gravity-based accessibility measures for integrated transport-land use planning (GraBAM) (Papa and Coppola 2012) |
| Multi-criteria analysis (MCA) | Evaluation of the quality of cycling networks based on a set of pre-determined criteria related to safety, connectivity, and accessibility | Multiple criteria decision analysis (MCDA) (Belton and Stewart, 2002) |
| | | Explanatory spatial data analysis (ESDA) (Osama and Sayed 2016) |
| Stress level analysis or comfort analysis | Evaluation of the perceived level of stress, based on quantitative factors such as traffic volume and speed or the presence of bike lanes | Bicycle compatibility Index (BCI) (Harkey et al. 1998) |
| | | Bicycle stress map (BSM) (Mekuria et al. 2012) |
| Bikeability index | Evaluation of a range of factors that affect the quality of the bicycle experience in urban areas, taking into account both physical and social factors | Bikeability evaluation using street view imagery and computer vision (Ito and Biljecki 2021) |
| | | Munich bikeability index (Schmid-Querg et al. 2021) |

It is a quantitative method that assigns a level of service of the transportation system based on the level of comfort, safety and the quality of traffic flow for cyclists. Similar to all methods presented in Table 1, it is considered a macroscopic assessment method so far, since it mostly focuses on entire links or link segments of the transportation network, that stretch over a relatively long distance, or entire intersections. However, this macroscopic focus does not provide a detailed understanding of the quality of

smaller elements of the transportation network or bottlenecks within longer sections, disturbing an otherwise continuously safe and comfortable cycling experience. Thus, an adapted method for analyzing the service quality of bicycle paths on a microscopic level is proposed.

3 Methodology

The general workflow of the implemented process is illustrated in Fig. 1 and explained in detail in the following sections.

3.1 Implemented Method for Evaluating the BLOS

The method implemented to determine the service quality of bicycle paths is described in a national technical regulation called “Handbook for Road Infrastructure Design” issued by the German “Research Association for Roads and Traffic” (FGSV 2015). The method is based on the calculation of a Bicycle Level of Service (BLOS) depending on a disturbance rate DR of cyclists. This disturbance rate is evaluated based on the local width and slope of bicycle paths, bicycle traffic volume, amount of wide bicycles and influence of adjacent disturbances such as bus stops. The method is applicable to stretches of dedicated bicycle paths with high traffic volumes. Bicycle paths need to be divided into segments as soon as one of the parameters necessary for calculating the BLOS changes significantly. The method distinguishes between one-way and two-way bicycle paths. This paper focuses on concepts for the evaluation of one-way bicycle paths, as these are generally by far the most common types of bicycle paths. This method is chosen since an increasing availability of high-resolution data allows an automated evaluation of bicycle paths based on geometric and semantic features. Limitations of this method and possibilities to extend it with

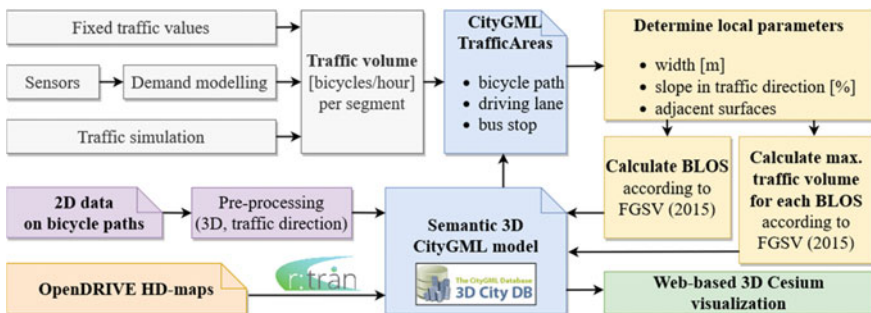


Fig. 1 General workflow for determining the bicycle level of service from relevant parameters

Table 2 Additional width required due to the local slope according to FGSV (2015)

| Slope (%) | w_{A1} (m) |
|---------------------------|--------------|
| > 6 | 0.45 |
| $4 < \text{slope} \leq 6$ | 0.30 |
| ≤ 4 | 0 |

additional concepts and information are discussed in Sect. 5 of this paper. Slopes increase the width requirement of a cyclist due to lateral swaying or the need to get off the bicycle. Thus, a fictional width w_{f1} of bicycle paths due to local slope (Eq. 1) results from the actual width w minus the additional width w_{A1} required due to the local slope according to Table 2.

$$w_{f1} = w - w_{A1} \quad (1)$$

The fictional width w_{f2} of bicycle paths due to the amount of wide bicycles such as cargo bikes (required if more than 15% of all bicycles are wide bicycles) results from the actual width w minus the additionally required width w_{A2} set at 0.3 m (Eq. 2).

$$w_{f2} = w - w_{A2} \quad (2)$$

The smallest fictional width w_f (due to slope or wide bicycles) is chosen (Eq. 3).

$$w_f = \min \begin{cases} w_{f1} \\ w_{f2} \end{cases} \quad (3)$$

The overtake rate OR is evaluated with respect to traffic volume qB , mean speed of bicycles V and corresponding standard deviation σ according to Eq. 4. The disturbance rate for one-way bicycle paths DR_O (Eq. 5) results from the overtake rate OR , based on number of bicycles per hour qB times the factor for disturbances due to overtakes f_{DO} according to Table 3.

$$OR = \frac{2 \cdot qB \cdot \sigma}{V^2 \cdot \sqrt{\pi}} \quad (4)$$

$$DR_O = OR \cdot f_{DO} \quad (5)$$

Punctual, local disturbances due to bus stops near bicycle paths DR_P are set at 1 for all segments within a certain proximity of bus stops. The disturbance rate DR for each segment then results from the addition of the disturbance rate of punctual disturbances DR_P (if available) plus the disturbance rate from overtakes DR_O (Eq. 6).

$$DR = DR_O + DR_P \quad (6)$$

Table 3 Factor for disturbances depending on local fictional width according to FGSV (2015)

| w_f (m) | f_{DO} | qB (bicycles/hour) |
|----------------------|--|--|
| ≥ 2 | 0 $0.25 \cdot (0.01 \cdot qB - 1)$ 0.5 | ≤ 100 $100 < qB < 300$ ≥ 300 |
| $1.80 \leq w < 2.00$ | 1.0 | |
| $1.60 \leq w < 1.80$ | 2.0 | |
| < 1.60 | 4.0 | |

BLOS values of segments are categorized and color-coded according to Table 4. The mean disturbance rate DR_m for the length L of an entire bicycle path then can be calculated as mean of the disturbance rates DR_i of all segments weighted according to respective segment lengths L_i (Eq. 7). Both values (Eqs. 6 and 7) can be translated to BLOS scores. Values of Eq. 6 give BLOS scores for each (high-resolution) segment, while Eq. 7 gives an aggregated (mean) value over an entire bicycle path.

$$DR_m = \frac{\sum_{i=1}^n L_i \cdot DR_i}{L} \tag{7}$$

The values in Table 4 correspond to the following definitions specified within the technical regulation (FGSV 2015):

- A: All cyclists have unrestricted freedom of movement. Changes in the line of travel within the cross-section or changes in speed are not required.
- B: Freedom of movement is hardly restricted. Changes in the line of travel within the cross-section or changes in speed are rare.
- C: Freedom of movement is repeatedly restricted by other cyclists. Changes in the line of travel within the cross-section or changes in speed are regularly required.
- D: Freedom of movement is significantly restricted by other cyclists. Changes in the line of travel within the cross-section or changes in speed are often required.
- E: Freedom of movement is constantly restricted by other cyclists. Constant changes in the line of travel within the cross-section or changes in speed are required.

Table 4 Bicycle Levels of Service (BLOS) of one-way bicycle paths depending on disturbance rates DR (for each segment) or DR_m (for an entire bicycle path) according to FGSV (2015)

| BLOS | Disturbance rate (DR or DR_m) |
|------|-------------------------------------|
| A | < 1 |
| B | < 3 |
| C | < 5 |
| D | < 10 |
| E | ≥ 10 |

In order to calculate the maximum traffic volume (qB_{max}) of bicycle path segments acceptable to reach a certain BLOS, Eqs. 4 and 5 are combined and rearranged to form Eq. 8. Where DR_{max} is the maximum disturbance rate per BLOS (cf. Table 4) and f_{DO} depends on the local fictional width w_f according to Table 3. For $w_f \geq 2.0$ the corresponding f_{DO} is set at 0.5. The mean speed of bicycles V and corresponding standard deviation σ are considered. Potential punctual disturbances are not considered in this evaluation. Resulting maximum traffic volume values are rounded down to give whole numbers.

$$qB_{max} = \frac{DR_{max} \cdot V^2 \cdot \sqrt{\pi}}{2 \cdot \sigma \cdot f_{DO}} \quad (8)$$

3.2 Requirements Towards Input Data

Until now this method has mostly been used manually, which is labor and time intensive both for gathering relevant input information on bicycle paths and calculating BLOS values from this information. Thus, so far this method is mostly used for limited spatial extends of specifically relevant segments. In order to automate this process and to be able to calculate BLOS values for bicycle paths of entire cities, information provided by semantic 3D streetspace models are beneficial. Table 5 lists parameters required for this method and compares available information provided by such models. The method and implemented process require geo-referenced 3D data with lane-level accuracy. As the results of the quality assessment strongly depend on the width of individual bike lanes, surface-based representations (using explicit coordinates) used to calculate this parameter must be available with centimeter accuracy since variations in width along a bicycle path must be detected in a high resolution to be usable for the implemented method. Average widths for longer sections (as available in OSM data or procedurally generated models) are not sufficient for the presented process. Additionally, a corresponding linear representation of bicycle lanes in 3D is required in order to calculate local slopes and segment lengths. While slopes could also be derived from digital terrain models, true 3D representations are required to represent bicycle paths on bridges or through underpasses. Linear and areal representations must be consistent in order to avoid errors in the width estimation process. With respect to semantic capabilities, functions of traffic lanes (e.g., bicycle paths) as well as traffic directions must be known. For relatively flat areas (with slopes less than 4%) information on slopes and traffic direction is not relevant. In addition to geometric information on bicycle path outlines and slopes, individual bicycle lanes should be linked with corresponding dynamic data on bicycle traffic volumes. Different methods for including this data are presented in Sect. 3.4. Bicycle counting sensors need to have information to which lane they belong. Depending on the specific use case, up-to-date information or planned scenarios are required. OpenDRIVE data (HD-maps) are increasingly available for a number of cities and often

Table 5 Parameters considered by the implemented method and availability of information in 3D CityGML streetspace models derived from OpenDRIVE data

| Parameter | Considered by the used method (FGSV 2015) | Availability in 3D CityGML model |
|-------------------------|---|----------------------------------|
| Width [m] | ✓ | Is calculated |
| Slope [%] | ✓ | Is calculated |
| Traffic direction | ✓ | ✓ |
| Traffic volume [b/h] | ✓ | Is linked |
| Wide bicycles (>15%) | ✓ | Is assumed |
| Adjacent bus stops | ✓ | Are derived |
| Other adjacent surfaces | – | Could be derived |
| Shared usage | – | ✓ |
| Change of direction | – | Could be derived |
| Change of slopes | – | Could be derived |
| Speed limit [km/h] | – | ✓ |
| Surface material | – | ✓ |
| Surface smoothness | – | – |
| Perceived comfort | – | – |

contain the required information. However, since the standard is based on a parametric representation of geometries, this data cannot be used directly for lane width calculations. Thus, the open-source converter *r:trän* (Schwab et al. 2020) is used to derive CityGML 3.0 streetspace models from OpenDRIVE data, which contain surface-based and corresponding linear network representations of bicycle paths with geo-referenced and explicit 3D coordinates. The original data was gathered using Mobile Mapping systems and contains the required high-resolution of geometric and semantic information.

3.3 *Deriving Width, Slope and Adjacent Surfaces of Bicycle Paths from 3D Streetspace Models*

Key variables necessary for this method are the width and slope of bicycle paths. Since these parameters can change rapidly and potentially just over a short distance in the course of a bicycle path, it is essential to be able to calculate lane widths and slopes at short intervals. Vitalis et al. (2022) use a method developed by Hoffmanns (2020) for deriving road widths from polygonal representations and corresponding centerlines. A similar approach is chosen in order to calculate the width of bicycle paths at a high resolution as illustrated in Fig. 2. This is implemented using the software Feature Manipulation Engine (FME).

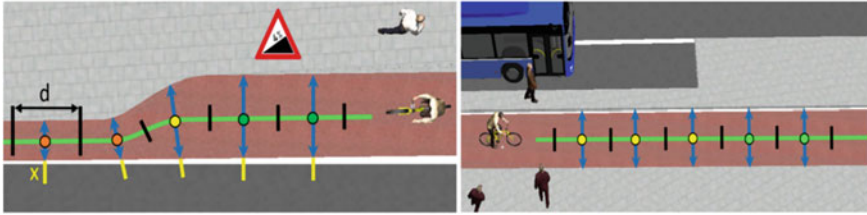


Fig. 2 Width calculation of bicycle paths at short intervals and determining adjacent surfaces

The following processing steps are implemented in order to derive the width and slope of bicycle paths at short intervals as well as information on adjacent surfaces.

1. Filter TrafficAreas with “function = bicycle path or combined footpath / cyclepath” (CityGML codelist values = 3 or 4) or “usage = bicycle” (CityGML codelist value = 6).
2. Calculate centerline (green) of each TrafficArea and its length and assign an ID per centerline using respective FME transformers. Alternatively, directly use linear TrafficSpace representations, which correspond to lane centerlines and are available in CityGML 3.0 data derived from OpenDRIVE data using r:trån.
3. Split each centerline into segments of length d (indicated with black dashes) and assign IDs per segment.
4. Calculate the slope of each segment using start and end elevation and length of each segment (the length of segments at the start or end may be shorter than d). If the original geometry already contains 3D information (which is the case for datasets given in OpenDRIVE and converted to CityGML), the slope can be directly derived. Otherwise, information from a corresponding digital terrain model can be incorporated. Information on the traffic direction is required (and also available) to evaluate the slope in traffic direction, since the method considers additional space needed by cyclists at positive inclines greater than 4% (cf. Table 2).
5. Create the centerpoint of each segment.
6. Create orthogonal lines (blue) in each centerpoint and extend to boundary of bicycle path polygons.
7. Calculate length of each orthogonal line (equal to lane width w at each centerpoint)
8. Extend orthogonal lines by x meters and test for overlap with adjacent surfaces (yellow extensions of orthogonal lines indicate, that an adjacent driving lane is detected). This method ensures, that driving lanes are detected even if they are not directly adjacent to bicycle paths, but within a range of x meters. Similarly, other relevant adjacent surfaces such as pedestrian sidewalks or parking lanes can be determined.
9. Buffer bus stop polygons by y meters and intersect with adjacent centerpoints (yellow points in the right image of Fig. 2).

Each centerpoint now contains information on local width, slope in traffic direction, adjacent surfaces and nearby bus stops (if available), a segment ID and the ID of the original CityGML TrafficArea. These parameters are then used to calculate BLOS values and corresponding maximum traffic volumes for each centerpoint as well as aggregated mean BLOS values for each CityGML TrafficArea according to the method presented in Sect. 3.1.

3.4 Including Bicycle Traffic Volumes

Information on bicycle traffic volumes (bicycles per hour) can be included in different ways. In every case, these values are assigned to corresponding city objects in order to be used as input data for the presented method.

1. Fixed bicycle traffic volumes

In case there is no information on actual bicycle traffic volumes available, the method can be applied multiple times using various fixed values (e.g. 50, 100, 250 bicycles/h). In this context, it needs to be stated, that bicycle traffic volumes and traffic flow differ between section and intersection areas. This means results calculated using fixed traffic volumes for an entire network often do not reflect reality. However, in this way it can be determined up to which capacity the quality of individual bicycle path segments are of a certain level of service.

2. Individual bicycle traffic volumes derived from sensors

Alternatively, detailed information on actual traffic volumes can be derived from sensors such as bicycle counting stations, which are available in several cities. Usually, the location of these sensors is known, which allows a direct relation of counting results to specific bicycle lanes in the semantic 3D city model. Even if not every bicycle lane may be linked with real-world sensor information, exemplary stations can give information on the general scale and magnitude of bicycle traffic volumes in certain parts of a city, which then can be used as input for the presented bicycle path quality analysis. Typical averages or maximum capacity utilization can be derived from such analysis and linked with corresponding semantic city objects. Information derived from sensors distributed in a city can also be used as input for demand modelling techniques to estimate bicycle traffic volumes. Cities such as Hamburg provide open-access to IoT servers of bicycle counting sensors via a standardized OGC SensorThings API (<https://iot.hamburg.de/>).

3. Individual bicycle traffic volumes derived from simulations

Bicycle traffic volumes derived from real-world counting sensors may not be available for every bicycle path (segment). However, approximate numbers can be simulated using traffic simulation software and then linked to semantic bicycle path objects.

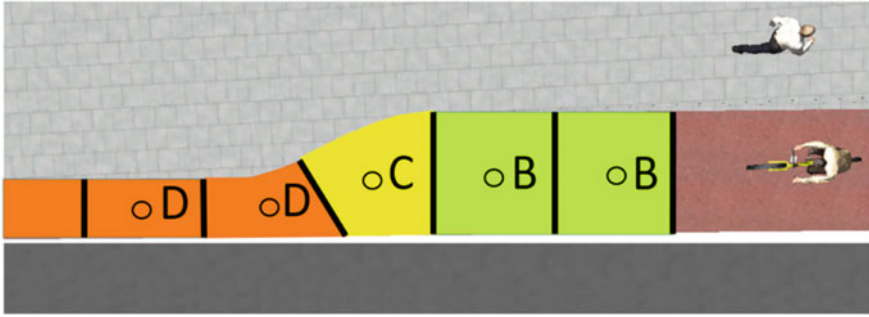


Fig. 3 Coloring bicycle path segments according to quality results of corresponding centerpoints

3.5 *Evaluating the Amount of Wide Bicycles (E.G. Cargo Bikes)*

Detailed information on the percentage of wide bicycles may not be available. However, since there are only two options that can be taken into account (share of wide bicycles over or under 15%), the method can be applied for both cases to compare results.

3.6 *Linking Results with Semantic 3D City Objects*

CityGML provides the concept of generic attributes for storing information not considered within the definitions of the original standard. Alternatively, a built-in mechanism called Application Domain Extension (ADE) to extend the data model of the standard with application-specific concepts is available, which could be developed for the presented application. In this study, results on the quality of bicycle paths (BLOS) determined using the method described in this paper, are stored as generic CityGML attributes with corresponding CityGML TrafficAreas (bicycle lanes). Furthermore, segments are colored according to quality categories (cf. Table 4) as depicted in Fig. 3, using the possibility of CityGML features to have appearances. This is done by transferring BLOS values of individual centerpoints to corresponding surface representations of bicycle path segments.

3.7 *Web-Based Visualization of Results in Combination with 3D City Models*

The 3DCityDatabase (3DCityDB) is a CityGML compliant open-source solution for importing, managing, analyzing, visualizing, and exporting virtual 3D city models

(Yao et al. 2018). The corresponding 3DCityDB Web-Map-Client is an application for web-based visualization of 3D city models using the Cesium virtual globe, which additionally offers the possibility to link city objects with semantic data for interactive exploration. Multiple layers (e.g., buildings, vegetation, road infrastructure, etc.) can be included and an incorporated tiling mechanism allows the visualization of large 3D city models. A web-based visualization is created using the 3DCityDB Importer/Exporter tool, including bicycle paths colored according to individual BLOS values (cf. Table 4). Since currently, only CityGML version 2.0 data is supported by the 3DCityDB (v4.3), results are provided in CityGML versions 2.0 and 3.0. The possibility to communicate analysis results for existing or planned scenarios in an interactive and openly accessible web-based visualization has potential for improved public participation. However, resulting BLOS values need to be interpreted correctly and can be misleading. Bicycle paths, for example, are usually constructed in such a way, that at peak traffic volumes, a BLOS of category D is reached. Without this knowledge, analysis results may give a wrong impression on the actual service quality of bicycle paths to the public.

4 Results

The method and process described in Sect. 3 is applied to data available for multiple cities including Hamburg and Munich. Since streetspace and bicycle path data according to CityGML 3.0 is available for all of these examples, the method is easily transferable.

1. Different bicycle traffic volumes for the same scenario

Figure 4 shows results of the presented method in a web-based Cesium visualization combined with a corresponding semantic 3D city model. Information on bicycle paths available in the CityGML format is used to calculate BLOS scores of the same scenario for different traffic volumes. This is done using different fixed values (e.g., 50, 100, 150, 200, 250 bicycles per hour).

Additionally, bicycle traffic volumes provided by bicycle counting sensors are available in the research area, providing information on typical average and peak bicycle traffic volumes. Multiple layers colored according to those results are integrated within this visualization. In this example, bicycle paths are split into 2 m segments, resulting in a high resolution of calculated BLOS scores. The top image in Fig. 4 shows results for a traffic volume of 50 bicycles per hour, while the bottom image shows results of the same bicycle paths for 250 bicycles per hour.

2. Same bicycle traffic volume for current and planned scenario

Figure 5 shows a direct comparison of BLOS scores of existing bicycle paths with a planned scenario in the same area with increased bicycle path widths. Both scenarios are calculated with the same traffic volume (150 bicycles per hour in this example), showing the improved BLOS for wider bicycle paths. Visualizations of these analyses

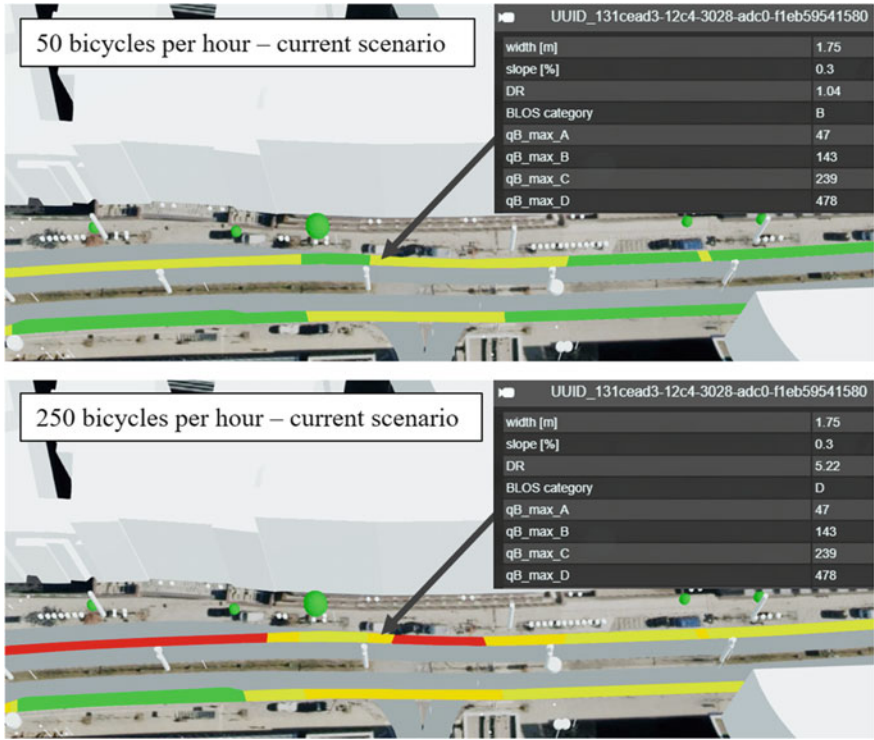


Fig. 4 Web-based Cesium visualization of BLOS results for different bicycle traffic volumes combined with a corresponding 3D city model. Green blobs represent trees

can be useful in order to present the impact of planned scenarios to the public, demonstrating improved bicycle path qualities according to the presented method.

3. Maximum traffic volume capacity for each BLOS

Using Eq. 8, maximum traffic volumes can be determined with which a bicycle path segment of a certain (fictional) width is still categorized within a specific BLOS category. For example: Assuming a typical mean speed of cyclists V of 18 km/h with a standard deviation σ of 3 km/h and a local (fictional) width of a segment of 1.75 m (corresponding to a factor for disturbances due to overtakes f_{DO} of 2.0, cf. table 3), a disturbance rate DR of under 1 (equal to category A, cf. table 4) can be achieved for a bicycle traffic volume of up to 47 bicycles per hour. Similarly, maximum capacities for all other BLOS categories are calculated (exemplary results are visible in Fig. 4). These values can be calculated per bicycle path segment. Additionally, the minimum value of all segments can be determined in order to evaluate the maximum capacity of an entire bicycle path.

4. Comparison of calculated bicycle path widths with recommended minimum widths



Fig. 5 Comparison of BLOS values at 150 bicycles per hour of the current state of bicycle paths with a planned scenario

Most countries have guidelines on the design of bicycle paths with regard to minimum widths that should be available in order to ensure usability and safety. In addition to evaluating BLOS values, calculated widths of bicycle paths can be compared with such standard widths specified within infrastructure design guidelines and regulations. This allows to determine the percentage of cycle paths (in relation to length) that adhere to these guidelines and to identify segments that do not fulfil them. Typically, bicycle path widths of at least 1.5 m for one-way lanes are required with a recommended minimum width of 2 m. Within the research area, over 28 km of bicycle path data is available. Widths are calculated at an interval of 0.6 m resulting in 47,655 individual bicycle path segments. Segments are categorized by width ranges as summarized in Table 6, showing that most of the evaluated bicycle paths are within the recommended ranges. Widths of under 1 m mostly result from tapered geometries of start and end segments of bicycle paths (as visible in Fig. 5) and are thus filtered from this evaluation. Alternatively, start and end segments can be filtered with topological information on neighboring segments. Similar metrics can be used to compare bicycle infrastructure of different cities.

Table 6 Percentage of bicycle path widths within a certain range in a citywide research area

| Width range (m) | Nr. of segments | Length (km) | Percentage (%) |
|--------------------|-----------------|-------------|----------------|
| ≥ 2 | 16227 | 9.74 | 34.1 |
| $1.5 \leq w < 2.0$ | 20658 | 12.39 | 43.4 |
| $1.0 \leq w < 1.5$ | 9642 | 5.79 | 20.3 |
| < 1.0 | 1128 | 0.62 | 2.2 |

5 Discussion and Outlook

In this paper, a method for automatically evaluating the Bicycle Level of Services (BLOS) of bicycle path segments based on geometric and semantic parameters derived from semantic 3D city and streetspace models is presented. Requirements of input data and a process to calculate widths and slopes of bicycle paths in a high resolution are shown. Additionally, different ways for considering bicycle traffic volumes required for this evaluation are discussed. Results of this analysis are linked with semantic 3D city objects and visualized within an interactive and web-based Cesium visualization.

The implemented method is chosen due to the potential for automated evaluations of bicycle path qualities for large areas. Parameters relevant for other methods such as the perceived level of comfort of cyclists are not considered. Thus, this method should be combined with other approaches (such as MCA, connectivity analysis, bikeability, etc.) for evaluating the quality of bicycle paths in order to have a holistic result.

While the presented method is based on detailed studies (FGSV 2015), there are some limitations. BLOS is primarily focused on physical attributes of the transportation system and does not take into account social or cultural factors that may affect the perceived quality of bicycle infrastructure. It also does not consider the quality of the connections between different parts of the network or the accessibility of certain destinations that can be reached by bicycle. Results of this method highly depend on accurate calculations of bicycle path widths and slopes. Thus, in order to be able to evaluate accurate BLOS scores, the used input data must be available in great geometric detail and with explicit geometry representations. The 3D streetspace models used in this paper are derived from highly accurate OpenDRIVE data converted to explicit geometries according to CityGML 3.0 and thus provide this information with the required detail. Results of width and slope calculations additionally depend on chosen distance intervals (sampling rates) for which these evaluations are calculated. In this context, a step size of at most 2 m is recommended. Since BLOS values are categorized based on disturbance rates calculated from bicycle path widths, a difference in width of only a few centimeters may already result in a different BLOS score. While this results in high-resolution evaluations, it can be beneficial to compare results of adjacent segments in order to identify very short segments with results differing from those of neighboring segments and considering a smoothing mechanism. In case no information on elevation or traffic

directions is available, the method can still be used in relatively flat areas with slopes of less than 4% (since this is the threshold for which slopes have an influence on the calculation of a fictional width, cf. Table 2). As discussed earlier, semantic 3D streetspace models can provide additional information that could be relevant for the estimation of bicycle path qualities, which are not considered by the implemented method. This includes information on surface smoothness, change of slopes and directions, surface material or speed limits. Additionally, the method only considers bus stops as sources of local disturbances. As presented in this paper, information on adjacent surfaces can easily be determined and thus be extended to influences of adjacent pedestrian paths, driving and parking lanes or structures separating bicycle paths from other traffic members.

Information on bicycle paths and 3D city models can also be used for visibility analysis (Bassani et al. 2015). Either to determine which parts of a city are visible for cyclists (e.g., in an intersection area) or to evaluate the visibility of cyclists for other traffic members such as car drivers. This is relevant for evaluating the safety of cyclists and to identify potentially dangerous areas. Known positions of cyclists and driving lanes, in addition with information on traffic directions (and thus view direction of cyclists), can be used to calculate lines of sight, which then can be intersected with other city objects such as buildings, vegetation or parking cars. In the context of urban digital twins, an increasing availability of detailed information on the streetspace will allow these evaluations to be used for a number of cities and regions.

Acknowledgements We thank the City of Munich for the cooperation in the Connected Urban Twins (CUT) project funded by the Federal Ministry for Housing, Urban Development and Building of Germany (BMWSB).

References

- ASAM (2021) ASAM OpenDRIVE version. 1.7 user guide, association for standardization of automation and measuring systems (ASAM), <https://www.asam.net/standards/detail/opendrive/>
- Bassani M, Grasso N, Piras M (2015) 3D GIS based evaluation of the available sight distance to assess safety of urban roads. *The Int Arch Photogram Remote Sens Spatial Inf Sci XL-3/W3:137–143*. <https://doi.org/10.5194/isprsarchives-XL-3-W3-137-2015>
- Beil C, Ruhdorfer R, Coduro T, Kolbe TH (2020) Detailed streetspace modelling for multiple applications: discussions on the proposed CityGML 30 transportation model. *ISPRS Int J Geo-Inf* 9:603. <https://doi.org/10.3390/ijgi9100603>
- Belton V, Stewart TJ (2002) An integrated approach to MCDA. In: *Multiple criteria decision analysis: an integrated approach*. Springer US, Boston, MA, pp 331–343. https://doi.org/10.1007/978-1-4615-1495-4_11
- Büttner B, Kinigadner J, Ji C, Wright B, Wulfhorst G (2018) The TUM accessibility atlas: Visualizing spatial and socioeconomic disparities in accessibility to support regional land-use and transport planning. *Netw Spat Econ* 18:385–414
- Carter DL, Hunter WW, Zegeer CV, Stewart JR, Huang H (2007) Bicyclist intersection safety index. *Transp Res Rec* 2031:18–24

- FGSV: Handbuch für die Bemessung von Straßenverkehrsanlagen [Manual for the design of road traffic facilities] (2015) Forschungsgesellschaft für Straßen-und Verkehrswesen e.V. (FGSV), <https://www.fgsv-verlag.de/hbs>
- Gehrke SR, Akhavan A, Furth PG, Wang Q, Reardon TG (2020) A cycling-focused accessibility tool to support regional bike network connectivity. *Transp Res Part d: Transp Environ* 85:102388
- Harkey DL, Reinfurt DW, Sorton A (1998) The bicycle compatibility index: a level of service concept, implementation manual
- Hoffmanns W (2020) Breedtescript BGT wegdelen, https://github.com/willemhoffmans/bgt_weg_breedte/
- Ito K, Biljecki F (2021) Assessing bikeability with street view imagery and computer vision. *Transp Res Part c: Emerg Technol* 132:103371. <https://doi.org/10.1016/j.trc.2021.103371>
- Kazemzadeh K, Lareshyn A, Winslott Hiselius L, Ronchi E (2020) Expanding the scope of the bicycle level-of-service concept: a review of the literature. *Sustainability* 12. <https://doi.org/10.3390/su12072944>
- Kellstedt DK, Spengler JO, Foster M, Lee C, Maddock JE (2021) A scoping review of bikeability assessment methods. *J Commun Health* 46:211–224. <https://doi.org/10.1007/s10900-020-00846-4>
- Kolbe TH, Kutzner T, Smyth CS, Nagel C, Roensdorf C, Heazel C (2021) OGC City Geography Markup Language (CityGML) part 1: conceptual model standard. Open Geospatial Consortium
- Labetski A, van Gerwen S, Tamminga G, Ledoux H, Stoter J (2018) A proposal for an improved transportation model in CityGML. *Int Arch Photogramm Remote Sensing Spatial Inf Sci XLII-4/W10*:89–96. <https://doi.org/10.5194/isprs-archives-XLII-4-W10-89-2018>
- Landis BW (1994) Bicycle interaction hazard score: a theoretical model. *Transp Res Rec* 1438:3–8
- Mekuria MC, Furth PG, Nixon H (2012) Low-stress bicycling and network connectivity. Mineta Transportation Institute Publications. <https://transweb.sjsu.edu/research/Low-Stress-Bicycling-and-Network-Connectivity>
- Osama A, Sayed T (2016) Evaluating the impact of bike network indicators on cyclist safety using macro-level collision prediction models. *Accid Anal Prev* 97:28–37. <https://doi.org/10.1016/j.aap.2016.08.010>
- Papa E, Coppola P (2012) Gravity-based accessibility measures for integrated transport-land use planning (GraBAM). *Access Instrum Plan Pract* 117:124
- Schmid-Querg J, Keler A, Grigoropoulos G (2021) The Munich bikeability index: a practical approach for measuring urban bikeability. *Sustainability* 13. <https://doi.org/10.3390/su13010428>
- Schwab B, Beil C, Kolbe TH (2020) Spatio-Semantic Road Space Modeling for Vehicle-Pedestrian Simulation to Test Automated Driving Systems. *Sustainability* 12:3799. <https://doi.org/10.3390/su12093799>
- Transportation Research Board, National Academies of Sciences, Engineering, Medicine: highway capacity manual 7th edition: a guide for multimodal mobility analysis. The National Academies Press, Washington, DC (2022) <https://doi.org/10.17226/26432>
- Vitalis S, Labetski A, Ledoux H, Stoter J (2022) From road centrelines to carriageways—a reconstruction algorithm. *PLoS ONE* 17:e0262801. <https://doi.org/10.1371/journal.pone.0262801>
- Wierbos MJ, Knoop VL, Hänseler FS, Hoogendoorn SP (2019) Capacity, capacity drop, and relation of capacity to the path width in bicycle traffic. *Transp Res Rec* 2673:693–702. <https://doi.org/10.1177/0361198119840347>
- Yao Z, Nagel C, Kunde F, Hudra G, Willkomm P, Donaubaauer A, Adolph T, Kolbe TH (2018) 3DCityDB—a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. *Open Geospatial Data Softw Standards* 3:1–26. <https://doi.org/10.1186/s40965-018-0046-7>

Virtual and Augmented Reality and Visualization

Virtual Reality Experience Analysis from Point Cloud Data



Diego Aneiros-Egido , Jesús Balado , Ha Tran ,
and Lucía Díaz-Vilariño 

Abstract This study explores the implementation of point cloud visualization in virtual reality environments to evaluate immersion based on point cloud characteristics. Two visualization methods were utilized, including points and meshes, and the method was tested on point clouds from three popular datasets: Paris-Carla-3D, Toronto-3D, and Stanford 3D Indoor Scene. The point-based visualization better preserved the original point cloud's geometries and colors, but visualization through surfaces formed by points added greater realism to the experience. However, the illumination options of the points resulted in a loss of realism. Mesh visualization modified both the geometry and colors, producing models that were less realistic than those created by the point visualization. Occlusions affected the realism of all scenes, causing a loss of information and influencing the generation of erroneous illumination. Immersion was hindered by the large number of points displayed on the screen simultaneously, which significantly reduced Frames Per Second. Despite this, no problems related to Virtual Reality sickness were observed beyond a brief initial adaptation period.

Keywords LiDAR · Photogrammetry · Game Engines · Unreal Engine · Indoor environment · Urban environment

This article was selected based on the results of a double-blind review of the full paper.

D. Aneiros-Egido · J. Balado (✉) · L. Díaz-Vilariño
GeoTECH, CINTECX, Universidade de Vigo, 36310 Vigo, Spain
e-mail: jbaldado@uvigo.gal

H. Tran
Department of Infrastructure Engineering, The University of Melbourne, Parkville 3010,
Australia

1 Introduction

Virtual Reality (VR) is being used in numerous applications including entertainment industry, architectural visualizations, training/simulation, and remote operations (Oprea et al. 2019). VR can be extremely valuable in situations when physical presence is not possible (Newman et al. 2022), either due to inaccessible location or unavailable equipment. Other advantages of VR simulations are the generation of richly detailed, complex and vivid controlled environments (Kisker et al. 2021).

In addition to presence sensation, many applications also seek to elicit strong emotional responses (Zibrek et al. 2019), mainly through first person VR experiences (Dunnagan et al. 2020). Since the effectiveness of simulations relies on creating realistic experiences, more realistic scenarios will produce greater emotional responses (Slater et al. 2020). However, the generation of true-to-life scenarios requires a great deal of time and effort. LiDAR technology and photogrammetric techniques can help in this generation, as survey time is reduced considerably (Stefano et al. 2021), while modelling processes can be automated to obtain realistic 3D scenarios (Shirowzhan et al. 2020).

The aim of this work is to evaluate the realism of point clouds for generating photorealistic VR environments. Point clouds acquired from urban and indoor areas are loaded into Unreal Engine and visualized using a VR headset. Characteristics influencing point cloud visualization are studied for each case study: geometry, color, illumination, density, and occlusions. Furthermore, the user experience in VR is also evaluated in terms of fluidity, realism, and sickness symptoms.

The paper is organized as follows. Section 2 compiles the state of the art on the generation of 3D models from point cloud data. In Sect. 3, the methodology is explained. Results are presented and analyzed in Sect. 4. Discussion is in Sect. 5. Section 6 concludes this work.

2 Related Work

LiDAR technology have been a key technology for digitalization of built and natural environments. To date, LiDAR point clouds are widely used for generation of 3D models and digital replicas of physical environments, which serve various applications, ranging from urban planning, automated manufacturing to preservation of built heritages, and vegetation inventories (Sithole and Vosselman 2004; Balado et al. 2020; Mani et al. 2015). Over the last few decades, classification and segmentation of urban scenes and 3D urban reconstruction from airborne laser scanning (ALS) and Terrestrial laser scanning (TLS), as well as mobile laser scanning (MLS) point clouds have been extensively developed (Balado et al. 2018; Park and Guldmann 2019; Rutzinger et al. 2010). These 3D reconstructed models of urban objects of cities and landscapes can be represented in exchangeable and standardized geographic information system (GIS) formats, such as The City Geography Markup Language

(CityGML), which are favorable and ready-to-use in many real-world applications (Kolbe et al. 2005). In general, the quality of laser scanning point clouds is a major factor influencing on not only point cloud-based visualization of the scenes, but also the geometric quality of virtual 3D reconstructed urban and landscape models (Oude Elberink and Vosselman 2011).

Building information modelling (BIM) is another key application of LiDAR technology. There are extensive efforts both in academia and in industry for reconstruction of virtual 3D surface-based and/or volumetric models of built assets, such as interior and exterior buildings (Becker et al. 2009; Díaz-Vilariño et al. 2015; Tran et al. 2019) as well as other physical facilities and infrastructures (e.g., bridges, railways, tunnels, and communication utilities) (Tran et al. 2021; Elberink and Khoshelham 2015; Fathi and Brilakis 2011). The reconstruction of BIM models contains the generation of not only geometric elements, but also semantic and topological elements, which conform with different BIM standards, such as such as OGC Indoor Geography Markup Language (IndoorGML) and Industry Foundation Class (IFC). Similar to LiDAR-based reconstruction of virtual 3D urban models, the LiDAR point clouds captured the same environments with different qualities may lead to different realistic levels of outcome models. In fact, noisy and occluded point clouds are likely to produce inaccurate 3D models with high level of incompleteness and incorrectness (Tran et al. 2019).

VR technologies have shown a great potential in enhancing human perceptions and interpretation of 3D immersive environment (Nehmé et al. 2021). The technologies provide a high realistic visualization of 3D digital contents and benefits in correction of their representation distortion. With the recent advancement in VR devices and technologies, there is an increasing demand in developing VR based application using LiDAR point clouds for documenting and interpreting natural and built environments (Nehmé et al. 2021; Ramousse et al. 2023). With the recent development of modern VR devices and technologies, there is an increasing demand on developing VR based application using LiDAR point clouds for documenting and interpreting real environments. Franzluebbbers et al. (2022) develop an interactive VR application for semantic interpretation of a point cloud of physical objects and environments. Tredinnick et al. (2016) propose a progressive feedback- driven rendering method for the reconstruction of physical scenes based on a VR system. In (Bozorgi and Lischer-Katz 2020), the authors propose combination of 3D laser scanning and VR technology, which will support intuitive human-scaled engagement in documenting and analyzing a large historical landmark of the desert city in Kerman, Iran for culture heritage preservation. In general, similar to other well-developed LiDAR research areas, e.g., 3D urban reconstruction and BIM, these methods highly rely on faithfulness and precision of the input point clouds in order to generate high-fidelity representation of the physical environments in VR environments.

However, the aforementioned works have not provide a comprehensive comparison and analysis the relation between quality of point clouds and 3D models to the realistic of intuitive experience in VR visualization. In this paper we will visualize point clouds of urban and indoor environments and relate the user experience to the characteristics of point clouds.

3 Methodology

In order to generate a 3D model for Virtual Reality (VR) experience, we first texture the input point cloud by creating a mesh model which aligns the input points with their corresponding colors. In this way, millions of discrete points are joined together and represent as a surface that covers the objects. Generally, these processes transform the point cloud by joining the points forming triangles (mesh visualization) (Vincke et al. 2019) between them, thus reducing the number of data while maintaining almost the same amount of information as the original cloud. There are also processes that simply add some thickness to each point until the point cloud appears as a united surface (point visualization) (Richter and Döllner 2014).

In this paper two different texturing methods are used, and the results are compared. The first method is based on meshes that are generated through Python. The other is based on an Unreal Engine plugin for point visualization. In both cases, Unreal Engine is used as the graphics engine to generate the 3D model and the VR visualization.

3.1 Mesh Visualization

The visualization of the mesh involves the creation of surface reconstructions from the point cloud data $P(X,Y,Z,R,G,B)$, being XYZ the geometric information and RGB the natural color. This process is performed in MeshLab using the Screened Poisson Surface Reconstruction algorithm (Kazhdan and Hoppe 2013). To calculate the surface normals, pre-calculations are done for $k = 50$ neighbors (Weinmann et al. 2015), and these are reoriented based on the terrain direction. The mesh is then textured to obtain a color representation, and MeshLab has a built-in method for generating a texture map for each triangle. The color information from the input point cloud is transferred to the new mesh texture. To use the mesh in Unreal Engine, a new material is created and linked to the texture map (shown in Fig. 1). The material characteristics are set to “Metallic” = 0, “Specular” = 0, and “Roughness” = 1. In addition, the “Static mesh LODGroup” is set to SmallProp to enable visualization by Level of Detail and save graphic resources.

3.2 Point Visualization

The “LiDAR Point Cloud” plugin is an extension for Unreal Engine (Epic Games) that facilitates the import, visualization and processing of point clouds generated by LiDAR devices. This extension can import point clouds in “.las” or “ASCII” format without the need to manually perform the texturing process. This extension also supports importing point clouds in “.txt” format if you specify the order and range

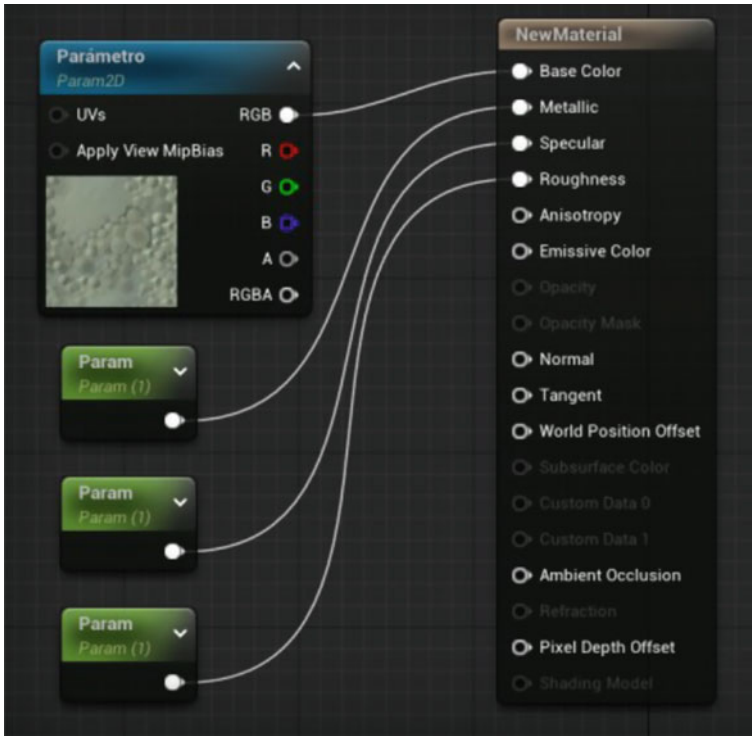


Fig. 1 Material properties

of the correct values that form the cloud. To install this extension, last version of Unreal Engine 4 or 5 is required and must be enabled in the plugin manager after downloading it from the store integrated in the “Epic Games” launcher. Thereafter, point clouds can be directly dragged into the workspace in Unreal Engine without further configuration.

3.3 VR Implementation

Once the two methods of importing and visualizing point clouds in the 3D model have been applied, the point clouds are moved to their initial position XYZ in the virtual space through translation options. To make it possible to visualize the 3D models with the VR glasses, it is necessary to install specific software.

There are plugins to identify the VR hardware to be used, so graphic engines can detect them and associate the controls when starting the simulation. In case the graphic engine cannot find a specific extension for the VR kit, “Open XR” plugin in Unreal Engine can be employed to obtain generic drivers. This plugin is suitable

for any generic VR device, allowing the implementation in the simulation all the functionalities available in the VR hardware. In addition, depending on the operating system, additional auxiliary software may be required. In this case “Windows Mixed Reality” is used with Windows 10 to connect to the VR hardware.

The steps mentioned so far are sufficient to visualize the 3D model in VR without any collisions (traversing surfaces) and being able to move freely without restrictions in the XYZ axes (“fly”) over the model with the help of the mouse and keyboard. For greater immersion and realism in the VR experience, mobility is then restricted so that the avatar can only move by walking on the floor of the 3D model and cannot traverse solid objects.

3.4 Definition of Collisions and Navigable Area

Applying collisions to point clouds prevents the user from traversing objects during the simulation. Collisions are generated by Unreal Engine indicating a precision threshold. The higher the accuracy the more complicated the mesh processing will be. A collision accuracy of 10 cm for indoor environments and 40 cm for urban environment is set. In case the collision mesh is not generated as desired, it is possible to manually insert a custom collision mesh, using polygons or change the number of vertices of the collision mesh.

In addition to having collisions, it is advisable to delimit accessible zones to eliminate blockages of the player or falls out of the model limits. The delimitation of zones is done by means of the object “Nav mesh Bound Volume” which is dragged to the navigation area and generates a polygon representing the accessible zone (Fig. 2). The object can be modified by varying its scale, position and vertex properties. It is also possible to generate several objects of this class to cover the entire navigable area.

To generate the avatar that represents the user in the immersive visualization of the 3D model, the “MotionControllerPawn” entity is imported. If this object is not available, the VR content pack can be also imported (Fig. 3). Then the “MotionControllerPawn” is dragged, and the avatar placed at the start location of the simulation. The displacement is made by the movement of the VR helmet but also by jumping to locations set by the controls. For a more realistic movement, particularly in indoor environments, it is necessary to consider the dimensions of the avatar. The dimensions can be changed in the scale settings on each axis. A default height of 182 cm is selected by according to A century of trends in adult human height (2016).

Once the navigable space and collisions have been implemented, the user puts on the VR glasses and walks through the scenario (for point and mesh visualization), trying to visualize distant and nearby objects. As far as possible, the user follows the same route with both visualizations, trying to get closer to the objects and identifying unintended collisions.

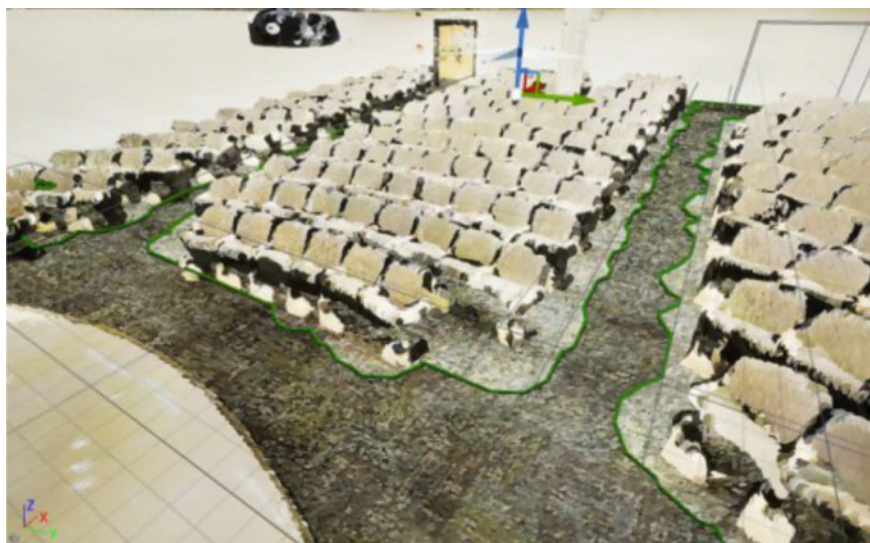


Fig. 2 Navigable areas of the point cloud

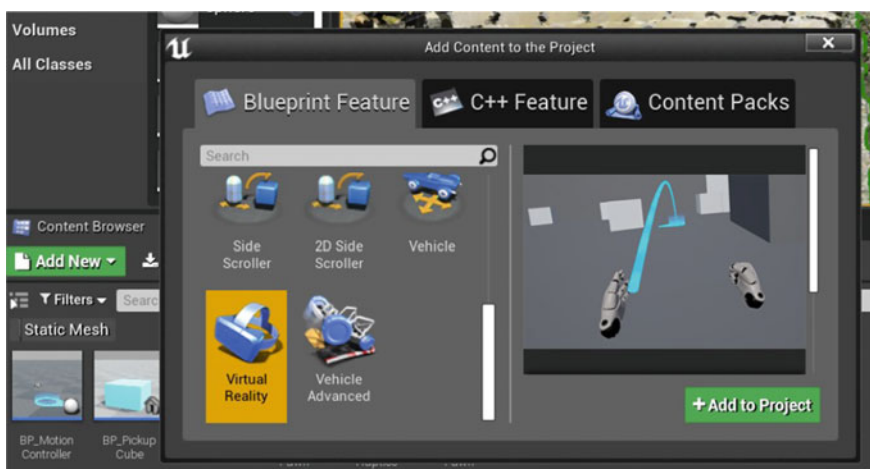


Fig. 3 Package for VR motion

Table 1 Number of points per case study

| Case study | Number of points |
|-----------------------|------------------|
| Paris-Carla-3D | 56 million |
| Toronto-3D | 74 million |
| S3DIS auditorium | 7 million |
| S3DIS lounge | 1 million |
| S3DIS corridor | 2 million |
| S3DIS conference room | 1.5 million |
| Bathroom | 1 million |

4 Experiments

4.1 Case Study

The presented methods were tested on two urban case studies and one indoor case study. The urban cases corresponded to the Paris-Carla-3D (Deschaud et al. 2021) and Toronto-3D datasets (Tan et al. 2020), generated from Mobile Laser Scanning (MLS) systems. The boundaries of both point clouds were cropped to remove the areas with low point density. The indoor case study was obtained from Stanford 3D Indoor Scene (S3DIS) (Armeni et al. 1702) and five corresponding locations were selected: auditorium, lounge, corridor, conference room, and bathroom. The survey was conducted with a Matterport Camera. Number of points are compiled in Table 1.

4.2 Results and Analysis

The quality of visualization can be assessed by objective and subjective methods. Objective methods are also known as Quality Assessment and some of the most representative are PSNR, SSIM, BRISQUE, NIQE, or PIQE. However, numerous studies have shown objective quality assessments are no related to a better human perception (Alexiou et al. 2018; Silva Cruz et al. 2019; Uchida et al. 2020). For that reason, the result analysis was done by subjective observations. Pairs of visualizations (mesh vs point) with a similar perspective were selected and compared. Afterwards, the VR experience was analyzed.

The results of loading the point clouds into Unreal Engine for the outdoor environments are shown in Fig. 4 with mesh visualization and point visualization. The indoor environments are shown in Fig. 5. A distinguishing feature between indoor and outdoor point clouds was their size, both in number of points and in coverage of the area depicted. For outdoors, priority was given to a visualization of several meters, where realism was obtained at medium distances. For indoors, on the other hand, a visualization at short distances, room by room, was valued. The analysis of the

results is based on the following five key characteristics of point cloud visualization: Geometric precision, color, illumination, point density and occlusions.

Geometric precision: In all visualizations, a large difference in the geometric definition of some objects was observed. In outdoors, the meshing method smooths the surfaces excessively, even eliminating objects the size of cars. In Fig. 4b, the street looks empty, while in Fig. 4f, the street looks crowded (although most of the points corresponded to noise). Similarly, the cars in Fig. 4c, d, g, h can be observed. This excessive smoothing was due to the setup parameters of the Screened Poisson Surface Reconstruction, which were adjusted to obtain a result independent of the large point density variation. The central areas of the urban scenes in both case studies had enough precision to easily recognize road markings and windows, but the meshing method produces unrealistic shapes on roofs and trees. The point visualization preserved the geometry better since no modifications were made to point clouds as in the meshing method, but it was more difficult to recognize objects far



Fig. 4 Mesh visualization of Paris 3D (a, b) and Toronto 3D (c, d). Point visualization of Paris 3D (e, f) and Toronto 3D (g, h)

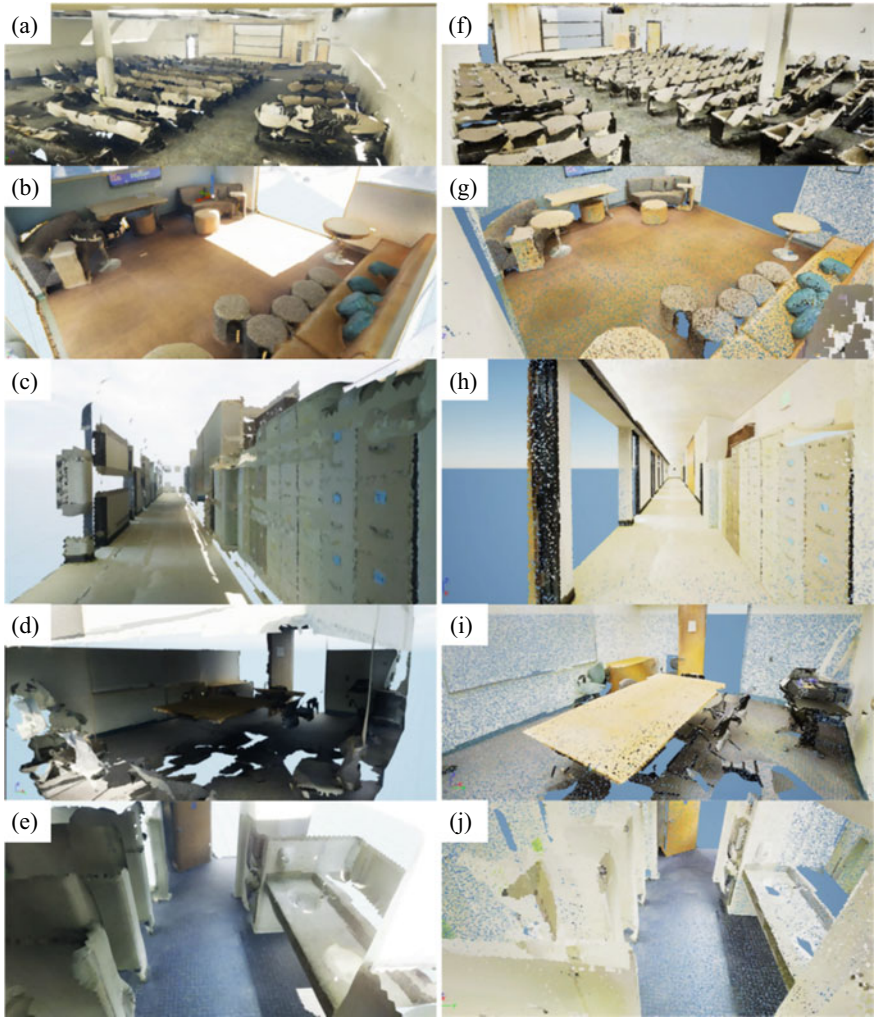


Fig. 5 Mesh visualization (left) and point visualization (right) of S3DIS: **a, f** auditorium, **b, g** lounge, **c, h** corridor, **d, i** conference room, and **e, j** bathroom

away from the Point of View (POV). In indoor areas, the result of the visualization was similar between both methods due to the size of the case studies. The meshing eliminated small objects (few centimeters in size) in this case too, but it can be considered a problem of point cloud resolution.

Color: Color is a key element to obtain a realistic scene, and even more so when some objects cannot be recognized by their geometrical shape (Estes et al. 1983). The meshing method also showed limitations in color due to the texture map generation. Colors were mixed and blurred with other colors during mesh generation especially

in urban areas (Fig. 4), where a significant color shift was observed in the coloring of trees and some vehicles.

Lighting: By default, the precomputed illumination was generated in mesh visualization but not in point visualization. The existence of illumination gave a great realism to the scene, both in urban and indoor areas. Point visualization without illumination causes a loss of depth information, although natural survey shadowing was lightly preserved (floor of Fig. 5h, j). Precomputed illumination also had some drawbacks. Since illumination is a geometry-dependent technique, errors in the mesh generation and occlusions lead to the appearance of new light inputs (wall open spaces) in the scene (floor in Fig. 5.c).

Occlusions: Occlusions negatively affected both visualizations equally. The lack of information resulted in user rejection in terms of realistic immersion. The most notable example was the meeting room (Fig. 5d, i), where there was a considerable occlusion under the table that allows viewing through. Also, the non-digitization of parts of objects, in this case the table legs, is something that affects object recognition very negatively (Sun et al. 2021).

Point density: While the generation of meshes eliminated this problem, point visualization enhanced the visualization of isolated points, resulting in a visualization through surfaces and a consequent loss of realism. In all the case studies visualized using points, point density variations were observed close to the user's POV (Fig. 4e–h and Fig. 5f–j). Outdoors, the density also particularly affected areas away from the MLS trajectory.

5 Discussion

VR tests were conducted on a CPU AMD Ryzen 7 4800H, 16 GB RAM DDR4, Nvidia RTX 2060 6 GB GDDR5 with VR headset HP Reverb G2. Smoothness is one of the most important characteristics for VR immersion since low Frame per Seconds (FPS) rate is not immersive and annoying to the user. Smoothness is closely related to the number of simultaneously displayed points on the screen (Schütz et al. 2020). While the mesh visualisation showed an adequate and constant FPS for a correct visualisation, in the point visualisation the FPS suffered drops that produced user dissatisfaction (Table 2). A possible solution would be to implement LoD visualization (Oosterom et al. 2022) according to the distance to the POV.

Collisions were another limitation on VR experience affecting navigation. In mesh visualization, the collisions automatically generated by Unreal Engine prevented the user from getting close to the objects. To minimize this problem, the maximum number of vertices in the collision mesh generated by the program was limited. In the case of point visualization, each point generated a collision and therefore a barrier to navigation. Although desirable for the built environment points, the noise

Table 2 FPS per environment and visualization method

| Case study | FPS |
|----------------------------|-------------|
| Urban mesh visualization | 60 stables |
| Urban point visualization | 30 unstable |
| Indoor mesh visualization | 60 stables |
| Indoor point visualization | 35 unstable |

points generated undesirable collisions. One solution to the noise problem was the application of a Statistical Outlier Removal (SOR) filter.

The visualization quality is highly dependent on the survey sensor but also on the method used (point or mesh). While point visualization preserves shapes and colors better, mesh visualization corrects for density variation and small occlusions. The recommendation of the use of illumination depends on the completeness of the input data, although illumination provides more realism, light can also enter through occlusions, causing the opposite effect.

In first VR visualizations of point clouds (by both methods) some symptoms of VR sickness (Peng et al. 2020) (dizziness, disorientation, and fatigue) were felt by the user, although without reaching more severe symptoms (headaches, or nausea). This was associated more with the user's adaptation to the VR visualization than to the point cloud. Although VR sickness is a negative effect, several studies indicate that 80–95% of users experience these symptoms (Hancock et al. 2008), so it is not something exclusively associated with VR viewing of point clouds. Other common problems in VR displays, (chromatic aberration, distorted images, or Screen Door Effect) are more related to the quality of the VR headset and were not identified.

6 Conclusions

In this work, indoor and urban point cloud visualization was implemented in Virtual Reality (VR) environments to evaluate immersion in relation to point cloud characteristics. Two visualization methods were implemented (using points and meshes) and navigable areas and collisions were defined for better realism in terms of mobility. The method was tested on point clouds from three well-known datasets: Paris-Carla-3D, Toronto-3D, and Stanford 3D Indoor Scene.

The experience of each visualization method varied. The experience was highly dependent on the method employed. In the point-based visualization, geometries and colors of the original point clouds were better preserved, however the visualization through surfaces (formed by points) added a great deal of realism to the experience. The limited lighting options applied to points resulted also in a loss of realism. Mesh visualization modified both the geometry and colors, producing less realistic models than those created by point visualization. Artificial lighting and non-visualization through surfaces were clear advantages of this mesh visualization.

Occlusions affected the realism of all scenes, causing a loss of information and influencing the generation of erroneous illumination. Immersion was hindered by the large number of points displayed on the screen simultaneously, which significantly reduced FPS. Despite this, no problems related to VR sickness were observed beyond a brief initial adaptation period.

Future work will explore the use of non-photorealistic techniques such as Eye-Dome Lighting (Noël et al. 2020), Ambient Occlusion (Ren and Song 2022), and Multi Feature-Rich Synthetic Color (Balado et al. 2023) to improve immersion.

Acknowledgements This work was partially supported by human resources grant ED481D-2023-005 funded by Xunta de Galicia and RYC2020-029193-I funded by MCIN/AEI/10.13039/501100011033 and FSE “El FSE invierte en tu futuro”, by grant ED431F 2022/08 funded by Xunta de Galicia, Spain-GAIN, and by the projects PID2021-123475OAI00 funded by MCIN/AEI/<https://doi.org/10.13039/501100011033/> FEDER, UE, and PCI2022-132943, funded by MCIN/AEI/https://doi.org/10.13039/501100011033 and by the European Union “NextGenerationEU”/PRTR. The statements made herein are solely the responsibility of the authors.

References

- A century of trends in adult human height. *Elife* 5:e13410 (2016). <https://doi.org/10.7554/eLife.13410>
- Alexiou E, Ebrahimi T, Bernardo MV, Pereira M, Pinheiro A, Da Silva Cruz LA, Duarte C, Dmitrovic LG, Dumic E, Matkovics D, Skodras A (2018) Point cloud subjective evaluation methodology based on 2D rendering. In: 2018 Tenth international conference on quality of multimedia experience (QoMEX), pp 1–6. <https://doi.org/10.1109/QoMEX.2018.8463406>
- Armeni I, Sax S, Zamir AR, Savarese S (2017) Joint 2d-3d-semantic data for indoor scene understanding. arXiv preprint [arXiv:1702.01105](https://arxiv.org/abs/1702.01105)
- Balado J, Díaz-Vilariño L, Arias P, González-Jorge H (2018) Automatic classification of urban ground elements from mobile laser scanning data. *Autom Constr* 86:226–239. <https://doi.org/10.1016/j.autcon.2017.09.004>
- Balado J, Sousa R, Díaz-Vilariño L, Arias P (2020) Transfer learning in urban object classification: online images to recognize point clouds. *Autom Constr* 111:103058. <https://doi.org/10.1016/j.autcon.2019.103058>
- Balado J, González E, Rodríguez-Somoza JL, Arias P (2023) Multi feature-rich synthetic colour to improve human visual perception of point clouds. *ISPRS J Photogramm Remote Sens* 196:514–527. <https://doi.org/10.1016/j.isprsjprs.2023.01.019>
- Becker S, Haala N (2009) Quality Dependent Reconstruction of Building Façades. In: Rothmel K, Fritsch D, Blochinger W, Dürr F (eds) *Quality of Context*. Springer, Berlin, Heidelberg, pp 174–184
- Bozorgi K, Lischer-Katz Z (2020) Using 3D/VR for research and cultural heritage preservation: project update on the virtual. *Preserv Digital Technol Cult* 50. <https://doi.org/10.1515/pdct-2020-0017>
- Deschaud J-E, Duque D, Richa JP, Velasco-Forero S, Marcotegui B, Goulette F (2021) Paris-CARLA-3D: a real and synthetic outdoor point cloud dataset for challenging tasks in 3D mapping. *Remote Sensing* 13. <https://doi.org/10.3390/rs13224713>
- Díaz-Vilariño L, Conde B, Lagüela S, Lorenzo H (2015) Automatic detection and segmentation of columns in as-built buildings from point clouds. *Remote Sensing* 7:15651–15667. <https://doi.org/10.3390/rs71115651>

- Dunnagan CL, Dannenberg DA, Cuales MP, Earnest AD, Gurnsey RM, Gallardo-Williams MT (2020) Production and evaluation of a realistic immersive virtual reality organic chemistry laboratory experience: infrared spectroscopy. *J Chem Educ* 97:258–262. <https://doi.org/10.1021/acs.jchemed.9b00705>
- Elberink SO, Khoshelham K (2015) Automatic extraction of railroad centerlines from mobile laser scanning data. *Remote Sensing* 7:5565–5583. <https://doi.org/10.3390/rs70505565>
- Epic Games: LiDAR Point Cloud Plugin, <https://docs.unrealengine.com/4.27/en-US/WorkingWithContent/LidarPointCloudPlugin/>
- Estes JE, Hajic EJ, Tinney LR, Carver L (1983) Fundamentals of image analysis: analysis of visible and thermal infrared data. *Manual of Remote Sensing* 1:987–1124
- Fathi H, Brilakis I (2011) Automated sparse 3D point cloud generation of infrastructure using its distinctive visual features. *Adv Eng Inform* 25:760–770. <https://doi.org/10.1016/j.aei.2011.06.001>
- Franzluebbbers A, Li C, Paterson A, Johnsen K (2022) Virtual reality point cloud annotation. In: 2022 IEEE conference on virtual reality and 3D user interfaces abstracts and workshops (VRW), pp 886–887. <https://doi.org/10.1109/VRW55335.2022.00294>
- Golparvar-Fard Mani, Vahid B, de la Garza Jesus M (2015) Segmentation and recognition of highway assets using image-based 3D point clouds and semantic texton forests. *J Comput Civil Eng* 29:04014023. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000283](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000283)
- Hancock PA, Vincenzi DA, Wise JA, Mouloua M (2008) Human factors in simulation and training. CRC Press
- Kazhdan M, Hoppe H (2013) Screened poisson surface reconstruction. *ACM Trans Graph* 32. <https://doi.org/10.1145/2487228.2487237>
- Kisker J, Gruber T, Schöne B (2021) Behavioral realism and lifelike psychophysiological responses in virtual reality by the example of a height exposure. *Psychol Res* 85:68–81. <https://doi.org/10.1007/s00426-019-01244-9>
- Kolbe TH, Gröger G, Plümer L (2005) CityGML: interoperable access to 3D city models. In: van Oosterom P, Zlatanova S, Fendel EM (eds) *Geo-information for disaster management*. Springer, Berlin Heidelberg, pp 883–899. https://doi.org/10.1007/3-540-27468-5_63
- Nehmé Y, Dupont F, Farrugia J-P, Le Callet P, Lavoué G (2021) Visual quality of 3D meshes with diffuse colors in virtual reality: subjective and objective evaluation. *IEEE Trans Visual Comput Graphics* 27:2202–2219. <https://doi.org/10.1109/TVCG.2020.3036153>
- Newman M, Gatersleben B, Wyles KJ, Ratcliffe E (2022) The use of virtual reality in environment experiences and the importance of realism. *J Environ Psychol* 79:101733. <https://doi.org/10.1016/j.jenvp.2021.101733>
- Noël F, Flugekvam Nordang S, Jaboyedoff M, Derron M-H (2020) Identifying past rockfall trajectories and runout distances from detailed 3D terrain model: the case of the Mel de la Niva mountain, Switzerland. In: EGU General assembly conference Abstracts, p 11446. <https://doi.org/10.5194/egusphere-egu2020-11446>
- Oprea S, Martinez-Gonzalez P, Garcia-Garcia A, Castro-Vargas JA, Orts-Escolano S, Garcia-Rodriguez J (2019) A visually realistic grasping system for object manipulation and interaction in virtual reality environments. *Comput Graph* 83:77–86. <https://doi.org/10.1016/j.cag.2019.07.003>
- Oude Elberink S, Vosselman G (2011) Quality analysis on 3D building models reconstructed from airborne laser scanning data. *ISPRS J Photogramm Remote Sens* 66:157–165. <https://doi.org/10.1016/j.isprsjprs.2010.09.009>
- Park Y, Guldmann J-M (2019) Creating 3D city models with building footprints and LIDAR point cloud classification: a machine learning approach. *Comput Environ Urban Syst* 75:76–89. <https://doi.org/10.1016/j.compenvurbsys.2019.01.004>
- Peng Y-H, Yu C, Liu S-H, Wang C-W, Taele P, Yu N-H, Chen M.Y (2020) WalkingVibe: reducing virtual reality sickness and improving realism while walking in VR using unobtrusive head-mounted vibrotactile feedback. In: *Proceedings of the 2020 CHI conference on human factors*

- in computing systems. Association for computing machinery, New York, NY, USA, pp 1–12. <https://doi.org/10.1145/3313831.3376847>
- Ramousse F, Lavoué G, Baert P, Bhoowabul V, Fleury S, Ravey B, Gay A, Catherine M, Helfenstein-Didier C (2023) ReVBED: a semi-guided virtual environment for inducing food craving in a binge-eating therapy process. In: ACM international conference on interactive media experiences (IMX)
- Ren L, Song Y (2022) AOGAN: A generative adversarial network for screen space ambient occlusion. *Comput Visual Med* 8:483–494. <https://doi.org/10.1007/s41095-021-0248-2>
- Richter R, Döllner J (2014) Concepts and techniques for integration, analysis and visualization of massive 3D point clouds. *Comput Environ Urban Syst* 45:114–124. <https://doi.org/10.1016/j.compenvurbsys.2013.07.004>
- Rutzinger M, Pratihast A, Oude Elberink S, Vosselman G (2010) Detection and modelling of 3D trees from mobile laser scanning data
- Schütz M, Mandlbürger G, Otepka J, Wimmer M (2020) Progressive real-time rendering of one billion points without hierarchical acceleration structures. *Comput Graphics Forum* 39:51–64. <https://doi.org/10.1111/cgf.13911>
- Shirowzhan S, Lim S, Trinder J, Li H, Sepasgozar SME (2020) Data mining for recognition of spatial distribution patterns of building heights using airborne lidar data. *Adv Eng Inform* 43:101033. <https://doi.org/10.1016/j.aei.2020.101033>
- Silva da Cruz LA, Dumić E, Alexiou E, Prazeres J, Duarte R, Pereira M, Pinheiro A, Ebrahimi T (2019) Point cloud quality evaluation: towards a definition for test conditions. In: 2019 eleventh international conference on quality of multimedia experience (QoMEX), pp 1–6. <https://doi.org/10.1109/QoMEX.2019.8743258>
- Sithole G, Vosselman G (2004) Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. *ISPRS J Photogramm Remote Sens* 59:85–101. <https://doi.org/10.1016/j.isprsjprs.2004.05.004>
- Slater M, Gonzalez-Liencres C, Haggard P, Vinkers C, Gregory-Clarke R, Jelley S, Watson Z, Breen G, Schwarz R, Steptoe W, Szostak D, Halan S, Fox D, Silver J (2020) The ethics of realism in virtual and augmented reality. *Frontiers Virtual Reality* 1. <https://doi.org/10.3389/frvir.2020.00001>
- Stefano FD, Chiappini S, Gorreja A, Balestra M, Pierdicca R (2021) Mobile 3D scan LiDAR: a literature review. *Geomat Nat Haz Risk* 12:2387–2429. <https://doi.org/10.1080/19475705.2021.1964617>
- Sun X, Wang P, Wang C, Liu Y, Fu K (2021) PBNNet: Part-based convolutional neural network for complex composite object detection in remote sensing imagery. *ISPRS J Photogramm Remote Sens* 173:50–65. <https://doi.org/10.1016/j.isprsjprs.2020.12.015>
- Tan W, Qin N, Ma L, Li Y, Du J, Cai G, Yang K, Li J (2020) Toronto-3D: a large-scale mobile lidar dataset for semantic segmentation of urban roadways. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp 202–203
- Tran H, Khoshelham K, Kealy A, Díaz-Vilariño L (2019) Shape grammar approach to 3D modeling of indoor environments using point clouds. *J Comput Civ Eng* 33:04018055. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000800](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000800)
- Tran H, Nguyen TN, Christopher P, Bui D-K, Khoshelham K, Ngo TD (2021) A digital twin approach for geometric quality assessment of as-built prefabricated façades. *J Build Eng* 41:102377. <https://doi.org/10.1016/j.jobe.2021.102377>
- Tredinnick R, Broecker M, Ponto K (2016) Progressive feedback point cloud rendering for virtual reality display. In: 2016 IEEE virtual reality (VR), pp 301–302. <https://doi.org/10.1109/VR.2016.7504773>
- Uchida T, Hasegawa K, Li L, Adachi M, Yamaguchi H, Thufail FI, Riyanto S, Okamoto A, Tanaka S (2020) Noise-robust transparent visualization of large-scale point clouds acquired by laser scanning. *ISPRS J Photogramm Remote Sens* 161:124–134. <https://doi.org/10.1016/j.isprsjprs.2020.01.004>

- van Oosterom P, van Oosterom S, Liu H, Thompson R, Meijers M, Verbree E (2022) Organizing and visualizing point clouds with continuous levels of detail. *ISPRS J Photogramm Remote Sens* 194:119–131. <https://doi.org/10.1016/j.isprsjprs.2022.10.004>
- Vincke S, de Lima Hernandez R, Bassier M, Vergauwen M (2019) Immersive visualisation of construction site point cloud data, meshes and BIM models in a VR environment using a gaming engine. *Int Arch Photogramm Remote Sens Spatial Inf Sci XLII-5/W2:77–83*. <https://doi.org/10.5194/isprs-archives-XLII-5-W2-77-2019>
- Weinmann M, Jutzi B, Hinz S, Mallet C (2015) Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS J Photogramm Remote Sens* 105:286–304. <https://doi.org/10.1016/j.isprsjprs.2015.01.016>
- Zibrek K, Martin S, McDonnell R (2019) Is photorealism important for perception of expressive virtual humans in virtual reality? *ACM Trans Appl Percept* 16. <https://doi.org/10.1145/3349609>

Visualisation of 3D Uncertainties for Subsurface Infrastructure Using Augmented Reality



Simon Quaade Vinther, Frida Dalbjerg Kunnerup, Lars Bodum, Lasse Hedegaard Hansen, and Simon Wyke

Abstract The damage of subsurface infrastructure under the auspices of excavation is a long-standing global problem, which causes great financial losses as a consequence of project delays, disruptions of public supply, and the increased life-cycle costs of utility lines. The primary causes of excavation damage are attributed to the lack of reliable utility information and inadequate approaches to communicating the positional uncertainties to the end users. Accordingly, this study presents a deterministic uncertainty-aware approach for visualising subsurface infrastructure in 3D using augmented reality (AR). The prototype was presented and evaluated in a focus group interview with five respondents with experience from the utility sector. The participants agreed, that the insufficient availability of vertical coordinates for the cables at present constitutes the biggest challenge. However, they emphasised the future potential of the AR solution in the prospect of ongoing improvements in data quality prompted by the new Danish data model for exchanging utility information.

Keywords Uncertainty visualisation · Augmented reality · Subsurface utilities · Excavation damage

1 Introduction

Within the utility sector, availability of accurate information regarding the position of the underground infrastructure is a fundamental prerequisite. This information is pivotal in ensuring an efficient design phase, optimising communication among stakeholders in the construction process, and preventing damage to cables. In particular, the avoidance of excavation damage is a vital aspect, due to the huge economical losses they cause on a global scale. The expenses amount to approximately

This article was selected based on the results of a double-blind review of the full paper.

S. Q. Vinther · F. D. Kunnerup · L. Bodum (✉) · L. H. Hansen · S. Wyke
Aalborg University, 9000 Aalborg, Denmark
e-mail: lbo@plan.aau.dk
URL: <https://www.en.plan.aau.dk/>

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
T. H. Kolbe et al. (eds.), *Recent Advances in 3D Geoinformation Science*, Lecture Notes in Geoinformation and Cartography, https://doi.org/10.1007/978-3-031-43699-4_7

270million (GBP) in the UK and 30billion (USD) in the US on an annual basis Hansen et al. (2021a).

However, the total economic loss is far more extensive since less serious incidents and the derived consequential costs generally are omitted from the statistics—including losses as a consequence of disruptions and the increased life-cycle costs of the supply infrastructure Li et al. (2015). Based on data from the study by Makana et al. (2016), it was ascertained that the ratio between direct and indirect costs was 1:29. The primary causes of excavation damages are according to Li et al. (2015) and Su et al. (2013) attributed to (1) the lack of reliable, complete, and accurate utility information and (2) inadequate approaches to communicate the inherent, positional uncertainties associated with cable data to the end users, e.g. excavator operators. This is seconded by Tanoli et al. (2019), who describes how missing depth information and inaccurate utility-location data are primary reasons for utility strikes, in addition to the nature of utility markings which are removed during excavation when the topsoil is removed.

Insufficient information available and unawareness of positional uncertainties thus constitutes contributory causes to the instilling of false confidence, which potentially can induce unintentional damages when digging-actors perform excavation activities Li et al. (2015). However, these circumstances can simultaneously cause the opposite effect, where the consistent lack of reliable data induces a more cautious and lengthy approach to the excavation work which can prompt delays in the projects in question.

Prompted by the above considerations, the aim of this study is to investigate how the dissemination of uncertainties for utility data can be improved through 3D visualisation in augmented reality (AR). The remainder of this paper is structured as follows. First, related studies, the conceptual framework and the utilisation of augmented reality for utility line visualisation are presented in Sect. 2, respectively. Subsequently, Sect. 3 comprises an exposition of the chosen case and the appertaining data.

Afterward the methods of the study and accordingly the development of the prototype are presented in Sect. 4, followed by an evaluation of the results in Sect. 5. Finally, Sect. 6 encompasses the conclusions and suggestions for future research.

2 Literature Review

In recent years, geospatial research related to uncertainties has primarily focused on the development of typology frameworks and conceptual models. This foundational knowledge has functioned as a precursor for newer research areas that to a greater extent aim at conveying the uncertainty aspect by means of different visualisation techniques. The intention of these new approaches is to utilise the communicative value of graphic representation Smith Mason et al. (2017).

At present, the scientific work that revolves around the visualisation of uncertainty with a specific focus on utility lines is sparse. The current practice in terms of conveying the quality of spatial data sets is mainly related to the utilisation of

metadata. The VISTA project at the University of Leeds, however, illustrates two specific examples, where the visual variables; blur and color, are utilised to indicate the positional accuracy of 2D utility data Beck et al. (2009); Boukhelifa and Duke (2007). Furthermore, a 2D color graphic band for utility lines was proposed by Lanka et al. (2001) in order to handle positional uncertainties associated with three different sources of information; existing drawings, explored location information, and Ground Penetrating Radar (GPR). Also within other scientific domains the concept of visualisation of uncertainty has been investigated. See as an example Weiskopf (2022) who performs a literature review on uncertainty visualization of biological data.

In terms of 3D visualisation, Li et al. (2015) and Su et al. (2013) model uncertainties for utility lines with a basis in probabilistic error bands. In the article by Li et al. (2015), GPR, GPS, and GIS are integrated into a system that makes it possible to map underground infrastructure in an uncertainty-aware and non-destructive way, whereas Su et al. (2013) presents a conceptual framework based on a classification of four different pipeline measurement methods.

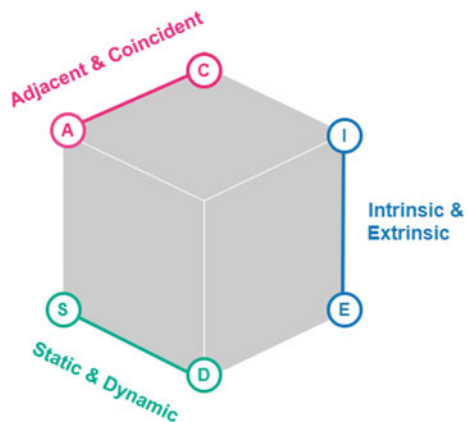
2.1 Uncertainty Visualisation Framework

The conceptual framework for this paper is based on the Uncertainty Visualisation Cube presented by Kinkeldey et al. (2014a). The main objective of the model is to function as a structure for navigating the various visualisation techniques for communicating uncertainty—including their context-specific applicability in relation to the varying groups of end users.

The axes of the cube, illustrated in Fig. 1, represent three dichotomies for the systematisation of uncertainty visualisation approaches; intrinsic/extrinsic, coincident/adjacent, and static/dynamic.

The first categorisation distinguishes between intrinsic techniques, which comprise manipulation of the existing symbology through changes to the visual variables,

Fig. 1 Uncertainty visualisation cube Kinkeldey et al. (2014a)



and the extrinsic approaches, which instead add new objects to depict the uncertainty (e.g. grid or buffer). In a literature review by Kinkeldey et al. (2014a), it was concluded that the majority of the 44 examined studies utilised an intrinsic approach with regard to either the variable color (e.g. MacEachren et al. (1998)) or transparency (e.g. Slocum et al. (2003)). However, a few studies that focused on extrinsic uncertainty visualisation were also identified—including Kinkeldey et al. (2014b), where a grid-based approach was utilised.

The coincident/adjacent distinction refers to the organisation of the display—that is, whether the uncertainty parameter is represented in an integrated or separate display. While most of the studies in the review by Kinkeldey et al. (2014a) utilised coincident techniques, a number of them involved a comparison of the two approaches (e.g. Viard et al. (2011)).

In the static/dynamic dichotomy, a distinction is made between the traditional, static maps and dynamic maps where animations or a high degree of interaction are utilised. An example of a display with dynamic elements could be the ability to switch between showing data and the appertaining data uncertainty Kinkeldey et al. (2014a). According to Kinkeldey et al. (2014a), most of the studies included in the review applied traditional static visualisations. However, Gerharz and Pebesma (2009) and Slocum et al. (2003) implemented interactive interfaces and non-interactive animations, respectively.

2.2 *Augmented Reality for Utility Line Visualisation*

The research concerning the utilisation of AR within subsurface utility engineering is prominent—both in relation to the assessment of different visualisation methods, system development, and evaluation of AR solutions for specific prerequisites and target audiences Hansen (2009); Hansen et al. (2021a, b); Schall et al. (2009); Su et al. (2013). This enhanced version of reality with a graphic overlay display can facilitate processes by providing the necessary overview, both in the planning phase and at the excavation site Schall et al. (2009). AR solutions can thus help to strengthen the exchange of information from the virtual world to the real world and thereby improve the utilisation of utility information Fenais et al. (2019). Behzadan et al. (2015), in addition, points out that visualisations enhance a user's cognition and learning experience, as well as communicate information about complex phenomena.

According to Fenais et al. (2019), the data used in AR applications applied within the field of underground construction of utility lines has three main uses:

1. Visualisation of objects that have not yet been constructed (design and planning).
2. Visualisation of objects that are hidden—including blocked or buried elements.
3. Visualisation of metadata associated with the objects in question.

So far, the majority of the studies in this field have been focusing on the exposure of underground infrastructure—for instance, the research by Muthalif et al. (2022), which revolves around the utilisation of Mixed Reality to mitigate visual perceptual

challenges related to the visualisation of underground utilities. However, the inherent uncertainties of the cables which constitute a significant part of the appertaining metadata are generally omitted in the visualisations. The aim of this research is thus to present an uncertainty-aware method for visualising utility lines in augmented reality.

Solutions that visualise subsurface utilities exist in the present day in commercial products such as VGIS¹ and Trimble SiteVision². These solutions can collect and visualise different forms of data, relying on technology such as 3D scanning, photogrammetry, and GIS data. A common denominator between the products is their use of handheld smart devices connected to an external GNSS receiver and IMU, which enables high-accuracy georeferencing of the data in augmented reality.

The possibility of georeferencing geographical data in augmented reality through the use of GNSS and IMU technologies has thus been thoroughly demonstrated by commercial products to be attainable in practical settings. This study will not seek to replicate this vital aspect of a functioning system but will instead focus on the lesser studied aspect of uncertainty visualisation in augmented reality. Data uncertainty is a fundamental issue, which is not addressed in either commercial products or in research related to augmented reality.

Fundamentally, the goal of this study is to build a prototype solution that works in a similar manner to existing commercial products, though the focus will be on uncertainty visualisation.

3 Data and Case Description

The Danish register of utility owners (Ledningsejerregisteret, LER) was established in 2005 with the overall aim of easing the administrative burden associated with the exchange of utility data and thereby reducing the total costs induced by excavation damage. Within the Danish utility field, a raft of changes has been implemented in the past few years, which both relate to the legislative, administrative, and technical aspects. The new system is referred to as LER 2.0 (Lovforslag 2022).

Figure 2 shows a process diagram that illustrates the change in the workflow for the exchange of utility data when transitioning from the current LER version to the new LER 2.0.

A significant difference emerges in the delivery phase (steps 3–4). At the present moment, the utility owners send the information directly to the requester in varying formats, whereas the future practice is deliveries of data in a uniform GML format which is structured according to the defined data model for LER 2.0. Furthermore, a number of requirements are imposed in order to improve and standardise the utility data—comprising a demand for information regarding vertical and horizontal coordinates, dimensions of the utility line, accuracy classes, etc. These changes became effective from 1 July 2023 (Høring 2019).

¹ <https://www.vgis.io/>.

² <https://sitevision.trimble.com/>.

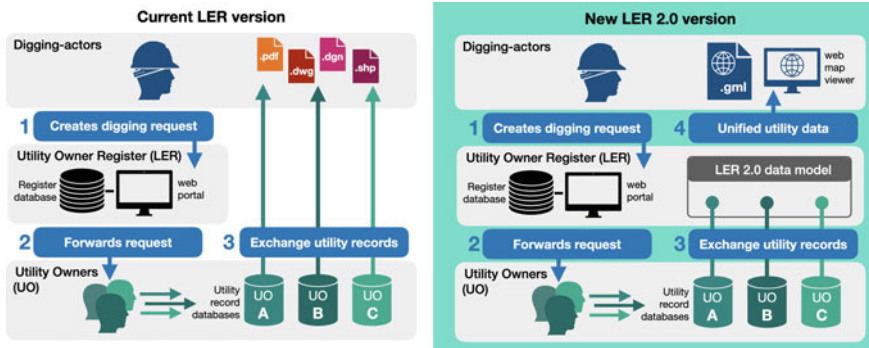


Fig. 2 The workflow for the current LER and the new LER 2.0 version (Hansen 2009)

The new requirements, in particular the information regarding the positional accuracy of the utilities, help to increase the credibility of utility data and provides an opportunity to improve the communication of uncertainties. In this study, the accuracy classes thus form the basis of the uncertainty visualisations. The accuracy classes are determined as an overall assessment of various parameters—including the burial method, materials of the utility line, and the procedure used for measuring—and are divided into five categories (Vejledning 2021):

1. The utility line deviates up to 0.25 m.
2. The utility line deviates up to 0.50 m.
3. The utility line deviates up to 1.00 m.
4. The utility line deviates up to 2.00 m.
5. The utility line deviates more than 2.00 m.

The available data thus constitute an expedient basis for utilising a deterministic approach to uncertainty visualisation as opposed to the related studies by Li et al. (2015) and Su et al. (2013), which both are based on probabilistic error bands and Scholtenhuis et al. (2018), who classified the utility data with reference to four identified location quality parameters; standards/handbooks, estimated through professional experience, surveyed and unknown.

3.1 Aalborg Case

In order to use LER 2.0 data in the development part of this project, an excavation inquiry for utility information in the Danish city of Aalborg was carried out. The content of this is illustrated in Fig. 3. As previously mentioned, the obligation to deliver data in LER 2.0's GML format effective from 1 July 2023. Therefore not all utility lines in the area are included. However, it is assessed that there is sufficient data to demonstrate a prototype.



Fig. 3 An overview of utility data in the center of Aalborg

4 Method

The prototype development of this study is made using Unity, which is a platform-independent game engine intended for the development of games in both 2D and 3D. A primary reason for this is Unity's built-in AR Foundation framework, which enables rapid AR prototype development. In addition, Unity offers a variety of other functionality needed for the development, including a method for porting the prototype to both Android and iOS.

As the georeferencing aspect is omitted from this study, a simple emulation functionality has been conceived. To ensure accurate alignment of AR visualisations with the real world, the plane detection functionality offered by Unity's AR Foundation framework is utilized. Consequently, this prototype is only suited for flat surfaces, preferably those with textures, in order to minimize any undesirable visual fluctuations or inconsistencies caused by the absence of a stable anchor. Given that the focus of this research is on the uncertainty visualisation aspect, this is deemed acceptable.

Since this emulation method doesn't depend on georeferencing, it requires a method to identify the specific underground infrastructure that is to be visualised. To achieve this, polygons intersecting the LER data were manually defined to encapsulate areas representative of different situations that might showcase visualisation during different circumstances, as exemplified in Fig. 4. Specifically, some polygons were defined for their high density of underground utilities, some for simplicity, and some for a representative balance. Seven of these polygons were defined, each with its own unique identifier. These polygons will later be used to select the intersecting lines for visualisation.

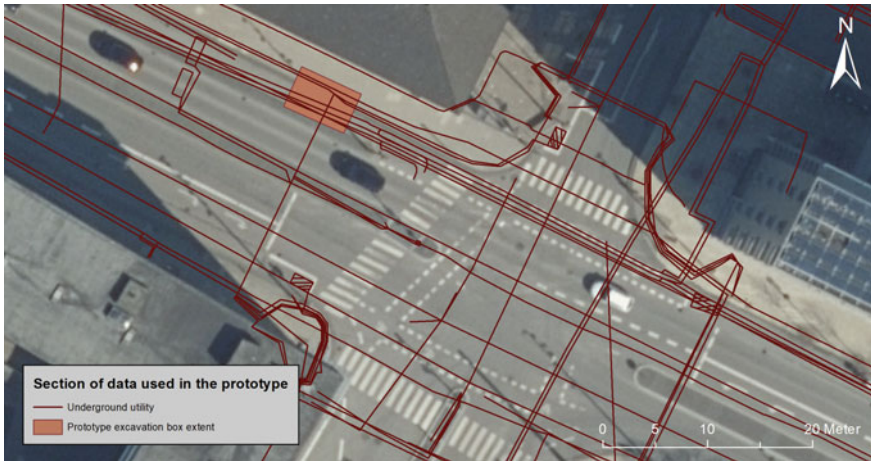


Fig. 4 Section of the data used for prototype development

4.1 *Creating the Baseline*

Since the goal is to give the target group of digging-actors a better understanding of the underground utilities and their uncertainties, the proper conveyance of effective depth cues is paramount. For this purpose, the underlying visualisation method used is the excavation box method, which has been demonstrated as being effective for advancing depth perception in a limited vicinity Eren and Balcisoy (2018); Schall et al. (2009).

By use of the touch display, the user can select a location on the screen, where the excavation box should be generated. This is done by translating the point on the screen to the estimated plane surface utilising the plane detection part of the AR Foundation framework. Following this, an intersection between the bounds of the excavation box and the LER data can be made to find a selection of the underground utilities that are to be visualised. In the case of the prototype, the user-selected coordinates are defined in local space, and the intersection is instead made between one of the seven predefined polygons and the LER data.

From the user-selected location, points containing the bounds of the excavation box are extrapolated and used to create the geometry of the hole. Figure 5 shows an excavation box that has been generated from a set of extrapolated points. Depth lines at 0.5 meter intervals have been added to provide a numerical depth cue for the user.

In order to visualise the underlying 2.5D LER data in AR, it has to be transformed into 3D geometry. This is done by extruding the lines, which represent pipes, such that their virtual dimensions reflect their physical ones, which is made possible by the fact that the features have a diameter attribute in the LER data. In the case of absence of either the depth or diameter attributes, default values are used based on the

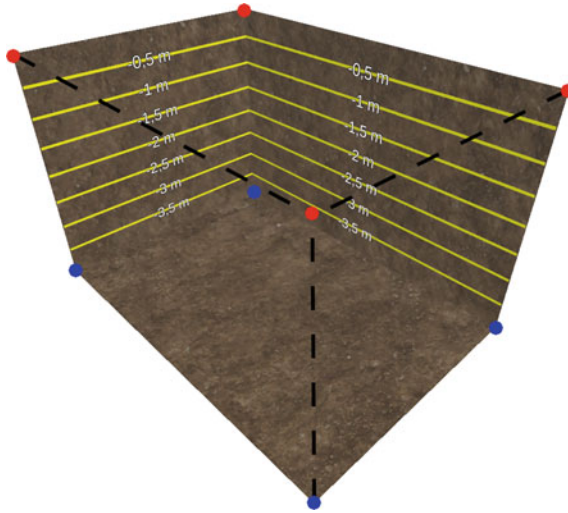


Fig. 5 The baseline excavation box method

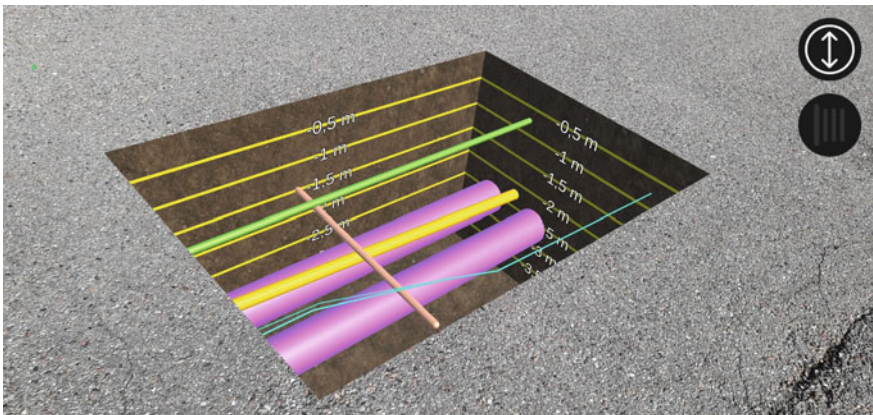


Fig. 6 The baseline excavation box method visualised with utility lines in augmented reality

pipe type. Figure 6 shows an excavation box containing underground infrastructure visualised in AR. The excavation box effect, where objects are only visible when viewed through the hole in the ground, is achieved by using stencil buffer shaders.

4.2 Adding Uncertainties

The novelty of this study is the addition of uncertainty visualisation, which in this case is enabled by the accuracy class field in the LER data model. The overall goal

is to create an intuitive graphical 3D representation of the uncertainties, as opposed to LER's standard non-graphical method (see Fig. 3). The visualisation methods developed for this are put in the context of Kinkeldey's Uncertainty Visualisation Cube.

4.2.1 Coincident/Adjacent

The currently publicly available viewer for LER data relies on an adjacent view to display the accuracy classes in plain text. As the purpose of this research is to create a more intuitive understanding of the uncertainties, a coincident visualisation method will be employed. This entails that the uncertainties will be an integrated part of the visualisation alongside the underground infrastructure, rather than in a separate view, such as table-based metadata.

4.2.2 Extrinsic/Intrinsic

It is assessed that an extrinsic visualisation method is most suitable. The pipe objects already carry information about the physical pipe diameter and the colorisation represents the pipe type (e.g., gas, water, electricity). As such, it would be problematic to add additional intrinsic symbology. An additional intrinsic parameter such as transparency might cause the pipes to be overlooked, while texture patterns might be misinterpreted to have other industry-specific connotations. In both cases, it would be more difficult to interpret the pipe type, and the geographical extent of the pipes would be difficult to intuit. For this reason, an extrinsic visualisation method is used, which involves adding supplementary objects to represent the uncertainty.

As the goal is to convey the geographical extent of the uncertainties, size is naturally the most intuitive visual variable that can be used. The size of the 3D geometry that is generated to represent the uncertainty is therefore based on the maximum extent of the accuracy class attribute—i.e., the uncertainty of a pipe classified as '<1.00m' will have a radius equal to the pipe extruded by 1 m.

As the nature of the uncertainties causes them to extend beyond the pipes, and thus obscure them, the visual variable of transparency is applied to the uncertainty geometry as depicted in Fig. 7. In the prototype, uncertainties are generally visualised using the same color in an effort to reduce information overload.

4.2.3 Static/Dynamic

While AR inherently possesses dynamic traits in the sense that user movement results in an altered view, the prototype will contain several additional dynamic aspects.

An anticipated challenge that made itself apparent during the prototype development was information overload as a result of densely packed infrastructure. The combination of multiple uncertainties, each represented by a transparent cylinder,

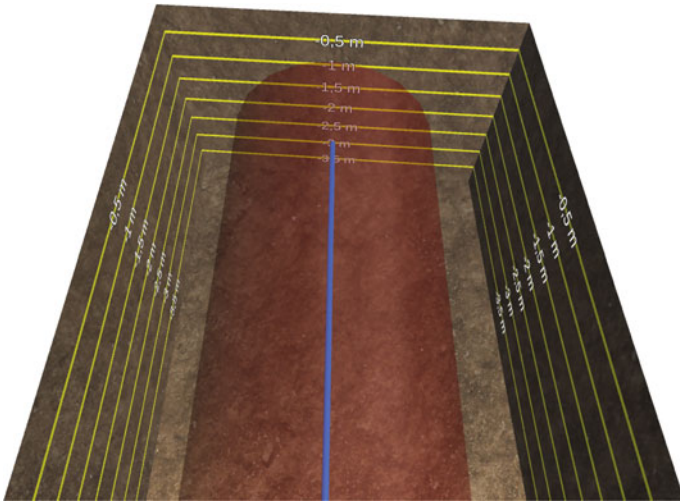
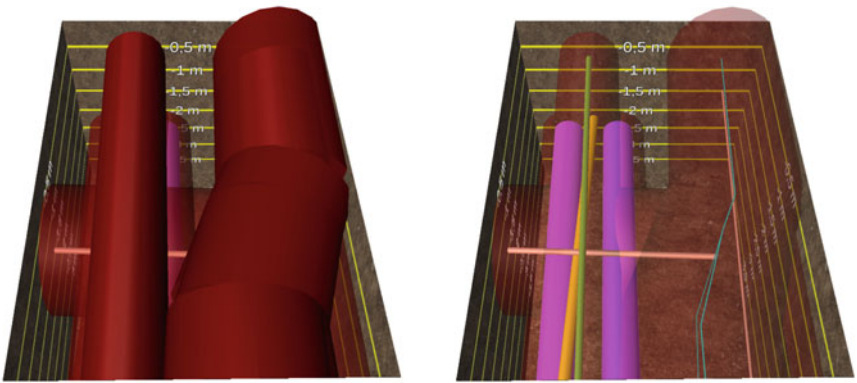


Fig. 7 A single utility pipe with its uncertainty visualised



(a) Stack-method, where multiple transparent uncertainties are layered on top of each other.

(b) Top-method, where only the first user-facing layer of transparent uncertainties are rendered.

Fig. 8 Two different visualisation methods

can in some cases cause the impression of an opaque object, as seen in Fig. 8a. This occurs when multiple pipes are run in parallel to each other and are thus registered at the same or nearly the same locations.

The method developed for overcoming this challenge is the addition of dynamically being able to switch to a different mode of visualisation, as is seen in Fig. 8b. This visualisation method is referred to as ‘top’, while the method depicted in Fig. 8a is referred to as ‘stack’. The top visualisation method only renders the first layer

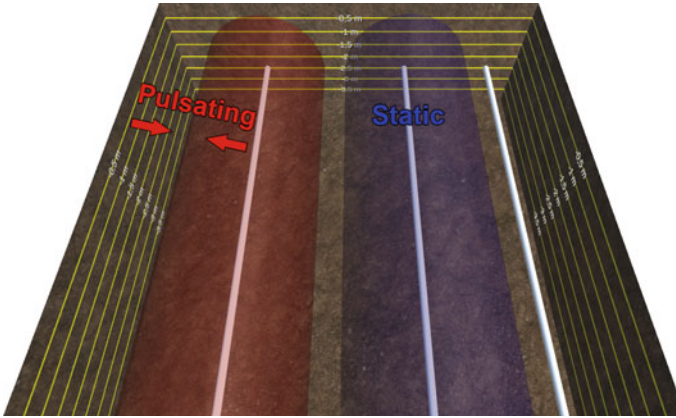


Fig. 9 Three different visualisation methods for utility lines with >2.00 m uncertainty. From left to right is seen a pulsating dynamic uncertainty, an alternatively coloured static uncertainty, and the omission of uncertain visualisation

of uncertainty, rather than rendering them all stacked on top of each other. When using this setting, the uncertainties of a high density of pipes will not obscure the underlying infrastructure, and it is thus a tradeoff between the completeness of information and a reduction in information overload. In cases with less infrastructure, it might be advantageous to visualise uncertainties stacked on top of each other, as the uncertainty of the underlying pipes would otherwise be unknown.

Another challenge arises from the fact that the accuracy classifications include the ' >2.00 m' and 'unknown' values. These uncertainties cannot be represented accurately by geometry as with the other classifications, since no meaningful bounds can be defined. As both classifications share this trait, they are treated and visualised as being of a single classification. To convey this level of inaccuracy, an additional visual variable is needed. In this study, three different methods for visualisation of uncertainties have been developed for this purpose, as depicted in Fig. 9.

The first of these is a pulsating animation that alternates the uncertainties diameter between 1,33 and 2 m following a sinus function. The second is a static uncertainty of 2 m that has an alternative color relative to the more accurate uncertainties. In both these cases, the intent is to draw the user's attention to the fact that data with excessively high uncertainty is present. The third visualisation method is simply to omit the visualisation of the uncertainty, with the rationale that uncertainties of this magnitude provide little practical information, while still contributing unnecessarily to information overload. Once the user is aware of the problematic data, this gives them the option to simply leave it out, giving a clearer picture of the remaining infrastructure.

An overview of the settings that can be dynamically switched between is seen in Table 1.

Table 1 Available settings

| | | |
|-----------|----------------|-----------------------|
| | Layer handling | >2.00 m uncertainties |
| Setting 1 | Top | Pulsating |
| Setting 2 | Stack | Alt. color |
| Setting 3 | Off | Off |

5 Results and Evaluation

The visualisation methods were presented in a focus group interview consisting of five participants who all had some experience with underground utilities, either in the form of practical field work or the planning phase.

Initially, a short presentation was held to convey the motivation of the study, the principles behind the visualisations, and the framework and delimitations of the tool. The demonstration then commenced with a series of staged scenarios, each taking place in various virtual locations with different circumstances and employing different visualisation methods. To conclude the focus group interview, the respondents were asked a series of general questions regarding the tool’s potential uses and possibilities for further development.

5.1 *Feedback Concerning the General Setup and Visualisation Method*

The initial scenario consisted of a baseline visualisation without uncertainties (Fig. 6), as to better evaluate the basic setup. The participants found the depth indicator lines helpful and suggested vertical lines to assist the users in horizontal distance estimation. The use of the standardised LER color scheme for the utility lines was found appropriate, though the participants did point out that sharp sunlight in a practical setting might make it difficult to interpret the colors.

5.2 *Feedback Concerning Uncertainty Visualisation*

When presented with the different visualisation methods for utility lines with >2.00 m uncertainties, the participants initially misinterpreted the ‘pulse’ visualisation as being between an interval (i.e. 0.5–1.5 m uncertainty). While this confusion was easily cleared up, it did highlight the issue of visualising an uncertain uncertainty. Between the ‘pulse’ and ‘blue’ visualisation methods, they all preferred the ‘pulse’ method, due to its ability to catch the eye and signal to the user that they should be aware of the high degree of uncertainty. The participants expressed the need to be

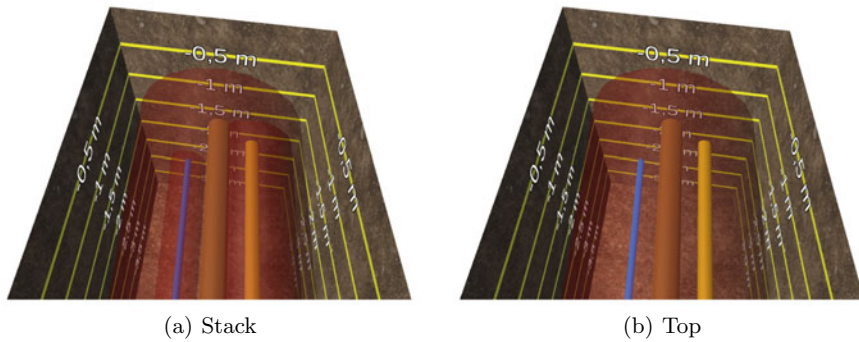


Fig. 10 A scenario in the focus group interview with stack and top

able to toggle the >2.00 m uncertainty between on and off. An intrinsic visualisation alternative was proposed, where the utility lines ‘blink’ on and off.

Two different scenarios were presented for the purpose of evaluating the ‘top’ and ‘stack’ methods. In the scenario depicted on Fig. 8, the ‘top’ method was preferred by the participants, while the ‘stack’ method was preferred in the scenario depicted on Fig. 10. This was consistent with the initial hypothesis. The participants likewise reasoned that while the ‘stack’ method provided more information in simple scenarios with few utility lines, it is subject to information overload when the utilities are more densely packed.

Finally, one of the presented scenarios exhibited a single utility type with relatively low uncertainties, as depicted in Fig. 11. In this simple scenario, the participants found the visualisation useful and easy to comprehend.

5.3 Thoughts on Potential Uses and Possibilities for Further Development

There was consensus among the participants that the inadequate data quality of the available utility data at present is the biggest challenge. However, with the use of standard depths in cases where depth data is not available and the prospect of ongoing improvements in data quality, the respondents were generally positive about the tool’s potential.

Apart from its potential use for contractors and digging actors in the field, one respondent suggested using the tool earlier in the process, during the design and planning phase. Another respondent noted an interest in having the ability to plot and view pipelines at the project stage.

In terms of more practical concerns, the participants repeatedly expressed a desire to be able to mix and match different settings beyond what was already possible, to reduce information overload. This included the option to turn off different types of

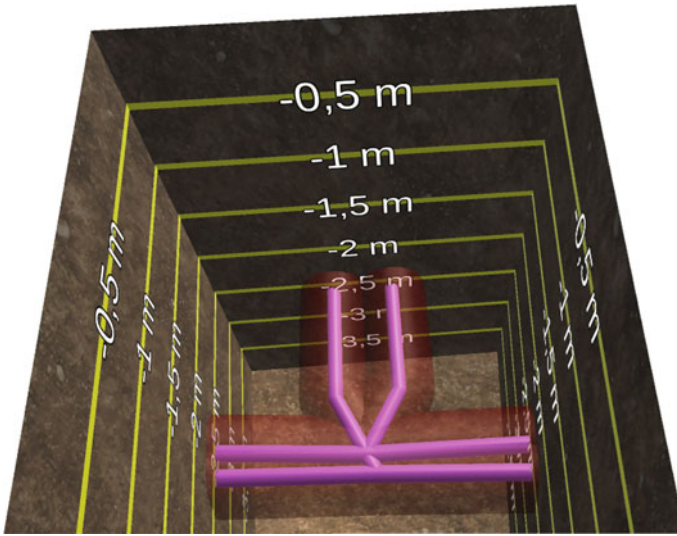


Fig. 11 A scenario in the focus group interview with a single utility type

utilities to get a better overview, as well as being able to change the uncertainty colors to reflect the utility type instead of a uniform red color. The participants also expressed a need for signature explanation as well as a means to select and display metadata of the utility lines.

6 Discussion and Outlook

In excavation work, the depth of the subsurface infrastructure is crucial, which creates a need to move visualisations into the third dimension. In this regard, visualising subsurface infrastructure in augmented reality shows promise in terms of enhancing the intuitive understanding of the hidden objects' position, extent, and uncertainty. The focus of this study was to visualise the uncertainties for the purpose of making digging-actors aware of them so that they might better interpret the level of caution they need to take. For this purpose, an uncertainty-aware method for visualising utility lines in augmented reality was proposed. The participants of the focus group interview had a positive outlook on the potential of the methods employed in this study, though further research and development are needed for practical utilisation. The developed system lacks the georeferencing component, making it unsuited for even early-stage field testing. In further research, the involvement of digging-actors is advisable to establish their needs and ensure fit-for-purpose visualisation methods. In addition to this, more thorough evaluation methods should be employed—including usability experiments in which the users have to perform specific tasks. In this manner,

the visualisation techniques can be compared in terms of parameters such as error rate and completion times.

Further research is needed to develop methods suitable for handling layered uncertainties, as this is the most significant contribution to information overload in the context of subsurface infrastructure and augmented reality. Segmentation and generalisation of infrastructure in a manner that conveys as much information as possible while still being comprehensive will likely play a role. In relation to this, it should be highlighted that the repeated request for dynamic visualisation options by the participants hints that no single option for visualisation is sufficient to provide a comprehensive overview of every situation. Given the nature of AR, interactive and dynamic methods that are less commonly employed in 2D visualisation may also be advantageous.

A prerequisite for visualising subsurface infrastructure is adequate depth data availability. The data used for the prototype proved to contain only sparse depth data, which resulted in the default depth values being frequently used. This is a significant barrier to a complete and useable system. According to several participants of the focus group interview, it is common practice to put certain subsurface infrastructure at certain depths. In future research, this tacit knowledge should be further investigated, such that default depths may provide a better estimate of actual depth. It may also be advantageous to add a visual variable to indicate when the depth is not based on direct data.

References

- Beck AR, Cohn AG, Parker J, Boukhelifa N, Fu G (2009) Seeing the unseen: delivering integrated underground utility data in the UK. In: The international archives of the photogrammetry, remote sensing and spatial information sciences, pp 1–8. https://www.isprs.org/proceedings/XXXVIII/3_4-C3/Paper_GeoW09/paper07_beck.pdf
- Behzadan AH, Dong S, Kamat VR (2015) Augmented reality visualization: a review of civil infrastructure system applications. *Adv Eng Inf* 29:252–267. <https://doi.org/10.1016/j.aei.2015.03.005>
- Boukhelifa N, Duke DJ (2007) The uncertain reality of underground assets. In: The international archives of the photogrammetry, remote sensing and spatial information sciences. https://www.isprs.org/proceedings/XXXVI/4-W45/PDF/04_Boukhelifa.pdf
- Eren MT, Balcisoy S (2018) Evaluation of X-ray visualization techniques for vertical depth judgments in underground exploration. *Vis Comput* 34:405–416. <https://doi.org/10.1007/s00371-016-1346-5>
- Fenais A, Ariaratnam ST, Ayer SK, Smilovsky N (2019) Integrating geographic information systems and augmented reality for mapping underground utilities. *Infrastructures* 4(60). <https://doi.org/10.3390/infrastructures4040060>
- Gerharz LE, Pebesma EJ (2009) Usability of interactive and non-interactive visualisation of uncertain geospatial information. *Geoinformatik* 4:223–230
- Hansen LH, Pedersen TM, Kjems E, Wyke S (2021b) Smartphone-based reality capture for subsurface utilities: experiences from water utility companies in Denmark. In: The international archives of the photogrammetry, remote sensing and spatial information sciences. XLVI-4/W4-2021, pp 25–31. <https://doi.org/10.5194/isprs-archives-XLVI-4-W4-2021-25-2021>

- Hansen LH (2009) Augmented reality for subsurface utility engineering: exploring and developing 3D capture and AR visualization methods for subsurface utilities. Aalborg Universitetsforlag. <https://doi.org/10.54337/aau466407995>
- Hansen LH, Fleck P, Stranner M, Schmalstieg D, Arth C (2021a) Augmented reality for subsurface utility engineering, revisited. *IEEE Trans Vis Comput Graph* 27(11):4119–4128. <https://doi.org/10.1109/TVCG.2021.3106479>
- Høring af udkast til bekendtgørelse om registrering af ledningsejere og udlevering af standardiserede ledningsoplysninger gennem Ledningsejerregistret. The Danish agency for data supply and infrastructure (2019). <https://www.ft.dk/samling/20181/almdele/EFK/bilag/202/2027328/index.htm>
- Kinkeldey C, MacEachren AM, Schiewe J (2014a) How to assess visual communication of uncertainty? A systematic review of geospatial uncertainty visualisation user studies. *Cartographic J* 51(4):372–386. <https://doi.org/10.1179/1743277414Y.0000000099>
- Kinkeldey C, Mason J, Klippel A, Schiewe J (2014b) Evaluation of noise annotation lines: using noise to represent thematic uncertainty in maps. *Cartography Geogr Inf Sci* 41(5):430–439. <https://doi.org/10.1080/15230406.2014.949868>
- Lanka M, Butler A, Sterling R (2001) Use of approximate reasoning techniques for locating underground utilities. *Tunn Undergr Space Technol* 16:13–31. [https://doi.org/10.1016/S0886-7798\(01\)00058-X](https://doi.org/10.1016/S0886-7798(01)00058-X)
- Li S, Cai H, Kamat VR (2015) Uncertainty-aware geospatial system for mapping and visualizing underground utilities. *Autom Constr* 53:105–119. <https://doi.org/10.1016/j.autcon.2015.03.011>
- Lovforslag nr. L 110 (2022) Folketinget 2021–22, Forslag til Lov om Ledningsejerregistret. The Danish agency for data supply and infrastructure. https://www.folketingstidende.dk/samling/20211/lovforslag/L110/20211_L110_som_fremsat.pdf
- MacEachren AM, Brewer CA, Pickle LW (1998) Visualizing georeferenced data: representing reliability of health statistics. *Environ Plann A: Econ Space* 30(9):1547–1561. <https://doi.org/10.1068/a301547>
- Makana L, Metje N, Jefferson IF, Rogers CDF (2016) What do utility strikes really cost? School of engineering. University of Birmingham. <http://rgdoi.net/10.13140/RG.2.2.14386.50883>
- Muthalif MZA, Shojaei D, Khoshelham K (2022) Resolving perceptual challenges of visualizing underground utilities in mixed reality. In: *The international archives of the photogrammetry, remote sensing and spatial information sciences*. XLVIII-4/W4-2022, pp 101–108. <https://doi.org/10.5194/isprs-archives-XLVIII-4-W4-2022-101-2022>
- Schall G, Mendez E, Kruijff E, Veas E, Junghanns S, Reitinger B, Schmalstieg D (2009) Hand-held augmented reality for underground infrastructure visualization. *Pers Ubiquitous Comput* 13(4):281–291. <https://doi.org/10.1007/s00779-008-0204-5>
- Scholtenhuis LL, Duijn X, Zlatanova S (2018) Representing geographical uncertainties of utility location data in 3D. *Autom Constr* 96:483–493. <https://doi.org/10.1016/j.autcon.2018.09.012>
- Slocum TA, Cliburn DC, Feddema JJ, Miller JR (2003) Evaluating the usability of a tool for visualizing the uncertainty of the future global water balance. *Cartography Geogr Inf Sci* 30(4):299–317. <https://doi.org/10.1559/152304003322606210>
- Smith Mason J, Retchless D, Klippel A (2017) Domains of uncertainty visualization research: a visual summary approach. *Cartography Geogr Inf Sci* 44(4):296–309. <https://doi.org/10.1080/15230406.2016.1154804>
- Su X, Talmaki S, Cai H, Kamat VR (2013) Uncertainty-aware visualization and proximity monitoring in urban excavation: a geospatial augmented reality approach. *Vis Eng* 1(1):2. <https://doi.org/10.1186/2213-7459-1-2>
- Tanoli WA, Sharafat A, Park J, Seo JW (2019) Damage prevention for underground utilities using machine guidance. *Autom Constr* 107:102893. <https://doi.org/10.1016/j.autcon.2019.102893>
- Vejledning om lov om registrering af ledningsejere og udlevering af ledningsoplysninger gennem Ledningsejerregistret. The Danish agency for data supply and infrastructure (2021). <https://retsinformation.dk/eli/retsinfo/2021/9658>

- Viard T, Caumon G, Lévy B (2011) Adjacent versus coincident representations of geospatial uncertainty: which promote better decisions?. *Comput Geosci* 37(4):511–520. <https://doi.org/10.1016/j.cageo.2010.08.004>
- Weiskopf D (2022) Uncertainty visualization: concepts, methods, and applications in biological data visualization. *Front Bioinform* 2. <https://doi.org/10.3389/fbinf.2022.793819>

Immersive Virtual Reality to Verify the As-built State of Electric Line Networks in Buildings



Julius Knechtel, Weilian Li, Yannick Orgeig, Jan-Henrik Haunert, and Youness Dehbi

Abstract Immersive virtual reality (IVR) allows viewing abstract concepts and entities in a three dimensional (3D) visuospatial environment. In this paper, we innovatively introduced IVR technology into the verification of the as-built state of electric line networks in buildings. On the one hand, using a reasoning-based estimation of electric networks as a starting point, we demonstrated the usability of IVR technology for verifying installed utilities in buildings. On the other hand, we established the communication between the Reasoner and the practitioner and also simulated the verification action of electric line networks in buildings in the real world. The principal findings of this work pave the way for a subsequent and systematic evaluation of the different reasoning strategies for estimating and generating the as-built state of building utilities.

Keywords Immersive virtual reality · Electric line network · As-built BIM · Reasoning · GIS

This article was selected based on the results of a double-blind review of the full paper.

J. Knechtel (✉) · W. Li · Y. Orgeig · J.-H. Haunert
Institute of Geodesy and Geoinformation, Geoinformation Group, University of Bonn,
Bonn, Germany
e-mail: knechtel@igg.uni-bonn.de

W. Li
e-mail: li@igg.uni-bonn.de

Y. Orgeig
e-mail: orgeig@igg.uni-bonn.de

J.-H. Haunert
e-mail: haunert@igg.uni-bonn.de

Y. Dehbi
Computational Methods Lab, HafenCity University Hamburg, Hamburg, Germany
e-mail: youness.dehbi@hcu-hamburg.de

1 Introduction

In the last decade, Building Information Modelling (BIM) has been used as a supporting technology for smart cities in the whole life cycle of a building, i.e., design, construction, operation and management, and renovation or demolition (Costin et al. 2018; Pezeshki and Ivani 2018; Moretti et al. 2021). While as-planned BIM models corresponding to the design and planing of new construction sites exist en masse, as-built BIM models for existing constructed objects are scarce. Existing buildings in Europe have a particularly high proportion of old buildings and architectural heritage (EU Commission 2022). For most of them no BIM models exist, and even the design drawings are missing, which poses challenges for maintenance and renovation. With the upsurge of smart cities and digital twins, surveying techniques like 3D laser scanning are used to scan buildings and generate the corresponding BIM data (Yang et al. 2020) leading to digital twins of existing real objects. However, the structure and elements hidden in the wall, i.e., the layout and routing of pipes and wires, cannot be observed by such techniques.

Considering the smart city scenario, however, the in-depth management of building facilities at a granular level requires highly detailed and semantically rich digital building models (Xia et al. 2022; Zhu and Wu 2021). In this context, Dehbi et al. (2022) proposed an incremental constraint-based reasoning approach for estimating the as-built electric line routing in buildings. This method allows to unambiguously determine the electric network incrementally with a minimum number of local measurements. The experimental results on real-world buildings show that the approach can save up to 80% of the electric line verification effort and effectively support the extension of existing BIM models with such hidden utilities. In this manner, existing semantically rich 3D city models, e.g. CityGML (Gröger et al. 2012), could be further enriched by the newly acquired building utilities. Nevertheless, the reasoning algorithm still lacks an intuitive visual interface and interaction, leading to some cognitive difficulties for inspectors in determining the layout of the electric line networks in buildings.

For the task at hand, we opt for immersive virtual reality (IVR) in order to fill the gap between the Reasoner, which suggests an electric line for measurement to the human inspector, and the actual localization in the existing buildings of the particular line. In this context, IVR allows for viewing abstract concepts and entities in a 3D visuospatial environment which fits well into the verification of the as-built state of electric line networks. With the advances in computer graphics and the rise of consumer-grade VR devices (e.g. Oculus Quest), IVR characterized by immersion, interaction, and imagination has driven its shift from gaming to more diverse applications (Hamilton et al. 2021), such as civil engineering (Getuli et al. 2020), architecture design (Lach et al. 2020; Schiavi et al. 2022) and disaster education (Sermet and Demir 2019). To the best of our knowledge, no previous work has been focused on applying IVR to verify electric line networks in buildings, although this offers significant benefits: (1) IVR provides a visuospatial perspective, which allows inspectors to perform the verification of electric line networks in a high-fidelity

environment; (2) IVR additionally offers the possibility of optically simulated measurement of electric line networks; (3) IVR allows for testing and evaluating different approaches to retrieve the electric network of an existing building in a controlled environment.

The main contribution of this article is the innovative utilization of IVR technology for the verification of the as-built state of electric line networks in buildings. First, we briefly introduce the algorithm of reasoning-based estimation of electric line networks. Second, a communication mechanism is established between the VR and Reasoner. Third, the current proposal and the determined electric line networks are visualized to assist inspectors in identifying the spatial layout of the electric line networks in buildings. Fourth, the verification process is exemplarily simulated using ground truth data of existing buildings.

The remainder of this article is structured as follows: Sect. 2 discusses related work whereas Sect. 3 gives insights into the introduced approach. In Particular, Sect. 3.1 proposes the study workflow of this article, Sect. 3.2 describes the algorithm corresponding to the Reasoner, Sect. 3.3 presents the communication mechanism between VR and Reasoner, and Sect. 3.4 introduces the electric line visualization and interactive verification. Subsequently, Sect. 4 discusses the implementation of the prototype system and experiment results. Section 5 finally summarizes the article and gives an outlook for future research.

2 Related Work

In recent years, VR has captured increasing attention in academia and industry owing to its tremendous potential for various applications. Generally speaking, VR is not a new concept or term, Whyte (2002) refers to the Oxford English Dictionary (OED) pointing out that VR was first used in the 1980s and defines it as a technology that uses computerized clothing to synthesize a shared reality. Subsequently, some scholars started working on VR and pointed out that VR corresponds to the use of computer technology to create the effect of an interactive 3D world in which the objects have a sense of spatial presence (Bryson 2013; Kamińska et al. 2019; Wang et al. 2018). This generally refers to desktop-based VR (DVR), which uses a computer monitor as platform to create a 3D virtual environment without any tracking equipment to support it (Chen et al. 2007; Makransky and Petersen 2019). It only allows the participants to rely on their spatial abilities and a traditional keyboard and mouse hardware to perceive and manipulate the 3D virtual environment (Li et al. 2023). However, mouse and keyboard interaction limits the freedom and realism of the participant's interaction with the environment. Furthermore, DVR is essentially a 2D projection of a 3D scene; Participants generally have a lower degree of sensory experience and cannot intuitively perceive the distance and direction of geographic objects (Boyce et al. 2019; Zhang et al. 2020).

With the advances in computer graphics and the rise of consumer-grade VR devices (e.g., HTC Vive, Oculus Quest, Pico, etc.), DVR accessed through

head-mounted displays (HMD) offers a high level of immersion, which provides the participant with a sense of “being” in a real-world (Radianti et al. 2020). Immersion, presence, and interactivity are regarded as the core characteristics of IVR. The advantage of immersion is particularly clear in the questions of spatial perception (Hruby et al. 2020), the sense of presence has been found to improve procedural memory and spatial tasks (Keil et al. 2021), and interactivity is believed to be the most fundamental aspect of effective learning (Martirosov and Kopecek 2017; Udeozor et al. 2021). Given these advantages, IVR is already being applied in many areas, such as industrial design (Berg and Vance 2017), fire training (Ooi et al. 2019), psychology education (Kamińska et al. 2019), and medical care (Snoswell and Snoswell 2019).

However, some researchers remain negative toward the application prospects of IVR. On the one hand, they question the real value of IVR and whether people can actually benefit from it. IVR implementation is greatly limited by the hardly accessible hardware (Dimitriadou et al. 2021; Li et al. 2023; Zhang et al. 2023). On the other hand, IVR still has many pressing technical issues that need to be resolved, such as limited field of view (FOV) (Wheelwright et al. 2018), motion sickness (Fu et al. 2021; Çöltekin et al. 2020), efficient interaction modes (Li et al. 2023), etc. The FOV is critical in determining the range visible to the human eye in a VR environment, directly influencing the sense of immersion. Limited FOV diminishes user immersion significantly, while increasing FOV requires higher screen size and resolution. Motion sickness is partly caused by a low frame rate, which causes the picture to lag behind the interactive action. Compared with keyboard and mouse, handle-based interaction is more challenging for users who have never used VR, which could affect their motivation during practice.

Nevertheless, most researchers are still optimistic about the value of using IVR. Particularly with the rising quality and availability of geospatial data and the growing popularity of 3D geovisualization, IVR has been identified as a promising approach to improve interactive geographic applications (Keil et al. 2021) or applications in civil engineering-related fields. Particularly over the past two decades, there has been a growing trend in the construction sector toward adopting BIM owing to numerous benefits during the design, planning, construction, and management of new buildings (Volk et al. 2014), which has driven the transition from paper-based to 2D drawings-based and also to the digital model-based construction. To the authors’ best knowledge, BIM-enabled IVR allows users to take building design or construction into a high-fidelity 3D virtual environment that incorporates detailed and semantically-rich building information, without the limitations imposed by viewing 2D drawings and examining the design space (Wang et al. 2018). For example, Boton (2018) proposed a comprehensive framework to support constructability analysis meetings with immersive VR-based collaborative 4D simulation. Banfi et al. (2019) used a Scan-to-HeritageBIM process to create a VR experience of an archaeological site. Getuli et al. (2020) optimized the workspace planning process by simulating a construction activity using IVR and BIM technologies. Although the process of generating building models from point clouds, for example from laser scanners, is a well-established process (e.g. Ochmann et al. (2019)), the enrichment of the resulting model with the

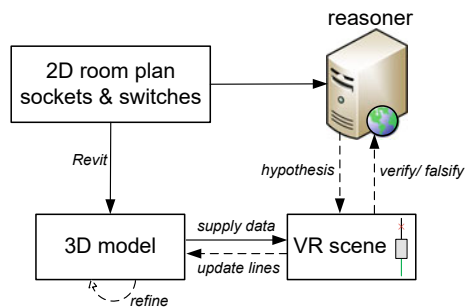
underlying hidden utilities is an open research field. Krispel et al. (2017) generate a hypothesis for the electric network using a laser scanner and panoramic images, but no verification is performed. Dehbi et al. (2022) introduced a method based on Mixed Integer Linear Programming (MILP) for the generation of a hypothesis for the electric line network in buildings and its subsequent verification. The latter is based on different reasoning strategies and is performed using a wire detector, saving up to 80% of the possible measurements.

3 Methodology

3.1 Study Workflow

Figure 1 shows the workflow of our study. We deploy a reasoning-based estimation algorithm of electric line networks called Reasoner. *Revit* (Demchak et al. 2009), a building information modelling software for engineers, architects and designers, is used to create a 3D model of the underlying building which is then displayed in the VR scene. The corresponding 2D room plan serves as input for *Revit* as well as the position of sockets and switches in the housing. Both are also used as input parameters for the Reasoner to examine the possible installation zones, i.e., where the placement of electric lines is allowed with regard to the underlying norms and standards of the respective country. Additionally, a two-way communication mechanism is established between the Reasoner and the VR system. The Reasoner provides a hypothesis of the electric line network and a measurement proposal, which part of the hypothesis should be verified first. The inspector verifies whether the routing is correct in the VR scene and supplies the corresponding result to the reasoner. The latter incorporates this and, subsequently, provides an updated hypothesis as well as the next part of the line network to measure accordingly. This process will be iterated until the entire electric line network in the whole building is confirmed. Furthermore, the electric lines in the 3D model will be updated, to refine and enrich the corresponding BIM model.

Fig. 1 The proposed study workflow of this article. The solid line indicates the data is only supplied once, while the dotted line represents an iterative process



3.2 Reasoning-Based Estimation of Electric Line Networks

To enrich existing, e.g. Geographic Information System (GIS), models leading to semantically augmented BIM models, additional surveying techniques supported by reasoning methods are needed. In this context, Dehbi et al. (2022) introduced a method based on Mixed Integer Linear Programming (MILP) for the generation of a hypothesis for the electric line network in buildings and its verification based on different reasoning strategies. Regularities and repetitive patterns typically characterising man-made objects are also reflected by hidden structures such as electric lines. For the installation of such utilities, a set of standards and recommendations should be followed according to the particular country. Figure 2 exemplarily depicts the recommended zones for electric lines according to the standard DIN 18015–3 in Germany. On this basis, and enhanced with the positions of easily observable objects like sockets and switches, a search space structured as a graph can be established, allowing to sample hypotheses for the electric line network.

This structured search space serves as starting point for an incremental process. The aim is to obtain a verified model of the underlying network, i.e. it has been determined without any remaining ambiguity, using as few measurements as possible. To reach this goal, a position for the next measurement is proposed in order to verify or falsify the current hypothesis using an adequate sensor. Figure 3 illustrates the whole procedure of smart sampling and verification as provided by Dehbi et al.

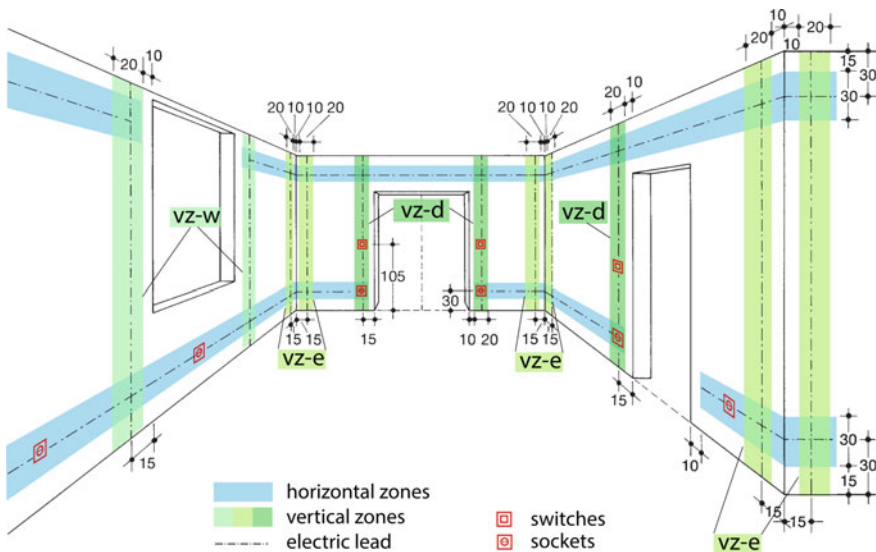


Fig. 2 Recommended installation zones for electric lines according to the German standard DIN 18015–3. Horizontal zones are depicted in blue and vertical zones in green colour. The graphic representation of sockets and switches indicate their preferred locations (Dehbi et al. (2017))

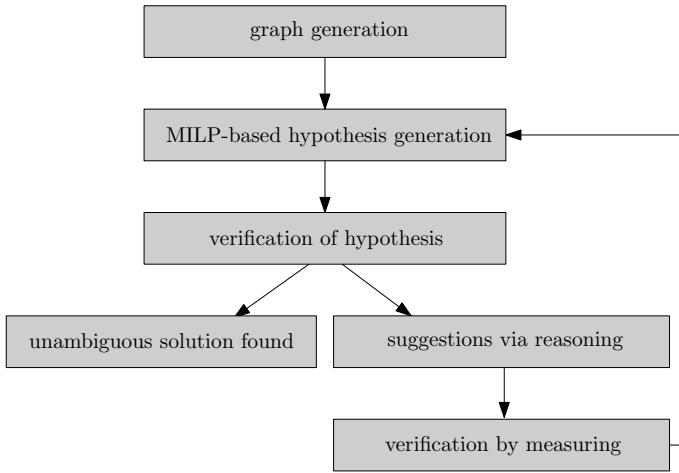


Fig. 3 Overview on the different steps of the incremental hypothesis sampling and verification method for electric networks from Dehbi et al. (2022)

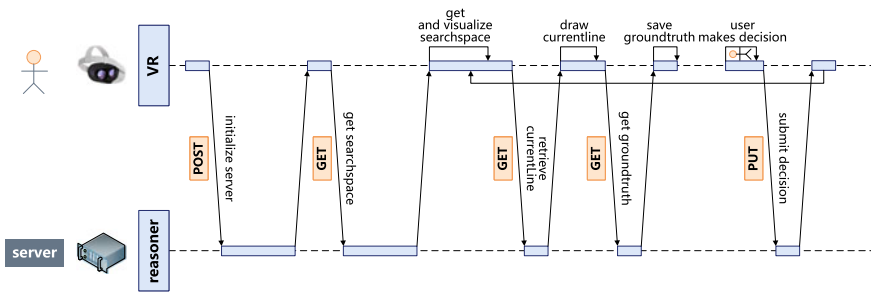


Fig. 4 Sequence diagram of the communication between the VR system and the reasoner as interface for a human inspector

(2022). Six different reasoning strategies have been proposed and evaluated in order to narrow the search space of possible hypothesis economising up to 80% of the measurement overhead.

3.3 VR and Reasoner Communication

To facilitate the process of verifying the as-built state of electric line networks in buildings for the inspectors, the Reasoner is deployed to a server and a smooth communication between the Reasoner and the VR system is established (cf. Fig. 4).

At the beginning of our VR usage process, the user initializes the Reasoner using a POST request, which transfers some essential parameters such as the number of

rooms, the apartment layout, etc. After the initialization is completed, the next step is to retrieve the search space, consisting of all theoretically possible lines within the entire building, using a GET request and then include it into the 3D model of the VR device.

In the next step, we simulate the process of an inspector verifying the lines and making decisions in an immersive 3D environment with the help of two VR handles and the ground truth of the electric network of the underlying building. With a GET request, the line which is proposed for measuring is retrieved from the Reasoner and displayed in the VR scene. In the real world, the measurement of electric lines can be conducted as point-wise measurement utilizing handheld devices, i.e. wire detectors, which indicate whether an electric installation is present. To create the simulation of this line check process, an additional GET request is executed, which queries the ground truth of the currently proposed line. In our simulation, the wire detector will then show in different colours if the current proposed line is present in the wall or not. Subsequently the users can form their decision, whether to verify or falsify the proposed line. After this decision is transmitted to the Reasoner through the PUT endpoint, a new assumption using the newly acquired information is calculated and, thus, the cycle of the working phase starts again. This repetitive process continues until the Reasoner provides an unambiguous and plausible hypothesis for the electric network in the building.

3.4 Electric Line Visualization and Interactive Verification

Figure 5 shows our electric line visualization and interactive verification solution, which aims to support the inspector intuitively and to quickly identify the electric line routing. For visualization purposes, we use *Revit* to create a 3D model based on the 2D room plan, which is the basis for representing the electric line network. The left corner of the building is selected as an anchor point, and all coordinates of the electric lines are transformed with reference to this anchor point. Subsequently, the search space representing all possible electric lines in the complete building is visualized. As mentioned, the Reasoner delivers hypotheses based on reasoning strategies aiming at performing as few measurement steps as possible. Therefore, the line proposed in the current iteration is also displayed in the 3D model in combination with a box collider for the subsequent detection with the VR ray pointer, to simulate the line checking using a wire detector.

For the interaction of the human inspector, the left and right handles are used to detect and confirm the currently proposed line, respectively. The electric line changes its color when the left handle detects the line by collision with the aforementioned box collider, the right handle enables to confirm and submit the user decision to the Reasoner with the help of a request panel. If the simulated wire detector detects a wire at the location of the currently proposed line, the latter changes its color to green accordingly, if there is no line, the representation turns red. This process, hence, simulates the verification of the as-built state of electric line networks in real world buildings.

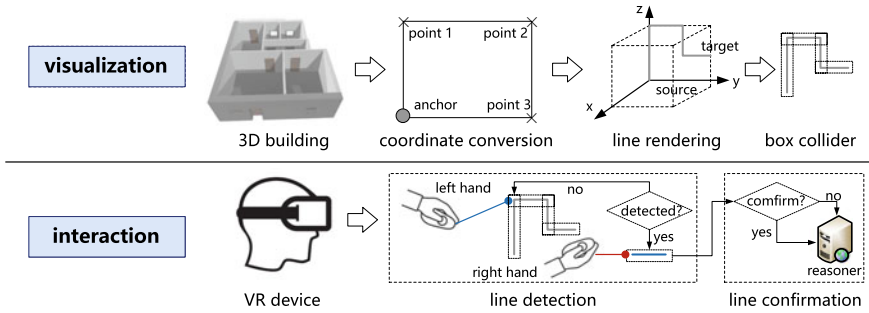


Fig. 5 The workflow for electric line visualization and interactive verification

4 Experiment Results

This section gives insights into the performed experiments and demonstrates the feasibility of our approach. For the sake of replicability, Sect. 4.1 describes the according environment and the used dataset as well as the hardware and software specifications. Section 4.2 presents the visualization engine dedicated to the 3D representation of the electric line as visual interface for the human practitioner. Section 4.3 describes the verification process using the VR device and interacting with the visualization engine.

4.1 Experiment Environment

The setup for implementing our electric line verification approach consists of two main components: the sever for the Reasoner and the VR environment. An *Apache Tomcat server*, an open-source application server, in version 9.0.16 has been set up to host the Reasoner, including two Intel Xeon E5620 processors clocked at a frequency of 2.4GHz with a memory of 8 GB. The *Gurobi Optimizer*¹, which is a solver using mathematical optimization and, hence, necessary to solve the MILP utilized in the reasoning algorithm, was applied in version 9.5.1. The VR system was implemented on a Lenovo Legion R9000P2021H, with a AMD Ryzen 7 5800H processor with Radeon Graphics, 16 GB memory and a NVIDIA GeForce RTX 3060 6 GB Laptop GPU. As 3D rendering engine we used *Unity v2018.3.7f1*, a cross-platform game engine. Additionally, we worked with the *Oculus Quest 2*² as VR device, while implementing the VR Program using the *SteamVR software development kit (SDK)* and *Virtual Reality Toolkit (VRTK)*.

¹ <https://www.gurobi.com/>.

² <https://www.meta.com/quest/products/quest-2/>.

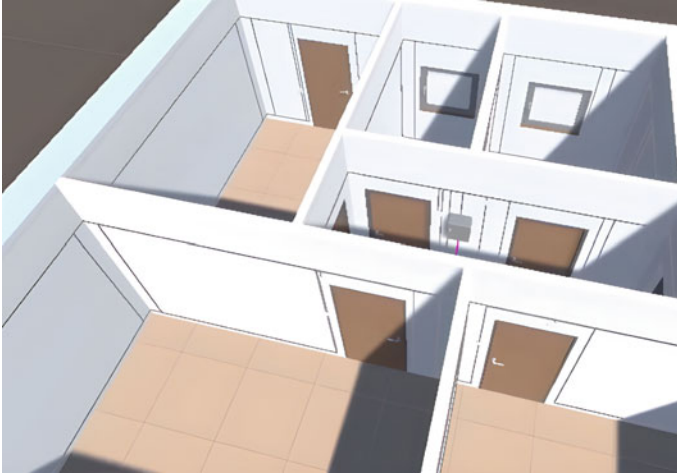


Fig. 6 An excerpt of the visualization in *Unity* of all possible electric lines in the housing considered in our experiment

4.2 *Electric Line Visualization*

Our experiments have been conducted on a real-world scenario of a housing without an existing BIM model. Hence, a 3D model of the housing including the opening locations has been generated using *Revit* based on an existing 2D floorplan. The housing object (cf. Fig. 6) consists of 6 rooms with 25 consumers, i.e. sockets and switches, which are distributed over 4 circuits. Furthermore, background knowledge on the as-is course of the electric lines is available and used as ground truth in order to simulate the measurements conducted by the human inspector. This would of course not be necessary using an Augmented Reality (AR) environment instead of the VR system. In this case, the verification will be performed by a wire detector on-site rather than in a virtual scene.

After applying the first GET request to the server in order to obtain the complete search space, the Reasoner provides the coordinates of the source and target points of each electric line segment, which are converted and rendered as gray lines on the wall. Figure 6 shows all theoretically possible electric lines in the considered apartment. This intuitive 3D visualization helps the inspector to quickly identify and recognize the course of the electric lines.

As aforementioned, the search space, i.e. all possible zones where electric lines could be possibly installed, is defined based on the underlying norms and standards of the particular country. For example, these zones are present next to the corners of a room. Since the Reasoner provides merely the end points of each electric line segment, a visualization of the direct connection of such points would under certain circumstances lead to a false visualization of the underlying electric line as depicted in Fig. 7a. Hence, in order to ensure a correct visualization, an additional corner point

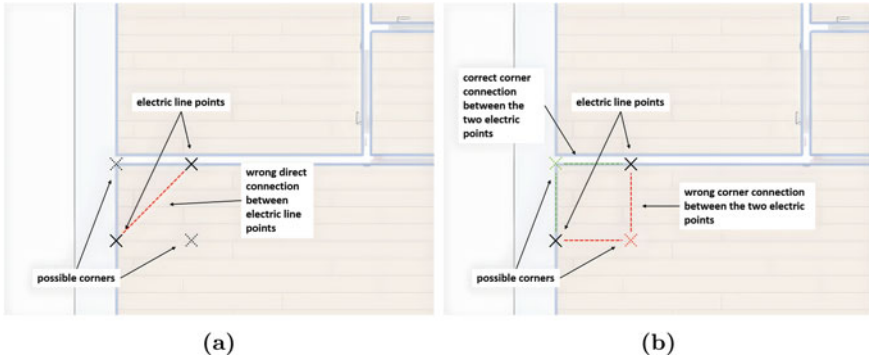


Fig. 7 Challenges in displaying an electric line sharing two adjacent walls with a direct connection **a** have been addressed incorporating the corner point for a correct visualization **b**

is required. Its placement is, however, not unambiguous. To determine the correct point, we reduce the problem into two dimensions since the height of the line is not relevant in this context. After checking the line of interest, i.e. an electric line sharing two adjacent walls, two corner candidate points (black dotted crosses) are available as shown in Fig. 7a. A comparison with the corner points of the building room reveals that the right point is leading to correct visualization of the considered electric line (green dotted) as depicted in Fig. 7b.

4.3 Electric Line Verification

Apart from the 3D visualization of all possible lines in the apartment, with the help of VR handles we also simulated the verification of the as-built state of electric line networks in buildings in the real world, as shown in Fig. 8. The left handle will activate a pointer and create a laser, the latter can detect the current line drawn in magenta in the apartment, which will then change its color based on the ground truth solution, simulating a measurement (cf. Fig. 8a). The corresponding color of the current line turns green if the recommendation line is verified (cf. Fig. 8b), otherwise it turns red.

The right handle is used for the decision confirmation. The inspector can share his decision about the current line with the pointer to the Reasoner, as shown in Fig. 8c. Once the Reasoner receives the decision, it will calculate an updated hypothesis, which will be queried and rendered in the VR environment again. The previously confirmed or rejected line will be displayed in green or red, respectively, as shown in Fig. 8d.

As aforementioned, the Reasoner is able to provide hypotheses performing six different reasoning strategies. In our experiments with the VR system, we followed the “NearPath” strategy from Dehbi et al. (2022). Having the search space as a graph where the nodes represent consumers or intersection points of recommended

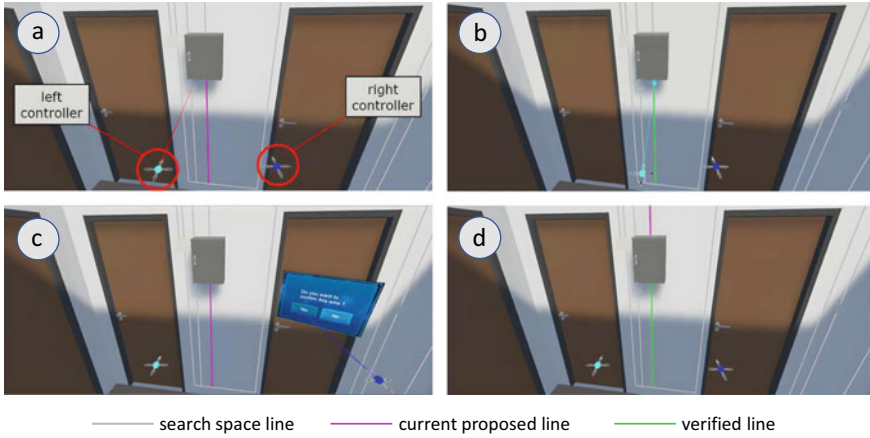


Fig. 8 Screenshots of the interactive verification of electric lines using the controllers from the VR device. The hypothesis sampled from the search space (grey) and suggested from the Reasoner (magenta) has been verified (green) by the inspector pressing the according VR button reflecting the real wire detector

zones and the edges correspond to connections between these nodes, this strategy can be explained as follows: For every visited node, it is checked whether it has an incident edge that neither is part of the current hypothesis, i.e. electric line, nor has been labelled as unused. The first such edge found is returned as the suggestion for the next measurement. In our experiment with our IVR system we were able to simulate the whole verification process based on this reasoning strategy and, thus, demonstrate the practical feasibility of this approach. Furthermore, our implemented system paves the way to compare the different reasoning methods from Dehbi et al. (2022) to each other and assess their user-friendliness and practicability for real use cases. This will be subject of future investigation.

5 Conclusion and Outlook

In contrast to the generation of BIM Models from design for new construction sites, which is a well established process, the extraction of the as-built state of existing buildings from observations is an open research field. In this paper we demonstrated the general usability of IVR technology in combination with a reasoning-based estimation approach for electric networks by Dehbi et al. (2022) for verifying the course of electric lines in buildings. For this purpose, we designed and implemented a VR system based on a 3D model stemming from a 2D room plan and information on the position of sockets and switches. Additionally, we established the communication between the Reasoner and the VR scene to mimic the verification action based on simulated measurements issued from an existing ground truth. In this context,

we demonstrated the practical feasibility of the approach presented by Dehbi et al. (2022). IVR turned out to be an innovative and good mean to interact with suggestions smartly sampled by different reasoning strategies and, hence, guide the inspector until all installed electric lines have been entirely verified without further ambiguities in a 3D digital twin of the building at hand.

Beyond the practical feasibility demonstration of the reasoning method the presented IVR system paves the way for a subsequent and systematic evaluation of the different reasoning strategies to assess their user-friendliness and practicability for real use cases. For this assessment, the design and implementation of a user study will be subject of future work. An ongoing work consists in the extension of the proposed system by an Augmented Reality (AR) component in order perform the verification process within the real environment supported by the visualization of the according digital twin and using an appropriate device, i.e. a wire detector.

References

- Banfi F, Brumana R, Stanga C (2019) Extended reality and informative models for the architectural heritage: from scan-to-BIM process to virtual and augmented reality. *Virtual Archaeol Rev* 10(21):14–30
- Berg LP, Vance JM (2017) Industry use of virtual reality in product design and manufacturing: a survey. *Virtual Reality* 21:1–17
- Boton C (2018) Supporting constructability analysis meetings with immersive virtual reality-based collaborative BIM 4d simulation. *Autom Constr* 96:1–15
- Boyce MW, Rowan CP, Shorter PL, Moss JD, Amburn CR, Garneau CJ, Sottolare RA (2019) The impact of surface projection on military tactics comprehension. *Military Psychol* 31(1):45–59
- Bryson S (2013) Virtual reality: a definition history—a personal essay. arXiv preprint [arXiv:1312.4322](https://arxiv.org/abs/1312.4322)
- Chen CH, Yang JC, Shen S, Jeng MC (2007) A desktop virtual reality earth motion system in astronomy education. *J Educ Technol Soc* 10(3):289–304
- Çöltekin A, Lochhead I, Madden M, Christophe S, Devaux A, Pettit C, Lock O, Shukla S, Herman L, Stachoň Z et al (2020) Extended reality in spatial sciences: a review of research challenges and future directions. *ISPRS Int J Geo-Inf* 9(7):439
- Costin A, Adibfar A, Hu H, Chen SS (2018) Building information modeling (BIM) for transportation infrastructure—literature review, applications, challenges, and recommendations. *Autom Constr* 94:257–281
- Dehbi Y, Knechtel J, Niedermann B, Haunert JH (2022) Incremental constraint-based reasoning for estimating as-built electric line routing in buildings. *Autom Constr* 143(104):571
- Dehbi Y, Haunert JH, Plümer L (2017) Stochastic and geometric reasoning for indoor building models with electric installations—bridging the gap between GIS and BIM. In: *Processes 12th 3D geoinformation conference, ISPRS annals of photogrammetry, remote sensing and spatial information sciences, vol IV-4/W5*, pp 33–39
- Demchak G, Dzambazova T, Krygiel E (2009) *Introducing Revit architecture 2009: BIM for beginners*. Wiley, New York
- Dimitriadou A, Djafarova N, Turetken O, Verkuyl M, Ferworn A (2021) Challenges in serious game design and development: educators experiences. *Simul Gaming* 52(2):132–152
- EU Commission (2022) Eu building stock observatory. <https://energy.ec.europa.eu/topics/energy-efficiency/energy-efficient-buildings/eu-building-stock-observatory>. Accessed 12 May 2023

- Fu L, Zhu J, Li W, Zhu Q, Xu B, Xie Y, Zhang Y, Hu Y, Lu J, Dang P et al (2021) Tunnel vision optimization method for VR flood scenes based on gaussian blur. *Int J Digit Earth* 14(7):821–835
- Getuli V, Capone P, Bruttini A, Isaac S (2020) BIM-based immersive virtual reality for construction workspace planning: a safety-oriented approach. *Autom Constr* 114(103):160
- Gröger G, Kolbe TH, Nagel C, Häfele KH (2012) OGC city geography markup language (CityGML) encoding standard. <https://www.ogc.org/standards/citygml>. Accessed 12 May 2023
- Hamilton D, McKechnie J, Edgerton E, Wilson C (2021) Immersive virtual reality as a pedagogical tool in education: a systematic literature review of quantitative learning outcomes and experimental design. *J Comput Educ* 8(1):1–32
- Hruby F, Sánchez LFÁ, Ressel R, Escobar-Briones EG (2020) An empirical study on spatial presence in immersive geo-environments. *PFG-J Photogrammetry, Remote Sens Geoinformation Sci* 88:155–163
- Kamińska D, Sapiński T, Wiak S, Tikk T, Haamer RE, Avots E, Helmi A, Ozcinar C, Anbarjafari G (2019) Virtual reality and its applications in education: survey. *Information* 10(10):318
- Keil J, Edler D, Schmitt T, Dickmann F (2021) Creating immersive virtual environments based on open geospatial data and game engines. *KN-J Cartography Geog Inf* 71(1):53–65
- Krispel U, Evers HL, Tamke M, Ullrich T (2017) Data completion in building information management: electrical lines from range scans and photographs. *Vis Eng* 5(1):4
- Lach E, Benek I, Zalewski K, Skurowski P, Kocur A, Kotula A, Macura M, Pamuła Z, Stankiewicz M, Wyrobek T (2020) Immersive virtual reality for assisting in inclusive architectural design. In: 6th International conference on man-machine interactions (ICMMI 2019). Springer, pp 23–33
- Li W, Zhu J, Dang P, Wu J, Zhang J, Fu L, Zhu Q (2023) Immersive virtual reality as a tool to improve bridge teaching communication. *Expert Syst Appl* 217(119):502
- Makransky G, Petersen GB (2019) Investigating the process of learning with desktop virtual reality: a structural equation modeling approach. *Comput Educ* 134:15–30
- Martirosov S, Kopecek P (2017) Virtual reality and its influence on training and education-literature review. *Ann DAAAM Proc* 28
- Moretti N, Ellul C, Cecconi FR, Papapesios N, Dejaco MC (2021) Geobim for built environment condition assessment supporting asset management decision making. *Autom Constr* 130(103):859
- Ochmann S, Vock R, Klein R (2019) Automatic reconstruction of fully volumetric 3d building models from oriented point clouds. *Int Soc Photogrammetry Remote Sens (ISPRS) J Photogrammetry Remote Sens* 151:251–262
- Ooi S, Tanimoto T, Sano M (2019) Virtual reality fire disaster training system for improving disaster awareness. In: Proceedings of the 2019 8th international conference on educational and information technology, pp 301–307
- Pezeshki Z, Ivani SAS (2018) Applications of BIM: a brief review and future outline. *Arch Comput Methods Eng* 25:273–312
- Radianti J, Majchrzak TA, Fromm J, Wohlgenannt I (2020) A systematic review of immersive virtual reality applications for higher education: design elements, lessons learned, and research agenda. *Comput Educ* 147(103):778
- Schiavi B, Havard V, Beddiar K, Baudry D (2022) BIM data flow architecture with AR/VR technologies: use cases in architecture, engineering and construction. *Autom Constr* 134(104):054
- Sermet Y, Demir I (2019) Flood action VR: a virtual reality framework for disaster awareness and emergency response training. In: ACM SIGGRAPH 2019 posters, association for computing machinery, p 27
- Snoswell AJ, Snoswell CL (2019) Immersive virtual reality in health care: systematic review of technology and disease states. *JMIR Biomed Eng* 4(1):e15.025
- Udeozor C, Toyoda R, Russo Abegão F, Glassey J (2021) Perceptions of the use of virtual reality games for chemical engineering education and professional training. *High Educ Pedagogies* 6(1):175–194
- Volk R, Stengel J, Schultmann F (2014) Building information modeling (BIM) for existing buildings—literature review and future needs. *Autom Constr* 38:109–127

- Wang P, Wu P, Wang J, Chi HL, Wang X (2018) A critical review of the use of virtual reality in construction engineering education and training. *Int J Environ Res Public Health* 15(6):1204
- Wheelwright B, Sulai Y, Geng Y, Luanava S, Choi S, Gao W, Gollier J (2018) Field of view: not just a number. In: Kress BC, Osten W, Stolle H (eds) *Digital optics for immersive displays*, international society for optics and photonics, vol 10676. SPIE, p 1067604
- Whyte J (2002) *Virtual reality and the built environment*. Routledge
- Xia H, Liu Z, Maria E, Liu X, Lin C (2022) Study on city digital twin technologies for sustainable smart city design: a review and bibliometric analysis of geographic information system and building information modeling integration. *Sustain Cities Soc* 84(104):009
- Yang L, Cheng JC, Wang Q (2020) Semi-automated generation of parametric BIM for steel structures based on terrestrial laser scanning data. *Autom Constr* 112(103):037
- Zhang G, Gong J, Li Y, Sun J, Xu B, Zhang D, Zhou J, Guo L, Shen S, Yin B (2020) An efficient flood dynamic visualization approach based on 3D printing and augmented reality. *Int J Digit Earth* 13(11):1302–1320
- Zhang J, Zhu J, Dang P, Wu J, Zhou Y, Li W, Fu L, Guo Y, You J (2023) An improved social force model (ISFM)-based crowd evacuation simulation method in virtual reality with a subway fire as a case study. *Int J Digit Earth* 16(1):1186–1204
- Zhu J, Wu P (2021) Towards effective BIM/GIS data integration for smart city by integrating computer graphics technique. *Remote Sens* 13(10):1889

3D Data Mapping with Augmented Reality



Ming-Chun Lee

Abstract This paper discusses two experimental Augmented Reality (AR) projects conducted by collaboration between the University of North Carolina at Charlotte and the Levine Museum of the New South in the City of Charlotte, USA between 2018 and 2021. These projects employ 2D mapping, 3D procedural modeling, and marker-based mobile AR techniques for data visualization focused on social and economic issues in Charlotte on a neighborhood scale. AR offers an interactive method to expand visualization capabilities in GIS. These projects show that mobile AR applications can support local community events that are aimed at expanding overall public participation with a goal of increasing awareness of neighborhood changes over time through 3D data visualization. Surveys from the demonstration events show that a majority of event participants learned new information about the demographic and economic changes over the past two decades in their communities through the AR experiences with 3D data mapping.

Keywords 3D Geovisualization · Augmented reality · Immersive technology · Data visualization · Procedural modeling

1 Introduction

Spatial information in 3D enables users to view and analyze data in detail and precision that cannot be achieved in 2D. 3D allows real-world views that improve users' orientation and increase their ability to perform a wide variety of tasks related to their geographic locations. Traditional 2D static maps have a limited exploratory capability. Modern 3D geovisualization allows for more interactive data displays, including the ability to explore different layers of a map, to zoom in or out, and to

This article was selected based on the results of a double-blind review of an extended abstract.

M.-C. Lee (✉)

David R. Ravin School of Architecture, University of North Carolina at Charlotte, 9201 University City Blvd, Charlotte, NC 28223-0001, USA

e-mail: ming-chun.lee@charlotte.edu

change the visual appearance of any given dataset in a map, usually on a computer display, a tablet, or a personal mobile device (Brink et al. 2017).

Today, GIS, aided by the power of information technologies, has evolved into a vital tool for science-based problem-solving. GIS has been used to study urban development and its associated issues for as long as its existence. Recent advancement in GIS offers new possibilities for the study of urban planning and community design in particular with new technologies for spatial data visualization. Among many technological developments, two crucial trends have led to new ways of data mapping and visualization, including (1) 3D GIS: conventional 2D maps are being replaced by interactive 3D models generated by attributes stored in GIS; (2) integration with immersive technologies, such as Augmented Reality (AR): AR creates immersive experiences with a perception of being physically present in a non-physical world. GIS can greatly enhance the accuracy and realism of virtual scenes with geographically referenced 3D features.

Immersive technologies, such as AR and Virtual Reality (VR), can serve as powerful instruments to create geovisualization in community planning and development (Broschart et al. 2013). As in the case of AR, which is practically a new visualization framework, it goes beyond conventional methods of public participation that typically only employ passive one-way communication with the community. In contrast, AR enables interactive ways to visualize human built environments with overlays between both non-computerized and computerized contents in urban planning and community design (Broschart et al. 2013; Reinwald et al. 2014; Carmignani and Furht 2011).

This paper discusses two experimental AR projects conducted by collaboration between the Levine Museum of the New South, a local history museum in the City of Charlotte, and a research center at the University of North Carolina at Charlotte. The main goal of these projects is to explore the new junction between the conventional mapping methods and the two new emerging digital visualization techniques: 3D GIS and AR, for data processing and geovisualization.

2 Augmented Reality

Augmented Reality (AR) combines real and virtual environments together through an interactive 3D experience. A viewer's visual perception is formed through digitally generated contents by a computer or a mobile device. This type of immersive visualization can be experienced in various ways, such as by using smart glasses, head-mounted displays, or mobile devices among others. Several key features of these AR systems include: (1) virtual contents generated by these digital devices and the real physical reality are combined and overlapped; (2) these systems operate interactively in real-time and in 3D (Azuma 1997; Azuma et al. 2001; Greengard 2019).

Initial research on AR and other related technologies, such as VR, can be traced back to the 1950s (Maffei et al. 2016). However, it was not until 1997 when the first

general definition of AR surfaced. Azuma argued that an AR experience is made up of three key components: (1) a combination of real and virtual environments; (2) an interactive user interface operating in real-time; (3) an overlaid visual perception registered in 3D (Azuma 1997). Although AR has been used in research projects since the 1990s with limited applications, only in the recent years have more mobile AR projects began to emerge due to in a large part the availability of AR-enabled software engines and web browsers (Fisher-Gewirtzman et al. 2017; Leeuwen et al. 2018; Ma et al. 2020). In addition, the current supply of relatively powerful and affordable mobile devices with high-resolution cameras and high-speed data connection has made AR applications more appealing to general end users (Szczepańska and Kaźmierczak 2021; Gnat et al. 2016).

3 Technical Framework

These research projects were intended to investigate the application of AR in 3D data mapping. The technical framework of these projects includes a set of mapping and modeling processes, which utilize a number of different software programs, including 3D GIS data visualization using ArcGIS Desktop, 3D modeling in ArcGIS CityEngine (a procedural modeling program) (Metze 2020; Ribeiro et al. 2014), and AR app development by Unity (a game engine).

These projects began with assembling GIS data from several online open data sources, including a database maintained by Mecklenburg County in North Carolina and Census datasets from the National Historical GIS (NHGIS) website. A set of 2D maps were later created by putting together datasets using both ArcMap and ArcGIS Pro (developed by Esri) to generate a series of digital representations of the social and economic conditions for the selected neighborhoods in the City of Charlotte for these projects.

In addition, a set of 2D maps to show the geography of these neighborhoods were also created. These maps contain several shared geographic features of the neighborhoods, including streets, parcel lines, neighborhood boundaries, creeks, and green open spaces. The main purpose of these maps is to allow participating community members to easily recognize these study areas.

In order to create the 3D contents needed for the intended AR immersive experience, a set of 3D models were then generated to visualize these social and economic datasets for the study sites. This process was completed by using ArcGIS Pro to develop a set of 3D symbology by extruding 2D polygons in the maps with variables as heights (commonly known as Z-values) that represent certain demographic, social, or economic measures, such as population counts, household incomes, and property values over a certain period of time.

These 3D symbols were then converted into solid 3D models as Multipatch geometries in ArcGIS Pro. A Multipatch geometry is a GIS feature that can store a collection of patches to represent the boundary of a 3D object as a single row in a database. A File GeoDatabase was then created to package all needed GIS datasets, including

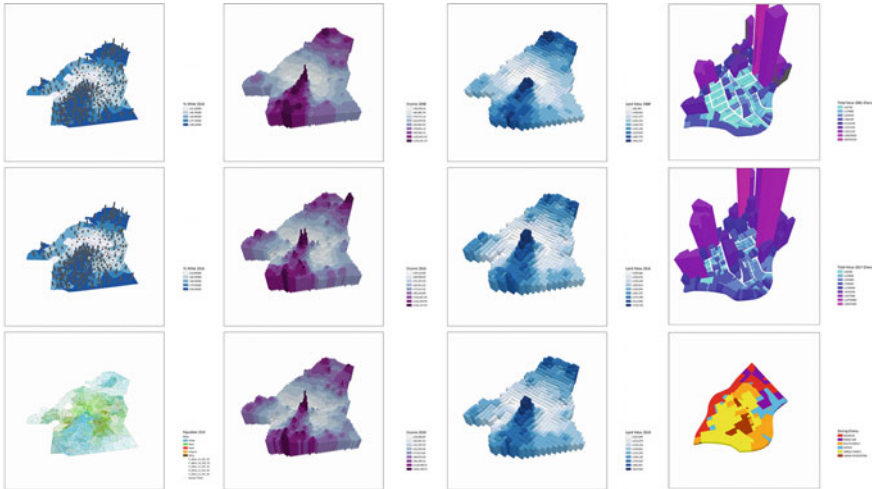


Fig. 1 A series of 3D Multipatch models to visualize social and economic datasets of the participating neighborhoods over time were created in ArcGIS Pro

these resulting 3D Multipatch geometries. This last step was essential to allow these 3D GIS models to later be imported into other 3D modeling platforms for further processing (Fig. 1).

To improve visual effects of these 3D GIS models, ArcGIS CityEngine, a rule-based procedural 3D modeling program by Esri, was used to apply colors and transparency to these 3D models. These Multipatch GIS models stored in the File GeoDatabase were imported into CityEngine and visual effects were added using a simple color procedural ruleset provided by Esri.

The final step of the projects was to import these enhanced 3D models from CityEngine into Unity, a game engine and mobile app development platform. These CityEngine models were first exported as Autodesk FBX (.fbx) files and then imported into Unity to create 3D immersive scenes (Fig. 2). Vuforia AR SDK was added into Unity as assets to enable the development of AR mobile apps using the 3D assets imported from CityEngine and the 2D maps from ArcMap as the AR image targets (generally known as AR markers). The final resulting AR mobile apps detect these image targets or markers (the 2D maps of study neighborhoods) and then display and align seamlessly the 3D models on top of these maps to create an immersive illustration of 3D effects for the GIS datasets.



Fig. 2 The technical workflow of the AR 3D mapping projects

4 Demonstration Projects

Two marker-based AR mobile apps were developed in order to demonstrate and test the applicability of this AR-enabled 3D data visualization framework. These mobile apps were used in two separate exhibits as demonstrations, one curated by City Building Lab in the David R. Ravin School of Architecture at the University of North Carolina at Charlotte and the other by the Levine Museum of the New South, a history museum in the City of Charlotte, whose exhibits are focused on life in the North Carolina Piedmont after the American Civil War.

Both two AR apps were developed by the research team, Augmented Reality Team (ART), a research group in City Building Lab at the University of North Carolina at Charlotte. ART conducts research on immersive visualization technologies and their applications in community engagement and public participation. The key mission of the research group is to explore a new junction between the conventional mapping methods and many new emerging digital techniques for data processing and visualization, including 3D GIS, integration with AR, and integration with remote sensing imagery among others.

4.1 Mapping (In)-Equity

Mapping (In)-Equity was a map exhibit in conjunction of Urban Complexities Symbolism organized by City Building Lab at the University of North Carolina at Charlotte in March 2018 (City Building Lab 2019). More than 200 participants from the region attended this one-day event, including planning practitioners, community activists, real estate developers, etc. This exhibit was aimed to discover a possible new connection between the traditional mapping methods and many emerging techniques for data processing and visualization, such as 3D GIS, integration with immersive technologies, and incorporation with imagery. Participants used the AR app developed by ART on a tablet or a smart phone to interact with 2D printed maps in the exhibit hall to view social and economic data associated with selected local neighborhoods present in these maps. They were able to see these datasets as 3D immersive scenes, such as household incomes, from different years to understand changes over time in these neighborhoods (Fig. 3).



Fig. 3 Mapping (In)-Equity exhibit at the University of North Carolina at Charlotte where an AR app was introduced for GIS data visualization

4.2 #HomeCLT: People. Places. Promises

#HomeCLT: People. Places. Promises was a multi-media story-telling exhibit, curated by the Levine Museum of the New South, to uncover stories of local neighborhoods in the City of Charlotte as they have grown and evolved over time through a combination of conventional museum installations and an AR immersive experience that brought the stories alive (Levine Museum of the New South 2019a, b). The first set of *#HomeCLT* was launched in February 2019. It included the stories of Enderly Park, Eastland Mall, Dilworth, Hidden Valley, and Sedgefield neighborhoods in the City of Charlotte. Through a mobile AR app developed by the project team, visitors were able to see changes in demographics and property values in these neighborhoods over the past two decades as 3D visual representations, which were made possible by AR graphics displayed on their mobile devices. The AR app detected the large map installed on the floor of the exhibit space and several other smaller maps mounted on the pedestals as a part of the exhibit installation. Users were able to see virtual contents overlaid on top of these maps. The AR app also interacted with other display panels to allow visitors to watch pre-recorded videos of oral history by selected residents on their mobile devices as a part of AR immersive experience.

The second iteration of *#HomeCLT*, *Brooklyn: Once a City Within a City*, launched in early 2020, added the historic Brooklyn neighborhood with additional AR-enabled contents. This addition of *#HomeCLT* explored the rise and demise of Charlotte's Brooklyn neighborhood, once the most thriving and vibrant black community in the Carolinas in the early twentieth century. The AR app for this second set of *#HomeCLT* expanded the immersive experience by including additional GIS datasets and using a variety of 3D visual effects for these different GIS datasets (Fig. 4).

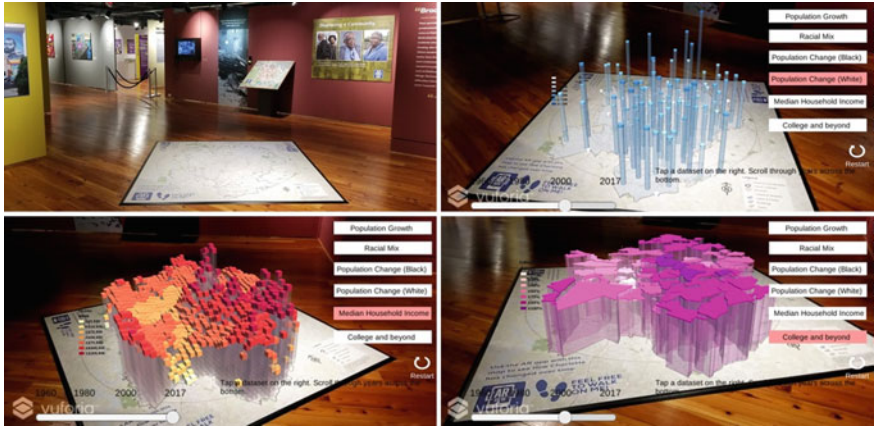


Fig. 4 Historic Brooklyn Neighborhood exhibit, the second iteration of #HomeCLT at the Levine Museum of the New South where an AR app was in use

5 Discussions and Conclusion

Landscape architect James Corner describes mapping “as equal to what is and to what is not yet... The unfolding agency of mapping is most effective when its capacity for description also sets the conditions for new eidetic and physical worlds to emerge” (Corner 2011). These AR apps and their demonstration exhibits explored the junction between conventional and emerging digital techniques for 2D/3D mapping, spatial data processing, and visualization. The recent history of the City of Charlotte, North Carolina is one of growth and expansion. It is also one of displacement and immobility. These characteristics often occur in the same geographic locations but impact people differently. By cross-referencing spatial, economic, and demographic data, these AR-enabled 3D data visualization projects revealed how social and economic systems may have perpetuated inequity through the built environment.

These demonstration projects and events described in this paper have allowed the project team to investigate the potential of AR in civic engagement through the use of mobile devices and 3D data visualization. Particularly, the team was able to develop a series of experimental AR mobile apps to enable members of local communities to explore the social and economic transformations over time in their neighborhoods through AR-enabled 3D data visualization.

All participants in these events were asked to complete a user survey to provide feedback about their experience with these AR apps. 211 survey results were collected and analyzed. These surveys provide the project team with valuable insights into the effectiveness of these AR apps in helping users understand GIS data as 3D representations. Several important results and lessons learned from these demonstration projects with these experimental AR apps are discussed as the following: (1) from the viewpoint of the general event participants, opportunities were given to them to learn about AR and gain first-hand experiences about its potential to promote civic

engagement and public education. The event participants appreciated the opportunity to be involved in these community events to learn about their own communities and meet their neighbors; (2) 93% of the event participants agreed that they learned new information about the history of their communities through these AR experiences; (3) for the event organizers, they appreciated the hands-on experiences about how AR can be an effective tool to engage with local community members as a way of storytelling or as a tool for educational purposes in general; (4) 95% of the event participants agreed that AR experiences made these events more interactive and informative.

An in-depth evaluation on the effectiveness of these AR-enabled visualization techniques for the study of 3D mapping is outside of the scope for this short paper, which can certainly be a necessary next-step for advancing this particular research. These exploratory projects discussed in this paper allow for a quick glimpse of the on-going development trends in the GIS world, especially in the area of 3D data mapping and visualization.

AR is constantly evolving as an immersive technology. Its potential as a new way of seeing through blending together our physical world with layers of virtual realities is enormous. However any possible use cases have to have a clear goal that is specific and geared towards its target users. As these digital 3D and immersive technologies are becoming more and more advanced, we researchers, scholars, and designers interested in understanding the dynamics of geospatial analytics should explore new methods and establish new workflows to utilize these tools in a holistic way in order to advance our work.

Acknowledgements These research projects were supported by the John S. and James L. Knight Foundation. The author of this paper thanks colleagues from the Levine Museum of the New South who provided insight and expertise that greatly assisted the research.

References

- Azuma R (1997) A survey of augmented reality. *Presence: Teleoperators Virtual Environ* 6(4):355–385
- Azuma R, Baillot Y, Behringer R, Feiner S, Julier S, MacIntyre B (2001) Recent advances in augmented reality. *IEEE Comput Graphics Appl* 21(6):34–47
- Broschart D, Zeile P, Streich B (2013) Augmented reality as a communication tool in urban design processes. *Proc Real Corp* 2013:119–126
- Carmignani J, Furtth B (2011) Augmented reality: an overview. *Handbook of Augmented Reality*, 3–46
- City Building Lab (2019) School of architecture. University of North Carolina at Charlotte. Available at: <https://www.facebook.com/CityBuildingLab-158386934209297/?ref=hl>
- Corner J (2011) The agency of mapping: speculation, critique and invention. In: *The map reader: theories of mapping practice and cartographic representation*. Dodge M, Kitchin R, Perkins C (eds). Oxford, UK, Wiley, 214
- Fisher-Gewirtzman D, Portman M, Natapov A, Hölscher C (2017) Special electronic issue: the use of virtual reality for environmental representations. *Comput Environ Urban Syst* 62:97–98

- Gnat M, Leszek K, Olszewski R (2016) The use of geoinformation technology, augmented reality and gamification in the urban modeling process. ICCSA 2016. Lecture notes in computer science. Springer, Cham, Switzerland, 9787
- Greengard S (2019) Virtual reality. MIT Press, Cambridge, MA, USA
- van Leeuwen JP, Hermans K, Jylha A, Quanjer AJ, Nijman H (2018) Effectiveness of virtual reality in participatory urban planning. In: Proceedings of the 4th media architecture biennale conference 2018 (Mab18), 128–136
- Levine Museum of the New South (2019) The Levine museum presents a variety of interactive and immersive exhibits. Available at: <https://www.museumofthenewsouth.org/exhibits>
- Levine Museum of the New South (2019) #HomeCLT: people. Places. Promises. Available at: <https://www.museumofthenewsouth.org/homeclt>
- Ma Y, Wright J, Gopal S, Phillips N (2020) Seeing the invisible: from imagined to virtual urban landscapes. *Cities* 98:102559
- Maffei L, Masullo M, Pascale A, Ruggiero G, Romero VP (2016) Immersive virtual reality in community planning: acoustic and visual congruence of simulated vs real world. *Sustain Cities Soc* 27:338–345
- Metze T (2020) Visualization in environmental policy and planning: a systematic review and research agenda. *J Environ Planning Policy Manage* 22(5):745–760
- Reinwald F, Berger M, Stoik C, Platzer M, Damyanovic D (2014) Augmented reality at the service of participatory urban planning and community informatics—a case study from Vienna. *J Commun Inf* 10(3)
- Ribeiro A, Duarte de Almeida J-P, Ellul C (2014) Exploring CityEngine as a visualization tool for 3D cadastre. In: 4th international workshop on FIG 3D cadastres, Dubai, United Arab Emirates, pp 197–218
- Szczepańska A, Kaźmierczak R, Myszkowska M (2021) Virtual reality as a tool for public consultations in spatial planning and management. *Energies* 14(19):6046
- van den Brink A, Van Lammeren R, Van de Velde R, Silke D (2017) *Imaging the future: geo-visualization for participatory spatial planning in Europe*. Wageningen Academic Publishers, Wageningen, The Netherlands, p 3

Creating a 3D Multi-Dataset Bubble in Support of OGC Testbed-19 and Metaverse Standards Prototypes



James Clarke, Steve Smyth, Rob Smith, and Jeremy Morley

Abstract Open Geospatial Consortium Testbeds are an annual research and development initiative that explore geospatial technology. For Testbed-19 we generate a multipurpose 3D dataset in support of the development of Augmented Reality/Metaverse architectures and infrastructure prototypes: Road Hazard Monitoring, Mixed Euclidean and Minkowski GeoPose Graphs, and GeoPose Graphs in a Minkowski Bubble. The dataset comprises: LiDAR capture, position and orientation, video/imagery from a survey vehicle, drone imagery, static camera and GoPro imagery, and derived data. This is extended by a simulated orbiting platform with high-resolution sensor. Our coverage is a bubble of radius 256 m, centered near the Ordnance Survey headquarters in Southampton, UK. The data is published in an open-source repository to support OGC projects, and a prototype of parts of the Ride Hailing use case of the Metaverse Standards Forum—Real/Virtual World Integration Domain Working Group. Open availability of this standard dataset will enable independent testing and benchmarking of architectures and algorithms for a variety of mobile applications.

Keywords OGC GeoPose · Open-source datasets · 3D data

This article was selected based on the results of a double-blind review of an extended abstract

J. Clarke (✉) · J. Morley
Ordnance Survey, Southampton SO16 0AS, UK
e-mail: james.clarke@os.uk

S. Smyth
OpenSitePlan, Kirkland, WA 98033, USA

R. Smith
Away Team Software, Crowthorne RG45 7AW, UK

1 Introduction

1.1 *Research Objectives*

The Open Geospatial Consortium (OGC) Testbeds are innovation program initiatives to prototype and execute solutions to location-related problems. They boost research and development and encourage location interoperability. The current iteration is Testbed-19 (2023), which involves six tasks: Geospatial in Space, Machine Learning, Geodatacubes, Analysis-Ready Data, Agile Reference Data, and High-Performance Computing. Solutions to these tasks will move into the OGC standards program, where they may be approved as new standards or extend existing ones.

Developing algorithms for mutually intervisible sensors in 3D space requires realistic comparisons of implementation completeness, accuracy, time and space complexity, and other properties of competing designs. Currently, despite the exponential increase in 3D data, there are a lack of available datasets that can fulfil this as detailed by the DPC Technology Watch Guidance Note on Preserving 3D (2023). The goal of this work is to capture and provision data to support development and benchmarking with a well-documented and realistic open-source dataset. While this fulfils some of the requirements in the Testbed-19: Geospatial in Space task (specifically for the OGC GeoPose standard; Sect. 1.2), it extends beyond their scope, and we aim for these data to be available for the development of any 3D community standards.

The Metaverse Standards Forum (2023) is a collaborative body comprising standards bodies and companies working to foster the development of interoperability standards for the metaverse (a 3D-enabled digital space that contains a virtual, immersible world). As part of this, the Real/Virtual World Integration working group has generated prototypes to realize one or more use cases and demonstrate the use of standards. The representation of position, orientation and inter-object relationships in their use cases further necessitates datasets such as ours.

1.2 *OGC GeoPose*

“GeoPose 1.0 is an OGC Implementation Standard for exchanging the location and orientation (“Poses”) of real or virtual geometric objects within reference frames anchored to the Earth’s surface (“Geo”) or within other astronomical coordinate systems. It defines the rules for the interoperable interchange of these poses. It specifies two basic forms for common use cases with no configuration options, an advanced form with more flexibility for more complex applications, and five composite GeoPose structures which support time series plus chain and graph structures”. OGC GeoPose Data Exchange Draft Standard (2023) of relevance to this

work is the reference frame graph, which represents the transformational relationships between reference frames (one or more pose chains that lead from the outermost frame to the innermost frame).

To assess the real-world utility for spatial computing scenarios and ease of use, we need to examine multiple implementations (<https://github.com/opengeospatial/GeoPose/tree/main/Implementations/>) and use cases as described in Sect. 1.4. GeoPose is used as it represents the poses and relationships of reference frames to fulfil these requirements, while having the advantage of being open and accessible.

The data collected will provide the geo-reference frames in which the GeoPoses of different objects and “actors” (sensors, camera yielding people) can be integrated and visualized, to facilitate these use cases.

1.3 Extending GeoPose to Minkowski Spacetime

The Testbed-18 Engineering report (2023) proposed extensions to current standards to cover broader needs in an astrophysical domain, accurately representing the effects of relative velocities approaching the speed of light, and the effect of gravity and acceleration. One goal of the collection of data is to make a first step in exploring the extension of GeoPose from absolute Newtonian space and time to the flat spacetime of special relativity.

This spacetime, characterised by the Minkowski metric, allows us to represent space and time in and between inertial frames (those at rest or with uniform relative velocity). The difference in results become apparent when the relative velocities between two reference frames in uniform relative motion approach a significant fraction of the speed of light (Macdonald 1983). In this case, the computation of temporal and geometric relationships between objects in different frames requires use of the Lorentz transformation to accurately represent events across reference frames (Rao 1988).

An approximation to Minkowski spacetime will be made using the platforms and sensors by taking the relative motion over brief time intervals to be unaccelerated, and to artificially set the speed of light to 102% of the speed of the fastest moving platform.

1.4 Use Case Prototypes

OGC Testbed-19: Implement a Prototype of the Road Hazard Monitoring [Away Team] The Testbed-18 (2023) Moving Features and Sensor Integration, and Testbed-17 (2023) Moving Features threads identified several road network use cases, e.g., wrong way vehicles and roadside litter monitoring. These use cases can be examined by aggregating geotagged video and sensor data from multiple sources. The data we capture will enable:

- Identification of moving vehicles from sensor data with GeoPose data captured by Away Team's 3D Compass Android app (2023).
- Identification of litter objects with video/LiDAR to monitor changes and predict collection location.
- Simulation of locale visibility from low-Earth orbit satellites for Earth Observation and communications.
- Processing of W3C WebVMT (2023) (an open format designed to synchronise video with an animated, annotated map) data with utilities developed in Testbed-17 Moving Features to display results in a web browser and demonstrate HTML integration.

Metaverse Standards Forum. Ride Hailing One of the Virtual/Real World Integration Domain Working Group's use cases is the Assisted Car/Human Urban Rendezvous. It comprises the following:

- A person (rider) requesting an available car. If one is available, it is sent to the rider.
- Virtual designations (e.g., a virtual sign) for car and rider when the two actors are mutually visible, to make themselves clear to each other. Each designation moves with the subject actor and always faces the other actor.
- Interface(s) for external services supporting rendezvous.
- Using existing physical structures to prototype intervisibility/occlusion calculations.

OGC Testbed-19 GeoPose in a Minkowski Bubble This will generate GeoPose graphs from the collected data to explore issues relating to implementation and extension of OGC GeoPose for use in a Minkowski spacetime.

The extension of the standard will be demonstrated with 2000s of dynamic pose data collected from moving entities in the Hillyfields (Southampton) Bubble. For purposes of the demonstration, the bubble will be fitted with variable physical constants. The speed of light in the bubble will be decreased to about 1.02 times the speed of the fastest moving platform.

The correctness and coherence of the flat spacetime extension will be visualised with multiple animations of the captured actor poses in a Minkowski spacetime adapted GeoPose graph. The animations will be captured as multiple video streams rendered from each of the inertial frames corresponding to the individual entities in uniform motion relative to the Bubble frame.

OGC Testbed-19 GeoPose in Euclidean Space, Time, and Minkowski Spacetime As well as extending the GeoPose graphs to Minkowski spacetime, this will demonstrate interoperability with GeoPoses in Euclidean 3-space plus absolute time.

For the purposes of the interoperability demonstration, a steerable visual imaging sensor on an orbiting platform in low earth orbit will be selected or simulated to provide a relationship between the Bubble reference frame (the region of locally Euclidean space plus time) and the reference frame of the orbiting imager. This

Table 1 The use cases and interactions/intervisibilities to fulfil requirements, with capture type in brackets. Position information for the drone, StreetDrone and orbital platform is assumed for all capture

| Use case | Interaction/Intervisibility |
|--|--|
| Road hazard monitoring (RH) | (Static camera ↔ StreetDrone (video), or GoPro ↔ StreetDrone (video)) StreetDrone → Litter (video, LiDAR) Static Camera → Litter (video) |
| Metaverse standards forum: ride hailing (Rendezvous; RV) | GoPro ↔ StreetDrone (video, LiDAR; occlusion and line-of-sight) |
| GeoPose in a Minkowski bubble (GM) | UAV → StreetDrone (video) |
| GeoPose in Euclidean space, time, and Minkowski spacetime (GEaM) | UAV → StreetDrone (video) Simulated orbital platform → StreetDrone (imagery) |

choice requires using a Minkowski metric to achieve accurate (<1 pixel) pointing of the imager toward photons arriving from the reference point of a moving vehicle in the Bubble.

1.5 Method

To support the prototypes, we collect a multipurpose dataset inside a 3D spherical region of radius 256 m, sunk below the surface of the Earth and centred near the Ordnance Survey headquarters in Southampton, UK. The bubble comprises data from visual and navigation sensors mounted on:

- A land vehicle, hereafter the StreetDrone.
- An Unmanned Aerial Vehicle (UAV).
- Mobile body-worn cameras.
- A simulated imaging platform in low earth orbit.
- Stationary cameras at two ground locations.

Equipment and capture plans are set out below. The capture plans were designed to orchestrate interactions and intervisibilities between the sensor platforms to support the use cases, as shown in Table 1.

1.6 The Ordnance Survey StreetDrone

The OS StreetDrone (Fig. 1) is a Renault Twizy modified to accommodate a suite of sensors. Two antennae on the roof allow us to collect multiple constellation position (GPS, Galileo, GLONASS), while a tactical grade Inertial Navigation System (INS;

Fig. 1 The OS StreetDrone research platform



OxTS xNAV650) and connection to OS Net (Ordnance Survey’s network of 120 UK reference stations (OS Net Overview 2023)) allow real-time kinematic (RTK) corrections to give cm-level positional accuracy. The INS also provides orientation (roll, pitch, yaw) data for the vehicle. The vehicle captures compressed video from up to eight mounted cameras that cover 360° with overlap, as well as capturing up to 1.2 million point/sec from the LiDAR (Hesai XT32).

The StreetDrone driving plan was the reference for all other capture and is therefore not time-critical but simply includes planned routes with different scenarios (e.g. simple drive and person-vehicle rendezvous).

1.7 UAV Capture

Survey imagery and video footage of the StreetDrone was captured by a Mavic2 drone at 50 m above ground level. The on-board sensors also include a GPS receiver, compass, accelerometer, and gyroscope. Imagery is captured using a 1.27 cm 12/48-megapixel chip that provides 84° field of view, achieving a theoretical ground sampling distance (pixel-to-pixel resolution) of 0.77 cm. Flight plans are set up for the initial imagery/LiDAR survey, while a combination of flight plan, “follow-me” mode and manual flying is utilized to observe the StreetDrone commencing when the ground vehicle moves.

1.8 *Simulated Orbital Platform*

To relate terrestrial and space-based position and orientation, a virtual satellite is modelled in low-Earth orbit at 500 km with a high-resolution imaging system to simulate 25 cm resolution imagery. The orbit is calculated from the method of producing two-line element sets (a data format for describing satellite orbits) from Rockwood et al. (2022). They describe an accurate method for generating orbital descriptions for notional systems with realistic terms for drag, and the first and second derivative of mean motion.

During the data capture the satellite moves at 7 km/s using the simulated orbit starting at 750 km over Longyearbyen, Svalbard at the time the StreetDrone starts moving, another waypoint overhead OS HQ 200 s later, with an end point over Dakar, Senegal. The velocity of the platform is sufficient to require the consideration of special relativity when considering the distance between the orbital platform's imager and other observed reference frames.

1.9 *Further Sensors and Actors*

Stationary smartphone cameras, and GoPro cameras mounted on chest straps allow multiple video capture of the StreetDrone and other passing vehicles and pedestrians, plus provide reference positions for pieces of litter. Litter comprises typical rubbish items which might be expected to be at the side of a road (e.g. cans, bottles) but also highway equipment (signs, A-frames, traffic cones).

The GoPros (which have GPS and IMU sensors) allow us to fulfil the rendezvous scenario, which mimics a taxi pick up; the StreetDrone pulls up to one of the actors wearing a GoPro, each capturing both their counterpart's image and location in the process. Where possible the actors and stationary cameras were placed where items of litter could be detected to provide another detection and object GeoPose.

2 Results

We performed 12 data collection runs over two days in either clear, sunny skies or dry, overcast conditions. The focus of each run was to collect data to fulfil car-rider rendezvous, litter detection, and Minkowski-spacetime extension/interoperability of GeoPose.

Table 2 shows the sensor capture status of each data collection run. Due to the complicated nature of the different platforms and sensors, not all data sources were available for each run. Also shown is the use case to which the capture contributes. All the interactions and intervisibilities listed in Table 1 were satisfied during at least

one run. Note that IMU and UAV capture was successful for every run so is not included in Table 2. Run 7 was an aborted run, so no data exists for this.

The data artefacts available in the open-source repository (<https://github.com/HillyfieldsBubble/Releases>) are summarised in Table 3 and discussed in more detail below by category. Further metadata about the contents of the data, and naming

Table 2 Details of the sensors used in each run, and the use case(s) that the data contributes to. The use cases are Rendezvous (RV), Road Hazard (RH), GeoPose in a Minkowski Bubble (GM) and GeoPose in Euclidean Space, Time, and Minkowski Spacetime (GEaM)

| Capture | Camera (Static Phone) | Camera (GoPro) | Cameras (StreetDrone) | LiDAR (StreetDrone) | GPS-RTK (StreetDrone) | Use case |
|-----------|-----------------------|----------------|-----------------------|---------------------|-----------------------|------------------------|
| 230425_r1 | ✘ | ✘ | ✘ | ✔ | ✔ | GM GEaM |
| 230425_r2 | ✘ | ✘ | ✘ | ✔ | ✔ | GM GEaM |
| 230425_r3 | ✔ | ✘ | ✘ | ✔ | ✔ | GM GEaM, RV |
| 230425_r4 | ✔ | ✘ | ✘ | ✔ | ✔ | GM GEaM |
| 230425_r5 | ✔ | ✘ | ✘ | ✔ | ✔ | GM GEaM |
| 230425_r6 | ✘ | ✘ | ✔ | ✔ | ✔ | GM GEaM |
| 230425_r7 | ✘ | ✘ | ✘ | ✘ | ✘ | None |
| 230425_r8 | ✔ | ✔ | ✔ | ✔ | ✘ | GM GEaM RH RV |
| 230425_r9 | ✔ | ✘ | ✔ | ✔ | ✔ | GM GEaM RH |
| 230426_r1 | ✘ | ✔ | ✔ | ✘ | ✔ | GM GEaM RH RV |
| 230426_r2 | ✔ | ✘ | ✔ | ✘ | ✔ | GM GEaM RH |
| 230426_r3 | ✔ | ✔ | ✔ | ✘ | ✔ | GM GEaM RH RV |

conventions/indexing, can be found in the repository readme and files alongside the data.

Positioning/Orientation The data was output as one csv file per run detailing time, position mode (“integer” when RTK available or other values when there is a dropout), position accuracy (E, N, altitude), heading, pitch and roll (with uncertainties). Where there were GPS-RTK dropouts (due to obstructions or potential communications problems) the INS unit uses a proprietary algorithm to ensure that the position remains accurate, although this degrades for the duration of the dropout. The degradation reaches a lower-limit accuracy of typically 0.3—1.0 m.

Imagery/Video We include the frames and video from the forward facing and rear facing cameras on the StreetDrone vehicle. Uncompressed images were captured, however bandwidth issues from the cameras mean that the output frame rate is not constant, and many dropouts are experienced (most noticeably for run 1 on 26/04/2023). To resolve this, we captured run 3 on 26/04/2023 in compressed format at 5 Hz.

Image and video content was reviewed for personally identifiable information such as faces, signs, and vehicle number plates of non-participants and the actors’ identities obfuscated.

For each morning and afternoon capture session, the UAV performed imagery surveys from which a 3D model could be constructed using AGI Metashape photogrammetry software. In each run the UAV was flown following the Street-Drone vehicle. Figure 2 shows an example run flight path with position at 0.1 s intervals.

Table 3 Summary of the data artefacts

| Type | Description |
|---|---|
| LiDAR (scan) | 10 Hz collection as packet file (pcap). Raw packets can be georeferenced or aggregated using SLAM algorithm |
| Point Cloud | Point cloud map from individual scans. Compressed Laz format |
| StreetDrone video | Uncompressed or compressed at 5 frames per second. 2 MP, front and rear facing capture. MP4 format |
| StreetDrone imagery | Individual frames from both cameras, compressed as PNG |
| Drone point cloud | $\sim 2 \times 10^7$ points covering full extent. Laz format |
| Drone video | Video capture for each run from 50 m altitude. MP4 format |
| INS output—GPS/IMU (Position/orientation) | Output roll, pitch, yaw, latitude, longitude, GPS status with accurate timings at 100 Hz, as csv file |
| Static camera video | Video capture of StreetDrone and other passing vehicles/ pedestrians. MP4 format |
| WebVMT | Synchronised position and video data, recorded from phones and StreetDrone. VMT extension |

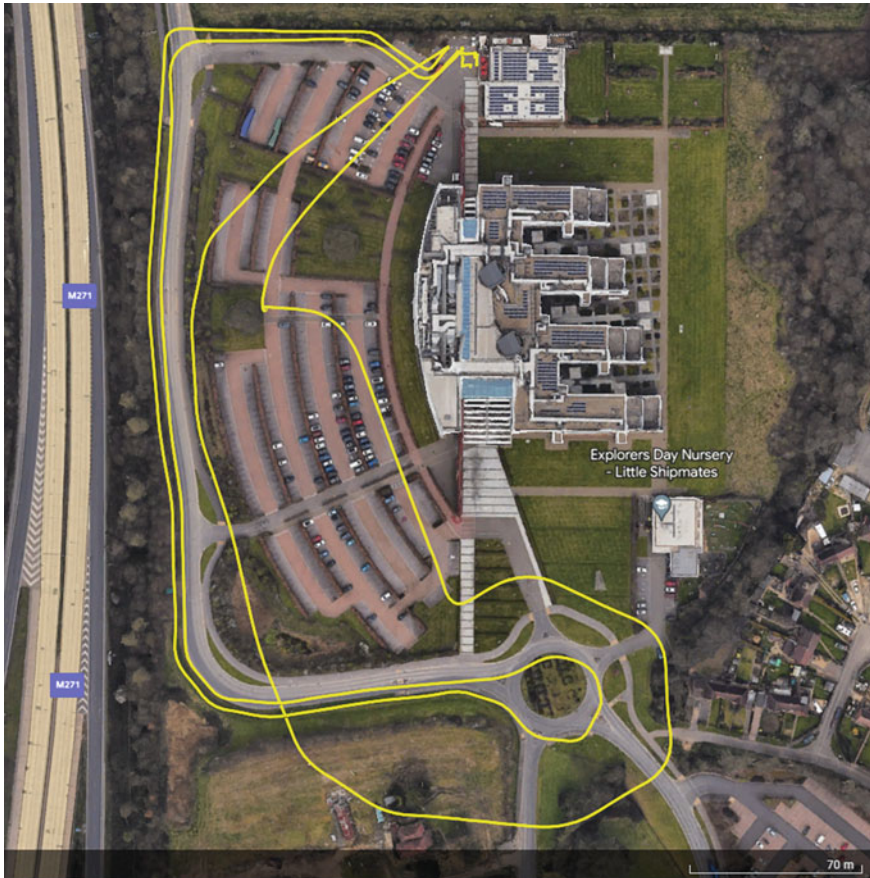


Fig. 2 Flight path: An example flight path for the UAV during a data capture run

LiDAR Prior to the data collection, we undertook boresight calibration to ensure that 6-degree-of-freedom dimensions between LiDAR and INS unit axes were correct (even small discrepancies can make georeferencing a point cloud impossible).

Point cloud data (~1.2 million points/second with rpm of 600) were collected for all runs on the first day with no interruptions and noise from precipitation. The capture data is included in the repository as raw pcap files.

Derived Data Further data were derived from the raw capture. These are detailed below.

Point Cloud Packet capture files can be used to generate a point cloud map with Simultaneous Localisation and Mapping (SLAM) algorithms, SLAM aggregation of the capture results point clouds with good relative accuracy, but for absolute accuracy (i.e. data alignment with control positions in the real-world) the point cloud needs georeferencing. For this we used OXTS's GeoReferencer software. While it would

be possible to generate point cloud from all runs, that with the best position quality was used—in this case run 1 on 25/04/2023. The point cloud was down sampled using GDAL to 10% of the points in consideration of size but has not been processed beyond this. Thus, there is no classification but there is the presence of moving vehicle noise. Given that we are using GPS-RTK in the georeferencing process, we achieve a typical accuracy $\approx 3\text{--}5$ cm against real-world pre-collected ground control points (mm accuracy). These ground control points have also been included in the dataset.

Figure 3 shows another point cloud collected prior to this work with the Street-Drone but included in the repository as supplementary data as it contains less noise. The whole route is shown, from the south-eastern roundabout on Adanac Park, Southampton, UK, to the northern access of the car park.

WebVMT Geotagged video files were generated from actors' mobile phones using the TrkdCam app designed by Away Team Software. The users collected video, while the app gathers position and IMU data synchronises these to generate WebVMT. Thus, any feature identified in the video can have its position identified and tracked.

Detected Targets An artificial neural network (ANN) was trained using portions of the Microsoft Common Objects in Context (MS COCO) dataset (Lin et al. 2014). This network was used as a detector for locating certain entity classes, such as people, cars, trucks, and bicycles, in the video imagery obtained via the various imaging sensors.



Fig. 3 A high-level view of the point cloud generated in this work

3 Data Quality

Characterising the individual data elements using ISO 19157 concepts depends on establishing sensor characteristics, intermediate processing, and fitness for use in specific use cases. For example, accurately pointing an imaging sensor on an orbiting platform at the StreetDrone depends on accurate time and spatial navigation predictions for both the spacecraft and the StreetDrone. These, in turn, depend on the precision, accuracy, and sampling rate of the corresponding navigation systems, as well as a mechanism for time synchronization, using for example, Einstein time synchronisation. One important aspect of the publication of this dataset will be an evaluation of the suitability of both ISO 19115 (2014, 2019) (metadata) and ISO 19157 (2013) (quality) to characterise the data products in a way that supports individual use cases in a practical way.

4 Conclusions

We captured 12 runs of data with different combinations of sensors. Despite some sensor downtime, the positioning and orientation, LiDAR capture, and imagery and video data support the requirements of three projects in the OGC Testbed-19, prior Testbeds, and a prototype of the Ride-Hailing Use Case of the Metaverse Standards Forum Real/Virtual World Integration Domain Working Group. To enrich the datasets we provide a georeferenced point cloud map, and 3D mesh from the UAV and StreetDrone, and WebVMT geotagged videos. These data are published in an open-source repository with a CDLA-Permissive-2.0 license, with the intention of being used for research and development by the 3D geospatial and wider community.

While the data capture for this work has been completed, data will be added to the repository for the duration of the prototypes' lifetimes to extend this dataset and further strengthen the potential of this benchmark geography. We aim to provide semantic segmentation and classification of the point cloud, and to provide a mesh model and a potential conversion to CityGML.

References

- Away Team 3D Compass App. Retrieved from <https://awayteam.co.uk/products/3dcompass/about>. Accessed on 7 Jul 2023
- DPC Technology Watch Guidance Note 2021—Preserving 3D. Retrieved from <https://www.dpconline.org/docs/technology-watch-reports/2479-preserving-3d/file>. Accessed on 7 Jul 2023
- ISO 19157 (2013) Geographic information—data quality
- ISO 19115-1 (2014) Geographic information—metadata—part 1: fundamentals
- ISO 19115-2 (2019) Geographic information—metadata—part 2: extensions for acquisition and processing

- Lin TY et al (2014) Microsoft COCO: common objects in context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer vision—ECCV 2014. Lecture notes in computer science, vol 8693. Springer, Cham, pp 740–755
- Macdonald AL (1983) Clock synchronization, a universal light speed, and the terrestrial red-shift experiment. *Am J Phys* 51:795–797
- Metaverse Standard Forum. Retrieved from <https://metaverse-standards.org/>. Accessed on 7 Jul 2023
- OGC GeoPose Data Exchange Draft Standard. Retrieved from <https://www.ogc.org/standard/geomapose/>. Accessed on 7 Jul 2023
- OS Net Overview. Retrieved from <https://www.ordnancesurvey.co.uk/business-government/tools-support/os-net>. Accessed on 7 Jul 2023
- Rao KNS (1988) The rotation and Lorentz groups and their representations for physicists. Illustrated edn. Wiley, p 353
- Rockwood T, Steeger G, Stein M (2022) Generating realistic two-line element for notional space vehicles and constellations. [arXiv:2203.04204](https://arxiv.org/abs/2203.04204) [astro-ph.EP], [arXiv:2203.04204v1](https://arxiv.org/abs/2203.04204v1) [astro-ph.EP]. <https://doi.org/10.48550/arXiv.2203.04204>
- Testbed-17. Retrieved from <https://www.ogc.org/initiatives/t17/>. Accessed on 7 Jul 2023
- Testbed-18. Retrieved from <https://www.ogc.org/initiatives/testbed-18/>. Accessed on 7 Jul 2023
- Testbed-18 Engineering Report. Retrieved from <https://www.ogc.org/initiatives/testbed-18/>. Accessed on 7 Jul 2023
- Testbed-19 Project Initiative Overview. Retrieved from <https://www.ogc.org/initiatives/t-19/>. Accessed on 18 May 2023
- The W3C WebVMT Format. Retrieved from <https://www.w3.org/2021/sdw/proposals/geotagging/webvmt/>. Accessed on 7 Jul 2023

GIS/BIM Integration

Assessment of the LoD Specification for the Integration of BIM-Derived Building Models in 3D City Models



Jasper van der Vaart, Jantien Stoter, Abdoulaye Diakité, Filip Biljecki, Ken Arroyo Ohori, and Amir Hakim

Abstract Although level of detail (LoD) is a central concept in 3D city modelling, specifying different LoDs in an unambiguous manner is not straightforward. To resolve this, a set of frameworks have been developed. This paper evaluates the suitability of the LoD framework of (Biljecki et al. 2016) for 3D building models that have been generated directly from BIM models. The output of two BIM shell extractors are tested on how well they can be defined by the framework. It was found that although BIM-derived models can be specified by the framework to a certain degree, the framework is not fully capable to also specify lower quality models and to support all the output that may come from BIM shell extractors. This can be resolved by either addressing issues in the shell extractors' output or in the framework itself. The results of this research can be used to improve the LoD framework and to adjust the shell extractors output to better comply with unambiguous definitions of building models at different LoDs and could be a first step to standardise the conversion of BIM models at different LoDs to be used in urban applications.

Keywords LoD framework · 3D city model · BIM envelope extractor

This article was selected based on the results of a double-blind review of an extended abstract.

J. van der Vaart · J. Stoter (✉) · K. A. Ohori · A. Hakim
Delft University of Technology, Delft, ZH, The Netherlands
e-mail: j.e.stoter@tudelft.nl

J. van der Vaart
e-mail: J.A.J.vanderVaart@tudelft.nl
URL: <https://3d.bk.tudelft.nl/jvdvaart/>

A. Diakité
The University of New South Wales, Sydney, NSW, Australia

F. Biljecki
National University of Singapore, Singapore, Singapore

1 Introduction

Level of Detail (LoD) is a central concept in 3D city modelling (Biljecki et al. 2014). It describes how much an object has been abstracted from reality. The term originates from 3D computer graphics where it has a slightly different meaning. In computer graphics, this is a rather lenient phenomena primarily used for resource effective rendering (Luebke et al. 2003). In contrast, in 3D city models, different LoDs are used as collections of rules (i.e. specifications) for static visualisation, acquisition, modelling, generalisation and exchange of 3D data (Biljecki et al. 2014). The results of city scale analyses are influenced by the method and degree of object abstraction during acquisition and modelling (Biljecki et al. 2018; García-Sánchez et al. 2021; Peronato et al. 2016). A lenient, poorly applicable or vague ruleset regarding object abstraction can influence results in possibly unpredictable ways and can introduce uncertainty. Uncertainty can also be introduced when evaluating, comparing or exchanging city models that include abstracted objects that have not been clearly defined, see Fig. 1.

To reduce this uncertainty a ruleset is needed to unambiguously define 3D objects at different LoDs. These LoD-definitions are an interplay between the data acquisition methods (e.g. obtained from aerial, terrestrial or mobile measurements), modelling methods (e.g. manually or automatically) and the 3D data requirements of applications. The latter may vary between different applications. For example, LoD1 block models may be sufficient for noise simulations (Stoter et al. 2020), while the accurate roof structures of LoD2 models are needed for solar potential estimation of rooftops (Alam et al. 2016). More detailed building models with information about windows and doors stored in LoD3 models are important for estimating heat losses (Geiger et al. 2018).

Biljecki et al. (2016) refined the LoD specification of the CityGML 2.0 conceptual model to define such a ruleset for buildings, see Fig. 3. This specification has indeed reduced the vagueness of the four main CityGML LoDs. However, this refined framework was established when 3D city models were mainly a product of data acquisition through measurements and observations. In recent years, new ways have been developed to generate 3D data for 3D city models, one of which is the automated abstraction of BIM (Building Information Modeling) models. BIM models contain detailed information about the designs, planning, construction and exploitation of buildings and other constructions. IFC (Industry foundation classes)-files are an open and vendor-neutral standard for exchanging BIM models which contain information related to architecture, engineering and construction projects (Borrmann et al. 2018). The IFC-files store the majority of the data that is included in the source BIM file. BIM/IFC models are primarily utilised at an architectural scale. Thus, they often cover a smaller area than 3D city models while being more complex. The way buildings and other objects are modelled also differ between BIM and 3D city models. Buildings in BIM models are represented by a large collection of objects, while buildings represented in city scale models are primarily modelled by

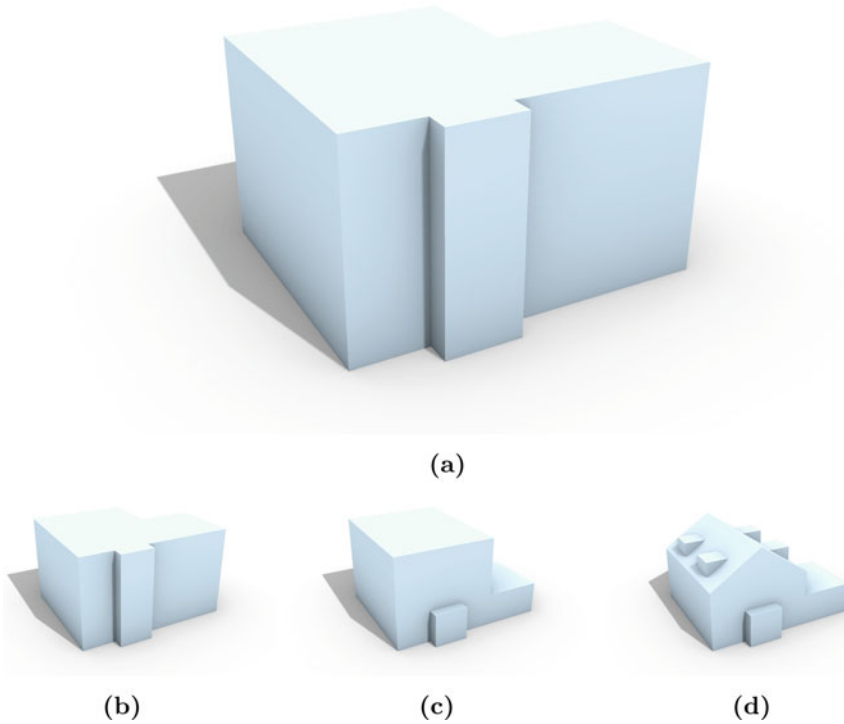


Fig. 1 A model without a clear supportive LoD framework can introduce uncertainty. For example, this model could be an abstraction of a building with a single flat roof (b), with multiple flat roofs at different heights (c) or it could even represent a building with a gable roof (d)

their shells, see Fig. 2. BIM data can however be abstracted into generalised building models/shells, which can be integrated in 3D city models to study the impact of their design on the environment and vice versa (Noardo et al. 2022).

At the time the framework by Biljecki et al. (2016) was developed, there was little practical experience available on the use of BIM models to generate abstracted models for their use in downstream 3D geo-applications. Since then, several BIM to GIS conversion methods and tools have been developed (and more are currently under development) to extract building-related concepts from BIM, such as building shells, building elements, storeys and rooms (Diakité 2023, van der Vaart 2022). These methods are quite diverse, following different rules and therefore result in vastly different outputs that often do not consider different LoDs.

BIM derived models were considered by Biljecki et al. (2016), but this was exclusively in LoD3.x. BIM models can in practice also be the direct source for further abstracted models than LoD3.x. These further abstracted BIM derived models may be required, because LoD3.x models might be too complex for certain applications. Utilising these too complex models may slow down the process unnecessarily while making the process more sensitive to errors.

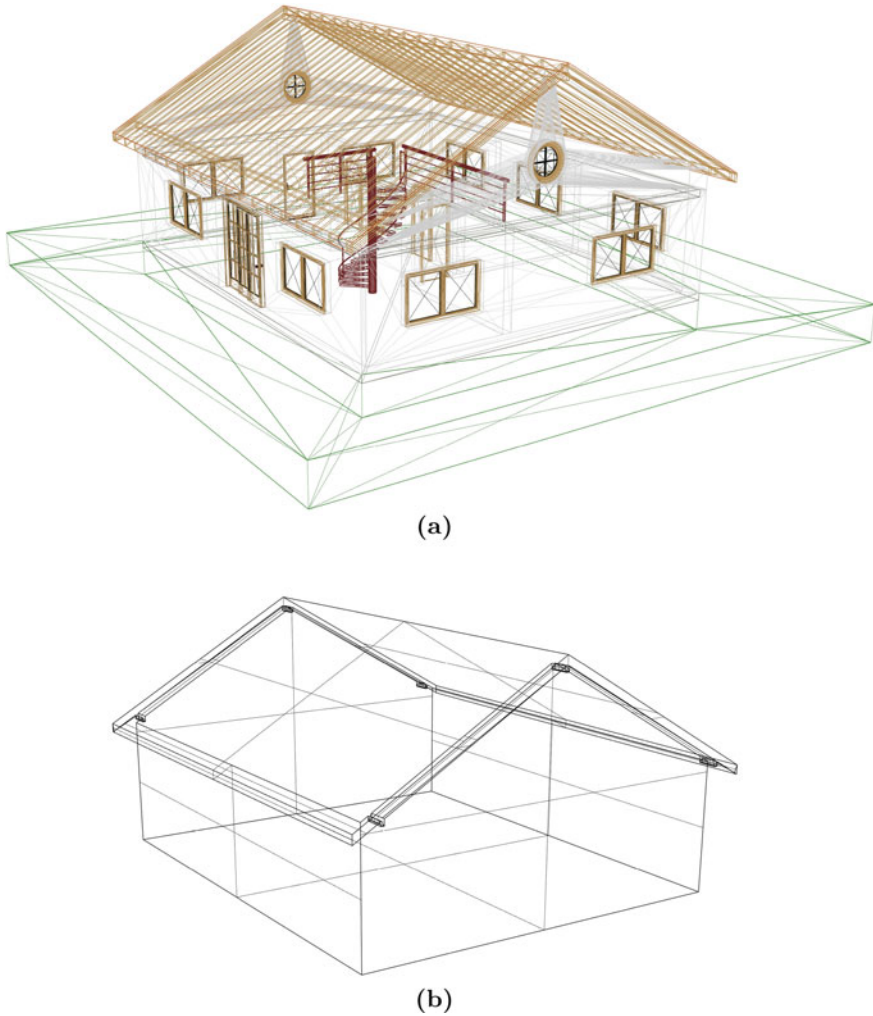


Fig. 2 The difference between a BIM model **a** and a city model **b** which are both based on the same building. The BIM model is not only more complex but it contains many differently modelled unique objects while the city model is represented by a single shell shape

One could reason that if the LoD3.3 shell could be generalised from a BIM model, the other LoD shells can be abstractions from this LoD3.3 shell, as is implied by Biljecki et al. (2016). However, experiences have shown that the quality of the input BIM model can impact the LoD shells that an automated abstraction process is able to extract. Thus, an input BIM model might not be suited to automatically extract LoD3.x output from while being sufficient for LoD2.x output or lower.

For these reasons, BIM derived output might not fit neatly into the existing LoD framework. Very little research on the fitting of BIM derived city models in LoD

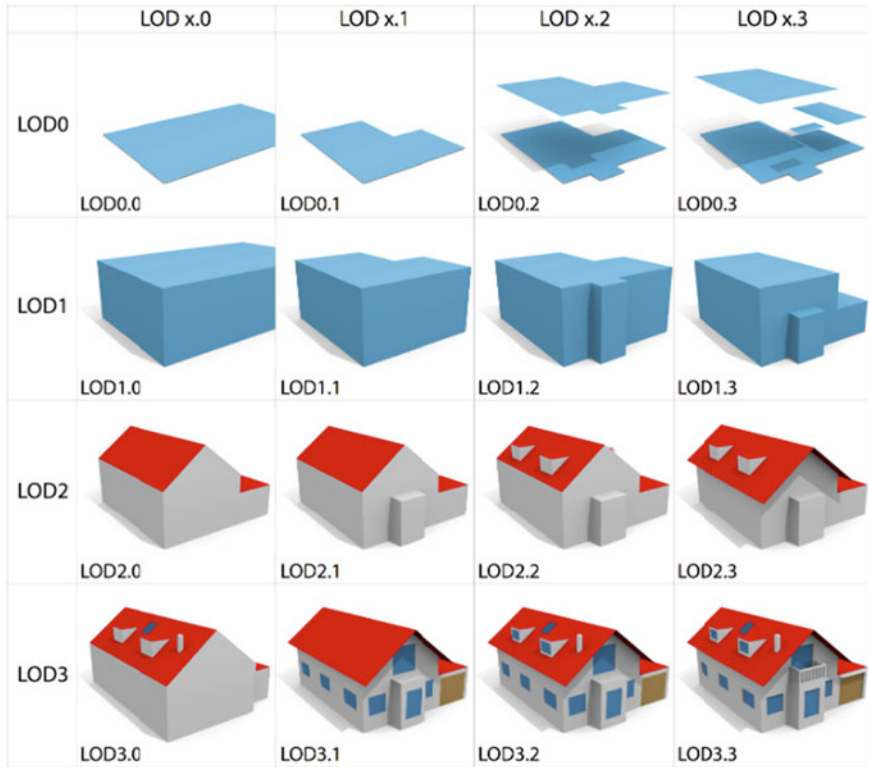


Fig. 3 The refined LoD specification of Biljecki et al. (2016)

frameworks has been done. Thus, it is unclear if BIM derived output can sufficiently be defined by the current LoD framework or if further refinements are needed to be able to define BIM-derived output in an unambiguous manner.

1.1 Goal and Scope

The goal of this paper is to evaluate how suitable the LoD framework of Biljecki et al. (2016) is for the integration of BIM-derived models in 3D city models. Based on this evaluation refinements can be proposed, or missing knowledge can be highlighted which requires further research. The evaluation in this paper is done for the entire LoD range and not limited to LoD3.x. The results of this research can be utilised to improve the LoD framework as well as BIM-to-Geo conversion methods and could be a first step to standardise the conversion of BIM models to building models at different LoDs to be used in urban applications.

To evaluate the suitability of the framework of Biljecki et al. (2016) for BIM-derived building models, we consider the output of two different BIM shell extractors according to the existing framework. These two extractors are the IFC_BuildingEnvExtractor (van der Vaart 2023b) and BIMShell (Diakit  2023). Both extractors are still in development but are in their current state able to extract high quality outer shells from most IFC-files.

1.2 The Software Tools

The IFC_BuildingEnvExtractor (van der Vaart 2023b) extracts multiple differently abstracted LoD shells from IFC-files. The classification/abstraction of the shells has been developed in such a way that the output complies as much as possible with the LoD specification of Biljecki et al. (2016). The possible number of extractable shells depends on the quality of the input model, see Fig. 4. The tool defines three levels of extraction where the extraction methods are similar per level. Low-level shells (LoD0.0 & 1.0) are generated by approximating a smallest bounding box around the input model. These shells can be extracted from any valid IFC-file regardless of how well it is constructed. Mid-level shells (LoD0.2, 1.2, 1.3 & 2.2) are generated by isolating the roof structure of the input model and either projecting or downward extruding it. Due to this process being based on the roofing structure, an accurate output requires an input BIM model that has the roof well modelled. High level shells (LoD3.2) are generated from the surfaces of objects that together construct the outer envelope of the BIM model. These shells can only be extracted accurately if the objects constructing the outer shell are well constructed in the original BIM model.

The tool only considers a subset of object types available in the IFC-format. When the default settings are used only the space dividing objects (van der Vaart 2022) are used for the extraction of the shells. These space dividing objects are: IfcBeam, IfcColumn, IfcCovering, IfcCurtainWall, IfcDoor, IfcMember, IfcPlate, IfcRoof, IfcSlab, IfcWall and IfcWindow objects. Additionally, if the user desires additional object types to be included they are able to set this manually when running the tool. Aside from this customisation, the tool requires limited user input: an input path, output path, and voxel size. The voxel size is used to generate a voxel grid to roughly filter the objects that are being processed. This parameter has usually negligible effect on the end result, but fine tuning it to certain types of buildings can improve computation speed.

BIMShell is another tool that also extracts building shells from BIM files, but currently it does not constrain itself to any LoD specification, see Fig. 5. Its main purpose is to reduce the size of the original BIM model and to automatically remove internal elements, while preserving as much as possible of their external appearance. As such, it is mainly meant for visualisation and applications that do not need more than the visual resemblance of a building model. The current version supports IFC and several other geometric and CAD formats as input (obj, 3ds, fbx, etc.). It produces two shells: a *raw shell* that corresponds to a collection of the exterior faces of the

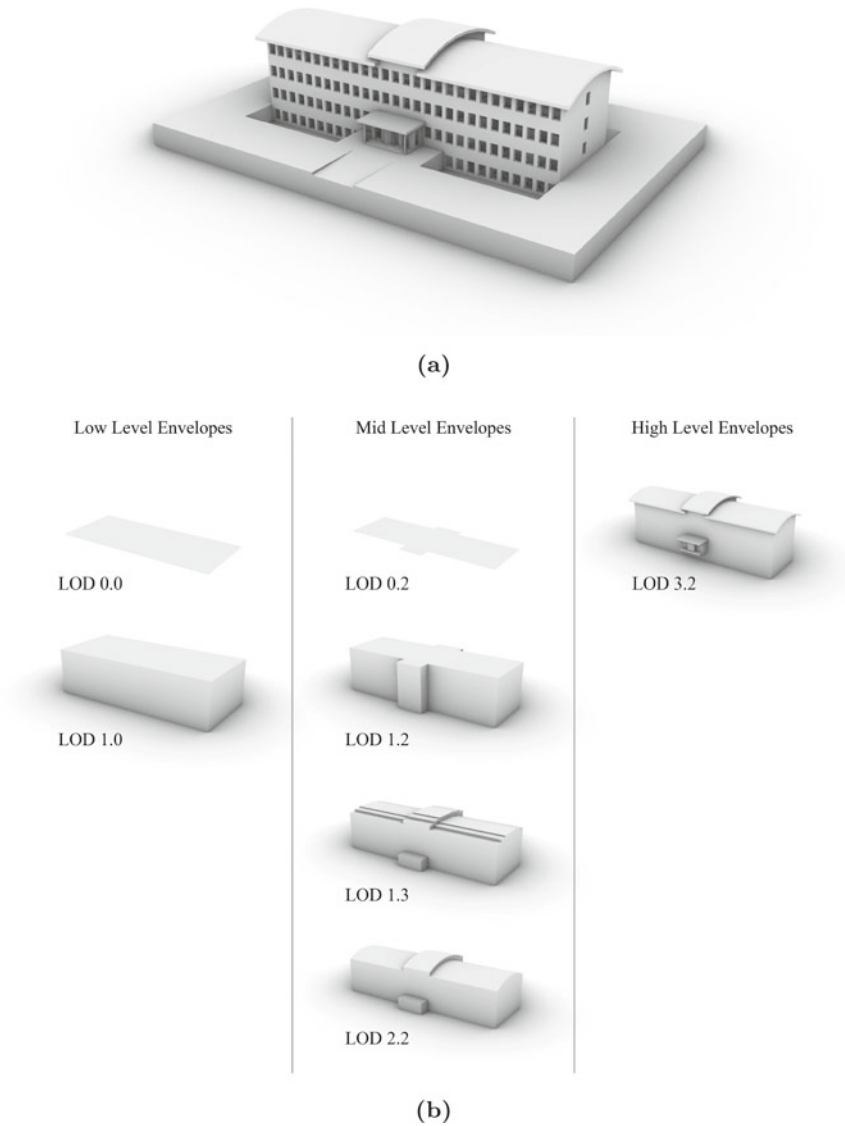


Fig. 4 The possible extracted shells of the IFC_BuildingEnvExtractor are categorised in three different levels of quality. Low level shells are accurately created from any quality model. Mid level shells are only accurately created from models that have well defined roofing structures. High level shells are only created from models that have well defined objects that construct the outer shell. All shells are generated from the model in Fig.4a

input model detected through a ray tracing process, and a *voxel shell* that is, as its name suggests, a watertight shell built from a voxelization of the input. The few parameters required by the tool include an error tolerance corresponding to the

resolution of the voxel grid used. It does not assume any condition and it does not require any constraint on the input models except that gaps that are larger than the tolerance are likely to cause the production of false shells (e.g. the output includes indoor faces).

2 Methodology

The evaluation of the LoD framework for BIM-derived building models is done by comparing output models created by the two software tools to models defined in the current framework. Biljecki et al. (2016) give a clear collection of rules to which every LoD has to adhere which makes it easy to see if the outputted model can be classified as one of the specified LoDs. This enables us to determine if an output model does not comply with the framework, why it does not comply, and how this can be resolved (by either addressing issues in the shell extractors' output or in the framework itself).

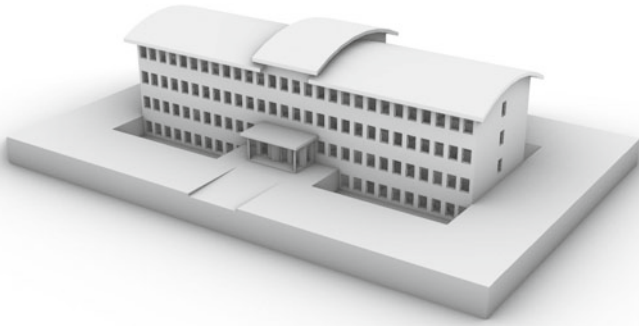
The primary evaluation that was done is testing a set of output models against the different rules per LoD. Figure 6 shows the followed rules as described in Biljecki et al. (2016). When considering the different LoDs of the existing framework, the written rules were followed as much as possible instead of interpreting the rules from the visual representation of the framework (Fig. 3). This is done to reduce subjectivity in the classification process.

This evaluation can be easily executed with the output of the IFC_BuildingEnvExtractor because its output models are already classified in LoDs according to the framework. This is not the case for BIMShell. Therefore, prior to testing BIMShell's output to the LoD rules, we first needed to define the most likely LoD of the output. This is done by testing the BIMShell output to the rules of every LoD in the framework. The output is considered the LoD with which it complies the most. Often it does not comply with all the rules related to the LoD it is classified in, as will be seen further.

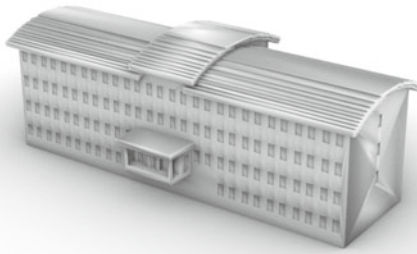
Biljecki et al. (2016) state that the 3D LoD shells are volumetric. This suggests that the 3D shapes are required to be valid watertight solids. To evaluate if the models are watertight solids, two different approaches were taken. Firstly, the output models were evaluated manually/visually to see if there were any glaring issues present. Secondly, the output models were checked by a small c++ program that searches for matching edges to guarantee water-tightness. This program relies on the CJT library (van der Vaart 2023a) to open CityJSON files and OpenCASCADE library (Open Cascade n.d.) to open OBJ files and check if the edges have neighbours.

The settings used for IFC_BuildingEnvExtractor in all the evaluations were the default settings with a voxel size of 1 by 1 m. The settings used for BIMShell in all the evaluations were a 0.2m tolerance with a detailed raw shell look.

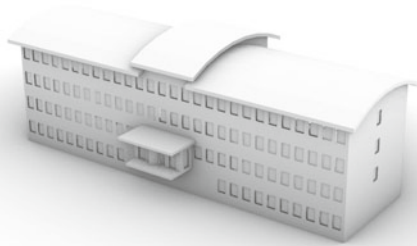
The input IFC models that have been used were picked to represent different building shapes while at the same time being fairly small and simple. This allows us to evaluate different building shapes while excluding potential issues with the created



(a)



(b)



(c)

Fig. 5 All the possible extracted shells of BIMShell. **b** is a voxelized shape. **c** is a collection of exterior extracted faces. The shells are generated from the model in Fig. 5a

| Requirements | Refined levels of detail | | | | | | | | | | | | | | | |
|--|--------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | 0.0 | 0.1 | 0.2 | 0.3 | 1.0 | 1.1 | 1.2 | 1.3 | 2.0 | 2.1 | 2.2 | 2.3 | 3.0 | 3.1 | 3.2 | 3.3 |
| Individual buildings | | • | • | • | | • | • | • | • | • | • | • | • | • | • | • |
| Large building parts (>4 m, 10 m ²) | | • | • | • | | • | • | • | • | • | • | • | • | • | • | • |
| Small building parts, recesses and extensions (>2 m, 2 m ²) | | | • | • | | | • | • | | • | • | • | • | • | • | • |
| Top surface ⁽⁰⁾ | | | S | M | S | S | S | M | | | | | | | | |
| Explicit roof overhangs (if >0.2 m) | | | | | | | | | | | | • | | • | • | • |
| Roof superstructures ⁽¹⁾ (larger than 2 m, 2 m ²) | | | | | | | | | | | • | • | • | | • | • |
| Other roof details (e.g. chimneys >1 m) | | | | | | | | | | | | | • | | • | • |
| Openings ⁽²⁾ (>1 m, 1 m ²) | | | | | | | | | | | | | R | W | • | • |
| Balconies (>1 m) | | | | | | | | | | | | | | • | • | • |
| Embrasures, other façade and roof details, and smaller windows (>0.2 m) | | | | | | | | | | | | | | | | • |

Fig. 6 Summary of the LoD framework rules by Biljecki et al. (2016). (0) Applicable only to LoD0.y and LoD1.y: S—Single top surface; M—Multiple top surfaces if the difference in height of the extruded building elements is significant (larger than 2 m). (1) It includes dormers and features of comparable size and importance (e.g. very large chimneys). (2) R-only openings on roofs; W-only openings on walls. In R, openings on dormers are not required

shells caused by exceptional cases that the software can not deal with. The evaluation assesses how the output models fit in the LoD framework. It does not evaluate the performance of the software tools. All models are openly available online, or can be easily recreated when desired.

An overview of the models can be found in Table 1. In the Appendix a visual representation of the models is included.

3 Results

Following the described method, the results of these analyses are split in two parts. The first part covers the fitting of the output of BIMShell into a suitable LoD from the framework of Biljecki et al. (2016), see Table 2. The second part tests how well

Table 1 Summary of the used input BIM models

| Input model name | Object count* | Building type** | Storeys | Has overhang (Y/N) | Source |
|----------------------|---------------|-----------------|---------|--------------------|-----------------|
| AC20-FZK-Haus | 105 | 1 | 2 | ✓ | KIT (n.d.) |
| AC20-Institute-Var-2 | 896 | 3 | 3 | ✓ | KIT (n.d.) |
| AC-20-Smiley-West-10 | 972 | 2 | 4 | ✓ | KIT (n.d.) |
| RE16_E3D_Building | 552 | 3 | 2 | ✗ | IBPSA (2021) |
| DigitalHub | 775 | 3 | 3 | ✗ | RWTH E3D (2022) |
| RAC_basic_sample | 450 | 1 | 2 | ✓ | Revit |

* object count done with BimVision. ** 1 = freestanding house, 2 = terraced house, 3 = office building

Table 2 Evaluation of the classification of the output of BIMShell

| Requirement | Raw shell | Voxel shell |
|--|-----------|-------------|
| Individual buildings | ✓ | ✓ |
| Large building parts (>4 m, 10 m ²) | ✓ | ✓ |
| Small building parts recesses and extensions (2 m, 2 m ²) | ✓ | ✓ |
| Top surface ⁽⁰⁾ | P | M |
| Explicit roof overhangs (if >0.2 m) | ✓ | ✓ |
| Roof superstructures ⁽¹⁾ (larger than 2 m, 2 m ²) | ✓ | ✓ |
| Other roof details (e.g. chimneys > 1 m) | ✓ | ✓ |
| Openings ⁽²⁾ (> 1 m, 1 m ²) | ✓ | ✓ |
| Balconies (> m) | ✓ | ✓ |
| Embrasures, other façade and roof details, and smaller windows (>0.2 m) | ✓ | ✓ |

S Single top surface; M Multiple top surfaces; P Precisely followed top faces

Table 3 Evaluation of the fit of the BIMShell raw shell output models to the existing LoD framework

| Input model name | Compliant LoD3.2 (Y/N) | Compliant LoD3.2* (Y/N) |
|----------------------|------------------------|-------------------------|
| AC20-FZK-Haus | ✗ | ✓ |
| AC20-Institute-Var-2 | ✗ | ✓ |
| AC-20-Smiley-West-10 | ✗ | ✓ |
| RE16_E3D_Building | ✗ | ✓ |
| DigitalHub | ✗ | ✓ |
| RAC_basic_sample | ✗ | ✓ |

* disregard of the volumetric/solid requirement

the output of BIMShell and the IFC_BuildingEnvExtractor fit to the classified LoD as defined in the framework, see Tables 3, 4 and 5. This part also evaluates if the output of both tools are solid 3D shapes, see Tables 6 and 7.

Table 4 Evaluation of the fit of the IFC_BuildingEnvExtractor output models to the existing LoD framework

| Input model name | Compliant LoD1.0(Y/N) | Compliant LoD1.2(Y/N) | Compliant LoD1.3(Y/N) | Compliant LoD2.2(Y/N) | Compliant LoD3.2(Y/N) |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| AC20-FZK-Haus | × | × | × | ✓ | × |
| AC20-Institute-Var-2 | × | × | × | ✓ | × |
| AC-20-Smilely-West-10 | × | × | × | ✓ | × |
| RE16_E3D_Building | ✓ | ✓ | ✓ | ✓ | × |
| DigitalHub | ✓ | ✓ | ✓ | ✓ | × |
| RAC_basic_sample | × | × | × | ✓ | × |

Table 5 Evaluation of the fit of the IFC_BuildingEnvExtractor output models to the existing LoD framework. The requirements related to wall and roof openings are ignored

| Input model name | Compliant LoD1.0(Y/N) | Compliant LoD1.2(Y/N) | Compliant LoD1.3(Y/N) | Compliant LoD2.2(Y/N) | Compliant LoD3.2(Y/N) |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| C20-FZK-Haus | × | × | × | ✓ | ✓ |
| AC20-Institute-Var-2 | × | × | × | ✓ | ✓ |
| AC-20-Smilely-West-10 | × | × | × | ✓ | ✓ |
| RE16_E3D_Building | ✓ | ✓ | ✓ | ✓ | ✓ |
| DigitalHub | ✓ | ✓ | ✓ | ✓ | ✓ |
| RAC_basic_sample | × | × | × | ✓ | × |

Table 6 Results of the solid check for the output of the BIMShell

| Input model name | Solid raw shell (LoD3.2) (Y/N) | Solid voxel shell (Y/N) |
|-----------------------|--------------------------------|-------------------------|
| AC20-FZK-Haus | × | ✓ |
| AC20-Institute-Var-2 | × | ✓ |
| AC-20-Smilely-West-10 | × | ✓ |
| RE16_E3D_Buildin | × | ✓ |
| DigitalHub | × | ✓ |
| RAC_basic_sample | × | ✓ |

3.1 Fitting of the BIMShell Output

We can see in Table 2 that both the raw shell and the voxel shell output of BIMShell comply with all the rules defined by Biljecki et al. (2016) for LoD3.3. This would suggest that both could be classified as LoD3.3. There is however one nuance that challenges this conclusion. The raw shell output of BIMShell follows the shape of the building very closely and does not exclude any detail. It also complies well with the description of LoD3.3: “an architecturally detailed model ... that contains features of size larger than 0.2 m, including embrasures of windows (i.e. making windows 3D),

Table 7 Results of the solid check for the output of the IFC_BuildingEnvExtractor

| Input model name | Solid LoD1.0 (Y/N) | Solid LoD1.2 (Y/N) | Solid LoD1.3 (Y/N) | Solid LoD2.2 (Y/N) | Solid LoD3.2 (Y/N) |
|----------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| AC20-FZK-Haus | ✓ | ✓ | ✓ | ✓ | ✓ |
| AC20-Institute-Var-2 | ✓ | ✓ | ✓ | ✓ | ✓ |
| AC-20-Smiley-West-10 | ✓ | ✓ | ✓ | ✓ | ✓ |
| RE16_E3D_Building | ✓ | ✓ | ✓ | ✓ | ✓ |
| DigitalHub | ✓ | ✓ | ✓ | ✓ | ✓ |
| RAC_basic_sample | ✓ | ✓ | ✓ | ✓ | ✗ |

awnings and similar features of comparable size” Biljecki et al. (2018). However, the voxel shell does not closely follow the shape of the building, nor its roofing structure. It results in a shape that has multiple flat top surfaces that represent the roofing structure but is not identical to it. So, while the raw shell could be considered LoD3.3, the voxel shell cannot. The way the roofing structure is modelled in the voxel shell output would presume it to be more closely related to LoD1.x. However, the rest of the shell is considered too detailed to fit with LoD1.x. In the end we were unable to find a suitable LoD in the framework and are thus unable to further analyse the framework’s suitability of the voxel shell output of BIMShell. Supporting voxel shells in the BIM-based LoD framework, might therefore require the addition of yet another LoD.

3.2 Evaluation of the Framework Fit for the Output

Table 3 shows if the output of BIMShell fully fits in the framework. Due to the challenging classification of the voxel shell we only evaluated the fit of the raw shell to the framework’s rules. It can be seen that when considering all the set rules it can not be properly considered LoD3.2 according to the framework. Table 6 summarises the main issue, i.e. none of the raw shells are closed solids. Gaps can exist, which mainly occur at places where small details were present in the input model e.g., windows and doors. If we exclude the volumetric/solid requirements the framework is much better suited to define the output of BIMShell.

Table 4 shows if the output of the IFC_BuildingEnvExtractor fully fits in the framework. As can be seen, many of the output models do not completely fit in the framework. The exception are two of the three evaluated office buildings. These two models show an output that is fully consistent with significantly more LoDs than the other input models.

None of the output models are compliant with the LoD3.2 shell. However, when we ignore the wall and roof opening requirements of the framework (see Sect. 4.4), the LoD3.2 output does fit the framework very well, see Tables 5 and 7.

4 Discussion and Evaluation of the Framework

The evaluation of the shell output highlighted a set of issues with the current framework when attempting to define BIM-derived building models in an unambiguous way. These issues are addressed and discussed in the following subsections.

4.1 Voxels

The fitting of BIMShell's voxel shell highlights the first issue of the framework of Biljecki et al. (2016) for BIM derived models: it does not mention voxelized shapes. This makes it impossible to classify the voxel shell output of BIMShell to a certain LoD within the framework. This shell was considered to be deviating so far from the defined LoDs that it could not be fit in a meaningful manner and this shell's classification was discarded.

The absence of voxelization in the framework could be seen as an issue. Voxelization can be utilized to create watertight geometries which can even be used to approximate shapes from incomplete and/or faulty BIM data sources (Huang et al. 2020; Mulder 2015). IFC-models have shown to often contain geometric issues that make it difficult for envelope extractors to generate high LoD models. These issues range from misplaced objects (van der Vaart 2022), to missing or corrupted geometry (Arroyo Ohori et al. 2018; Krijnen et al. 2020). With the help of voxelization, shell shapes could still be approximated when these issues occur. However, the fact that voxelization and voxel related shapes are a solution for creating shells from erroneous input models does not mean that these outputted shells are desirable in further analysis. The effects of these models on analysis (and therefore the need of such models in urban applications) has to be examined first, before it can be decided if these models should have their place in city scale models and are important to include in the LoD framework.

4.2 Non-watertightness

The raw shell output of BIMShell follows the framework geometrically quite well. The major exception is that the raw shell is not a volumetric solid, but a collection of surfaces. This is however an issue that cannot be attributed solely to the framework. A shell is needed to separate the outside from the inside of an building. A somewhat closed shape is also required to enable geometrical analysis in 3D such as volume-calculation and CFD-modeling.

BIMShell also occasionally outputs models that have seemingly closed shapes but include some residual surfaces of interior objects. The outer shell of the shape is closed, or close to being closed, but can not be considered solid due to these residual

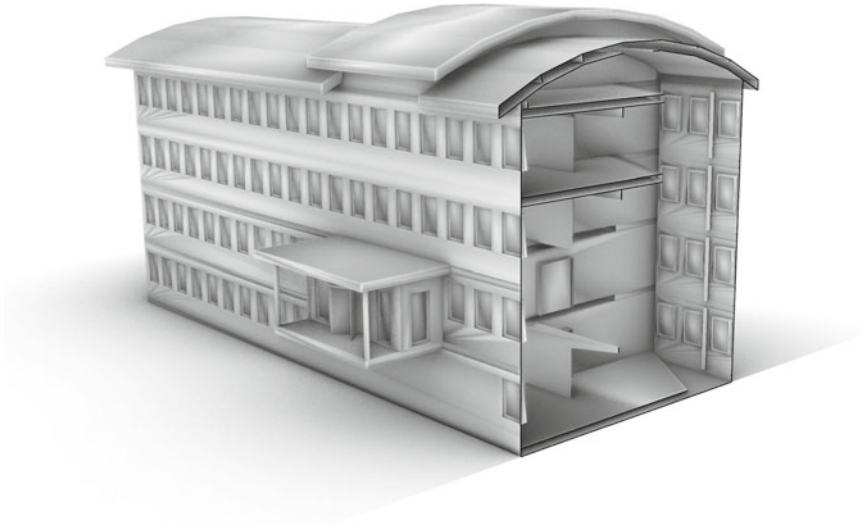


Fig. 7 Section of the BIMShell output of the AC20-Institute-Var-2 model. This section shows a seemingly closed shell while still having a subset of interior faces

surfaces. This means that the boundary between interior and exterior is present, but it is not completely explicit where it lies, see Fig. 7. Possibly for exterior analysis (e.g. exterior wind and sunlight analysis) these models could still be suitable. More research should be done on further usefulness of models that are not completely closed in urban applications to be definite about the inclusion of these models in the framework.

4.3 Footprints and Roof Outlines

The evaluation of the output of IFC_BuildingEnvExtractor shows that there are some ambiguities with the lower level LoD shell classification. The main issue is the use of footprints and roof outlines. Biljecki et al. (2016) imply that LoD0.x, LoD1.x and LoD2.x models are based on the footprint, while LoD0.x may optionally include the roof outline. However, at LoD2.x it is stated that “When roof overhangs are not available, the walls are usually obtained as projections from the roof edges to the ground, inherently increasing the volume of the building”. This exception was added due to the cost of collecting the required data to accurately determine the overhang. This exception is not mentioned in subsequent parts of the paper that presents the framework. It is thus not completely clear if a valid LoD2.x model is based on the footprint, the projected roof outline, or if both options are framework compliant. In any case, if only one is allowed it is not clearly dictated by the framework. If both are allowed it introduces even more vagueness because the user may not be aware of

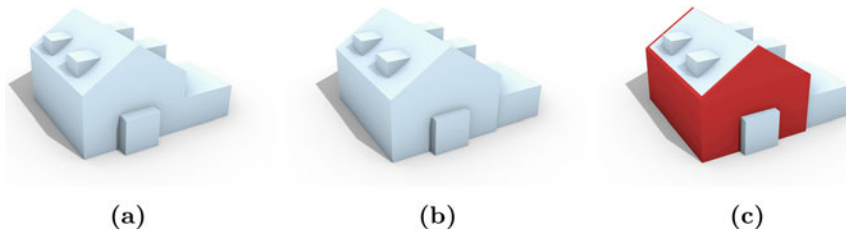


Fig. 8 Two LoD2.3 models displayed in **a** and **b**. Where **a** is a extrusion of the roofing structure and **a** is the same shape, but clipped to comply with the building's footprint. This results in the displayed in **c** highlighted shape to be removed

such a openness to this LoD and it cannot be specified which is used, the footprint or the roof outline. If the input model has a roofing structure with overhang, the geometry of the projected roof structure will differ from its footprint, see Fig. 8 and this can have an impact on the analysis in which the models are used. A solution to distinguish between both options is to include the explicit use of footprints or projected roof edges for extruded objects in the framework.

For the evaluation, we used the roof outline as base for the LoD2.x. For the LoD1.x the footprint was considered to be the base. Although the paper presenting the framework is not explicit about the flexible use of footprints or projected roof outlines in lower LoDs it seems to imply that this flexibility is not present. Due to this we presume that that lower LoDs should exclusively be based on the footprint to be considered compliant.

However, the LoD0.x and LoD1.x shells show that they would have been valid if they were allowed to be based on the roof outline instead of the footprint. Extending the reasoning of Biljecki et al. (2016) for LoD2.x, it could be said that creating LoD0.x and LoD1.x from BIM models based on only footprints would bring a larger cost with it as well. The computations to successfully create compliant shapes for footprints from BIM would cost a lot more time to execute, and might become unreliable or ineffective. It is easier to extract roof outline and extrude these downwards. Additionally the BIM model could be required to be made more precisely and robust, costing time and money. This could possibly be avoided if LoD0.x and LoD1.x were also allowed to be based on the projected roof outline.

Extending the flexibility of footprint/roof outline use, without clearly distinguishing between both, to LoD0.x and LoD1.x might make the fitting of BIM output into the framework more easy. It will however also introduce more ambiguity since it is not clear which 2D geometry has been used for the extrusion. This ambiguity is already present in practice, since LoD1.x and LoD2.x models are sometimes generated from 2D polygons that represent the projected roof outline and not the footprints e.g., 3D BAG (TU Delft and 3DGI n.d.) while this distinction is not supported in the framework. We therefore recommend to explicitly add this refinement regarding the used 2D primitive (footprint or roof outline) to the framework.

4.4 *Missing Semantic and Opening Information*

The LoD3.x output of IFC_BuildingEnvExtractor does not include the geometry related to wall and roof openings that are required by the framework. BIMShell has these surfaces present, but it does not semantically classify these surfaces.

This highlights an interesting issue with the framework for BIM-derived models. It is stated that openings have to be modelled, but it does not state how to correctly model these openings. The best option to do so is to model these openings both geometrically and semantically. In earlier work it is stated that indeed every object should be modelled both geometrically and semantically (Biljecki et al. 2014). However, in the paper presenting the framework it is explicitly mentioned that semantic data is not used for its construction (Biljecki et al. 2016). It is thus unclear if the openings should be modelled according to the earlier described method or if they are only required to be modelled geometrically.

Since the framework mentions it actively disregards semantics, we did so as well for the evaluation of LoD3.x output. However, if we would consider the framework to require the openings to be modelled both geometrically and semantically, both tested software tools would not generate valid output. One could reason that both the tools would then generate LoD2.3 as highest quality output because of these missing elements. But both outputs do include small roof details and balconies which are all exclusively part of a higher LoD (higher than LoD2.x).

IFC_BuildingEnvExtractor does not model the openings at all, not semantically and also not geometrically. LoD3.x models are often used for evaluations that require wall and roof openings to be properly modelled, such as heat loss estimations. However, there has been little research regarding the need for LoD3.x shapes in applications that do not require these openings. This is required to decide if LoD3.x models with detailed building elements but without openings should be supported by the framework.

5 **Conclusions and Proposed Refinements**

In this paper we evaluated how suitable the framework of Biljecki et al. (2016) is to unambiguously specify BIM-derived building models. This research was done by comparing the output of two BIM shell extractors with the LoDs defined in the framework: IFC_BuildingEnvExtractor and BIMShell.

The results highlight two issues that arise when attempting to fit BIM-derived building models in the framework and to define these models in an unambiguous manner. These issues also highlight some other general needs to resolve the ambiguity of the framework.

These are the definition of wall/roof openings in case of LoD3.x models and the definition of the used 2D primitive (roof outline or footprint) for LoD0.x, LoD1.x and LoD2.x models. The first issue can easily be resolved by clearly describing the

rules depending on the data needs from urban applications (either always requiring openings at LoD3.x that are both geometrically and semantically modelled or also allowing LoD3.x models containing detailed building elements without containing openings). The second issue requires to decide if both the roof outline and footprint should be allowed to be used as a base for LoD2.x models. If so, a clear and concise way of signifying which source is used has to be included. Additionally, the flexibility of roof outline and footprint use should be considered for lower LoDs as well. The IFC_BuildingEnvExtractor outputs LoD1.x shells that are based on the projected roof outline instead of the footprint. This is not framework compliant. However, roof outline based shapes will be more easy to generate from BIM-derived models, also for LoD1.x models. Additionally, roof outline based LoD1.x models are already used in practice. Apart from including the distinction of the used 2D primitive in the framework, more research is needed on the effect of using either of them in city scale analysis.

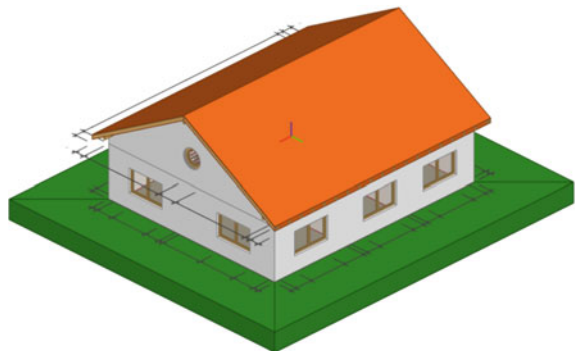
Finally, the framework does not have a place for voxelized shapes and non solid volumetric shapes. Further research is needed to see if such models are required in urban applications.

The results of this research can be used to improve the LoD framework and to adjust the shell extractors output to better comply with unambiguous definitions of building models at different LoDs. This could be a first step to standardise the conversion of BIM models at different LoDs to be used in urban applications.

Appendix

See Figs. 9, 10, 11, 12, 13 and 14 for a visual representation of the used models. More detailed information can be found in Table 1.

Fig. 9 Visual isometric representation of the AC20-FZK-Haus model



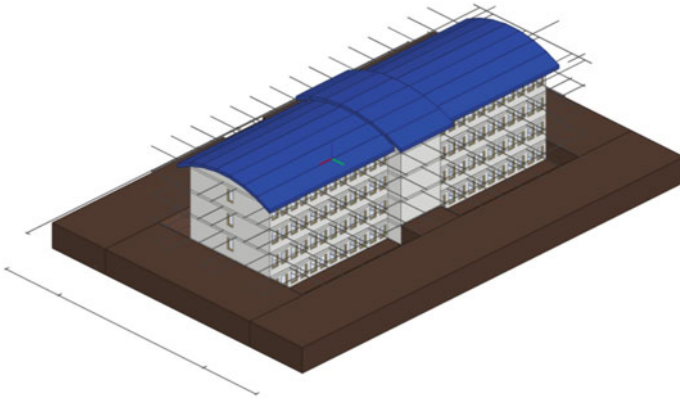


Fig. 10 Visual isometric representation of the AC20-Institute-Var-2 model

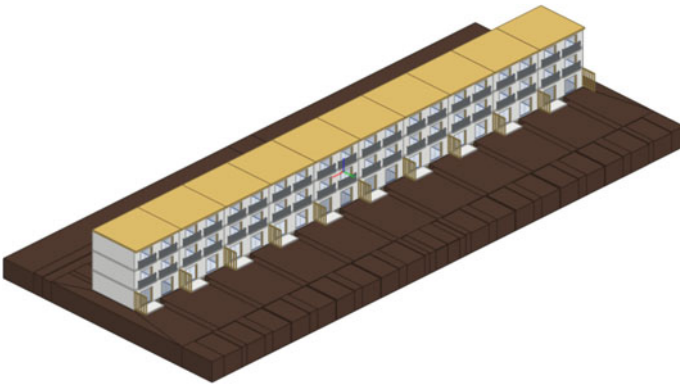


Fig. 11 Visual isometric representation of the AC-20-Smiley-West-10-Bldg model

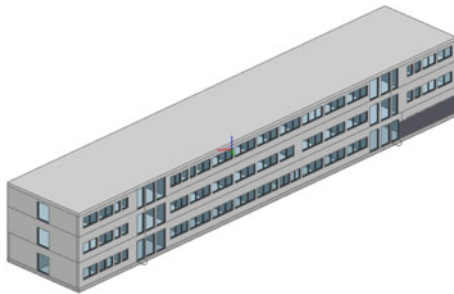


Fig. 12 Visual isometric representation of the RE16_E3D_Building_2x3_Testversion model

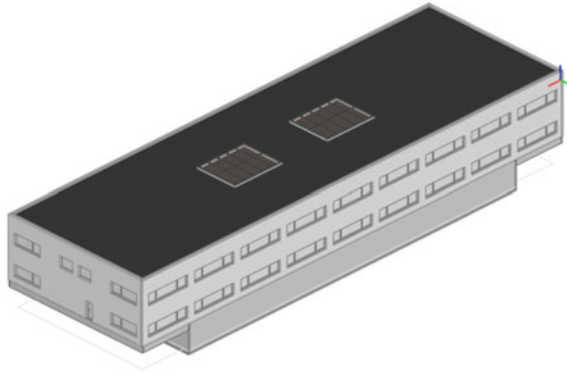


Fig. 13 Visual isometric representation of the FM_ARC_DigitalHub model

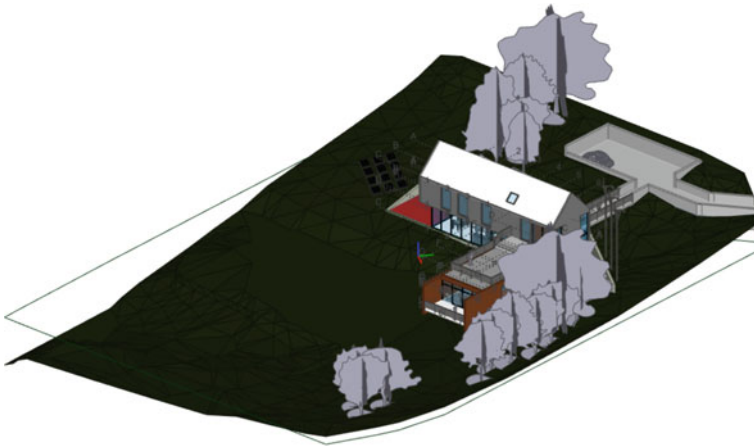


Fig. 14 Visual isometric representation of the RAC_basic_sample_project model

Acknowledgements This project has received funding from the European Union's Horizon Europe programme under Grant Agreement No.101058559 (CHEK: Change toolkit for digital building permit).

References

- Alam N, Coors V, Zlatanova S, Oosterom P (2016) Resolution in photovoltaic potential computation. ISPRS Ann Photogrammetry, Remote Sens Spat Inf Sci 4:89–96
- Arroyo Ohori K, Diakit  A, Krijnen T, Ledoux H, Stoter J (2018) Processing BIM and GIS models in practice: experiences and recommendations from a geo BIM project in The Netherlands. ISPRS Int J Geo-Inf 7(8):311

- Biljecki F, Ledoux H, Stoter J, Zhao J (2014) Formalisation of the level of detail in 3d city modelling. *Comput, Environ Urban Syst* 48:1–15
- Biljecki F, Ledoux H, Stoter J (2016) An improved LoD specification for 3d building models. *Comput, Environ Urban Syst* 59:25–37
- Biljecki F, Heuvelink GB, Ledoux H, Stoter J (2018) The effect of acquisition error and level of detail on the accuracy of spatial analyses. *Cartography Geogr Inf Sci* 45(2):156–176
- Borrmann A, Beetz J, Koch C, Liebich T, Muhic S (2018) Industry foundation classes: a standardized data model for the vendor-neutral exchange of digital building models. *Technology foundations and industry practice, Building information modeling*, pp 81–126
- Diakité A (2023) Bimshell. <https://bimshell.app/>. Accessed 04 July 2023
- García-Sánchez C, Vitalis S, Paden I, Stoter J (2021) The impact of level of detail in 3d city models for CFD-based wind flow simulations. *The international archives of the photogrammetry, remote sensing and spatial information sciences*, pp 67–72
- Geiger A, Benner J, Häfele KH, Hagenmeyer V (2018) Thermal energy simulation of buildings based on the city-GML energy application domain extension. In: *BauSIM2018–7. Deutsch-Österreichische IBPSA-Konferenz: Tagungsband*. Hrsg., P. Von Both, pp 295–302
- Huang J, Zhou Y, Guibas L (2020) Manifoldplus: a robust and scalable watertight manifold surface generation method for triangle soups. *arXiv preprint arXiv:2005.11621*
- IBPSA (2021) project1-wp-2-2-bim. <https://github.com/ibpsa/project1-wp-2-2-bim/find/master>
- KIT (n.d.) Kit ifc examples. https://www.ifcwiki.org/index.php?title=KIT_IFC_Examples
- Krijnen T, Noardo F, Ohori KA, Ledoux H, Stoter J (2020) Validation and inference of geometrical relationships in IFC. In: *Proceedings of the 37th international conference of CIB W*, vol 78, pp 98–111
- Luebke D, Reddy M, Cohen JD, Varshney A, Watson B, Huebner R (2003) Level of detail for 3D graphics. Morgan Kaufmann
- Mulder D (2015) Automatic repair of geometrically invalid 3d city building models using a voxel-based repair method. Master's thesis, Delft University of Technology
- Noardo F, Wu T, Arroyo Ohori K, Krijnen T, Stoter J (2022) IFC models for semi-automating common planning checks for building permits. *Autom Constr* 134:104,097, ISSN: 0926–5805
- Open Cascade (n.d.) Open cascade technology collaborative development portal. <https://dev.opencascade.org/>. Accessed 20 May 2023
- Peronato G, Bonjour S, Stoeckli J, Rey E, Andersen M (2016) Sensitivity of calculated solar irradiation to the level of detail: insights from the simulation of four sample buildings in urban areas. In: *Proceedings of PLEA 2016, 32th international conference on passive and low energy architecture*, CONF
- RWTH E3D (2022) Digitalhub. <https://github.com/RWTH-E3D/DigitalHub>
- Stoter J, Peters R, Commandeur T, Dukai B, Kumar K, Ledoux H (2020) Automated reconstruction of 3d input data for noise simulation. *Comput, Environ Urban Syst* 80(101):424
- TU Delft, 3DGI (n.d.) 3d bag viewer. <https://3dbag.nl/en/viewer>
- van der Vaart J (2022) Automatic building feature detection and reconstruction in IFC models. Master's thesis, TU Delft
- van der Vaart J (2023a) Cjt. <https://github.com/jaspervdv/CJT>. Accessed 20 May 2023
- van der Vaart J (2023b) Ifc_buildingenvextractor. https://github.com/jaspervdv/IFC_BuildingEnvExtractor. Accessed 20 May 2023

IFC Georeferencing for OSM



Helga Tauscher, Dominik Heigener, Subhashini Krishnakumar,
Thomas Graichen, Rebecca Schmidt, and Julia Richter

Abstract Digital building models are increasingly available and used as sources to inform geospatial data sets. This poses the requirement to spatially locate building models in the geospatial context. In this paper, we present a case study on transferring georeference during conversion from digital building models in IFC (Industry Foundation Classes) to OpenStreetMap (OSM). First, we provide a condensed overview of how coordinates in IFC's local engineering coordinate systems are related to geospatial reference systems with examples for different constellations. Second, we demonstrate a simple method to enrich IFC datasets with georeferences using existing OpenStreetMap outlines. In the third part we describe two substantially different methods to convert engineering coordinates into geospatial coordinates and show how these methods are implemented in two opensource software packages. Finally, we verify and compare the methods with a set of sample IFC and OSM data.

Keywords IFC · OSM · Georeferences

This article was selected based on the results of a double-blind review of an extended abstract.

H. Tauscher (✉)
HTW Dresden, 01069 Dresden, Germany
e-mail: helga.tauscher@htw-dresden.de

D. Heigener
Bauhaus-Universität Weimar, 99425 Weimar, Germany

S. Krishnakumar
Bundesamt für Kartographie und Geodäsie, 04105 Leipzig, Germany

T. Graichen
Pinpoint GmbH, 09126 Chemnitz, Germany

R. Schmidt · J. Richter
Technische Universität Chemnitz, 09111 Chemnitz, Germany

1 Introduction

Digital building models, mainly in the form of IFC (Industry Foundation Classes) data sets, are increasingly used as sources to complement or inform geospatial data sets, in particular city models. This poses the requirement to spatially locate building models in the geospatial context.

Georeferencing in IFC has also received extended attention during the last years due to the expanding scope of the standard beyond buildings into the area of large-scale infrastructure buildings such as roads and railways. The increased requirements were addressed in the update from ISO 16739:2005 and ISO 16739-1:2018 towards IFC4 (ISO 2018) with additional entities and attributes.¹ Experts from the GIS and surveying area have pointed out the insufficiencies and contributed to the developments. For example Jaud et al. (2020; 2022) provided comprehensive reviews on the necessity and details of improved georeferencing capability in IFC. Markic et al. (2018) demonstrate the deficiencies of early IFC georeferencing capabilities for infrastructure projects with two example projects, a long tunnel and a rail track.

In this paper, we present a case study on transferring georeference during conversion from IFC to OpenStreetMap (OSM). First, we provide a condensed overview of how IFC2x3 and IFC4 relate coordinates in local engineering coordinate systems to geospatial reference systems with examples for different constellations (Sect. 3). Second, we demonstrate a simple method to enrich IFC datasets with georeferences using existing OpenStreetMap outlines (Sect. 4). In the third part we describe two substantially different methods to convert engineering coordinates into geospatial coordinates and show how these methods are implemented in two open source software packages (Sect. 5). Finally, we verify and compare the methods with a set of sample IFC and OSM data (Sect. 6).

As the concepts are transferable to other geospatial formats, the results are interesting beyond OSM. Complementing the existing literature, this paper can serve as a practical guide for the community operating at the BIM-GIS intersection to create their own georeferenced data, to consume existing georeferenced data, and to avoid common pitfalls.

2 Related Work, Sample Data and Software

In this section, we put our study into context by reporting the state of the art and related work, listing the sample data used and the software implementations considered.

¹ We refer to the respective ISO-approved versions 2.3.0.1 (IFC2x3 TC1) and 4.0.2.1 (IFC4 ADD2 TC1) when using the short forms IFC2x3 and IFC4.

2.1 Related Work

Clemen and Görne (2019) have proposed a classification of different levels of georeferencing in IFC. They also provide a checking tool.² A corresponding tool for enrichment (IfcGeoRefUpdater) has been discontinued and is not maintained anymore.

Diakite et al. (2020) try to derive geolocation information by matching building outlines from digital building models against the corresponding building footprints or vertical projections from official land surveying data with OpenStreetMap data as a fallback.

Zhu et al. (2021) try to “verify” geolocation by displaying them in ArcGIS online, Google Earth, FME and their own software stack OCCT-OSA. They use the KIT samples for verification neglecting that two of the buildings are fantasy buildings.

Ohuri et al. (2018) present a case study with georeferences being one facet. They describe the process to add georeferences with Revit 2018 and demonstrate an application to enrich the IFC site WGS84 geospatial position.³ This tool does not handle rotation via true north though.

Some authors convert IFC for large scale display in geospatial context, but do not mention georeference at all, e.g. Chen et al. (2018) discuss conversion to GLTF and Cesium tiles using the Opensource BIMserver (Chen et al. 2018).

The GeoBIM benchmark 2019 led by Francesca Noardo was an investigation of the software landscape around processing digital building models in buildingSMART IFC and 3D city models in OGC CityGML. Among others facets, Noardo et al. report on the georeferencing capabilities of the probed tools, both in reading (interpreting) and writing (creating) georeference information and transforming it between the two formats (Noardo et al. 2020).

The user guide of Mitchell et al. (2020) provides guidelines for a standardized setting up of georeferenced BIM models using the IFC format. Besides general concepts and calculation rules of georeferencing, two main embedding methods are shown for storing georeferencing data in IFC2x3 via property sets and IFC4 via IFC entities (Mitchell et al. 2020).

2.2 Sample Projects

We are using a set of seven different projects and their respective digital building models in IFC (Industry Foundation classes) format, many of which are publicly available. Of all the projects, we use an IFC4 version—either directly as available or converted from an IFC2x3 version. Model A contains a public administration building under construction. Model B (Smiley West) and C (FZK house) stem from the famous KIT dataset. Model D is taken from the well-known Schependomlaan

² <https://github.com/dd-bim/City2BIM/tree/master/IFCGeoRefCheckerGUI>

³ <https://github.com/tudelft3d/IfcLocator>

Table 1 Sample models with specific properties

| | Project | IFC version | Entities | CAD SW | Georef | GR version | Units |
|---|------------------------|-------------|-----------|--------------|--------|------------|-------|
| A | Public administration | IFC4 | 2,218,405 | Revit | o | IFC4 | m |
| B | Smiley west | IFC4 | 110,159 | Archicad | + | IFC2x3 | m |
| C | FZK house | IFC4 | 44,259 | Archicad | – | IFC2x3 | m |
| D | Schependomlaan | IFC2x3 | 714,485 | Archicad | – | IFC2x3 | m |
| E | Gymzaal Amersfoort | IFC4 | 32,183 | Revit | o | IFC4 | mm |
| F | Two-storey residential | IFC4 | 669 | hand-crafted | + | IFC2x3 | cm |
| G | University library | IFC2x3 | 757,413 | Archicad | – | IFC2x3 | m |

dataset, a Dutch semi-detached house. Model E is a Dutch sports hall, published by buildingSMART to demonstrate georeferencing in IFC4. Model F is a two-storey residential building, hand-crafted for testing purposes. Model G finally is a university library building with detailed information about interior equipment, such as book shelves, desks and co-working spaces.

The projects deliberately differ with regard to the IFC version, originating software, georeferencing information, structure and other content of the building models. See Table 1 for the main properties of each model. In column 4, we provide the number of entities as an indication of the size. Column 3 records the IFC and column 7 the georeference version (see Sect. 3 for more detail). In column 6 we noted whether we found no (–), deficient (o) or full (+) georeferencing information. Column 8 holds the length unit used for the geometry.

2.3 Software Implementations

The conversion algorithms described in Sect. 5 can be found in two different open source software applications. Both are implemented in Java and support the conversion of IFC to OSM as one part of a larger scope of functionality. In the course of conducting the case study, we were able to improve the software and consider the extraction of a shared library.

2.3.1 JOSM Indoorhelper

The first one is the Indoorhelper plugin⁴ for the JOSM editor⁵ which includes functionality to import IFC files and convert them to the Simple Indoor Tagging (SIT) Scheme.⁶ The Indoorhelper was developed to exploit additional indoor data sources for OSM and ensure that the imported data is compliant to the standardised indoor modelling scheme.

2.3.2 LevelOut Platform

The second one is LevelOut,⁷ a platform to convert building models in IFC format into various formats for map and navigation services, OSM (SIT) among others. The platform is based on the Opensource BIMserver and the conversion implemented as a BIMserver serializer plugin. The conversion is a two-step process, where an intermediate model holds all information extracted from the IFC and necessary to supply all target formats with the required input, including georeferencing information (Krishnakumar and Tauscher 2023).

3 Georeferencing Information in IFC

3.1 IFC Schema for Georeferencing Information

Georeferencing information can be represented in IFC with different depths and detail. The documentation of IFC4 (ISO 2018) contains *fundamental concepts and assumptions* about the project context. Thus, all project-related data sets have an instance of *IfcProject* that provides information about the overall context and a directory with including objects. The context definition contains the concept of global positioning of the project coordinate system for localization on the earth's surface. For this purpose, information is usually indicated by values for the easting, northing, elevation, geodetic datum, vertical datum, and orientation. This part of the context definition shows how a spatial coordinate reference from recognized coordinate reference systems can be added to the context of the geometric representation. Thereby, the geometric representation context of the shape representation builds an additional context definition. For this reason, information will be specified about the coordinate system orientation, true north direction, precision, as well as further information that applies to all project-related geometries. This is realized with a main geometric

⁴ JOSM Indoorhelper plugin: <https://wiki.openstreetmap.org/wiki/JOSM/Plugins/indoorhelper>.

⁵ JOSM, an extensible editor for OSM <https://josm.openstreetmap.de/>.

⁶ Simple Indoor Tagging is a data model to represent indoor building data directly in OSM https://wiki.openstreetmap.org/wiki/Simple_Indoor_Tagging.

⁷ LevelOut project: <https://bauinformatik.github.io/levelout/>.

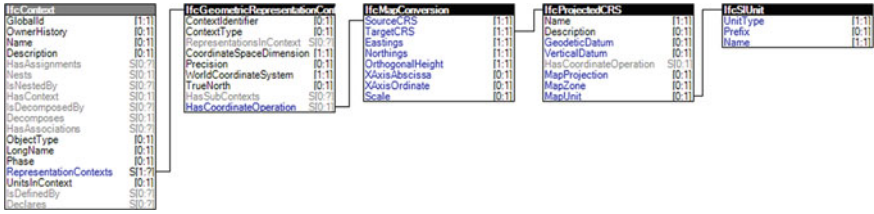


Fig. 1 Required IFC entities and attributes (blue) for “IFC4 style”, modified image

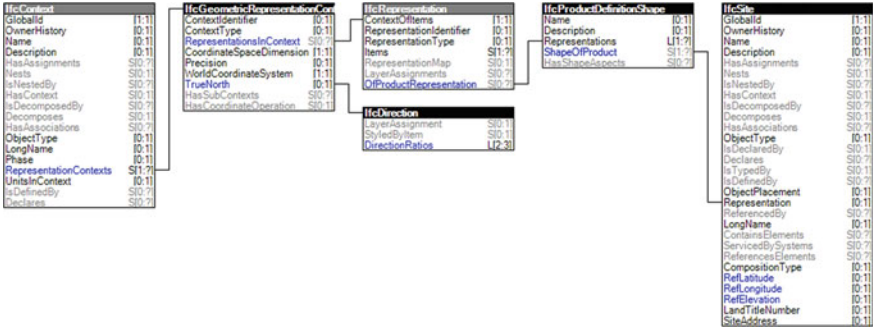


Fig. 2 Required IFC entities and attributes (blue) for “IFC2x3 style”, modified image

representation context for 3D models and 2D representations that can be extended by geometric representation sub contexts.

Based on the mentioned concepts, we are distinguishing two configurations, the improved version from IFC4 and the outdated version from IFC2x3 as a fallback for legacy data. Following the attempt made by Clemen and Görne (2019) to classify geoinformation in IFC into so-called Levels of Georeferencing (LoGeoRef), these correspond to LoGeoRef50 or a combination of LoGeoRef40 and LoGeoRef20. In the following, we call these the “IFC4 style” and the “IFC2x3 style”. IFC4 accommodates both styles, and IFC2x3 only the last style. IFC2x3 style georeferences always pertain to the geodetic system WGS84, while IFC 4 style georeferences can pertain to any coordinate reference system (CRS), including WGS84.

About the templates of the fundamental concepts and assumptions from the IFC4 specification, the following figures represent an extraction of IFC entities and attributes for each style. The figures follow the notation of instance diagrams as used within the IFC4 documentation for the sake of clarity. In addition, the required IFC attributes are highlighted in blue font of the respective IFC entity. Figure 1 shows the extraction for the “IFC4 style”, while Fig. 2 shows the extraction for the “IFC2x3 style”.

3.2 Georeferencing Examples in SPF

In the following, we show examples and discuss the encountered population of the georeferencing concepts as occurring in the sample data of Sect. 2.2.

Listing 1.1 shows the georeferencing information for project E, Gymzaal Amersfoort as an example of IFC4 style georeferences. This sample had been published by buildingSMART to demonstrate IFC4 georeferencing capabilities and document CAD export procedures. The location was specified roughly and exported with Autodesk Revit. We show the corrected and enhanced version, for details on correction and enhancement see Sect. 4.

Listing 1.1 Georeferencing within the Gymzaal SPF file

```
#120= IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,0.01,#117,#118);
#130= IFCSIUNIT(*,.LENGTHUNIT.,$,.METRE.);
#133= IFCPROJECTEDCRS('EPSG:28992','Netherlands, Amersfoort datum,
New System','AMERFORT-EP',$,$,$,#130);
#135= IFCMAPCONVERSION(#120,#133,149626.31591058132, 413717.68421662884,
0.,1.,6.12303176911189E-17,$);
#137= IFCPROJECT('2_ZvelcG1tR1bYc4wxe3h',#38,'',$,$,'','',(#120),#115);
```

Similar to this, project A, public administration, contains IFC4 style information. The proper export from CAD is described in Sect. 3.3.

Below listing 1.2 shows the georeferencing information for project B, Smiley West from the well-known KIT sample data set, as an example for IFC2x3 style georeferences. This is the only sample in the list which contained proper georeferences out of the box.

Listing 1.2 Georeferencing within the Smiley West SPF file

```
#66= IFCPROJECT('34407vICcwH8qAEnwJDjSU',#12,'Smiley West',$,$,
,$,$,#62,#9079),#49);
#9077= IFCDIRECTION((-0.537299608347,-0.843391445813));
#9079= IFCGEOMETRICREPRESENTATIONCONTEXT($,'Plan',3, 1.000000000000E-5,#9076,
#9077);
#9080= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Box','Plan',*,*,*, #9079,$,
.PLAN_VIEW.,$);
#9085= IFCSHAPE REPRESENTATION(#9080,'Box','BoundingBox', (#9084));
#9088= IFCPRODUCTDEFINITIONSHAPE($,$,#9060,#9085);
#9094= IFCSITE('20FpTZCqJy2vhVJYtjuIce',#12, 'Gel\X2\00E4\X0\nde',$,$,#84,
#9088,$,.ELEMENT., (49,1,59,680200),(8,23,27,528000),110.,$,$);
```

Similar to this, projects C, D, F and G contain IFC2x3 style information, even though some of them are actually IFC4 files. For projects D and G we found the georeference pertaining to a default location in the respective region, e.g. Schependomlaan (D) in the Netherlands, university library (G) in central Berlin. For project abc, georeference information was missing altogether. A method to enrich and correct the georeferencing information is shown in Sect. 4.

3.3 Export from CAD

For new models, the preferred method of georeferencing in IFC models is provided by the “IFC4 style” with the highest quality in terms of georeferencing information

available within an SPF file. Support for “IFC4 style” georeferences has only been added recently to CAD software because the entities and attributes necessary for this style were unavailable for previous IFC specifications, according to the implementation in software and certification processes.

Besides different BIM-enabled CAD software of the AEC (Architecture, Engineering, and Construction) industry, Autodesk Revit is a known representative for planning processes. Revit provides the SPF files export using an IFC exporter in a standard version or an externally installable open-source version. If the IFC export settings remain unconsidered, the “IFC2x3 style” is supported by default in case of available Revit data for the respective instances, leaving further data assignment settings unaffected. Furthermore, Revit’s IFC exporter supports the “IFC4 style” with specific geographic reference settings. For this purpose, an appropriate EPSG (European Petroleum Survey Group Geodesy) code and the coordinate base must be indicated. The EPSG code is used for accessing information about the coordinate reference system. The coordinate base is defined by using a Revit-specific coordinate base point, such as the project base point.

For example, project A, public administration, has been created using Autodesk Revit. The predefined *MVD IFC4 Reference View [Architecture]* has been used for generating an SPF file. The EPSG code 25833 has been specified because the project base point contains UTM coordinates that are located in a specific UTM zone.

4 IFC Enrichment

Where the information necessary for georeferencing (as described in Sect. 3) is not available, we enrich the IFC data, either by re-exporting with adjusted CAD settings (Sect. 3.3) or preferably with direct editing of the IFC-SPF (STEP physical file, a human-readable text file). In this section, following the elaboration of georeferencing concepts in IFC2x3 or IFC4 style, we describe a method to obtain the parameters for the population of the concept’s attributes from OSM building outlines. This way, the engineering CRS is positioned in WGS84, which can be written in either of the styles. If professional surveyor’s or cadastral reference information is available pertaining to other reference systems, then this should be used with priority and must be written in IFC4 style. Often though, this is not available, particularly for existing data stocks that should be retroactively integrated at the geospatial scale for management, navigation, analysis or other ubiquitous purposes.

This section provides a reusable method to enrich IFC with georeferencing information. We also contribute enriched data to the public domain.

4.1 Existing Georeferencing Information and Actual Location

First, we reviewed the existing georeferencing information in the files and gathered information about the actual location of the buildings.

Project A, the public administration, was enriched with surveyor's information in IFC4 style.

Project B, Smiley West, has a correct IFC2x3 style georeference, corresponding to the address contained in the model. However, the IFC 4 version as published by KIT⁸ is not valid against IFC4 ADD2 TC1. The file, exported from ArchiCAD20, contains numerous entities of type `IfcMaterialProperties` which are invalid and superfluous. Although unrelated to georeferencing we published the fixed file to make it available for further uses.

Project C is a fantasy building with a location on a free spot of the KIT (Karlsruhe Institute of Technology) campus, which seems plausible in general, but a bit random and not fully intentional in the details, in particular regarding the rotation/true north which seems to differ from the surrounding buildings.

Project D, the pertinent Schependomlaan project,⁹ originally contained an IFC2x3 style geolocation at N 52° 9' E 5°23' which is the city center of Amersfoort rounded to the minutes. This seems to be a common method for CAD software to populate the geolocation attributes during IFC export if only a city is given as the rough project location. Given that the architectural office is located in Amersfoort (according to IFC data), the default project location is likely set to the city of Amersfoort in the CAD software. The actual project location is only given as the city of Nijmegen and to find the respective address we employed Google street view to locate the iconic building along the street called Schependomlaan.

Project E is a sports hall in Amersfoort, Hambakendreef 2A, Netherlands, provided to demonstrate new georeferencing features in IFC4 and test and document a process to populate of the new attributes from Autodesk Revit.¹⁰ The IFC was exported from Revit 2020 with standard RV MVD settings. In Revit, geolocation in WGS84 was initially determined from the given address and manual positioning on a large scale map, hence with a very rough location. The Revit IFC exporter then converted the WGS84 location to the Amersfoort CRS-EPSG 28992, resulting in a location of the WCS origin (eastings, northings = 149,692, 413,790) situated on a green space north of the building. However, the project base point is at the south most point of the building almost 100m away from the determined location. For a corrected version, the WGS84 coordinates of the south-west point of the OpenStreetMap building polygon have been transformed to Amersfoort CRS with EPSG

⁸ Institute for Automation and Applied Informatics (IAI)/Karlsruhe Institute of Technology (KIT), https://www.ifcwiki.org/index.php?title=KIT_IFC_Examples.

⁹ Schependomlaan data set: <https://github.com/buildingSMART/Sample-Test-Files/tree/master/IFC%202x3/Schependomlaan>.

¹⁰ BuildingSMART IFC4 georeferencing demo : <https://github.com/buildingSMART/Sample-Test-Files/tree/master/IFC%204.0/Example%20project%20location>.

4833 reversed. Further, issues with units from the original IFC Exporter version have been corrected manually.

Project F is a hand-crafted sample project, a fictitious building and was enriched with a georeference to a random building plot in Weimar.

Project G has an original georeference to N 52°, 31' E 13°, 24', which is a location in central Berlin on Museumsinsel, a touristic area. This seems to be a similar effect as observed in project D, just with a German default location, not a Dutch one.

4.2 Georeferencing Method

With this method, we generate IFC2x3 style georeferences.

First, we find the building polygon in OSM. Then, we identify two prominent outline points on a line that is horizontal in the engineering CS. We locate the respective points (hence geospatial coordinates) on the OSM outline polygon and can compute the angle.

In addition we ideally need an OSM node in the place of the Cartesian coordinate system origin, but this will rarely be the case, since the CS origin is usually on an axis intersection in the CAD plan. Axis, running in the interior of walls will naturally not lie on the building outline and thus constitute no node in common OSM data. We can locate the Cartesian CS origin by looking at the file in a CAD software or IFC viewer, there is usually some sort of axis cross that can be toggled visible. Using a building outline point P close by the origin, its Cartesian coordinates specify its relative location with respect to the origin. Hence, we can use these coordinates to calculate the angle and distance to the origin and apply these (taking the rotation from the first step into account in addition) to the geospatial coordinates of the OSM node corresponding to the chosen point P.

Finally, we can convert rotation and location to the required form for writing it to the IFC attributes.

Figure 3 shows the Schependomlaan (project D) building outline polygon in OSM. We chose two points on the northern edge for calculating the rotation: Pt 1 51.8419207, 5.8359716; Pt 2 51.8419708, 5.8361939.

The resulting bearing is 69.95911558532617 deg resulting in a normalized vector with abscissa 8.744136117400103E-7 and ordinate 2.3971081424637476E-6. The OSM node of the SW corner of the building outline is located at 51.8417616, 5.8361467. From the IFC viewer we know its Cartesian coordinates 5050, 750 (mm) in the Engineering CS. From there, the origin (project base point) can be calculated to 51.84173969993845, 5.836081382150285 and in IFC representation, as a tuple of degrees, minutes, seconds, fraction of seconds we get (51, 50, 30, 262,919), (5, 50, 9, 892,975).

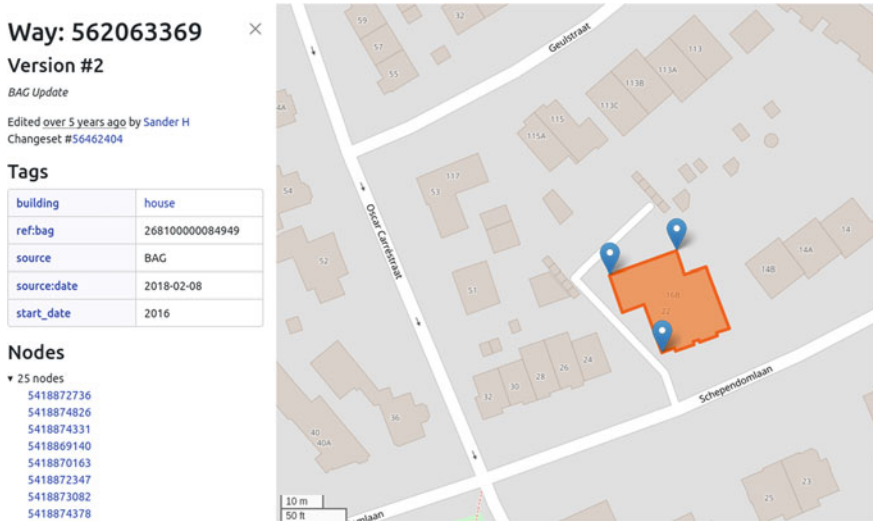


Fig. 3 OSM outline of Schependomlaan building with OSM nodes used for georeferencing: SW corner, two points on northern edge of the building outline

4.3 Discussion

The method applies similarly to IFC4 style references with the exception that coordinates have to be transformed to a suitable mapped CRS. This has been done for Gymzaal Amersfoort, project E, which had only rough location when originally published. At the time of original publication, the export from Revit was not correct in terms of units. Thus, besides the exact location, units for the projected CRS were also fixed.

When using this method, it should be cared for validating the files and updating author information when changing IFC data sets. CAD vendors voice legitimate complaints about manipulated invalid files attributed to their certified software exports.

Also, one could discuss the proper place to put georeference: It is commonly assumed that georeferencing information should be contained with the IFC file. Many authors pose this as obvious or unavoidable. Certainly this is true when the geoinformation is provided as part of the original authoring. For after-the-fact enrichment this might be disputed, as the georeference might be specific to the target application, e.g. GIS, mapping or navigation application, and authority over the “correct” location is undetermined.

5 Coordinate Conversion

Given IFC data with more or less precise georeference information, the detailed building information can be leveraged in geospatial contexts. At some point, the geometry given in Cartesian coordinates with a georeferenced engineering CRS has to be transformed to geospatial coordinates. Depending on whether applications operating in the geospatial context can only handle geospatial CRS and coordinates or also georeferenced engineering CRS, this will happen in a preparatory data-transformation step, e.g. export or conversion into a geospatial format, or in the geospatial application itself upon import of the engineering data. In this section we discuss this conversion step and demonstrate it by example of creating OSM data from IFC.

5.1 Conversion Algorithms

With a georeference, the engineering coordinate system is anchored at a point on a well-defined model of the earth's surface and oriented with respect to that geospatial model (datum). While this does provide reliable input for any conversion, it does not answer the more fundamental question of how to interpret the Cartesian coordinates with respect to a target CRS which may differ from the reference CRS. The most obvious interpretation from an engineer's point of view is certainly a topocentric coordinate system with horizontal surfaces in the engineering CS corresponding to planes parallel to a reference ellipsoid's tangent plane in the CS origin and the y-axis pointing north (Sect. 5.2). A second interpretation is with the horizontal surfaces corresponding to ellipsoidal surfaces.¹¹ And finally, we can also treat the coordinates as small numbers in a projected coordinate system, assuming that engineers plan their constructions on a map. Each of the three approaches yields a different conversion algorithm which will be discussed in the following sections. Further, we will demonstrate the three approaches by looking at two open source software implementations for IFC-to-OSM conversion as listed in Sect. 2.3.

¹¹ Interestingly, this could be seen as an appropriate model for construction processes. As these processes follow the laws of gravity, concrete floats curve like the ocean in the large scale and buildings are erected vertically using plumbs. However, in the small scale the simplicity of Euclidean geometry wins over the precision of spherical geometry.

5.2 Topocentric Cartesian CS

The topocentric coordinates can be converted to geodetic via geocentric coordinates as described in the EPSG method 9837.¹² Even though it is the most obvious interpretation, it is not yet implemented in any of the software.

5.3 Ellipsoidal Surfaces

Knowing the local distance d and bearing b from Cartesian point (x, y) to Cartesian origin (x_0, y_0) the point can be converted to geodetic format by mapping d and b to the ellipsoidal surface. For this, d is converted into meters and b into degrees defined clockwise from y-axis. Then the searched latitude and longitude values (ϕ, λ) can be determined by calculating a new point at given distance and bearing to the known geodetic origin (ϕ_0, λ_0) as follows:

$$\phi = \arcsin\left(\sin \phi_0 * \cos \frac{d}{R} + \cos \phi_0 * \sin \frac{d}{R} * \cos b\right)$$

$$\lambda = \lambda_0 + \text{atan2}\left(\sin b * \sin \frac{d}{R} * \cos \phi_0, \cos \frac{d}{R} - \sin \phi_0 * \sin \phi\right)$$

with

$$b = 90^\circ * -\text{atan2}(y - y_0, x - x_0)$$

$$d = \sqrt{(y - y_0)^2 + (x - x_0)^2}$$

and R being the earth's radius. This approach is implemented in the JOSM Indoorhelper in the [ParserGeoMath.cartesianToGeodetic method](#). Although this procedure can be applied regardless of how the CRS origin is specified, via IFC2x3 or IFC4 style georeferences, the Indoorhelper is currently only able to extract and process IFC2x3 style georeferences.

5.4 Projected CRS

When interpreting the cartesian coordinate system as being based on a cartographic map, hence as a projected CRS, we can easily resort to the various existing conversion

¹² For a depiction of the topocentric coordinate system refer to the description of the first part of the conversion: <https://proj.org/en/9.2/operations/conversions/topocentric.html>.



Fig. 4 Calculating rotation from true north when converting via projected CRS

libraries such as proj4j which handle conversion between CRS' and can even resolve them with their parameters from the EPSG code. However, the correct calculation of the rotation appears errorprone in implementations. It is challenging because it has to take into account the different angle's reference axis and orientations as well as the true north directions as shown in Fig. 4.

This approach is implemented in the LevelOut platform in the [ProjectedOriginCRS.cartesianToGeodetic method](#) and in the [GeodeticOriginCRS.cartesianToGeodetic method](#). These methods are applied depending on how the CRS origin is specified, via IFC2x3 or IFC4 style georeferences.

6 Verification with OSM Data











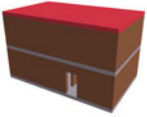



Finally, we verified the algorithms with those sample models which have an actual georeference either originally contained or received through enhancement as described in Sect. 4. Table 2 shows the resulting OSM data in geospatial context. Project C is left out because of its fantasy nature. The verification at this stage was carried out by visual inspection using JOSM and Indoor=.¹³ A more thorough numerical validation measuring the deviation between the resulting OSM geometries and the original OSM polygons could yield a more substantive evaluation.

7 Conclusion

This use case study provides insight with practical examples in the population of georeferencing information in IFC as well as its interpretation and usage for conversion of geometry in Cartesian into WGS84 coordinates. Both the population and interpretation being based on freely available OSM data and open source implemen-

¹³ Indoor= <https://indoorequal.org>.

Table 2 Sample models with location in geo-context

| IFC | LevelOut platform | JSOM Indoorhelper |
|-----|---|--|
| A |  |  IFC4-style georef not supported |
| B |  |   |
| D |  |   |
| E |  |  IFC4-style georef not supported |
| F |  |  geometry type not supported |
| G |  no spaces and spaceboundaries |  |

tation, this provides low-barrier guidance for spatially integrating BIM and GIS data as a base for further studies on applications using the integrated data.

There are various limitations and gaps in the current study. First of all there is no implementation of the topocentric coordinate system.

Second we did not do any numerical validation yet. The resulting fitting and deviation of the building outlines in IFC and OSM could be compared and evaluated.

Third, the level of detail and topology might vary between the OSM outline and the IFC outline such that it is not possible to identify sufficiently corresponding points. The variance might be to a degree that choice of corresponding points outweighs the choice of conversion algorithm.

Finally, geodetic height deserves attention as well in the future.

Acknowledgements The authors would like to acknowledge the financial support of the German Federal Ministry of Digital and Transport (BMDV) through grant 19F1095A. We would like to thank the Chemnitz University of Technology for providing building reference data and contributions to the OpenStreetMap project, such as IndoorHelper. The outcome of this paper has also been supported by the City of Dresden by providing the demonstration object *Public Administration*, which is gratefully acknowledged. We also like to thank the contributors of the buildingSMART sample files, Schependomlaan Nijmegen and Gymzaal Amersfoort.

References

- buildingSMART International: IFC 4.0.2.1 Documentation (2017). https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/
- Chen Y, Shooraj E, Rajabifard A, Sabri S (2018) From IFC to 3D tiles: an integrated open-source solution for visualising BIMs on cesium. *ISPRS Int J Geo-Inf* 7(10). <https://doi.org/10.3390/ijgi7100393>
- Clemen C, Görne H (2019) Level of georeferencing (LoGeoRef) using IFC for BIM. *J Geodesy, Cartography Cadastre* 2019(10):15–20. https://jgcc.geoprevi.ro/docs/2019/10/jgcc_2019_no10_3.pdf
- Diakite AA, Zlatanova S (2020) Automatic geo-referencing of BIM in GIS environments using building footprints. *Comput, Environ Urban Syst* 80:101,453. <https://doi.org/10.1016/j.compenurbysys.2019.101453>
- ISO 16739:2005 (2005) Industry foundation classes, release 2x, platform specification (IFC2x platform). Technol Rep 16739. International Organization for Standardization, Geneva, Switzerland. IFC 2.3.0.1 (IFC2x3 TC1)
- ISO 16739-1:2018 (2018) Industry foundation classes (IFC) for data sharing in the construction and facility management industries. Part 1: data schema. Technol Rep 16739–1. International Organization for Standardization, Geneva, Switzerland. IFC 4.0.2.1 (IFC4 ADD2 TC1)
- Jaud Š, Clemen C, Muhič S, Borrmann A (2022) Georeferencing in IFC: meeting the requirements of infrastructure and building industries. *ISPRS Ann Photogrammetry, Remote Sens Spatial Inf Sci X-4/W2-2022*:145–152. <https://doi.org/10.5194/isprs-annals-X-4-W2-2022-145-2022>
- Jaud Š, Donaubaauer A, Heuncke O, Borrmann A (2020) Georeferencing in the context of building information modelling. *Autom Constr* 118:103,211. <https://doi.org/10.1016/j.autcon.2020.103211>
- Krishnakumar S, Tauscher H (2023) Floor plan extraction from digital building models. In: Proceedings of FOSSGIS (academic track) 2023. Berlin, Germany, pp 146–152. https://files.fossgis.de/Konferenz/fossgis_tagungsband_2023_digital.pdf#page=148
- Markič Š, Donaubaauer A, Borrmann A (2018) Enabling geodetic coordinate reference systems in building information modeling for infrastructure. In: Proceeding of the 17th international conference on computing in civil and building engineering (ICCCBE). Tampere, Finland. <http://programme.exordo.com/icccbe2018/delegates/presentation/373>
- Mitchell J, Bennett D, Brown N, Gregory L, Karlshøj J, Nisbet N, Parslow P (2020) User guide for geo-referencing in IFC. <https://www.buildingsmart.org/wp-content/uploads/2020/02/User-Guide-for-Geo-referencing-in-IFC-v2.0.pdf>
- Noardo F, Harrie L, Arroyo Ohoiri K, Biljecki F, Ellul C, Krijnen T, Eriksson H, Guler D, Hintz D, Jadidi MA, Pla M, Sanchez S, Soini VP, Stouffs R, Tekavec J, Stoter J (2020) Tools for BIM-GIS integration (IFC georeferencing and conversions): results from the GeoBIM Benchmark 2019. *ISPRS Int J Geo-Inf* 9(9):502

- Ohuri KA, Diakit  A, Krijnen T, Ledoux H, Stoter J (2018) Processing BIM and GIS models in practice: Experiences and recommendations from a GeoBIM project in The Netherlands. ISPRS Int J Geo-Inf 7(8). <https://doi.org/10.3390/ijgi7080311>
- Zhu J, Wu P (2021) A common approach to geo-referencing building models in industry foundation classes for BIM/GIS integration. ISPRS Int J Geo-Inf 10(6). <https://doi.org/10.3390/ijgi10060362>

Merging BIM, Land Use and 2D Cadastral Maps into a Digital Twin Fit—For—Purpose Geospatial Infrastructure



Dimitra Andritsou, Sofia Soile, and Chryssy Potsiou

Abstract Digital Twin (DT) technology is the tool for monitoring, management and intervening in a timely manner to prevent disasters in urban areas, transforming them into smart cities. By enabling real-time data flow through utilizing a vast network of interconnected sensors and smart devices the functioning of buildings, facilities and utility networks can be monitored, managed and optimized to achieve a more stabilized, fair and sustainable urban environment. Good land management and optimal exploitation can also be achieved. The paper presents an innovative crowdsourced methodology for a fast, low cost and reliable structuring of the necessary geospatial infrastructure for a digital twin of an urban neighborhood by merging and visualizing individual BIMs with available open data, such as open orthophotos and cadastral maps, planning and building regulations, as well as existing land uses, smart devices and metric data derived from the Google Earth Pro and Street View platforms. The above-mentioned data if transferred into an open-source platform, such as Tandem, are ideal for tracking, managing and improving the residents' living, from an economic, hygienic, safety and ecological point of view. Data derived from smart devices is a prerequisite for predicting and preventing various problems and setbacks.

Keywords BIM · Digital twin · Smart devices · Crowdsourcing · Fit-for-purpose geospatial infrastructure · Open data

This article was selected based on the results of a double-blind review of the full paper.

D. Andritsou (✉) · S. Soile · C. Potsiou
School of Rural and Surveying Engineering, National Technical University of Athens, 9 Iroon Polytechniou Street, Zographos 157 80, Greece
e-mail: andimitra@hotmail.gr; texasenterprisesmsg@hotmail.com

S. Soile
e-mail: ssoile@survey.ntua.gr

C. Potsiou
e-mail: chryssyp@survey.ntua.gr

1 Objective of the Research

This paper presents the first stage of a research that aims to propose a methodology for structuring a Digital Twin (DT) of an urban neighborhood of Greece, where old and new constructions may exist, to be used for urban management purposes using crowdsourcing techniques. For the new constructions it is assumed that BIMs will be available and will be easily integrated into the DT of the neighborhood, while a lack of available data may hinder the creation of BIMs for the old constructions.

Therefore, the research first investigates the options to utilize geospatial information derived from open data and any other sources.

In the case that architectural plans regarding the inner structure of old buildings are not available, Building Information Models (BIMs) of only the outer shell of the existing old constructions are created. For that, the following data are used: available orthophotos and 2D cadastral maps derived from the cadastral agency and approximate measurements derived from Google Earth Pro and StreetView, as well as field inspections. Once such individual BIMs are completed, 3D topographic surfaces that represent the surrounding area of the city blocks and the road network are made; next, they are enriched with (a) land-use information (e.g., residential or commercial use on the ground level and the floors of each building) and ownership status (e.g., private/public), (b) urban and construction regulations, height restrictions and 2D cadastral maps, (c) information about existing smart devices on the façades (e.g., alarm systems, air-conditioning units, electric network control panels, lighting, ventilation shafts, intercoms, surveillance cameras, heating and energy providing units, solar panels, etc.), and (d) any additional street furniture information about the surrounding area (e.g., greenery, traffic signs, lighting poles, parking areas, bicycle railings, trash bins and benches, etc.).

Next, all information is transferred into the opensource platform of Tandem where all qualitative information will be classified, visualized, edited, updated and managed. Available BIMs of the newly built constructions and relevant information about these buildings may also be integrated into the platform. The information of the existing old constructions on the platform may be sporadically updated with the inner structure of each building as soon as such information will become available. Especially for that, such information about the inner architecture and floor plans of the old buildings is currently collected from the electronic building identity program in Greece, following each new transaction.

Motivations can be given in order to raise spirits for switching over to smart devices, in and outside of a building. To enable the creation of a DT of the urban neighborhoods to flourish, additional land policies, tools and incentives will be provided for a large-scale installment of smart devices into the buildings, such as the current campaign for replacement of old electrical devices with smart and ecological friendly ones (e.g., smart air conditioners, solar systems, refrigerators, etc.)

In the following chapters, a simple, fast, crowdsourced and low-cost methodology for the data collection and structuring of the above-described solution is proposed

and tested. The proposal may also be applicable to other countries after proper modification and adjustment in order to utilize open-sourced portals and free-to-use data in each country.

2 Related Works

The BIM and DT concepts have been recognized as semantically correlated, with the latter being the evolutionary form of the first (Douglas et al. 2021). There is a plethora of definitions describing the DT concept and function. The DT can be described as a virtual reproduction of a physical asset, built in the real world, in need of regular update to match the present state of the real object (Brilakis et al. 2019). As presented in various studies, the two concepts are extremely valuable in reforming not only the design but also the construction stage (Alonso et al. 2019; Lu et al. 2019; Madni et al. 2019; Redmond et al. 2012), while enhancing collaboration between different stakeholders (Madni et al. 2019).

The most prominent difference between the two, is that a DT entails a virtual model of its real prototype which gets updated in real time (Khajavi et al. 2019). The BIM standard does not support a bidirectional communication of data and changes between the virtual and the physical environment (Qi et al. 2018). Machine learning and automated procedures are an exclusivity of the DT technology (Khajavi et al. 2019), while offering a dynamic representation of built and environmental assets (Eyre and Freeman 2018). This results in receiving immediate update on the status of various elements and thus structuring a realistic maintenance plan (Khajavi et al. 2019). A DT is extremely valuable in the construction sector as it can provide a full virtual simulation analysis of the said project considering all the elements, parameters and conditions on site (Eyre and Freeman 2018; Haag and Anderl 2018). Thus it can pitch in organizing a predictive maintenance system (Khajavi et al. 2019). A DT of a building contributes in improving its utilization, operation and performance (Eyre and Freeman 2018). A DT of a neighborhood is a system of DT systems.

DTs contribute in monitoring the constant function and state of a physical element or system, utilizing real—time data acquired by sensors. The sensors are interconnected through a virtual and communicational route with a computer generated database. The user can edit, manage, sort, categorize and dispose the incoming data according to the needs of each project. The sensor generated data can be used to create virtual simulations of possible situations or outcomes, predict future implications, plan scenarios and alternate handlings to different problems and support decision making procedures for sustainable environmental, cadastral legal, structural and many other reasons. As it is already practiced in a lot of studies, the DT model can be constantly updated with new data through sensors, IoT devices (Khajavi et al. 2019) and augmented reality procedures (Eyre and Freeman 2018). The new trend and vision of the entire community, is the utilization of BIM and DT practices in order to continuously update the current urban representation status.

In literature, smart cities are described in many ways and through different scopes. Commonly they are characterized as a sustainable urban hub where all facilities, utilities and networks are interconnected and continually monitored. This is achieved through the exploitation of IoT, Industry 4.0, Augmented Reality (AR), Virtual Reality (VR), Artificial Intelligence (AI), Big Data, etc.

Digital technologies and virtual applications can lead to smart, sustainable and democratic urban planning and management (Dembski et al. 2020). A wide variety of scenarios can be designed and virtually played out utilizing DT technology. These scenarios may concern objects of different uses and sizes, from the city's entire hydrologic network to a room's lighting pulp. Nevertheless, the center of a smart city's function should be its residents (Dembski et al. 2020).

An Urban Digital Twin, for the German city of Herrenberg, has been designed to enable a collaborative and optimal urban planning (Dembski et al. 2020). A didactical concept, regarding the Digital Twin as the final goal in a project-based learning procedure has also been made (Wahbeh et al. 2020).

Cadastral and land administration sectors are in dire need of creating and including BIMs for land management and property registration purposes. An application merging the conceptual registry of a Land Administration Domain Model (LADM) with the 3D visualization possibilities of the BIM/IFC standards to present volumetric Rights Restrictions and Responsibilities (RRRs), all in an open platform, has also been developed (Andritsou et al. 2022).

BIM projects, that are already enriched with DT elements, have been interconnected with sensors placed in the real world and are constantly renewing their functionality status presenting an almost real-time simulation of their prototypes and their broader environment (Evangelou et al. 2022).

BIM utilization and IFC schema extensions entail a solution in tackling land management problems and coordinating information of properties, in high-rise buildings (Atazadeh et al. 2017). The combination of DT and CyberGIS technologies can measure the impact of infrastructure development in a smart city environment (Shirowzhan et al. 2020).

Central services of an urban center have been equipped with sensors and IoT devices in order to construct AI algorithms for an optimal management of the various operations in the city under study (Wagner et al. 2019).

Applications of DTs contribute in reducing traffic volume and gas emissions providing a more sustainable urban environment (Fuller et al. 2020).

The weather conditions, the behaviors of the workers, the state of the machinery and the equipment and the procedures of managing the project can all be virtually modelled providing a real-time chart of every micro change and intervention on the site (Afzini 2017).

Neighborhood management in terms of energy consumption, pollution reduction, security and safety controls, traffic management, garbage collection management, spatial planning monitoring, service provision, maintenance and management of green spaces, etc., can all be more efficiently achieved through bidirectional communication facilitated by DT technology.

3 Methodology

The steps of the proposed crowdsourced methodology are:

The office work refers to the (a) investigation of the available open data to be used for the neighborhood, (b) determination of the roles and responsibilities among professionals, team leaders and volunteers, the call for volunteers and the determination of their required skills, selection of the best trained volunteers to become team leaders, etc., (c) classification of the type of information to be collected for the particular neighborhood, and collection of information from the office, (d) structure of the 3D model of the neighborhood (construction of the building BIMs, 3D urban plot, 3D road network, (e) editing and corrections, insertion of additional elements on the building facades and the surrounding area, and (f) uploading of all information in the Tandem platform.

Tandem is a cloud—based platform that structures an accurate, reliable and BIM—based virtual twin through automated workflows (AUTODESK (2023). <https://intandem.autodesk.com/>). Tandem features a collaborative interface that helps all stakeholders and participants to be in tune with the project and its final outcome from the very beginning (AUTODESK (2023). <https://intandem.autodesk.com/>). Each project can be accessed through the cloud, any time, enabling the constant editing and management of each model. Inside the Tandem program all the various elements, components, spaces and systems of each project are all part of a homogenous and interconnected virtual environment. This platform provides a holistic and interactive presentation of the building model under study and its twin. The users can add, edit, manage, categorize, isolate, query and hide any element that suits each phase of the project.

The volunteers may be students or young people with technical skills that may have no prior experience in handling open data to create 3D building models or may have not utilized the Revit program before. The training of the volunteers is made by the professional experts and includes the data collection procedure and the use of Revit. Following the training, the best of the volunteers may become team leaders and may train other volunteers as well as they may supervise the work of all volunteers.

Following the training the collection of data phase starts by the volunteers under the supervision of the team leaders. The national cadastral orthoimages are used as the basemap to delineate the 2D building footprints on them.

Both Google Earth Pro and Streetview platforms are user friendly and provide qualitative and quantitative data of fairly good accuracy to create a fit-for-purpose digital twin of the neighborhood to be used for management purposes. Both platforms are used for obtaining:

- (1) Approximate elevation data, regarding the ground surface (see Fig. 1) and the building heights. Additional data about the building façades and the surrounding area are also obtained.
- (2) Approximate horizontal and vertical measurements are made on the facades of the buildings about all architectural elements (number of floors, windows,



Fig. 1 Approximate elevation measurement (159 m) on the ground surface, in Google Earth Pro Desktop Application

doors, etc.), derived from Google Earth Pro, to create LOD₃ information (see Fig. 2).

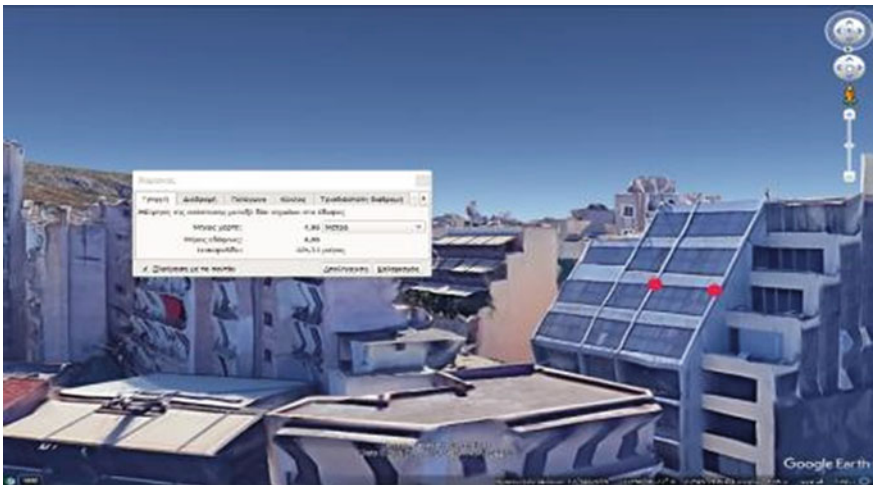


Fig. 2 Approximate measurement of a window’s height, in the Google Earth Pro



Fig. 3 In the Streetview platform, a building’s number of floors as well as the textures and materials of its various architectural elements are visible

- (3) Information about the textures and materials is also collected (see Fig. 3).
- (4) Information about the location and type of elements such as alarms systems, air-conditioning units, electric network control panels, lighting, ventilation shafts, intercoms, surveillance cameras, heating and energy providing units, solar panels, etc., on the facades and the roofs, as well as on the ground such as greenery, traffic signs, lighting poles, street furniture, parking areas, bicycle railings, trash bins and benches, etc., are collected (see Fig. 4).

Such elements are classified into the following categories as following:

- MEP (Mechanical, Electrical and Plumbing) category: Alarms for protection and prevention purposes, surveillance equipment, various control panels, intercom

Fig. 4 Examples of Digital Twin elements, in Streetview (Orange: MEP/Blue: HVAC/ Red: Urban elements)



and communication devices, various types of vents and lighting are part of this category.

- HVAC (Heating, Ventilation and Air Conditioning) category: Air conditioning and ventilation systems, solar panels, heating and energy providing units, are noted in this category.
- Street furniture category: public lighting poles, trash bins, traffic signs, benches and bicycle railings.

The aforementioned elements prepare for the transformation of the 3D fit-for-purpose representation of the urban environment to become a DT of the area once a feedback information from these devices will be enabled.

- (5) Information about existing land uses (such as the type: housing, shops, services) and additional information about the type of it. The classification of the detected land uses is made in accordance with the national cadastral regulatory frame.

Three distinct land use categories can be noted in the neighborhood's area:

- Services: Company offices and facilities, association offices, restaurants and hair salons, workshops, can be found in this category (see Fig. 5 left).
- Storages: A significant number of storages, storage rooms, repositories and stock-rooms exist in the neighborhood. Some of them seem to not be currently in use (see Fig. 5 middle).
- Stores-Retail: Department stores, bakeries (see Fig. 5 right), fire security/protection gear shops, motorcycle repair shops and clothing cleaning stores, are found in the area.

Once the data collection phase is over, the structuring of the 3D model of the neighborhood takes place. An in-depth demonstration and simulation of the REVIT toolboxes is given by the team leaders to all volunteers (Fig. 6).

The next phase of the proposed methodology entails the editing and correction of the crowdsourced BIMs and 3D representation of the neighborhood, by the involved

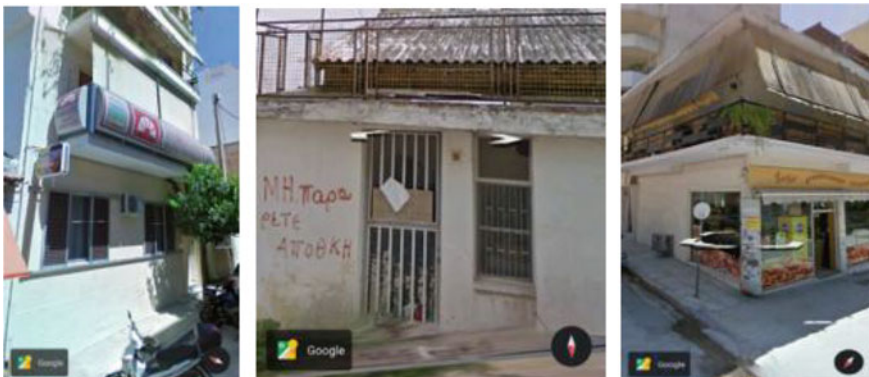


Fig. 5 Examples of an office (left), storage (middle) and bakery (right)



Fig. 6 Indicative BIM product, originally constructed under the instructions of the authors and corrected by the authors (in the Revit program) (Source Manolis Chasapis)

professionals. Mistakes regarding the modelled elements, their elevation and location are detected and corrected by the experts. Special attention should be paid on the modelling of the surrounding area of the buildings and the road network so that the lowest elevation of each building as derived from Google Earth Pro environment will fit with the elevations measured to model the surrounding area. Otherwise, some buildings may not be aligned to the lowest elevation surface causing visual and development inconsistencies (see Fig. 7 left). The correction of the aforementioned problems is done by the experts. Any deviations regarding the elevation and georeference are eliminated, resulting in a homogenous and cohesive final product (see Fig. 7 right).

The official 2D cadastral maps of each City Block containing land parcels are added as a new layer in each project, by the experts. The cadastral maps are obtained



Fig. 7 Wrong elevation of the building (Source Vasilis Spetsieris) (left); Right elevation of the building as corrected by the authors (right)

Fig. 8 Integration of the official 2D cadastral parcels by the experts



from the Official National Cadastral Geoportail, INSPIRE. The parcels, of each City Block, are depicted with purple colorations (see Fig. 8).

The official construction and height restrictions are modelled, by the experts, as third dimensional surfaces, visualizing the building regulations and construction limitations (see Fig. 9). The 3D surfaces are modelled in accordance with the regulatory framework.

The spatial extent of each land use is approximately modelled and visualized following the boundaries of the 2D architectural and topographic plans.

Semantic data are added for each single land use (see Fig. 10):

- the name and type of each use,
- the department that the use belongs to,
- the level that the use is located,
- the upper height limit that it reaches,
- the IFC (Industry Foundation Class) type that is to be exported as,
- the occupancy (private/public),



Fig. 9 Construction and height restrictions visualization as 3D surfaces by the experts

| Room Schedule of City Block 27 (Given Index) / 53 (Official National Index) | | | | | | | | | | |
|---|---------|------------------|-----------|--------------------|-----------|----------|------------|----------------------|-----------|-------------|
| A | B | C | D | E | F | G | H | I | J | K |
| Name | Level | Export to IFC As | Perimeter | Area | Occupancy | Occupant | Department | Comments | Image | Upper Limit |
| Store - Retail | Level 0 | ifcSpace | 42 | 108 m ² | Private | Private | Retail | Bakery | 2.png | Level 0 |
| Service | Level 0 | ifcSpace | 17 | 19 m ² | Private | Private | Services | Company Offices | 1.png | Level 0 |
| Service | Level 0 | ifcSpace | 42 | 79 m ² | Private | Private | Services | Construction - Build | 1.png (2) | Level 0 |
| Service | Level 0 | ifcSpace | 43 | 24 m ² | Private | Private | Services | Dining Area - Rest | 2.png (2) | Level 0 |
| Storage | Level 0 | ifcSpace | 16 | 15 m ² | Unknown | Unknown | Storage | Storage Shed - Rep | 1.png (3) | Level 0 |

Fig. 10 Table with semantic and metric information about land uses in an urban block

- the calculated area and perimeter,
- an image of the according use from the Streetview platform, and
- additional information in the comments section.

Enrichment of the collected information of the created 3D urban model of the neighborhood with data about existing smart devices in the neighborhood and urban elements that will be used for the creation of the neighborhood’s DT is made by the volunteers (see Fig. 11).

On-site inspections in the neighborhood are needed to correct gaps and errors or misinterpretations. The team leader’s check, update, complete and correct the information collected in the office, as well as the classification of the existing land uses and their location.

BIMs of newly built apartment buildings may also be integrated into the 3D model and uploaded on Tandem platform. Personal, protected data may not be integrated. Then the necessary geospatial infrastructure for a DT of the neighborhood is structured through a categorization and visualization of all land uses and all smart devices on the building facades and uploaded in the open-source Tandem platform where citizen and local administration may query and derive various data about the neighborhood, update the information and may use it for management purposes.

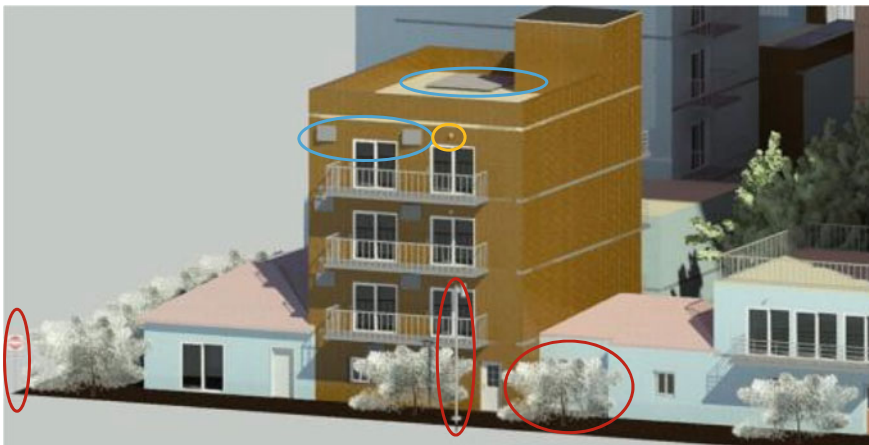


Fig. 11 Smart devices, in Revit (Orange: MEP/Blue: HVAC/Red: Urban elements)

4 Case Study

A neighborhood of Athens, Greece, comprises the testing area of this research. The structural profile of the area is defined by the interchange of complex multi-storey attached apartment buildings and single family detached houses (see Fig. 12).

For a unified and collective presentation of the neighborhood, the cloud-based visualization platform of Tandem (by Autodesk) is utilized by the experts (see Fig. 13).

The set of the edited and enriched with additional information city blocks are imported in the newly established (still in Beta mode) Tandem platform, by Autodesk.

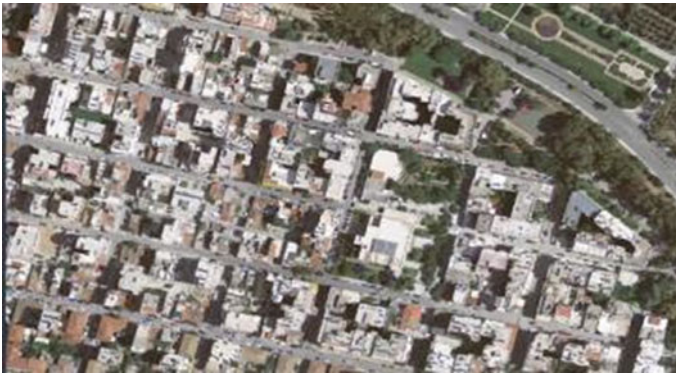


Fig. 12 Orthophoto of the greater neighborhood



Fig. 13 The 3D integrated BIM model of the neighborhood, in the Tandem platform

Tandem is a platform, that provides a cooperative and cohesive virtual environment. In this platform the experts have categorized the various elements, both semantically and visually, into various groups according to their type and functionality (see Figs. 14, 15 and 16).

Tandem provides an interoperable and highly detailed environment, where users may zoom and visualize all the various components in good resolution (see Fig. 17).

The prisms, depicting the various land uses in the area, are split into 3 different categories and accordingly presented with different colors (see Fig. 18).

Each category has a handful of sub categories, resulting in visualization with different color shades (see Fig. 19, Table 1).

A detailed inventory of semantic information, is created for the land use categories (see Fig. 20):

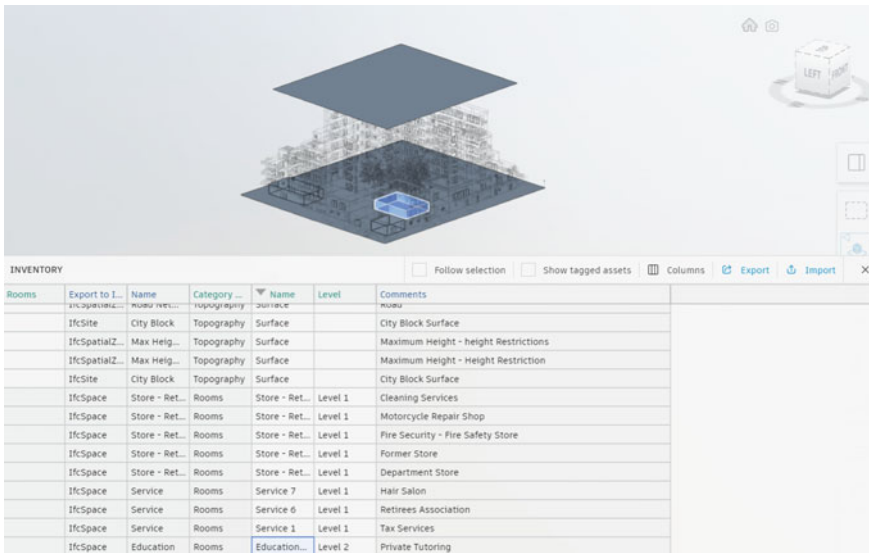


Fig. 14 Volumetric visualization and conceptual classification of a land use 3D space

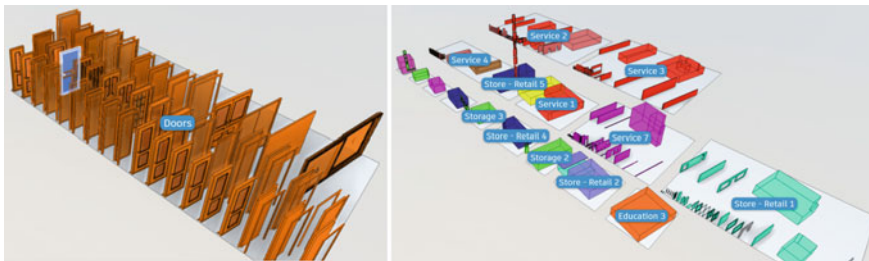


Fig. 15 Selected modelled components, filtered by user defined configurations



Fig. 16 MEP, HVAC and street furniture in the broader neighborhood

Fig. 17 Zoom and visualization of Smart devices and street furniture (e.g., electrical control panels, intercoms, traffic signs, lighting poles and trash bins) in the Tandem platform

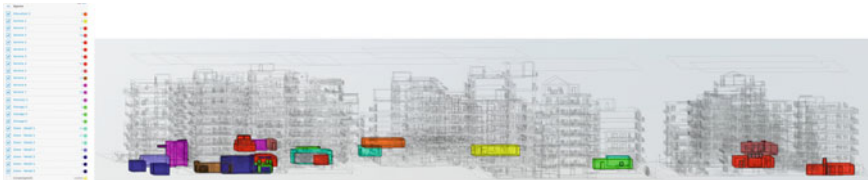
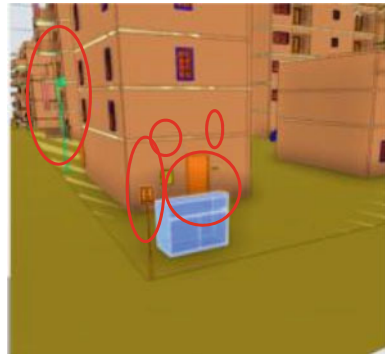


Fig. 18 The different colors show the location of each one of the 3 major land use categories

- the name, category and sub category of each use,
- the location of the space,
- the occupancy of the space (private/common),
- the calculated area and perimeter of the space,
- extra information about the use,
- an image of each space, captured in the Streetview platform,
- the limit offset, the upper limit and the unbounded height of the use's prismatic volume,
- and IFC related information.



Fig. 19 The different sub categories of the land use of the area under study

Table 1 Land use sub categories and their assigned colors

| Services | Color | Storages | Color | Stores | Color |
|-----------------|--------|------------|-------|---------------------------|---------|
| Educational use | Orange | Storages | Green | Bakery | Cyan |
| Companies | Red | Stockrooms | Green | Department store | Cyan |
| Offices | Purple | | | Motorcycle repair shop | Magenta |
| Restaurants | Brown | | | Fire protection gear shop | Magenta |
| Workshop | Yellow | | | | |

Smart devices are chosen to be visualized with a different color (see Figs. 21 and 22).

Each element of every category is enriched with semantic data, leading to informational inventories (see Fig. 23). Functional and operational information (see Fig. 24) is added to each smart device, highlighting its purpose and exploitation, in the real world (see Fig. 25).

For instance, the solar panels are added in the category ‘Equipment and Furniture’, in the ‘Assembly Code’ field. The system class, for each solar panel, is filled with the ‘Domestic Hot Water’ value, while in the ‘Classification’ panel the option ‘Solar heat collector’ is picked (see Fig. 26).

The figure displays two screenshots of software inventory tables. The top screenshot shows a table with columns: Name, Level, Classification, Category, System Class, Comments, Department, Image, Name, Number, Occupancy, and Occupant. The bottom screenshot shows a more detailed table with columns: Occupant, Type Name, Level, Limit Offs., Upper Lim., Area (m²), Computat., Perimeter, Unbounded, Volume (m.), Export to I., Export to L., IFC Predef., and IfcGUID.

Fig. 20 Inventory with information about all land use categories

5 Discussions

An assessment of the proposed method in terms of required time and costs is attempted. Table 2 shows a rough estimate of the required time for each stage of work for the creation of the 3D geospatial infrastructure for the DT of each urban plot of the neighborhood. Each urban plot includes about 13 apartment buildings of 4 to 5 floors each.

The required costs include only the reimbursement of the involved professionals and the per deem costs for the volunteers involved. Software and data used are open source, free of cost.

Fig. 21 Air conditioning unit (in yellow), surveillance cameras (in green), fire alarms (in green) and a liquid fuel tank (in transparent cyan)



Fig. 22 Smart devices and street furniture, in the neighborhood under study



Fig. 23 Classification and functionality of an air conditioning unit

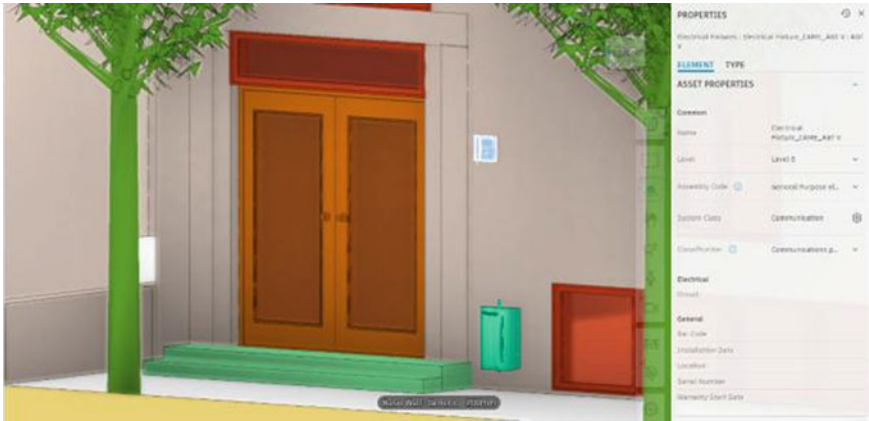


Fig. 24 Classification and functionality of an intercom



Fig. 25 Classification and functionality of a lighting pole

The management of Revit files created by different users is the largest challenge that can be faced. The correction of elevation errors may be a challenging task, as merged files are complicated and hard to be managed and altered. Therefore, such corrections should be made prior to the merging of individual BIMs. The proposed methodology may be proved of great value for the management of urban neighborhoods so that no neighborhood will be left behind, regardless of the availability of state funding. It is a good example of making the best use of available open data.

As a future improvement of the methodology, the software used by the authors to create the DT of a building in Evangelou et al. (2022) may be tested and used for



Fig. 26 Semantic categorization of a solar panel unit

Table 2 Duration for each methodology step

| Methodology steps | Duration—days required |
|---|-------------------------------|
| Office work by the experts (investigation of the available open data and call for volunteers) | 7 days |
| Training of the volunteers by the team leaders | 10 days |
| Structure of the 3D model of the neighborhood (construction of the building BIMs, 3D urban plot and road network) by each volunteer | 5 days for each urban plot |
| Editing and corrections, insertion of additional elements on the building facades and the surrounding area by the experts | 1, 5 days for each urban plot |
| Uploading of all information in the Tandem platform | 3 days |

tracking and monitoring the function and operation of all smart devices inside the buildings or in the surrounding area.

6 Conclusions

Merging BIM, 2D cadastral maps and land use information through the proposed methodology highlights the importance of the availability of open and free 2D data to be used as a basemap for 3D cadastral and urban management applications. It also includes training techniques for enabling crowdsourced geospatial data collection either for approximate compilation of BIMs (and integration of 3D cadastral data) of old constructions or for reporting any other geospatial information related to

urban management in a timely manner that will allow real—time community engagement in evidence based decision making through DT technology. Like in every other crowdsourced method, editing and checking of collected data is a necessity. The contribution and the roles of team leaders and volunteers are clearly defined. Additions and enrichments can be made continuously, in order to simulate and monitor in real—time the land use regulations and their implementation in the area under study. Thus visualization and monitoring of the development and the implementation of rules and regulations is enabled. The integration of smart devices provides dynamic monitoring of the functioning of the city blocks such as energy consumption, alarm notifications, remote lighting and energy controls, traffic management, garbage collection management, spatial planning and land use implementation monitoring, safety interventions, service provision betterment, maintenance and management of green spaces, etc.

Assessing the quality of the final result in terms of reliability and completion it is considered to be satisfactory and sufficient. The 3D model of the neighborhood is being thoroughly checked by the experts before it is inserted in the final platform of Tandem. The geometry, structure, georeference, sizing and height of the buildings is being revised, edited and corrected by the experts. The team leaders and volunteers utilized the open platforms of Google Earth Pro and Streetview to acquire the necessary information about the height and dimensions of the various buildings and its components. These platforms provide tools for approximate measurements but are more than adequate and reliable for a crowdsourced fit—for—purpose application. The benefits of such a fit—for—purpose approach for the necessary geospatial infrastructure for a DT include: (a) low data collection cost, (b) fast compilation procedure, (c) easy and quick updating, (d) community participation and engagement, and (e) broad applicability also in poor neighborhoods so that nobody will be left behind. The proposed methodology for the compilation of a fit—for—purpose DT infrastructure consists the result of an ongoing research on developing low—cost and appropriate tools for the fast and reliable implementation of several SDGs of the UN Sustainable Development Agenda 2030. Urban management applications may include: development monitoring, energy consumption monitoring and controlling policies, security monitoring, management of common use spaces, reporting of damages, traffic monitoring and control, etc.

The proposed methodology utilizes various programs and types of data that favor its reproducibility. The 3D model of the neighborhood is being constructed in the commercial software of the Revit program and then exported as an IFC file. The open standard of IFC is highly interoperable and promotes the reuse of the data in a plethora of platforms and programs, cloud or computer based. Then the IFC file of the 3D model is being inserted in the Autodesk platform of Tandem. If it was another platform to be chosen for the final outcome there would not be a problem, as the IFC standardization is compatible with the vast majority of open and computer based platforms. As the collected geospatial data by the volunteers are being thoroughly checked by the involved experts the issue of georeference of the buildings does not affect the reproducibility of the proposed methodology. The nature of Tandem

platform enables the use of the proposed crowdsourced methodology as several users can participate into a collaborative project.

Through the crowdsourced data collection phase 3D information on the land uses of the area under study is collected. On top of this data, applications on 3D Cadaster, land management and utilization can be constructed. By collecting 3D crowdsourced data about the various land uses of an area, a virtual open national hub for the betterment of land utilization can be created. Such data can also support an open land use registry, updated by the crowd and managed by experts. Crowdsourced data can be verified and checked by state professionals and then published in an open platform, for every citizen to see and manage, achieving a low—cost and easy way of enabling the crowd in land management procedures. Connecting with sensors it also allows interaction and interventions from a distance.

3D information about the land uses, height and building restrictions may transform a DT into a vital and complete 3D instance that may reform the status quo of 2D representation in the current cadastral portals and systems.

References

- Afzini MB (2017) Occupational health and safety in construction projects. Ph.D. thesis, De La Salle University, Manila, Philippines
- Alonso R, Borrás M, Koppelaar RHEM, Lodigiani A, Loscos E, Yöntem E (2019) SPHERE: BIM digital twin platform. *Proceedings* 20(9):1–6
- Andritsou D, Gkeli M, Soile S, Potsiou C (2022) A BIM/IFC—LADM solution aligned to the Greek legislation. In: *The international archives of the photogrammetry, remote sensing and spatial information sciences, XXIV ISPRS Congress, Nice, France, vol XLIII-B4–2022*, pp 471–477
- Atazadeh B, Kalantari M, Rajabifard A, Ho S, Ngo T (2017) Building information modelling for high-rise land administration. *Trans GIS* 21(1):91–113
- AUTODESK. <https://intandem.autodesk.com/>. Accessed on 2023
- Brilakis I, Fischer H, Fellow S (2019) Built environment digital twinning. Technical University of Munich, Germany. Retrieved from https://publications.cms.bgu.tum.de/reports/2020_Brilakis_BuiltEnvDT.pdf. Accessed on 15 May 2023
- Dembski F, Wössner U, Letzgus M, Ruddat M, Yamu C (2020) Urban digital twins for smart cities and citizens: the case study of Herrenberg, Germany. *Sustainability* 12(2307):1–17
- Douglas D, Kelly G, Kassem K (2021) BIM, digital twin and cyber-physical systems: crossing and blurring boundaries. In: *Proceedings of the 2021 European conference on computing in construction, Rhodes, Greece, vol 2*, pp 204–211
- Evangelou T, Gkeli M, Potsiou C (2022) Building digital twins for smart cities: a case study in Greece. In: *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences, 17th 3D GeoInfo Conference, Sydney, Australia, vol X-4/W2–2022*, pp 61–68
- Eyre J, Freeman C (2018) Immersive applications of industrial digital twins. In: *Proceedings of the 15th EuroVR international conference, London, UK*, pp 1–8
- Fuller A, Fan Z, Day C, Barlow C (2020) Digital twin: enabling technologies, challenges and open research. *IEEE Access* 8:108952–108971
- Haag S, Anderl R (2018) Digital twin—proof of concept. *Manuf Lett* 15(Part B):64–66
- Khajavi SH, Motlagh NH, Jaribion A, Werner LC, Holmström J (2019) Digital twin: vision, benefits, boundaries, and creation for buildings. *IEEE Access* 7:147406–147419

- Lu VQ, Parlikad AK, Woodall P, Ranasinghe GD, Heaton J (2019) Developing a dynamic digital twin at a building level: using Cambridge campus as case study. In: Proceedings of the international conference on smart infrastructure and construction 2019 (ICSIC): driving data-informed decision-making, pp 67–75
- Madni AM, Madni CC, Lucero SD (2019) Leveraging digital twin technology in model-based systems engineering. *Systems* 7(7):1–13
- Qi Q, Tao F, Zuo Y, Zhao D (2018) Digital twin service towards smart manufacturing. *Procedia CIRP* 72:237–242
- Redmond A, Hore A, Alshawi M, West R (2012) Exploring how information exchanges can be enhanced through Cloud BIM. *Autom Constr* 24:175–183
- Shirowzhan S, Tan W, Sepasgozar SME (2020) Digital twin and CyberGIS for improving connectivity and measuring the impact of infrastructure construction planning in smart cities. *ISPRS Int J Geo-Inf* 9(240):1–11
- Wagner R, Schleich B, Haefner B, Kuhnle A, Wartzack S, Lanza G (2019) Challenges and potentials of digital twins and industry 4.0 in product design and production for high performance products. *Procedia CIRP* 84:88–93
- Wahbeh W, Kunz D, Hofmann J, Bereuter P (2020) Digital twinning of the built environment—an interdisciplinary topic for innovation in didactics. In: *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences, XXIV ISPRS Congress, vol V–4–2020*, pp 231–237

Artificial Intelligence for the Automated Creation of Multi-scale Digital Twins of the Built World—AI4TWINNING



André Borrmann, Manoj Biswanath, Alex Braun, Zhaiyu Chen, Daniel Cremers, Medhini Heeramaglore, Ludwig Hoegner, Mansour Mehranfar, Thomas H. Kolbe, Frank Petzold, Alejandro Rueda, Sergei Solonets, and Xiao Xiang Zhu

Abstract The AI4TWINNING project aims at the automated generation of a system of inter-related digital twins of the built environment spanning multiple resolution scales providing rich semantics and coherent geometry. To this end, an interdisciplinary group of researchers develops a multi-scale, multi-sensor, multi-method approach combining terrestrial, airborne, and spaceborne acquisition, different sensor types (visible, thermal, LiDAR, Radar) and different processing methods integrating top-down and bottom-up AI approaches. The key concept of the project lies in intelligently fusing the data from different sources by AI-based methods, thus closing information gaps and increasing completeness, accuracy and reliance of the resulting digital twins. To facilitate the process and improve the results, the project makes extensive use of informed machine learning by exploiting explicit knowledge on the design and construction of built facilities. The final goal of the project is not to create a single monolithic digital twin, but instead a system of interlinked twins across different scales, providing the opportunity to seamlessly blend city, district and building models while keeping them up-to-date and consistent. As testbed and demonstration scenario serves a urban zone around the city campus of TUM, for which large data sets from various sensors are available.

Keywords Digital twin · Artificial intelligence · Point clouds · Building Information Modelling (BIM)

1 Overview

Today, technologies are available that allow capturing built environment with a wide range of technologies and from a broad range of platforms like UAVs, planes,

This article was selected based on the results of a double-blind review of an extended abstract.

A. Borrmann (✉) · M. Biswanath · A. Braun · Z. Chen · D. Cremers · M. Heeramaglore · L. Hoegner · M. Mehranfar · T. H. Kolbe · F. Petzold · A. Rueda · S. Solonets · X. X. Zhu
Technical University of Munich, Munich, Germany
e-mail: andre.borrmann@tum.de

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
T. H. Kolbe et al. (eds.), *Recent Advances in 3D Geoinformation Science*, Lecture Notes in Geoinformation and Cartography, https://doi.org/10.1007/978-3-031-43699-4_14

233

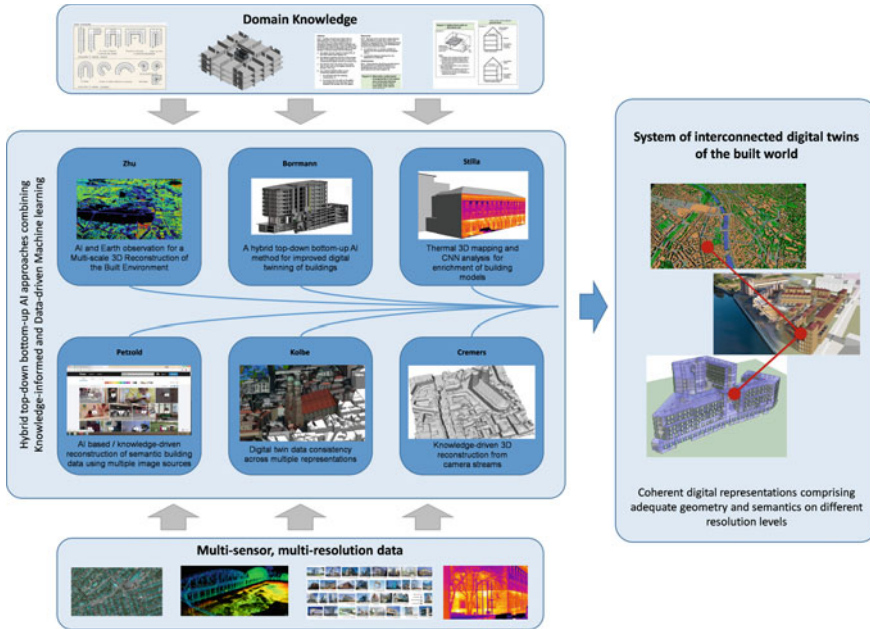


Fig. 1 Overall structure of the AI4TWINNING project including its subprojects and their interaction

satellites, mobile mapping cars, building cranes or robots on construction sites. The resulting point clouds or 3D mesh models provide a high level of geometric detail, and are well suited for realistic high-resolution 3D visualization. However, these urban data cannot be directly used for applications requiring semantically structured models, such as Building Information Modelling (BIM) and Urban Information Modelling (UIM).

The AI4TWINNING project that started in 2022 and runs for four years aims at realizing significant progress in automated creation of digital twins of the built world. The goal is to create digital replicas of buildings and infrastructure assets at varying, yet interconnected levels of detail comprising both geometry and rich semantics (Fig. 1).

The project implements a sandwich methodology where bottom-up data-driven methods are combined with top-down knowledge-driven approaches, thus combining the strengths of both approaches in order to achieve a new level of robustness in digital twinning. For the top-down direction, human knowledge on the structure of buildings and the functional relationships of their components are digitized and made available for computational processing. For the bottom-up direction, different sensing technologies on different scales are combined with a multitude of AI-based processing methods.

While data sources range from satellite-based earth observation, aerial images, radar, to UAVs terrestrial laser-scanning and thermal cameras, the applied AI-based

processing methods will include point cloud segmentation, image-processing, etc. A major challenge lies in fusing the information created through different paths and creating coherent digital twins from it. As modern machine-learning methods are based to a large extent on statistics, a major goal of the project is to represent the reliability of the different components of the resulting twin.

The envisioned digital twins will comprise different resolution levels – twins on the coarsest level will represent an entire city and model its buildings and infrastructure assets in comparatively low geometric-semantic resolution. The digital twins on the finest level, however, will model both the exterior and the interior of building in high geometric-semantic resolution, i.e. including individual building components in what is known as a building information model. As result, we aim at creating a network of interconnected digital twins at different resolution levels. Major emphasis is put on ensuring consistency across the different scales and resolutions.

For test, validation and demonstration purposes, the TUM City Campus and its surrounding districts provide a shared example and testbed across all participating groups, taking advantage of the fact that a multitude of different datasets are already available at the different groups, including optical and Radar satellite images, aerial images, StreetView images, 3D point clouds (from mobile mapping, airborne Laserscans, dense stereo reconstruction), 3D meshes, OpenDRIVE datasets, Open StreetMap data, Semantic 3D City Models, and Building Information Models.

The project is composed of six subprojects that are outlined below.

2 Subprojects

2.1 AI and Earth Observation for a Multi-scale 3D Reconstruction of the Built Environment

Recently, big Earth observation data amounts in the order of tens of Petabytes from complementary data sources have become available. For example, Earth observation (EO) satellites reliably provide large-scale geo-information of worldwide cities on a routine basis from space. However, the data availability is limited in resolution and viewing geometry. On the other hand, closer-range Earth observation platforms such as aircraft offer more flexibility in capturing Earth features on demand. This combination enables the creation of a multi-scale representation of the built environment.

This sub-project develops new data science and machine learning approaches tailored to optimal exploitation of big geospatial data as provided by the above mentioned Earth observation platforms, to provide invaluable information for a better understanding of the built world. In particular, we focus on 3D reconstruction of the built environment on an individual building level. From the AI methodology perspective, while the large-scale stream puts the focus on the robustness and transferability



Fig. 2 Multi-scale 3D reconstruction of the built environment. Left: LoD1 building models from TerraSAR-X imagery Sun et al. (2022). Right: LoD2 building models from aerial LiDAR point clouds Chen et al. (2023)

of the machine learning models to be developed, for the very high-resolution stream we particularly research the fusion of multi-sensory data, as well as hybrid models combining model-based signal processing methods and data-driven machine learning ones for an improved information retrieval.

In the realm of multi-scale building reconstruction, we have developed method prototypes expressly for synthetic aperture radar (SAR) Sun et al. (2022) and light detection and ranging (LiDAR) Chen et al. (2023, 2022). SAR imagery, impervious to weather conditions and accessible constantly, is notably advantageous for disaster response and regions with frequent cloud cover, while LiDAR offers greater flexibility in capturing detailed building geometry. Our SAR-based methodology has been validated on four urban data sets using TerraSAR-X images in both high-resolution spotlight and stripmap modes, exhibiting substantial computational efficiency while preserving exceptional accuracy in height estimation. On the other hand, our LiDAR-based technique perceives polygonal reconstruction as a classification problem, utilizing a graph neural network to categorize polyhedra produced through space partitioning based on their occupancy within the underlying building instance. The subsequent step involves extracting the building surface from these classified polyhedra. Our preliminary results illustrating multi-scale 3D reconstruction of the built environment are depicted in Fig. 2. We aim to further refine these methodologies to bolster their capability to handle challenges across diverse reconstruction scenarios. We also envisage the development of hybrid models that can more efficiently utilize the strengths of various data sources to generate large-scale, higher-resolution 3D reconstructions. This amalgamation would foster a more integrated model, equipped to navigate the complexities inherent in building reconstructions under differing conditions and scales. Ultimately, these advancements will serve to bridge the gap between extensive geospatial data and the extraction of meaningful, actionable insights from the built environment.



Fig. 3 Reconstruction of a building using model-based approach (on the left). Highlighted structures have close hidden representation (on the right)

2.2 *Knowledge-Driven 3D Reconstruction from Camera Streams*

While the reconstruction of the 3D world from moving cameras has advanced enormously over the last few years in terms of precision, density, robustness and speed, to date most power reconstruction algorithms merely generate a sparse point-cloud of the observed world. At the same time there are increasingly powerful model-based representations of the man-made world. Yet, there is a gap between the pointclouds generated by real-time capable visual SLAM algorithms and a fully model-based representation of the world as employed in architecture.

Thus, this subproject aims to develop methods that can create model-based reconstructions of the observed world from an input video. To this end, we combine highly accurate camera-based visual Simultaneous Mapping and Localization (SLAM) methods with the predictive power of deep networks. More specifically, we design and train networks to bridge the gap between purely data-driven point cloud reconstructions and model-based representations of man-made worlds - such as car models or building models. Appropriately designed and trained networks take as input images and/or point clouds and generate as output model-based representations of the observed world.

The subproject also emphasizes the importance of transferability. The goal is to develop a representation that can be effectively transferred from one scene to another, enabling the generalization of learned knowledge across different environments. The project will adopt an approach focused on learning representations of small chunks of 3D data. Instead of attempting to capture the entire scene or object in a single representation, we break down the 3D data into smaller, more manageable parts. By learning representations at a more granular level, the project can achieve transferability across different scenes and objects (Fig. 3).

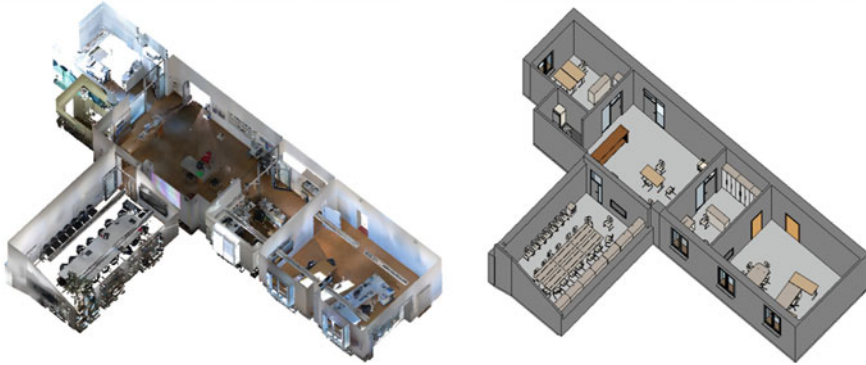


Fig. 4 Building digital twin model creation. Left: raw point cloud of the building's indoor environment. Right: a highly parameterized building digital twin model with rich semantics and coherent geometry

2.3 A Hybrid Top-Down Bottom-Up AI Method for Improved Digital Twinning of Buildings

With the increasing complexity and dynamic changes in the cities, the need for up-to-date information about buildings and indoor environments for operation and management has risen substantially. Digital twins are beneficial tools for digitally representing an asset's physical and functional properties and a shared data source for building information modeling that provides a reliable basis for decision-making throughout the project life cycle.

Over the past decade, there has been a significant increase in the utilization of advanced data acquisition techniques in the built environment, leading to a growing trend in the creation of digital representations of buildings with coherent geometry and rich semantic information. Despite advanced development in engineering and technology, automatic or semi-automatic building digital twin creation using point cloud data and RGB images is still an open topic in the engineering and design society which needs novel approaches.

The subproject focuses on developing knowledge-driven top-down approaches for digital twin generation (Fig. 4). While bottom-up approaches typically start from points and successively build up surfaces and volumes, they very often face problems when it comes to creating meaningful building objects and their relations. We therefore apply a top-down approach by fitting a highly parametrized building model to the observed data. Based on a typology of typical office and residential buildings in Germany, we develop a set of highly parametrized and modularized building models that provide sufficient degrees of freedom to allow modeling a wide range of different real-world buildings Mehranfar et al. (2023). These building models will be designed in such a way that they represent human knowledge on building design and engineering.

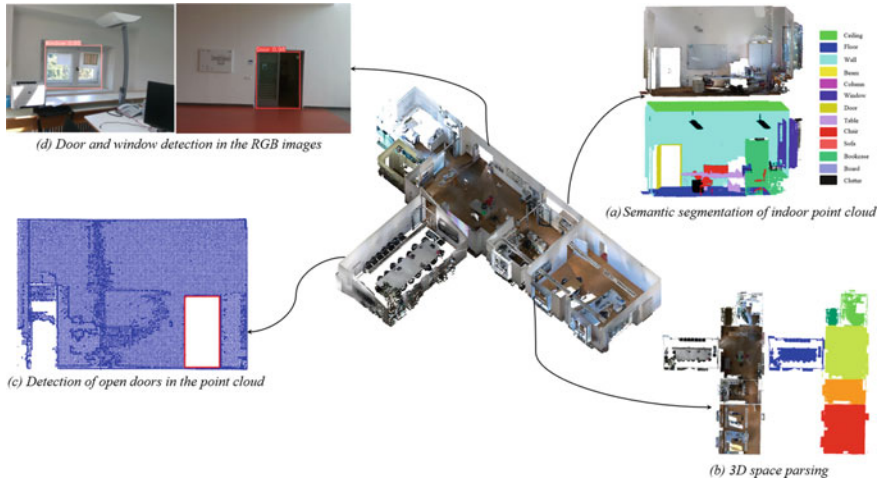


Fig. 5 Extracting semantic information for designing the highly parameterized building’s digital twin model

The process of designing the highly parametrized building models is initially involved by extracting the main structural elements of the building (eg. ceiling, wall and floors) and partitioning the 3D individual spaces within indoor point cloud Mehranfar et al. (2022). Major parameters (number of walls, windows, doors and columns) are also identified by a rough pattern-based interpretation of the point clouds and RGB images (bottom-up) (Fig. 5). To overcome the data challenges, the domain knowledge in construction and design will be aligned with the capabilities of artificial intelligence methods in scene understanding. The actual fitting of the model to the point cloud is approached as a high-dimensional optimization problem (top-down). For its solution, AI methods based on meta-heuristics are investigated. Thanks to applying the parametric modelling process, the resulting digital twin of individual buildings provide comparatively abstract geometry, but rich semantics and consistent logic (relationships, connection points etc.) which promise significant progress in the field of “Scan-to-BIM”.

2.4 AI Based/Knowledge-Driven Reconstruction of Semantic Building Data Using Multiple Image Data Sources

The usage of buildings and their changes over time at the district and city levels are essential parameters for urban planning and strategic urban development BBSR (2017). Digital twins of cities are increasingly implemented in municipalities, but currently focus mainly only on geometric representations. Alphanumeric data regarding building usage are partially incorporated from various analog and

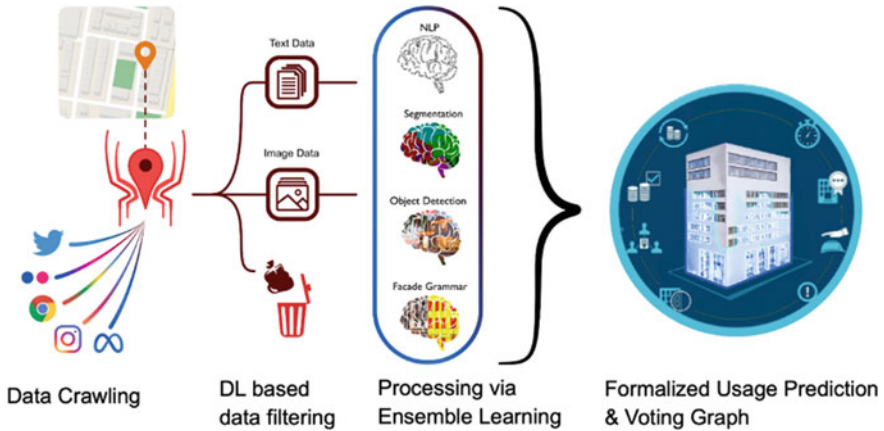


Fig. 6 The proposed pipeline. Data is obtained from openly available sources, filtered, and processed by multiple state of the art deep learning models. A formalized resulting prediction along its voting graph is delivered

digital sources of multiple authorities, often in time-consuming manual or semiautomatic procedures, and often they are not up to date. A key challenge is to keep these data up-to-date and consistent, as they are the basis for various applications.

This subproject aims to develop a data- and knowledge-driven methodology using Deep Learning methods combined with a Knowledge Base to reconstruct semantic building data.

We propose a multi-stage pipeline (Fig. 6) to improve the semantic enrichment process of digital twins by predicting the building usage from openly available sources such as Flickr, Twitter, and Google Street View. Since predicting building usage based on facade features is not always unique, additional sources such as comments, ratings, tweets, and hashtags to images are evaluated. It is composed of multiple state-of-the-art fine-tuned deep learning models as well as a novel grammar-based one.

The building architecture is expressed in terms of its underlying compositional logic Herzog et al. (2016); Cremers (2015); Bleek (2022); Berger et al. (1999); Kemp (2006); Robert and Wege zur Kunst (2009); Mitchell (1998). This “façade Grammar” (Fig. 7) is utilized to handle hybrid use buildings by analyzing patterns in its structure. To achieve this “Façade Grammar”, context-free grammars (CFGs) are implemented to formally describe this compositional logic. To expand upon existing datasets, a data crawler is implemented to gather additional data from open internet sources. This additional information is filtered, and the façade images and accompanying textual data (i.e., comments, reviews, tweets, and hashtags) are processed using several state-of-the-art deep learning models (segmentation, classification, NLP) via Ensemble Learning. Through this approach, a formalized prediction and its accompanying voting graph is provided to the digital twins group (Subproject Kolbe). Through the use of transfer learning, this pipeline is capable of being refitted for different regional and cultural characteristics. Several existing models have been investigated

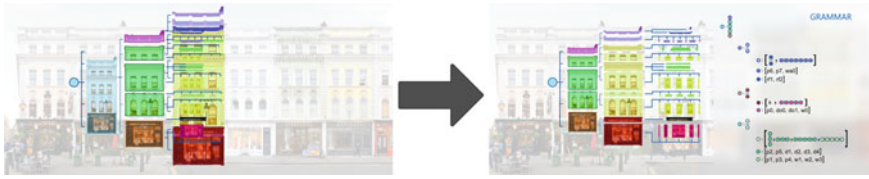


Fig. 7 “Facade Grammar”—parts and components of the facade and their relationships

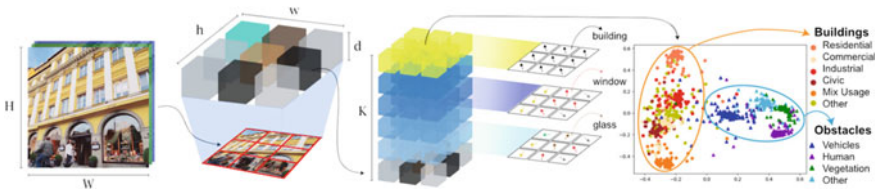


Fig. 8 Analysis of grammars, figure adapted from Garau et al. (2022)

and evaluated for segmentation, object detection, and classification. In our pipeline, we use and combine DeepLabv3 Chen et al. (2017) for segmentation, YOLO Redmon et al. (2016) for object recognition, and ResNext Xie et al. (2017) for classification.

CFGs approaches from image analysis Socher et al. (2011) and document analysis Chua et al. (2021) were explored and applied to a facade context in order to generate grammars. For the analysis of grammars, the framework Agglomerator Garau et al. (2022) was used to extract a representation of the part-whole hierarchy from visual features and determine the input distribution according to the conceptual-semantic hierarchical structure between façade parts and components (Fig. 8). In contrast to approaches that predict usage based only on image data, hybrid building usage can be predicted by analyzing part-whole trees.

A further aspect that was investigated was the quality of the freely available valid annotated datasets for training and evaluation. For example, the ETC Dataset, 104 rectified facade images of Haussmann’s renovated buildings in Paris, the CMP Facade Database with 602 images, and LabelMeFacade with 945 images were investigated. As a result, it was found that the existing datasets do not have a uniform standard, have different label classes, and are only partially rectified. Freely available valid datasets for training are insufficient, so a hybrid approach was taken using state-of-the-art models and fused datasets. The generation of annotated data using Deep Generative Models was carried out in collaboration with the subproject Prof. Zhu.

The next steps in the research project are developing methods to extract semantic information from texts (tweets, ratings, comments, hashtags) using NLP and the fusion of the results, including explanation as well as the integration of all “components” into the multi-stage pipeline and validation. The process to identify the usage of buildings can be carried out both at specific times and at time intervals to identify changes. For the setup of digital twins information about changes in use in the past will be also integrated.

2.5 *Thermal 3D Mapping and CNN Analysis for Enrichment of Building Models*

Thermal imaging cameras record electromagnetic radiation in the infrared range (IR), which is invisible to humans. This makes it possible to determine the characteristics of surfaces or detect objects that remain hidden in the visual range. An interesting field of application is the inspection of buildings in connection with current questions on the efficient use of energy. A common way for inspecting buildings concerning the thermal insulation is to take thermographs of the outer walls by an IR camera and evaluate visually the images in the recorded image geometry. But a direct three-dimensional spatial reference is not established for the measured values.

This deficiency becomes obvious when images of a complex building structure taken from different angles are combined, fused or the measured values of buildings are to be further processed and stored in an object-related manner. Surface-related IR-textures of facades allow a spatial analysis of thermal structures, but only allow very limited statements to be made about the geometry and material of the walls. So, knowledge of the inside and outside temperatures is necessary. The project aims to thermally record both external and internal building surfaces and to spatially locate the measurements with the aid of a building model. Therefore, an extended 3D thermal description of a building model needs to be created, and then subsequent analysis of thermal structures needs to be performed.

As part of creating the 3D thermal description, we introduce a thermal mapping algorithm for generating thermal textures, which is developed and implemented on two facades of a building. In our previous work Hoegner and Stilla (2018), we extracted the textures from thermal infrared image sequences by matching the images with 3D building models. But there were limited accuracy, blurred edges, and some projection errors because the geometry of building models is often simplified by removing small structures or overhangs. So to overcome those deficiencies, in this paper, we used a 3D thermal point cloud to project onto the facades of a building model. The thermal point cloud is prepared by Zhu et al. (2021) from mobile laser scanner point cloud extended with thermal intensities from TIR image sequences. The thermal mapping algorithm searches for nearest neighbor points in the point cloud to map the thermal intensities and generate the thermal textures. We use nearest neighbors instead of interpolation because we want to retain the original temperature values, although it results in some blank texels (where no corresponding neighbor points can be found). The method shows benefits such as occlusion detection and removal from textures for better temperature estimation of facade walls. We attempt three different approaches for searching the nearest point: minimize the angle to the normal vector of the texel plane, minimize the perpendicular distance to the normal vector, and minimize only the distance without considering the verticality. The second approach yields the finest quality textures with 0.04m mean perpendicular distance to normal vector and a reasonable computation time. Figure 9 (left) shows preliminary results of thermal textures generated for two facades. In the future, this method will be implemented on the entire university campus buildings by combining both indoor and

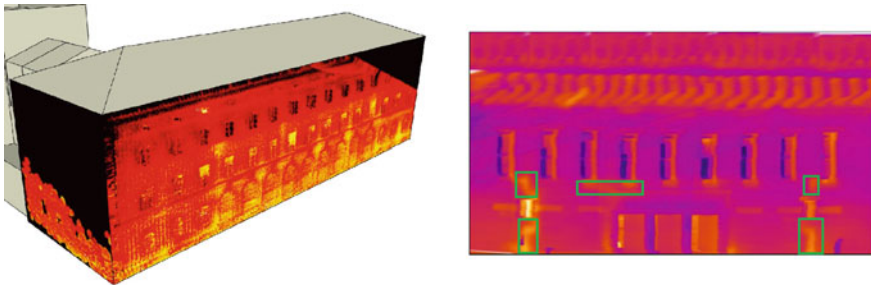


Fig. 9 Thermal mapping and analysis. Left: Thermal textures on a LoD2 building model. Right: Structures such as heating pipes, leakages, etc. are shown in green bounding boxes in a thermal texture

outdoor mapping. In this way, the building can be observed as a single unit instead of looking at individual surfaces, and we can spatially locate the measurements with a 3D reference to the building and link the inside structures with the outside more efficiently. The results should be consistent considering facades constructed by different materials at different periods of time and the mapping algorithm is expected to have a high detection rate with reasonably fast processing time. Performance evaluation of the methods will be carried out by comparing them with an annotated model.

2.6 *Digital Twin Data Consistency Across Multiple Representations*

With rapid developments on the technological front and improved data acquisition from various sources, different digital representations of urban objects created for varied applications exist next to each other. Though these models might represent the same real-world object, there is no explicit linking between them as they might be of different data types, data formats, geometry types, semantic information, etc. An Urban Digital Twin will not only consist of a single type of representation but must consider these multiple representations simultaneously. To create a 'Digital Twin' of the built environment, all these varied models created for different applications with different data structures, be it 3D point clouds, mesh models, semantic city models or Building Information Models (BIM), will need to be able to interact with each other. This is done by comparing or 'matching' models against each other. Once a match between corresponding models is established, it is possible to express a link between the models and/or the model components. A match also establishes the level of coherence between the models, which can become the foundation of an Urban Digital Twin. This can facilitate information flow between the different representations.

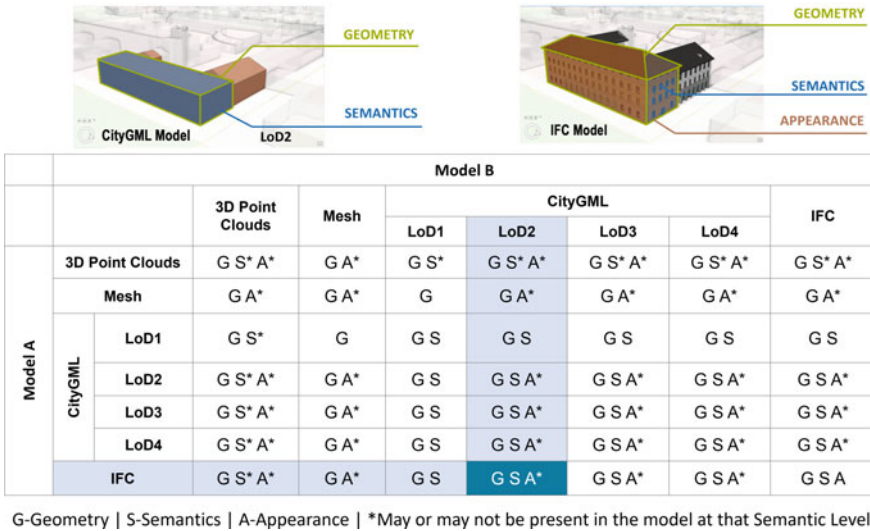


Fig. 10 Common comparable modelling aspects between the different types of representations

So far, there is no formal and computable framework available that can compare urban models over a broad range of representations on an objective basis. In the past, this has been addressed only partially by looking, e.g., for change detection between versions of urban objects as discussed in Nguyen et al. (2017) but this is mainly within the same type of representation. As discussed in Stadler and Kolbe (2007) each of the above-mentioned model types greatly differ from each other in terms of geometry, semantics, topology, appearance, etc. It is important to establish a common representation or a ‘baseline’ for the comparison and evaluation of models against each other. Before a match can be established, the common, comparable modelling aspects between the representations needs to be identified. For example, as seen in Fig. 10, if Model A (a CityGML LoD2 model) is compared with Model B (an IFC model), the comparable common aspects are geometry, semantics and possibly appearance. Model A would consist of boundary representations (BRep) while Model B would consist of Constructive Solid Geometry (CSG), both of which would need to be converted to a common geometry type. The semantic structure in both models is also very different and would need to be brought into a comparable framework.

In this subproject, we develop an ontology to express the linkage between urban models across different representations. Evaluation measures are developed based on this ontology, which describes the similarity and/or differences between various models. The match is expressed as a similarity measure for each modelling aspect, viz. geometry, semantics, topology, appearance, and change across time. For the above-mentioned representation types, understanding the heterogeneity of modelling aspects is key to establishing a match between models (Fig. 10).

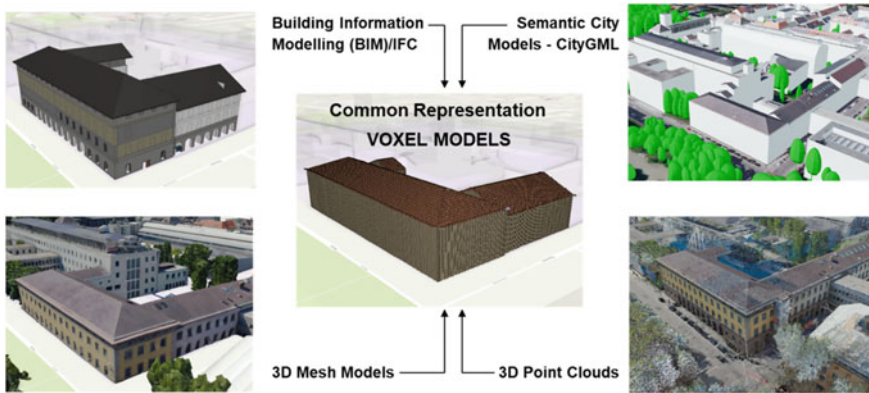


Fig. 11 RichVoxel models Heeramaglore and Kolbe (2022) as a common representation to match the different 3D city/building models

The first step in this process is to convert all models to a common representation type so that the integrity of the modelling aspects from the original representation is preserved. For example, any ‘common representation’ for semantic city models and BIM would require the transfer of not just geometry but also the semantic information as well as the organisational hierarchy and logical relationships between components. Though both the models, i.e., semantic city model (based on CityGML) and BIM, have semantics, the organisational structure of the information is very different in each model. The organisation of the information also needs to be translated into a cohesive and comparable structure. On the other hand, for 3D point clouds, the common representation would need to incorporate geometry, unstructured semantics, and possible appearance data. All the varying modelling aspects (as shown in Fig. 10) need to be translated without any loss, to the common representation for comparison. Therefore the common representation needs to have these characteristics - (a) unambiguous location, (b) geometric primitive that is compatible with points, polygons and solids, (c) ability to hold semantic information long with the organisation of information, (d) ability to support appearance information, etc. Voxels are an ideal primitive that support most of the above mentioned criteria as seen in Nourian et al. (2016).

Voxels in the field of 3D city modelling and urban research are not a new concept. Voxels have been used in various BIM/GIS applications such as path planning as seen in Wang et al. (2020), as volumetric modelling primitives Li et al. (2020), for space planning and semantic segmentation of 3D point clouds, Poux and Billen (2019), etc. Further analysis using voxels is also popular in many applications such as thermal analysis of buildings where voxelised IFC models have been used which is seen in Krijnen et al. (2021) or in Poux and Billen (2019) where voxels have been used to semantically segment 3D point clouds. But, the concept of a voxel as it exists is not enough to deal with the semantic organisation of city models especially those that follow the international standards of CityGML and IFC. For this, Heeramaglore

and Kolbe (2022) have developed the concept of a RichVoxel or a 'Semantically Enriched Voxel' which can map the information from the modelling aspects of all other representation types without loss of information.

For this joint case study in Munich, we employ the ontology to link the urban models given as separate datasets using a common representation in the form of 'Semantically Enriched Voxels' or 'RichVoxel' (Fig. 11) models. Once voxelised, the models are initially compared against the level of semantic information (explained in Heeramaglore and Kolbe (2022)) and a common baseline is established. The models are also compared against their spatial and semantic extents followed by a component to component match than an individual voxel match. The methods developed here perform qualitative and quantitative determination of the similarity measures for concrete datasets against modelling aspect as parameters which is explained in subsequent publications.

Acknowledgements The AI4TWINNING project is funded by the TUM Georg Nemetschek Institute for Artificial Intelligence for the Built World, which is thankfully acknowledged.



References

- BBSR (2017) Nutzungsmischung und die Bewältigung von Nutzungskonflikten in Innenstädten, Stadt- und Ortsteilzentren - Chancen und Hemmnisse. (<https://www.bbsr.bund.de/BBSR/DE/veroeffentlichungen/bbsr-online/2017/bbsr-online-23-2017-dl.pdf>), Accessed 26 Jul 2023
- Berger R, Berger E (1999) Bauwerke betrachten, erfassen, beurteilen: Wege zum Verständnis klassischer und moderner Architektur
- Bleek J (2022) Fassade und Ornament. (Brill Fink, 2022, 7)
- Chen Z, Ledoux H, Khademi S, Nan L (2022) Reconstructing compact building models from point clouds using deep implicit fields. *ISPRS J Photogram Remote Sens* 194:58–73
- Chen L, Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation
- Chen Z, Shi Y, Nan L, Xiong Z, Zhu XX (2023) PolyGNN: polyhedron-based graph neural network for 3D building reconstruction from point clouds. arXiv preprint [arXiv:2307.08636](https://arxiv.org/abs/2307.08636)
- Chua F, Duffy N (2021) DeepCPCFG: deep learning and context free grammars for end-to-end information extraction. *Doc Anal Recogn—ICDAR 2021*, 838–853
- Cremers J (2015) Die gestalterische Wirkung von Öffnungen in der Fassade. *Atlas Gebäudeöffnungen*, 24–31
- Garau N, Bisagno N, Sambugaro Z, Conci N (2022) Interpretable part-whole hierarchies and conceptual-semantic relationships in neural networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 13689–13698
- Heeramaglore M, Kolbe T (2022) Semantically enriched voxels as a common representation for comparison and evaluation of 3D building models. *ISPRS Annals Photogram Remote Sens Spatial Inf Sci X-4/W2-2022*. 89–96. 10.5194/isprs-annals-X-4-W2-2022-89-2022
- Herzog T, Krippner R, Lang W (2016) Fassaden atlas. (DETAIL, 2016, 12)
- Hoegner L, Stilla U (2018) Mobile thermal mapping for matching of infrared images with 3D building models and 3D point clouds. *Quant Infrared Thermography J* 15(2):252–270
- Kemp W (2006) *Architektur analysieren: Katalog Haus der Kunst München*. (Schirmer Mosel)
- Krijnen T, El-Diraby T, Konomi T, Attalla A (2021) Thermal analysis of IFC building models using voxelized geometries. In: *Proceedings of the 38th international conference of CIB W78*. (CIB W78 conference series), pp 437–446. <https://itc.scix.net/paper/w78-2021-paper-044>

- Li W, Zlatanova S, Gorte B (2020) Voxel data management and analysis in PostgreSQL/PostGIS under different data layouts. *ISPRS Ann Photogram Remote Sens Spatial Inf Sci VI-3/W1-2020*, 35–42
- Mehranfar M, Braun A, Borrmann A (2022) A hybrid top-down, bottom-up approach for 3D space parsing using dense RGB point clouds. *Proceedings of European conference on product and process modeling*
- Mehranfar M, Braun A, Borrmann A (2023) Automatic creation of digital building twins with rich semantics from dense RGB point clouds through semantic segmentation and model fitting. In *EG-ICE 2023; 30th international conference on intelligent computing in engineering*
- Mitchell W (1998) *The logic of architecture: design, computation, and cognition*. MIT Press
- Nguyen S, Yao Z, Kolbe T (2017) Spatio-semantic comparison of large 3D city models in CityGML using a graph database. In: *Proceedings of the 12th international 3D GeoInfo conference 2017 (ISPRS annals of the photogrammetry, remote sensing and spatial information sciences)*, ISPRS, 99–106
- Nourian P, Gonçalves R, Zlatanova S, Otori KA, Vu Vo A (2016) Voxelization algorithms for geospatial applications: computational methods for voxelizing spatial datasets of 3D city models containing 3D surface, curve and point data models. *MethodsX* 3:69–86
- Poux F, Billen R (2019) Voxel-based 3D point cloud semantic segmentation: unsupervised geometric and relationship featurizing vs deep learning methods. *ISPRS Int J Geo-Inf* 8(5)
- Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: *2016 IEEE conference on computer vision and pattern recognition (CVPR)*
- Robert H (2009) *Wege zur Kunst: begriffe und Methoden für den Umgang mit Architektur*. Schroedel
- Socher R, Lin C, Ng A, Manning C (2011) Parsing natural scenes and natural language with recursive neural networks. In: *Proceedings of the 28th international conference on international conference on machine learning*, pp 129–136
- Stadler A, Kolbe TH (2007) Spatio-semantic coherence in the integration of 3D city models. In: *Proceedings of the 5th international ISPRS symposium on spatial data quality ISSDQ 2007 in Enschede, The Netherlands, 13–15 June 2007 (ISPRS Archives)*, ISPRS
- Sun Y, Mou L, Wang Y, Montazeri S, Zhu XX (2022) Large-scale building height retrieval from single SAR imagery based on bounding box regression networks. *ISPRS J Photogram Remote Sens* 184:79–95
- Wang Q, Zuo W, Guo Z, Li Q, Mei T, Qiao S (2020) BIM voxelization method supporting cell-based creation of a path-planning environment. *J Construct Eng Manage* 146:04020080
- Xie S, Girshick R, Dollar P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: *2017 IEEE conference on computer vision and pattern recognition (CVPR)*
- Zhu J, Xu Y, Ye Z, Hoegner L, Stilla U (2021) Fusion of urban 3D point clouds with thermal attributes using MLS data and TIR image sequences. *Infrared Phys Technol* 113:103622

Development of a Geo to BIM Converter: CityJSON Importer Plugin for Autodesk Revit



Amir Hakim , Jasper van der Vaart, Ken Arroyo Ohori ,
and Jantien Stoter 

Abstract The integration of 3D city models and Building Information Models (BIM) in the context of GeoBIM has gained significant attention from both academia and industry. Harmonizing the distinct characteristics and goals of these models is crucial for successful integration. In this paper, we present the development of a plugin for Autodesk Revit, a popular BIM platform, which allows for the incorporation of 3D Geo-data encoded in CityJSON. The plugin, published as open source, enables the generation of individual geometries with associated city model attributes as parameters, facilitating analysing the impact of new or changed buildings (modelled in BIM) on the environment (captured in geo-data). Challenges addressed during development include georeferencing, data format import, handling different geometry approaches, hierarchy of attributes, code optimization, user-friendliness, and enhanced visualization. The plugin contributes to the seamless integration of geo- and BIM data, enhancing interoperability and supporting informed decision-making in the Architecture, Engineering, and Construction and urban domains.

Keywords Building information modelling · Georeferencing · Interoperability · CityJSON · 3D city models

1 Introduction

In recent times, the integration of 3D geoinformation, specifically 3D city models, with Building Information Models (BIM) has emerged as a significant subject of discussion to enhance urban applications. This convergence, commonly referred to as GeoBIM, has garnered substantial attention from diverse academic domains such as geoinformation, geomatics, construction, architecture, and urban planning. Furthermore, it has attracted interest from various non-academic entities including

This article was selected based on the results of a double-blind review of an extended abstract.

A. Hakim (✉) · J. van der Vaart · K. A. Ohori · J. Stoter
3D Geoinformation research group, Delft University of Technology, Delft, The Netherlands
e-mail: S.Hakim@tudelft.nl

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
T. H. Kolbe et al. (eds.), *Recent Advances in 3D Geoinformation Science*, Lecture Notes
in Geoinformation and Cartography, https://doi.org/10.1007/978-3-031-43699-4_15

government-affiliated institutions, National Mapping and Cadastral Agencies, and private enterprises (Noardo et al. 2020).

In the domain of GeoBIM, it is evident that 3D city models and BIM exhibit different goals and have therefore distinct characteristics (e.g., level of detail, designed versus measured geometries, and different semantic objects). This implies that in order to successfully integrate these models for any application, it is imperative to establish consistency and alignment in their features. Such harmonization usually entails ensuring that all data conform to the requirements of either the 3D city models or BIM. Alternatively, if a third-party use case is identified, they may adhere to a different set of specifications (Noardo et al. 2020).

Many studies have focused on making BIM models at lower levels of detail available in geo-applications. With the growing interest to take the environment into account in the design and construction phase of a building or infrastructure project, there is also a growing interest to make geo-data, often available as open data in large datasets, available in BIM applications.

This paper presents our solution to make 3D geo-data, encoded in CityJSON, available in a BIM application. CityJSON (Ledoux et al. 2019), a JSON-based encoding (JavaScript Object Notation) of the CityGML data model (version 3.0.0) is an open, standardized format for exchanging and storing digital 3D models of cities and landscapes. CityJSON has been accepted by the Open Geospatial Consortium (OGC) as a community standard. CityJSON establishes methods for describing prevalent 3D features and objects encountered in urban environments, including buildings, roads, rivers, bridges, vegetation, and city furniture, along with their interrelationships. A CityJSON file encompasses both the geometric and semantic information of city features within a specified area. It also introduces standard levels of detail (LoDs) to enable the representation of objects at various resolutions, tailored to diverse applications and objectives (Ledoux et al. 2019).

The primary objective of CityJSON is to provide an alternative to the GML encoding of CityGML, which often proves verbose, complex, and challenging to work with. CityJSON strives for user-friendliness, facilitating the reading and creation of data sets. Its design prioritizes programmers, allowing for the rapid development of supporting tools and APIs. Additionally, CityJSON emphasizes compactness, typically achieving a 6x compression of publicly available CityGML files. Moreover, it aligns with the requirements of web and mobile development. A CityJSON object, representing a city, adopts a flattened structure, where the CityGML hierarchy is simplified, including only the essential city objects that are the lowest level of the hierarchy. This approach significantly streamlines the storage of city models without sacrificing information.

For this research, we developed a plugin for Autodesk Revit as the BIM platform for incorporating 3D geo-data encoded in CityJSON. This decision was made since Autodesk Revit is the main commercial software in the field of BIM. The plugin that we developed is published as open source (Hakim 2023) and freely available from the Autodesk app store (Hakim 2023) (Figs. 1 and 2).

There are already other solutions available to import geo-data in BIM software. Existing Geo to BIM solutions for Autodesk Revit generate city models as a single

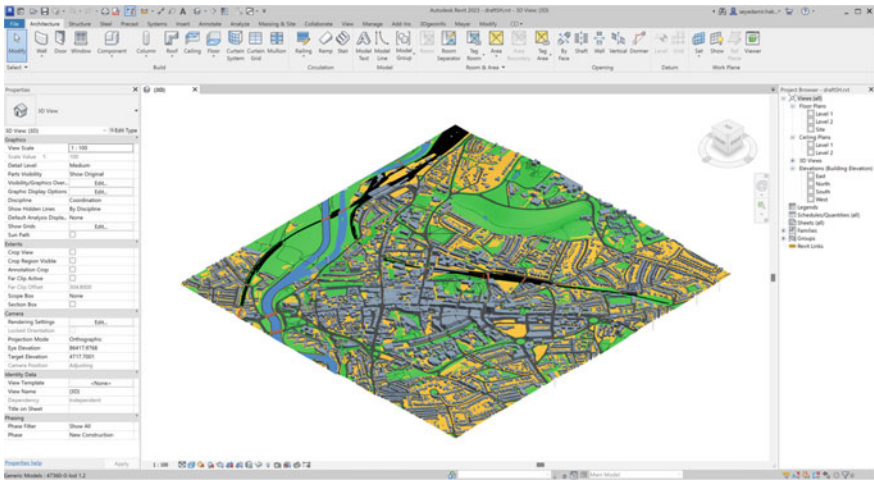


Fig. 1 3D City Model of Southampton, England created via CityJSON importer plugin

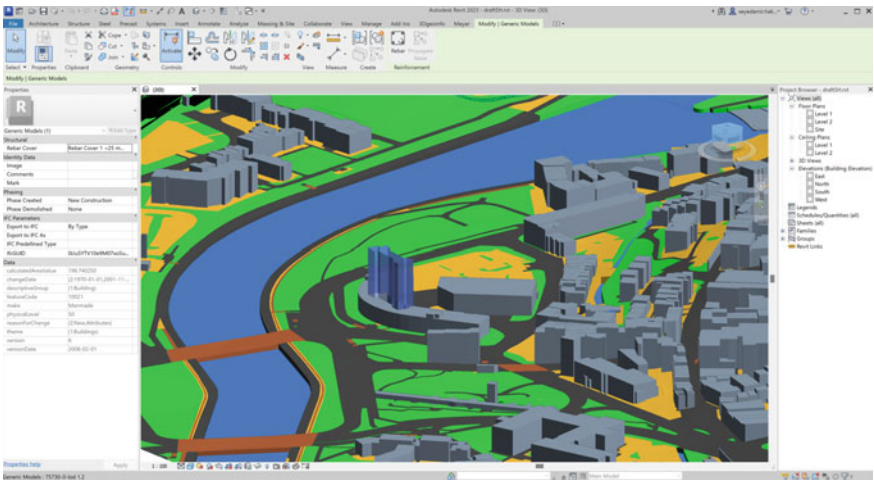


Fig. 2 Visualization and selected geometry information from Southampton City model in Autodesk Revit Software

object represented by a shell in BIM environment. In contrast, our plugin generates each geometry individually and associates city model attributes as parameters to the resulting models. By creating individual geometries, the plugin allows for a more detailed representation of the city models within the BIM environment. This granularity enables more precise analysis and evaluation of how each specific element in the BIM model interacts with the surrounding environment. In addition, it allows for easier isolation and manipulation during analysis. Users can focus on specific elements, make adjustments, and explore various scenarios, fostering a more flexible and detailed examination of the BIM model's influence on the surrounding context.

The plugin's advancements in directly incorporating 3D Geo-data encoded in CityJSON into Autodesk Revit, rather than converting it to IFC for import offers a more streamlined workflow by eliminating the need for an intermediate format conversion. This direct integration saves time and reduces complexities associated with data transformation and data loss of workflows that can read IFC files but that cannot implement customised choices. In addition, the plugin offers a specialized solution tailored explicitly for Autodesk Revit, optimizing the integration process and ensuring compatibility with Revit's features and capabilities. This targeted approach provides a more efficient and effective solution for Revit users compared to the more generic IFC-based method.

Based on a CityJSON file's coordinate referencing system (CRS) and metadata, the plugin reprojects and translates imported data from CityJSON file for implementation within the Autodesk Revit environment.

This paper starts with a description of related work. Then the challenges that needed to be addressed for the development of the plugin are described. We finalise the paper with conclusions and future work.

2 Related Works

The integration of 3D geoinformation with Building Information Models (BIM) in the context of GeoBIM has been a subject of considerable research and development. Numerous studies have explored different approaches to achieve seamless interoperability between these distinct models. In this section, we review some of the relevant related work in the field of GeoBIM and data integration, upon which our plugin is built.

2.1 GeoBIM Integration Tools Evaluation

Noardo et al. (2020) conducted a benchmark study on tools for GeoBIM integration, focusing on IFC (Industry Foundation Classes) georeferencing and conversions. The study evaluated various tools for importing 3D city models into BIM platforms, including methods for georeferencing and transforming spatial data. The authors highlighted the importance of harmonizing different data models and discussed the challenges and opportunities in achieving efficient GeoBIM integration.

2.2 Georeferencing Techniques in BIM

Jaud et al. (2020) investigated georeferencing techniques in the context of building information modeling. The study explored different methods for linking BIM models to their corresponding real-world coordinates and discussed the advantages and limitations of each approach. The authors emphasized the significance of georefer-

encing in larger-scale projects involving infrastructure and the potential benefits for construction and urban planning applications. Another study by Jaud et al. (2022) further examined the georeferencing of IFC models. The authors proposed an approach to meet the requirements of both infrastructure and building industries, enabling the placement of BIM models within a digital version of the site's encompassing environment. The study discussed the challenges in handling geospatial data within BIM platforms and provided recommendations for effective georeferencing.

2.3 GeoBIM Project Experience

Arroyo Ogori et al. (2018) presented a practical experience of processing BIM and GIS models in a GeoBIM project in the Netherlands. The study discussed the challenges and opportunities for integrating geo-data with BIM models and showcased the development of a GeoBIM plugin for a popular BIM software. The authors demonstrated how the plugin allowed for efficient manipulation and utilization of imported geospatial information within the BIM environment.

Deng et al. (2016) developed an instance-based method to map IFC from BIM and CityGML from GIS. They used the Semantic City Model, a reference ontology, to capture relevant information during mapping, ensuring accurate data exchange. The study also harmonized different levels of detail (LoDs) in CityGML for comprehensive mapping. The results demonstrated automatic data mapping between IFC and CityGML in various LoDs, enhancing interoperability.

2.4 Open-Source Plugin for CityJSON Support

Vitalis et al. (2020) developed an open-source plugin to support CityJSON in QGIS, a popular open-source GIS software. The plugin enabled the import and visualization of 3D city models encoded in CityJSON format within the QGIS environment. The study highlighted the importance of open data formats for seamless data exchange between platforms as required for GeoBIM workflows.

Vroegindeweij (2021) created an open-source Dynamo package which is a visual programming platform to facilitate the import of CityJSON objects into Autodesk Revit. His approach imports and visualizes geometries in each 3D city model file as a whole shell within the BIM environment instead of handling individual objects available in the 3D city model. In addition, basic knowledge of visual programming is needed for using the package.

The existing related works have emphasized the importance of seamless integration between 3D geoinformation and BIM models. Researchers have explored various techniques for georeferencing, data transformation, and data exchange between these models. Our work builds upon these studies by developing a plugin for Autodesk Revit that allows for the incorporation of 3D Geo-data encoded in CityJSON,

enabling the analysis of the impact of the BIM model on the environment and enhancing interoperability between the two domains.

3 Challenges that Needed to Be Addressed

This section describes the challenges that needed to be addressed in the development of the plugin to meaningfully import CityJSON data into Revit. These challenges are not limited to Revit and represent general challenges for making geo-data available for BIM applications.

3.1 Georeferencing

In a general context, the process of georeferencing an object encompasses the transformation of its geometric coordinates in a manner that guarantees precise placement at the correct geographical location on a map, while also achieving congruence with other features existing within the map's framework (Snyder 1987).

The geometric context of a BIM model is generally regarded as a localized, three-dimensional Euclidean space, wherein objects situated on the construction site are represented utilizing a Cartesian coordinate system with an origin in the project site and axes usually aligned with the features in the model (Jaud et al. 2020). In light of recent research developments, there have been notable updates to open BIM that have enabled the representation of models using a projected coordinate referencing system (CRS). This advancement is particularly advantageous for larger-scale projects, such as those involving infrastructure (Jaud et al. 2022).

Derived from Computer-Aided Design (CAD), BIM models often utilize local coordinate systems that are software-specific, leading to an inaccurate or entirely absent georeferencing information associated with these models when delivered. Properly georeferencing a BIM file makes it possible to link the (local) coordinates inside a BIM model with their corresponding real-world coordinates, and thus to place the model of a single building or construction within a digital version of the site's encompassing environment (Arroyo Otori et al. 2018).

Autodesk Revit offers multiple methods for establishing the geographical context of a model, including the internal origin, project base point, and survey point. Users employ various approaches to link their BIM models using these points (see Fig. 3).

The plugin specifically addresses the internal origin within the BIM software, granting users the option to either update or retain the Revit site location (internal origin) based on the CityJSON file location. If the user decides to update the internal origin of the project, the plugin also provides the capability to preserve the project base point and survey point (see Fig. 4).

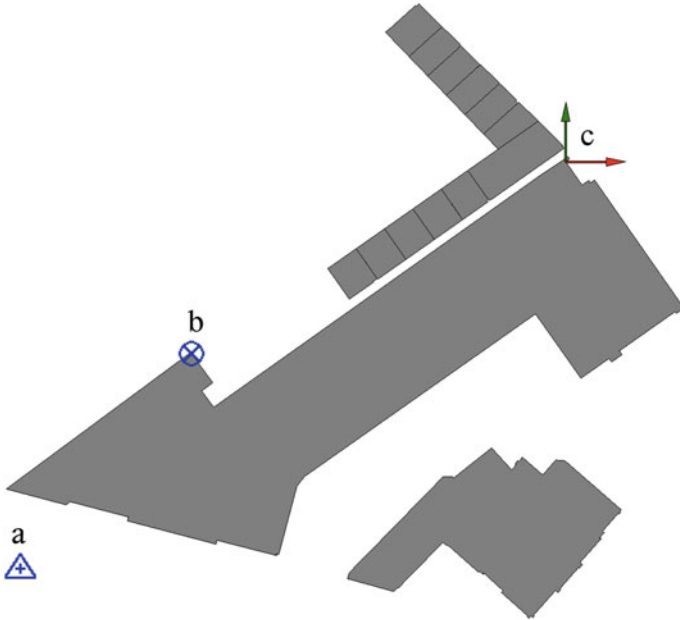


Fig. 3 Different origin points in Autodesk Revit environment: survey point (a), project base point (b), and internal origin point (c)

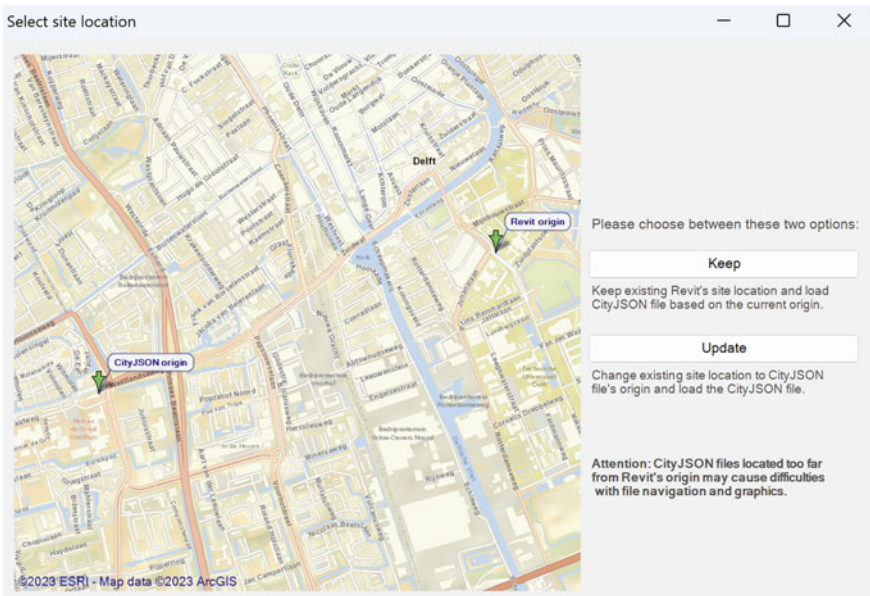


Fig. 4 CityJSON importer plugin's keep/update site location feature

3.2 Data Format

CityJSON's adoption of the JSON format offers advantages in terms of interoperability and data exchange. JSON is a lightweight, human-readable, and language-independent data interchange format, making it suitable for seamless communication between different systems and applications. Its textual representation enables easy comprehension and ensures flexibility in data handling.

The integration of CityJSON data with the Revit environment through .NET programming languages like C# and VB is an efficient approach that enables seamless manipulation and utilization of geo-data within architectural and construction workflows. These languages offer robust support for data manipulation, object-oriented programming, and efficient memory management, making them well-suited for handling complex geo-data and performing sophisticated calculations.

The integration process begins with reading and parsing the CityJSON data using established JSON libraries within C# (Newtonsoft.Json). This parsing allows for extracting and representing the relevant information from the CityJSON files, facilitating further analysis and processing.

Subsequently, the plugin applies various calculations and transformations to the processed geo-data. These operations include geometric manipulations like translation, rotation, or scaling, ensuring precise alignment of the data with the coordinate systems and other location references in the Revit environment.

To accomplish the integration with Revit, the RevitAPI is harnessed through C# to access the software's extensive object model and features. The RevitAPI provides the interface to interact with Revit's model and allows for the creation, modification, and manipulation of architectural elements. This process adheres to the principles of computer science, such as Abstraction, Modularity, Encapsulation, Data Structures, Algorithms, Object-Oriented Programming, Software Design Patterns, and Error Handling. This ensures consistent data flow and maintains the integrity of the Revit model.

3.3 3D Geometry Representation

The geometry structure in Autodesk Revit and CityJSON varies. In CityJSON, points are represented as vertices within the file, while each surface is formed by a set of connected points. Multiple surfaces can be combined to create multi-surfaces and solids. In contrast, direct shapes in Revit possess distinct surface or volume properties, which are based on multiple surfaces generated by sets of XYZ points. These differences in geometry representation between Revit and CityJSON highlight the varying approaches and properties associated with the two formats originating from two different domains that needed to be bridged.

In regard to representing 3D geometry objects in a BIM environment, three main approaches are commonly utilized and widely employed (see Fig. 5):

3.3.1 Boundary Representation (B-Rep)

Boundary Representation, also known as B-Rep or solid modeling, is a widely adopted technique for representing 3D objects. B-Rep describes objects as collections of surfaces, edges, and vertices, providing a detailed depiction of their boundaries. This method enables precise definition and manipulation of the object's geometry. B-Rep has found extensive use in various applications, including computer-aided design (CAD) systems and geometric modeling (Hoffmann 1993).

3.3.2 Sweep Solid (SS)

Sweep Solid represents another approach to 3D geometry representation. In this method, a 2D profile or cross-section is swept along a predefined path, resulting in the creation of a 3D solid. The resulting object maintains the shape and characteristics of the original cross-section as it moves along the specified path. SS proves particularly beneficial for generating objects with rotational or linear symmetries and finds applications in architectural modeling, industrial design, and computer graphics (Vandenbrande 2001).

3.3.3 Constructive Solid Geometry (CSG)

Constructive Solid Geometry, or CSG, constitutes a modeling technique that involves combining simple geometric primitives through set operations such as union, intersection, and difference. Primitives like cubes, cylinders, and spheres serve as fundamental building blocks for creating complex objects through Boolean operations. CSG offers a concise representation of objects and facilitates the construction of intricate shapes by combining simpler elements. This approach has been widely employed in computer graphics, virtual reality, and solid modeling applications (Requicha and Rossignac 1985).

As CityJSON utilizes the Boundary Representation (BRep) strategy for representing 3D geometry objects, we have opted to utilize the BRep to BRep approach, specifically involving the conversion of boundary representation (BRep) models from CityJSON format to BRep models in Autodesk Revit. This chosen methodology allows for the effective transfer of geometric information and enhances interoperability between different software platforms for 3D modeling and building information management. Additionally, seamless representation of 3D objects is ensured during the conversion process, minimizing inconsistencies or inaccuracies.

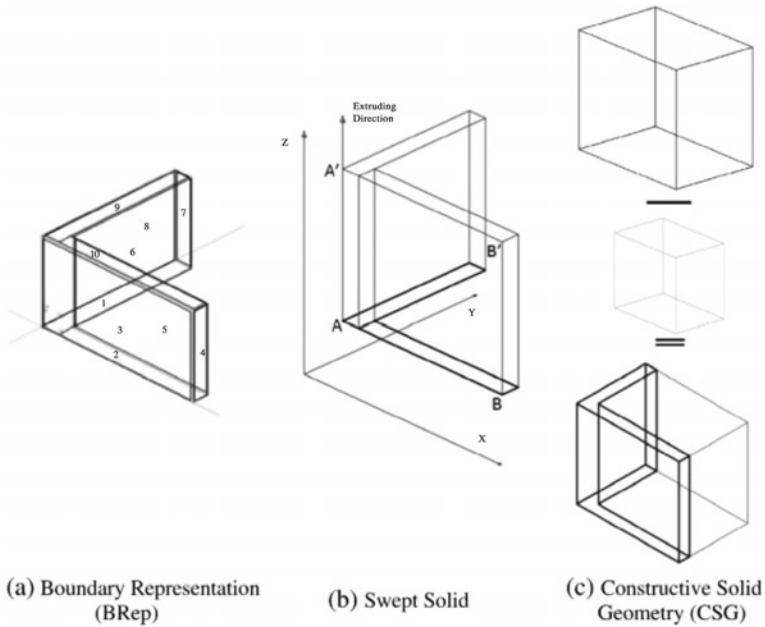


Fig. 5 BRep, swept solid and CSG (Deng et al. 2016)

3.4 Hierarchy of Attributes

The challenge of dealing with the hierarchy of attributes and their appropriate assignment is a significant aspect in the integration of CityJSON data into the Revit environment. CityJSON's hierarchical data model, with attribute inheritance and propagation, aims to optimize storage and processing efficiency. However, Revit's lack of inherent support for such hierarchical inheritance necessitates a carefully designed solution to ensure accurate representation of the hierarchical structure within the BIM environment.

In the context of related studies, the mapping of CityJSON's hierarchical data model has been recognized as a crucial undertaking, emphasizing the importance of addressing this issue to achieve a successful integration (Vitalis et al. 2020). Scientifically, the mapping process requires a systematic approach that considers the relationships and dependencies between attributes at different levels of the hierarchy.

To address the disparity between CityJSON's hierarchical data model and Revit's attribute assignment approach, a rigorous solution was developed. This solution entails individual assignment of relevant attributes to each geometry within the BIM environment. This process involves dissecting the hierarchical structure of CityJSON and mapping attributes to corresponding elements in the Revit model.

In the mapping process, advanced algorithms and data structures may be employed to efficiently handle the complexity of attribute propagation within the CityJSON

hierarchy. The analysis requires traversing the hierarchical data structure and identifying the relevant attributes to be assigned at each level. Techniques such as depth-first or breadth-first searches can be utilized to navigate the hierarchy and propagate the attributes accurately.

Additionally, the solution may require the development of custom data models or data mappings to reconcile the differences in attribute representations between CityJSON and Revit. By creating a specific mapping schema, scientists and engineers can systematically and uniformly link the attributes in CityJSON to the corresponding attributes in Revit, ensuring that no information is lost or misinterpreted during the integration process.

Furthermore, data validation and consistency checks are crucial steps to verify the accuracy and integrity of the attribute assignment. By conducting comprehensive validation tests, potential discrepancies or errors in the mapping can be detected and rectified, ensuring that the hierarchical structure is faithfully represented within the BIM environment.

The approach to handling attribute inheritance and assignment ensures that the integrated CityJSON data within the Revit environment retains its hierarchical structure and associated attributes accurately. This enables architects, engineers, and urban planners to leverage the benefits of both CityJSON's optimized data storage and Revit's powerful BIM functionalities. The developed solution fosters a seamless and efficient workflow, promoting collaboration and data exchange across different platforms and stakeholders in the architectural and construction domains.

3.5 Performance

During the software testing phase, the production of certain city models experienced delays for two primary reasons. Firstly, the transition between JSON information, C# libraries, and the Revit API posed challenges that required careful handling and integration. Secondly, some city models were significantly large in size, containing substantial amounts of data and information. To address these issues, the code underwent multiple reviews and optimizations to improve efficiency and streamline the processing of large city models.

Revit file sizes (for local, network, or cloud-based models), including linked Revit files, can become very large, up to 2 Gigabytes or more. When working with large-sized Revit files, several factors can significantly impact performance. These include file size, model complexity, hardware specifications, and network speed. These files with extensive model elements, such as high-detail families, numerous components, and intricate geometries, can lead to slow loading times, degraded system responsiveness, and an increased likelihood of system crashes. Creating a city model within Revit can lead to generation of large-sized files and entails performance implications that necessitate consideration.

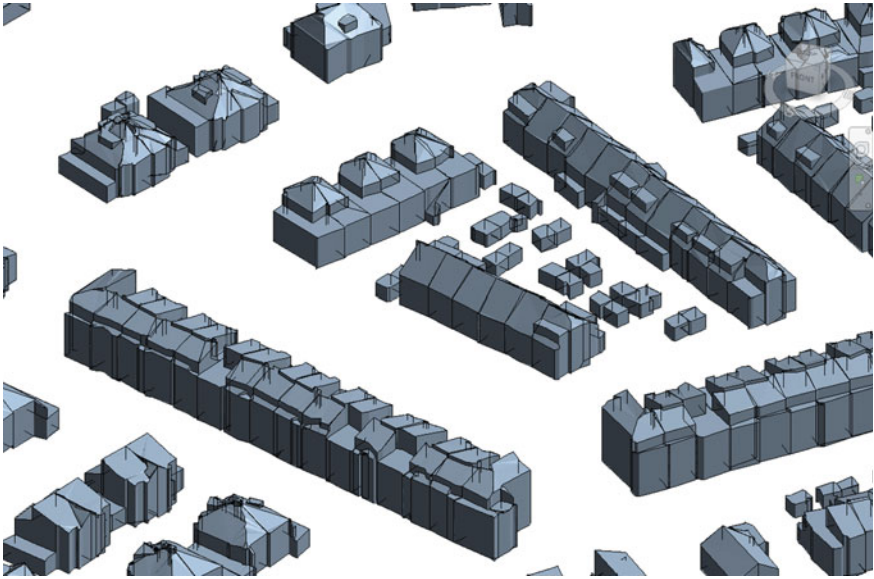


Fig. 6 Graphic glitches and distortions in an Autodesk Revit project with geometries located far from the origin

Geometries situated far from the startup location can trigger various graphic rendering artifacts in Revit. As the distance increases, the precision of rendering and representation diminishes, leading to visual inconsistencies, aliasing, and texture distortion. The software’s display algorithms may struggle to accurately portray intricate details of remote geometries, resulting in jagged edges, blurred textures, and misaligned elements, thereby compromising the visual quality and realism of the model (see Fig. 6).

To effectively address the graphic rendering artifacts experienced with geometries far from the startup location in Autodesk Revit, the plugin offers the “update site” option, as mentioned in the Georeferencing section. This adjustment significantly improves the precision of rendering and representation, thereby mitigating visual inconsistencies, aliasing, and texture distortion commonly associated with remote geometries.

3.6 Friendly for BIM Users

During the publication and deployment of the app on the Autodesk app store, the plugin was required to comply with all possible users’ errors. The design of the plugin focused on handling various scenarios such as opening incompatible file formats, dealing with corrupted files, and seamlessly opening existing files. Moreover, the user

interface was designed with the intention of being user-friendly and easily accessible for all BIM users. Additionally, the results obtained from the plugin automatically trigger the zooming of the 3D view for enhanced visualization and navigation through the project.

The plugin undergoes regular updates and redevelopment, driven by valuable feedback from BIM users. This commitment to continuous improvement ensures that the plugin stays up-to-date with the evolving needs and preferences of the BIM user community.

3.7 Enhanced Visualisation

As a BIM software, Revit can establish connections with other software applications to facilitate collaboration. The development team has addressed this aspect during the development process. For instance, they have incorporated the capability to assign distinct materials to each class of CityJSON objects within the BIM environment, allowing for enhanced visualization purposes.

4 Conclusion

In this paper, we have presented the development of a Geo to BIM converter in the form of a CityJSON importer plugin for Autodesk Revit. The integration of 3D city models and BIM in the context of GeoBIM has garnered significant attention from academia and industry. Harmonizing the distinct characteristics and goals of these models is essential for their successful integration, and our plugin aims to bridge the gap between these two domains.

The plugin allows for the incorporation of 3D Geo-data encoded in CityJSON into Autodesk Revit, a popular BIM platform. By generating individual geometries with associated city model attributes as parameters, the plugin enables users to analyse the impact of the BIM model on the environment during the design phase. This approach fosters a seamless workflow, facilitating the handling of GeoBIM information within a unified environment and enhancing interoperability between geospatial and BIM data. The plugin's direct incorporation of CityJSON data into Autodesk Revit offers a streamlined workflow, preserves attributes, enhances analysis, improves GeoBIM interoperability, and provides a dedicated Revit solution, surpassing the method of converting CityJSON to IFC for importation into Revit.

During the development process, several challenges were addressed, including georeferencing, data format import, handling different geometry approaches, hierarchy of attributes, code optimization, user-friendliness, and enhanced visualization. Each challenge was tackled systematically, employing advanced algorithms, data structures, and software engineering principles to ensure the robustness and efficiency of the plugin.

5 Future Development

The focus of this application revolves around generating generic models within a BIM environment based on CityJSON geometries. For future advancements, it would be highly beneficial to generate geometries using their semantic attributes specific to Autodesk Revit. For example, the topography of a site could be represented as a Toposurface or Toposolid within Autodesk Revit. While Toposolid generates a solid 3D volume to portray the topography, Toposurface generates a 2D surface that faithfully captures the terrain's contours and shape. Both options can be generated by utilizing points or imported data. By ensuring semantically correct generation of topography within the Revit environment, designers and engineers are empowered to seamlessly incorporate and analyze site topography in their BIM models, conduct volume calculations, and make informed design decisions.

In the existing version, the app performs a reprojection of CityJSON points to the EPSG 4326 coordinate reference system (CRS) as the base reference. Although directly altering the coordinate system within Revit is not possible, it is feasible to extract the CRS information from an imported CAD file. As a further development, the application will incorporate the functionality to reproject CityJSON geometries to the acquired CRS from the imported CAD file.

The CityJSON importer plugin for Autodesk Revit contributes to the seamless integration of geospatial and BIM data, supporting informed decision-making in the Architecture, Engineering, and Construction (AEC) and urban domains. By addressing the challenges and requirements of GeoBIM integration in a rigorous manner, the plugin opens new avenues for collaboration, visualization, and analysis, fostering advancements in the field of 3D city modeling and building information management.

Acknowledgements This project has received funding from the European Union's Horizon Europe programme under Grant Agreement No.101058559 (CHEK: Change toolkit for digital building permit).

References

- Arroyo Ohori K, Diakité A, Krijnen T, Ledoux H, Stoter J (2018) Processing BIM and GIS models in practice: experiences and recommendations from a GeoBIM project in the Netherlands. *ISPRS International Journal of Geo-Information* 7(8):311
- Deng Y, Cheng JC, Anumba C (2016) Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison. *Automation in Construction* 67:1–21
- Hakim, A. (2023). CityJSON Importer for Autodesk Revit, <https://apps.autodesk.com/RVT/en/Detail/Index?id=7787623024858844510>, Accessed: 25-07-2023
- Hakim, A. (2023). Github repository for cityjsonToRevit, <https://github.com/tudelft3d/cityjsonToRevit>, Accessed: 20-06-2023
- Hoffmann, C. M. (1993). Geometric and solid modeling: An introduction. Morgan Kaufmann
- Jaud Š, Clemen C, Muhić S, Borrmann A (2022) GEOREFERENCING IN IFC: MEETING THE REQUIREMENTS OF INFRASTRUCTURE AND BUILDING INDUSTRIES. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, p 10

- Jaud Š, Donaubaauer A, Heunecke O, Borrmann A (2020) Georeferencing in the context of building information modelling. *Automation in Construction* 118:103211
- Ledoux H, Arroyo Ogori K, Kumar K, Dukai B, Labetski A, Vitalis S (2019) CityJSON: A compact and easy-to-use encoding of the CityGML data model. *Open Geospatial Data, Software and Standards* 4(1):1–12
- Noardo F, Harrie L, Arroyo Ogori K, Biljecki F, Ellul C, Krijnen T, Stoter J (2020) Tools for BIM-GIS integration (IFC georeferencing and conversions): Results from the GeoBIM benchmark 2019. *ISPRS international journal of geo-information* 9(9):502
- Requicha AAG, Rossignac JR (1985) Constructive solid geometry: Representation and computation. *ACM Computing Surveys (CSUR)* 17(4):335–375
- Snyder, J. P. (1987). *Map projections—A working manual* (Vol. 1395). US Government Printing Office
- Vandenbrande, J., & Vos, R. (2001). Sweep solid generation: Algorithms for the non-manifold case. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (pp. 61-70). ACM Press
- Vitalis S, Arroyo Ogori K, Stoter J (2020) CityJSON in QGIS: Development of an open-source plugin. *Transactions in GIS* 24(5):1147–1164
- Vroegindeweij, M. (2021). GIS2BIM for Dynamo 2.x, <https://github.com/DutchSailor/GIS2BIM/wiki/Workflows-2.x-Nederland>, Accessed: 10-02-2023

3D Point Cloud Processing and Analysis

Efficient In-Memory Point Cloud Query Processing



Balthasar Teuscher, Oliver Geißendörfer, Xuanshu Luo, Hao Li, Katharina Anders, Christoph Holst, and Martin Werner

Abstract Point cloud data acquired via laser scanning or stereo matching of photogrammetry imagery has become an emerging and vital data source in an increasing research and application field. However, point cloud processing can be highly challenging due to an ever-increasing amount of points and the demand for handling the data in near real-time. In this paper, we propose an efficient in-memory point cloud processing solution and implementation demonstrating that the inherent technical identity of the memory location of a point (e.g., a memory pointer) is both sufficient and elegant to avoid gridding as long as the point cloud fits into the main memory of the computing system. We evaluate the performance and scalability of the system on three benchmark point cloud datasets (e.g., ETH 3D Point Cloud Dataset, Oakland 3D Point Cloud Dataset, and Kijkduin 4D Point Cloud Dataset) w.r.t different point cloud query patterns like k nearest neighbors, eigenvalue-based geometric feature extraction, and spatio-temporal filtering. Preliminary experiments show very promising results in facilitating faster and more efficient point cloud processing in many

This article was selected based on the results of a double-blind review of an extended abstract.

B. Teuscher · O. Geißendörfer · X. Luo · H. Li · K. Anders · C. Holst · M. Werner (✉)
Department Aerospace and Geodesy, Technical University of Munich, School of Engineering and Design, Munich, Germany
e-mail: martin.werner@tum.de

B. Teuscher
e-mail: balthasar.teuscher@tum.de

O. Geißendörfer
e-mail: o.geissendoerfer@tum.de

X. Luo
e-mail: xuanshu.luo@tum.de

H. Li
e-mail: hao_bgd.li@tum.de

K. Anders
e-mail: k.anders@tum.de

C. Holst
e-mail: christoph.holst@tum.de

potential aspects. We hope the insights shared in the paper will substantially impact broader point cloud processing research as the approach helps to avoid memory amplifications.

Keywords 3d point cloud · In-memory processing · Spatio-temporal · Nearest neighbor

1 Introduction

Point clouds significantly differ from other geospatial data in terms of their computational nature rendering efficient processing of point clouds in traditional geospatial infrastructures such as relational database management systems (RDBMS) or distributed key-value stores complex (Eitel et al. 2016; Lokugam Hewage et al. 2022; Pajić et al. 2018). The reason is rooted in the imbalance of data and metadata when a huge number of comparably small objects is to be organized. In relational databases, for example, a natural mapping would let each point occupy a row in a table inducing at least a primary key on top of the points. Otherwise, relational mapping of attributes would not be easily possible. For big data systems, key-value patterns dominate the organization of processing in line with MapReduce (Dean and Ghemawat 2008; Lee et al. 2012) as well as the theory of (almost) infinitely scaling systems. In both cases, managing the identities (primary keys in databases, keys in a key-value system) is similarly complex to manage the data itself (e.g., points represented as tuples of floating point values).

One proposal often seen in this context is spatial gridding or voxelization. If we would go ahead and cut the point cloud into grid cells, we can organize the data—both in distributed key-value stores or in relational data management systems—in terms of these blocks or 3D tiles. However, this implies that queries are first defined only up to these artificial data boundaries and refinement steps are needed if, for example, a query’s data demand extends beyond the current 3D tile.

Furthermore, this approach has limitations when it comes to varying point densities, which is, however, a very common aspect for terrestrial point clouds where density degrades with the distance from the sensor to the object and objects (after, for example, ground removal) are scattered in space. For focal point cloud analysis (e.g., point cloud analysis where a bound on the distance of relevant other points is given), this has been done by extending the grid to a grid overlapping neighbors (Brenner 2016).

The challenge of three-dimensional point cloud data and their volumes is amplified when the time dimension is added through repeated point cloud acquisitions. Neighborhood queries are then extended into the temporal domain (Anders et al. 2020; Kromer et al. 2015), for example, for spatiotemporal point cloud filtering.

In this paper, we propose and demonstrate that the inherent technical identity of the memory location of a point (e.g., a memory pointer) is both sufficient and elegant to avoid gridding as long as the point cloud fits into the main memory of the

computing system. With modern servers with commonly more than 1TB of main memory, this requirement is, however, not a severe limitation for many use cases. We demonstrate an open-source implementation of complex point cloud queries following this approach and give performance insights on selected queries. We test the feasibility and performance of these queries on multiple benchmark 3D point cloud datasets and share insights into a future deployment for high-performance computing (HPC) based point cloud analysis.

The remainder of the paper is structured as follows: The next Sect. 2 reviews related work on point cloud data management and point cloud processing. In Sect. 3, we derive an architecture for scalable point cloud data management in the Python environment. In Sect. 4, we classify various data modalities and common query patterns in order to derive an architecture for a point cloud processing engine. In Sect. 5, we evaluate scalability and performance on a variety of datasets. Section 6 comments on future potential directions and Sect. 7 concludes the paper.

2 Related Work

Point cloud data (containing x , y , z information) can be acquired via laser scanning and stereo matching of photogrammetry imagery, and has become an emerging and vital data in increasing aspects of real-world applications, ranging from forestry investigation (Eitel et al. 2016; Polewski et al. 2018), 3D city modeling (Huang et al. 2022; Xu et al. 2019), precision agriculture (Weis et al. 2008), to autonomous drones/cars (Yue et al. 2018). Especially, point cloud acquisition using the Light Detection And Ranging (LiDAR) method has been investigated in diverse levels of scale and accuracy, from satellite-based (e.g., the Geoscience Laser Altimeter System (GLAS) on ICESat (Zwally et al. 2002)), regional-scale Airborne Laser Scanning (ALS), to fine-grained Terrestrial Laser Scanning (TLS). The emerging availability of such sensors and data acquisition techniques resulted in a continuing growth in the amount of point cloud data, which poses great challenges for efficient data storing, processing, and management to the conventional spatial computing algorithms designed for 3D point data.

Point cloud processing is a highly challenging task due to an increasing point rate of available LiDAR scanners and the demand for handling the data in near real-time. The processing mainly involves registration to existing point clouds with the ICP (Besl and McKay 1992), RANSAC (Rusu et al. 2009) algorithms and segmentation tasks Qi et al. 2017a, b. Both purposes require features that are computed on images (Chen and Medioni 1991) and point clouds (Bueno et al. 2016) directly. Additionally, tasks are often solved with Deep Learning approaches (Choy et al. 2020; Gojcic et al. 2018; Zhang et al. 2022). Specifically, geometric features based on the mean, variance, number of points, density, and normal vectors (Ioannou et al. 2012) are used to describe local properties. Eigen-based features (Blomley et al. 2014) have gained interest for multiple years since they offer a higher-level specification of the surroundings. All features computed from the local neighborhood increase the infor-

mation content of the respective environment. Therefore, using robust and unique features are utilized for an unambiguous matching (Vetrivel et al. 2018), which is needed to identify corresponding regions.

Computing meaningful features in local neighborhoods is often a challenging piece of work. Structuring a point cloud with an octree (Elseberg et al. 2011; Kammerl et al. 2012) is a common approach used to describe point surroundings. Therein, respective tree leaf nodes summarise the information content of all points within its node boundaries of a pre-defined size. However, points lying close to the border of adjacent leaf nodes will not contribute to a high-level feature at the same time. An alternative is shown in (Gumhold et al. 2001) with building a graph structure to compute a triangulation to find line and edge patterns. Creating a graph enables a faster and more flexible search in a point neighborhood of an unsorted point cloud. Furthermore, using graphs is applicable in point cloud deep learning approaches (Fischer et al. 2021; Xie et al. 2020) as well.

In summary, quite challenging algorithms in terms of very large loops and many neighborhood retrievals are commonly needed to process point clouds.

3 Architecture

The point cloud query engine is designed following a few modern programming paradigms especially related to big geospatial data management and differs in its architecture from other approaches quite a bit. In this section, we highlight architectural choices, how they affect the performance of the system and give hints on future extensions and the path forward for this system.

3.1 Computational Infrastructure

The first and most important decision is to choose a generic programming platform that provides us with a good blend of performance, portability, usability, existing peer-reviewed libraries for geometry and linear algebra, flexibility, simple parallel programming features, and excellent Python interfacing. With these attributes, the choice for the core language has been made to be modern C++ as it provides us with type-generic programming, excellent OGC geometry implementations using `boost::geometry`, fast and powerful linear algebra using `Eigen3`, and annotation-based parallel programming in a shared-memory-parallel setting (SMP) based on `OpenMP`.

While not yet implemented, the choice of standard C++ containers throughout the implementation facilitates the use of distributed versions of these containers as well, such as provided by the `DASH` library for partitioned global address space model of computing, where a global address space across a supercomputer is simulated (Fürlinger et al. 2016). In this way, the library might work across large supercomputers coordinated using the Message Passing Interface (MPI).

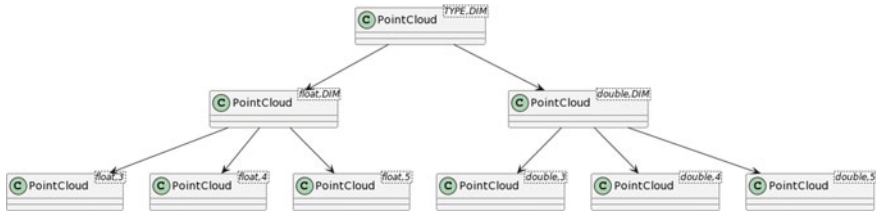


Fig. 1 Generic class architecture

All functionality has been implemented as a single C++ header file relating only to the boost and Eigen3 libraries. The implementation itself is almost fully generic as depicted in Fig. 1.

The main pointcloud class is implemented for all possible numeric types and all dimensions in the same code base. Template metaprogramming is used to implement functions for different dimensionalities, while the system is flexible enough to partially specialize certain dimensions. For example, a spatiotemporal algorithm can only be run on point clouds with at least four dimensions. Of course, such functions cannot be implemented as member functions of the generic PointCloud class, as any attempt to access the fourth coordinate in a 3D instantiation of a pointcloud will lead to a compilation error. But why do we go this way, if it is limited in this way? The reason is that in contrast to most other libraries, the type and dimension of the point cloud are constant at compile time and all loops or branches based on dimension (e.g., a loop going over the x, y, z coordinate in a 3D point cloud, but over x, y, z, t in 4D) can be easily unrolled by the compiler for unprecedented performance gains.

As said, we still want to implement special functions for less generic point clouds. In order to do so, we use free functions that get references to the point cloud. As we make sure that we request these functions only in a valid context, they will happily compile and at the same time be able to apply dimension and type-based optimizations otherwise not possible in this stage.

3.2 Algorithmic Infrastructure for Geometry and Spatial Indexing

Instead of reinventing the wheel or using non-generic implementations, we rely on the most modern family of peer-reviewed libraries for scientific computation in C++. Boost Geometry provides us with a complete and reasonably correct implementation of almost all OGC simple feature types and operations. It is generic in dimension and coordinate type in itself and we can forward the typespecifications with which the point cloud was instantiated with ease.

In addition, the indexing library of boost geometry provides us with a generic implementation of an R-tree indexing tree and associated query algorithms which

we instantiate with a default R^* balancing algorithm. This is a good approach for dynamic insertion of point especially in the case when many nearest neighbor queries are going to be performed. We initialize the tree using an STR bulk load algorithm, but in many situations, we will want to stream batches of points into the index to avoid the need for large contiguous areas of main memory for holding points while building the index. In this case, the first batch is used to STR bulk load a tree while subsequent points are inserted using the R^* approach of reinserting points on node splits.

This R-tree implementation provides a query interface with an output iterator (e.g., the query is actually a typical tree scan and emits one indexed value at a time) which we adapt to a C++ function call. The query predicate is built as a Boolean combination of contains, covered_by, covers, disjoint, intersects, overlaps, within and two special predicates. The first special predicate is the “satisfies” predicate which can be used to call a custom Boolean predicate during tree traversal that decides on whether points qualify for the result set in a rather arbitrary way and a “nearest” predicate which ensures that only the k nearest neighbors are retrieved. Note, however, that for performance reasons the output of the k nearest neighbors is not sorted by distance as in distance browsing - an aspect to keep in mind when using the system. It is worth noting that this setup has a huge advantage stemming from functional programming capabilities of modern C++ : when we implement the function that takes one result set element after another as a lambda function, we can capture variables from the calling scope. This can be used to, for example, aggregate the mean of all results without ever materializing the result set in memory.

3.3 Algorithmic Infrastructure for Linear Algebra

One of the query patterns we investigated while setting up this system is a quite classical one: from a given definition of the surroundings of a point, one builds up a $N \times 3$ matrix M , extracts the corresponding three eigenvalues $\lambda_0, \lambda_1, \lambda_2$ of the 3×3 self-adjoint matrix $M^t \cdot M$, normalizes them by their sum, orders them and calculates features including, but not limited to:

$$\begin{aligned} \text{linearity} &= (\lambda_0 - \lambda_1)/\lambda_0 \\ \text{planarity} &= (\lambda_1 - \lambda_2)/\lambda_0 \\ \text{scattering} &= \lambda_2/\lambda_0 \\ \text{omnivariance} &= \sqrt[3]{\lambda_0\lambda_1\lambda_2} \\ \text{anisotropy} &= (\lambda_0 - \lambda_2)/\lambda_0 \\ \text{eigen-entropy} &= -\lambda_0 \log_2(\lambda_0) - \lambda_1 \log_2(\lambda_1) - \lambda_2 \log_2(\lambda_2) \\ \text{trace by largest} &= (\lambda_0 + \lambda_1 + \lambda_2)/\lambda_0 \\ \text{change of curvature} &= \lambda_2/(\lambda_0 + \lambda_1 + \lambda_2) \end{aligned}$$

In our implementation, the Eigen3 library is used for eigenvalue extraction by QR decomposition, which could be further optimized by using a direct equation solution (with some loss in numerical precision) for compile-time constant-sized matrices. However, we skip such microoptimizations as the system is - at the moment - memory bound and a wall-clock performance improvement would not be possible.

3.4 *Python Bindings*

As Python is one of the most prominent glue languages, our system relies on pybind11 to efficiently (i.e., in most cases without copying memory or marshalling representations) communicate and share data between our library, numpy, and the user. However, as Python does not provide generic programming support, we instantiate the leaf nodes in Fig. 1 as individual Python classes. For each of these classes, we first bind specific API calls related to the dimension and type, followed by binding a generic API that all pointcloud classes share. This finalizes the solution to the problem that some operations can only be defined for certain dimensions of point clouds, as we will require those functions only from the function implementing the 4D Python binding.

3.5 *Data Handling*

All data in the library is organized as named columns, where the current version supports exactly one point cloud, and a variable number of named 1D columns representing the results of computations.

At this moment, the point information is available twice: once in an array, accessible via index, and once in the R-tree index as the leaf nodes contain the actual data. Fortunately, the array can be dropped if wanted, even more, a pattern of loading a dataset as a sequence of smaller numpy arrays (maybe a few hundred megabytes of data) always clearing the point cloud data structure afterward is possible. This has the effect that no large contiguous memory areas are occupied and the final memory consumption will be the cost of storing the points in the R-tree plus the size of the small numpy array that is used to provide the batches to the C++ implementation. A nice consequence is that the point cloud that you are going to use for queries can be a different one (e.g., a 3D grid if you are interested in a voxel-like computation). Most queries then loop over the given cloud, which can be either the original one or another one and compute features or other computations for each point in this cloud.

4 Data Modalities, Query Patterns and the API

Point clouds have different properties depending on how they are acquired and pre-processed. As we are providing a general implementation framework that is suitable for all point clouds, we select representative point cloud benchmark datasets of different natures for evaluation.

Photogrammetric point clouds are typically very dense 3D point clouds acquired from cameras in a 3D reconstruction setting where multiple images from different viewpoints are combined to a 3D point cloud structure.

Laserscanning point clouds show a comparably higher non-uniformity as many scanners are acquiring images steering the laser ray using a specific pattern. In addition, the spatial or angular resolution of laser scanners is often lower while the range and precision outperform photogrammetric methods.

In many cases, the sensor itself is mounted to a mobile platform and thus generates data along the trajectory of the mobile sensor platform. Methods of the family of Simultaneous Localization and Mapping (SLAM) use this mobile sensor information to jointly estimate the trajectory of the mobile sensor platform and a pointcloud representing the environment. While these approaches help to mitigate single scan weaknesses such as limited resolution and scan patterns, they introduce their own limitations due to strongly depending on the estimated pose of the sensor which is often subject to error accumulation through time, especially from integrating the movement using inertial measurement units.

Lately, neural rendering and especially Neural Radiance Fields (NeRFs) have been invented to generate a dense voxelized model of the environment from a limited number of observations. While the output is actually some average radiance observed for rays crossing a 3D volume, it can naturally be considered a point cloud by representing the volume with its center point. In a similar way, PointNeRF (Xu et al. 2022) directly models neural radiance fields with point clouds as opposed to vectors and we expect this exciting area of research to provide another category of point clouds in the future as occlusions can be modelled much better as opposed to scan-based methods.

Further, point clouds provide a way to represent geometry in a fully local way such that in some settings people sample point clouds from geometric models, including CAD and GIS datasets. Such sampling can reduce the complexity and generalize the sampled objects' geometric representations. For example, a random point on the surface of a mesh is often required in shape similarity computation (Osada et al. Oct 2002).

The general sampling nature of point cloud data and the individual samples being unbounded implies relating and aggregating the samples in order to draw meaningful information. The attributes of a single sample are almost never of immediate interest, but rather the neighborhood, which constitutes a segmentation of objects of interest. Hence range and neighborhood queries are at the core of point cloud analysis. Even more, given the importance of points contributing to the description of a neighborhood decreases with its distance to it (Tobler 1970).

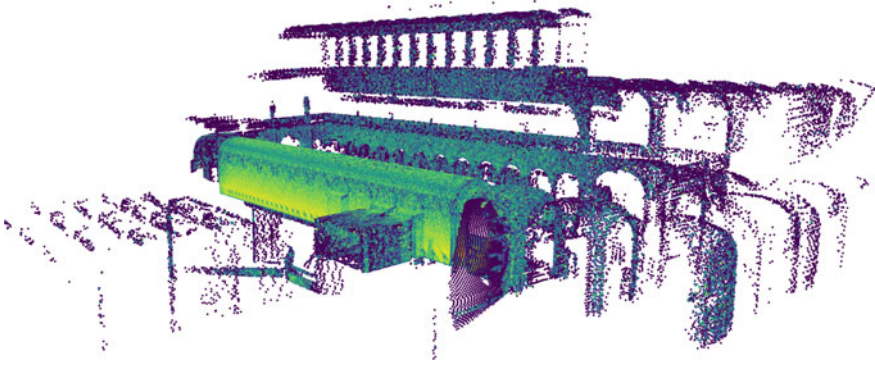


Fig. 2 ETH 3D point cloud dataset

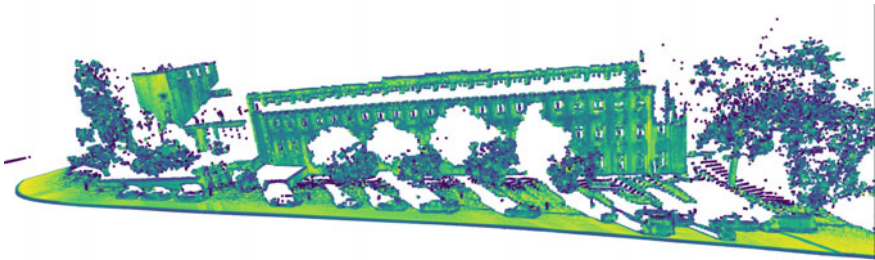


Fig. 3 Oakland 3D point cloud dataset

In this paper, we consider three 3D point cloud benchmark datasets (as shown in Fig. 2, 3, and 4) of different sizes to evaluate the performance of our in-memory point cloud processing library w.r.t the time and efficiency of point cloud query and feature extraction:

ETH 3D Point Cloud Dataset: The first dataset was collected in the main building (Hauptgebäude) of ETH Zurich (Pomerleau et al. 2012), where the laser scanner moved mainly along a straight line in a long hallway. This hallway is located on the second floor and surrounds the central place, which is then open over several floors. Herein, the hallway has a curved ceiling and a wall on one side. On the other side, there are pillars, arches and ramps. Those elements are the main interest of this dataset. ETH 3D point cloud datasets consist of in total 6,810,063 points.

Oakland 3D Point Cloud Dataset: This classic dataset was collected using Navlab11 equipped with side-looking SICK LMS laser scanners and used in push-broom (Munoz et al. 2009). The dataset was collected around the CMU campus in Oakland, Pittsburgh, PA, and contains labeled 3D point cloud derived from LiDAR data collected using a moving platform in an urban environment. Oakland 3D point cloud dataset contains 1,614,625 points.

Kijkduin 4D Point Cloud Dataset: The third dataset was recorded by a laser scanner observing one kilometer of coast at hourly intervals for about six months, in

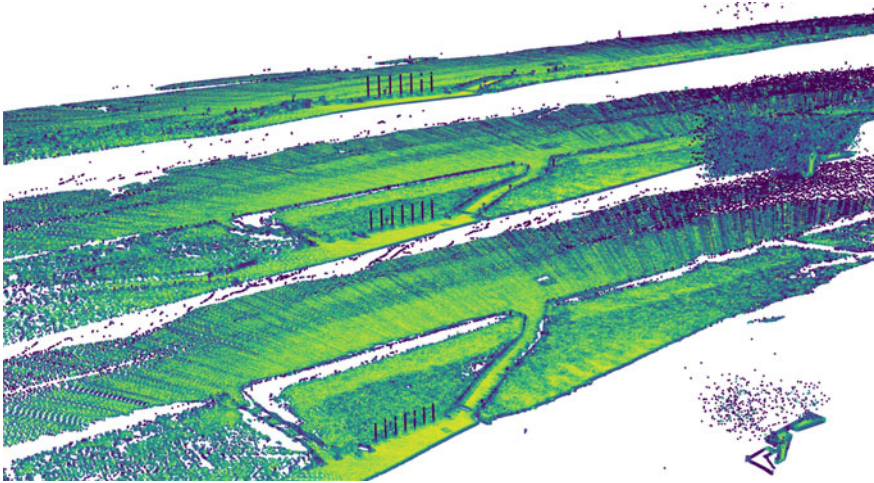
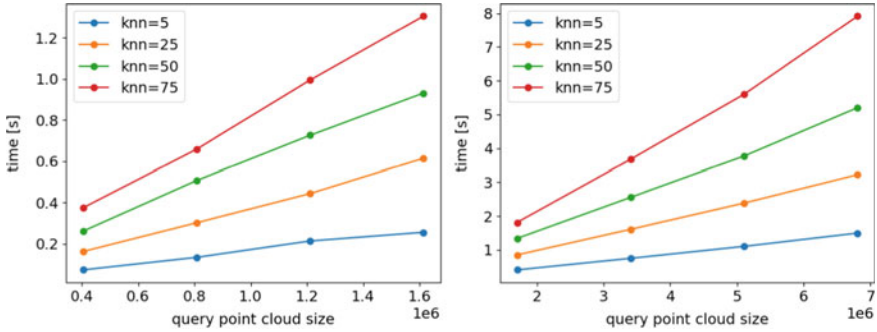


Fig. 4 Kijkduin 4D point cloud dataset with first three timestamps shown from bottom to top

the Kijkduin dune system in the Netherlands (Vos et al. 2022). This dataset includes more than 4,082 consecutive topographic scans of a varying size in the lower millions of points each, therefore, it can be regarded as a unique 4D spatiotemporal point cloud dataset in understanding the morphological behavior of the beach-dune system at hourly to monthly time scales. The total number of points in the dataset across all individual hourly scans adds up to about 6.6 billion spatiotemporal 4D points.

5 Evaluation

The following section gives hints on the performance and scalability of the system in typical scenarios. Specifically, we consider three types of point cloud queries: (1) the extraction of k nearest neighbors; (2) the extraction of eigen features; (3) spatio-temporal filtering of 4D point clouds. Moreover, we test the performance of these queries for smaller datasets on two different computing platforms, namely a business laptop (with an Intel(R) Core(TM) i7-8565U CPU, 16.0 GB DDR4 2400MHz memory and a 512GB PCIe 3.0 M.2 SSD) and a gaming PC workstation (with an Intel(R) Core(TM) i9-10900K CPU, 32.0 GB DDR4 2933MHz memory and 1TB PCIe 3.0 M.2 SSD). The big Kijkduin cloud is processed on a medium performance AMD-based server (AMD EPYC 7702P 64-Core Processor, 1 TB main memory, 500 TB disk array).



(a) Oakland 3D Dataset (1.6 million points) (b) ETH 3D Dataset (6.7 million points)

Fig. 5 Performance of simple kNN queries over the whole dataset

5.1 Extraction of K Nearest Neighbors

As a first query, we consider the raw extraction of k nearest neighbors of any point and perform simple aggregation in terms of finding the maximum distance from the query point on two benchmark datasets, namely the Oakland 3D dataset (with around 1.6 million points) and ETH 3D dataset (with around 6.4 million points). In terms of pseudo-code, the considered algorithm for a parameter k is given as follows:

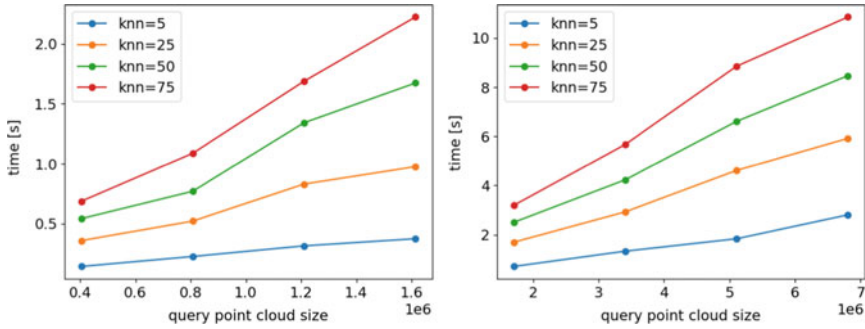
```

1 for p in points:
2   neighbors = knn(p, k)
3   d = max_dist(neighbors, p)
    
```

Figure 5 gives the performance of this query for varying datasets and values for parameter k. As expected, this query has an almost linear scaling behavior with respect to the size of the point cloud on a single machine, and this is observed by the straight lines in the figure. The increasing steepness of the lines is implied by the fact that not only more points are given, but also each neighborhood selection - despite the neighborhoods having the exact same size - traverses a deeper tree.

5.2 Extraction of Geometric Features

Geometric features are computed from a neighborhood. We define the following neighborhoods as they incur different query patterns: (1) kNN: the neighborhood consists of the k nearest neighbors; (2) cube: the neighborhood consists of a n-dimensional cube which might contain infinite coordinates, for example, a 2D box in 3D would be modeled as $(x_{min}, y_{min}, -\infty) \rightarrow (x_{max}, y_{max}, \infty)$; (3) sphere: all points that have a 2D or nD-Euclidean distance smaller than a given radius; (4) cylinder: all points within a cylinder.



(a) Oakland3D 3D Dataset (1.6 million points) (b) ETH 3D Dataset (6.7 million points)

Fig. 6 Performance of calculating 10 different eigenvalue-based features over the whole dataset

However, due to the nature of the R-tree, the query processing of all of these neighborhoods is essentially the same: first, we take the minimum bounding rectangle (MBR) of the query object (box, cylinder, sphere, etc.) and determine all points within this box with a tree traversal. During query processing, we evaluate the more specific geometry through additional “satisfies” constraints. All valid points passing all geometry tests are then collected as the relevant neighborhood. This neighborhood is now shifted to the origin either by computing the centroid of the neighborhood or by directly using the query point that was used to create the neighborhood. All results are combined into an $N \times 3$ matrix M . A self-adjoint 3×3 matrix is generated using from $M^t \cdot M$. As the eigenvalues of selfadjoint matrices are always real numbers, we can extract them, order them, and normalize them to sum up to one. The extraction is currently performed with a QR composition leading to good numerical stability, but faster algorithms would be available for 3×3 self-adjoint matrices. However, as we will see, the algorithms are not compute-bound such that this optimization would not pay off in terms of wall-clock time (but may save energy or provide more compute time in different scenarios).

These eigenvalues are then used together with the various derived geometry features from Sect. 3.3 to compute ten features for each point of the point cloud. As the performance evaluation in Fig. 6 shows, computing these features only adds a small overhead compared to bare kNN queries in Fig. 5.

Table 1 shows the detailed comparison of different query times of our proposed in-memory point cloud processing library on two different computing platforms in order to better understand the efficiency and scalability of the system. Herein, we compare the processing time (in second) of mainly four steps: (1) the initialization, which refer to a data reading operation; (2) the indexing, which create the R-tree for the whole point clouds; (3) the extraction of k nearest neighbors with k=25; (4) the extraction of ten eigen features based on k nearest neighbor results. One can easily identify two key findings from this comparison: first, a general trend of fast

Table 1 Performance of point clouds queries with two benchmark datasets on different computing platforms (i.e., a business laptop running Microsoft Windows and a gaming PC running Linux)

| Query time(s) | Oakland 3D (1.6M pts.) | | ETH 3D (6.2M pts.) | |
|----------------|------------------------|-------------|--------------------|-------------|
| | Laptop | Workstation | Laptop | Workstation |
| Initialization | 0.094 | 0.041 | 0.525 | 0.219 |
| Indexing | 0.483 | 0.226 | 3.327 | 1.174 |
| kNN radius | 3.137 | 0.646 | 30.458 | 3.511 |
| Eigenfeatures | 10.296 | 1.093 | 84.769 | 5.882 |

and efficient in-memory point cloud processing can be observed across different datasets and computing platforms; second, though the extraction of eigen features is still the most time-consuming operations, this becomes much more affordable and with even incremental addition times in a workstation. Note that the laptop was running a Microsoft Windows operating system to model a typical geoscientists mobile computing device while the workstation was running Linux.

5.3 Spatio-Temporal Filtering

The third query pattern is about spatio-temporal filtering. Especially in long-term observations, noise and small perturbations to the observed geometry shall be differentiated from relevant signals representing true change of the geometry.

We implement a special query designed for a spatio-temporal point clouds that does compute for each point of a query point cloud the average Z coordinate of a space-time cylinder with a given spatial radius, unbounded spatial height, and a configurable time window. In this way, a very massive point cloud is analyzed locally in space and time using a query grid of points of interest for which this value shall be computed. This query is also used to highlight the simplicity of extending our framework by experts from the field not firm in modern C++. Therefore, we discuss the function doing the computation right here:

```

1 namespace bg = boost::geometry;
2 template<typename pointcloud>
3 void boxfilter_4d(pointcloud &pcl, std::string prefix,
4                 double radius, double time_radius) {
5     pcl.named_attributes[prefix+“_boxfilter4d”] = std::vector<double>(pcl.cloud.size());
6     auto &a = pcl.named_attributes[prefix + “_boxfilter4d”];
7     auto & cloud = pcl.cloud;
8     #pragma omp parallel for
9     for (size_t i=0; i< cloud.size(); i++) {
10        typename pointcloud::point_type minPoint, maxPoint;
11        bg::set<0>(minPoint, bg::get<0>(cloud[i]) - radius);
12        bg::set<1>(minPoint, bg::get<1>(cloud[i]) - radius);
13        bg::set<2>(minPoint, -std::numeric_limits<double>::infinity());

```

```

14     bg::set<3>(minPoint, bg::get<3>(cloud[i]) - time_radius);
15     bg::set<0>(maxPoint, bg::get<0>(cloud[i]) + radius);
16     bg::set<1>(maxPoint, bg::get<1>(cloud[i]) + radius);
17     bg::set<2>(maxPoint, std::numeric_limits<double>::infinity());
18     bg::set<3>(maxPoint, bg::get<3>(cloud[i]) + time_radius);
19
20     typename pointcloud::box_type query_box(minPoint,maxPoint);
21     double sum=0, N=0;
22     pcl.rtree.query(bgi::intersects(query_box), boost::make_function_output_iterator
23     ([&](typename pointcloud::value_type const& v) {
24         N++;
25         sum += bg::get<2>(v.first.min_corner()); // z coordinate
26     }));
27     double mean = sum/N;
28     a[i] = mean;
29 }
30 }

```

This listing shows how easy it can be for a domain scientist to implement their own computational kernels over the point cloud. The infrastructure hides all Python interfacing, and the generic nature allows the implementation of a function that can be immediately used with, for example, `float` or `double` values for coordinates. References are used to make shortcuts to the cloud itself and to the output attribute vector (lines 5–7). Then, parallel processing is required with a parallel for loop based on OpenMP. The body itself prepares a query box with both finite and infinite coordinates (lines 11–21). Then, accumulators (in each thread of parallel execution) are initialized as double variables `N` and `sum` (line 22). Then, we increment `N` (line 24) and accumulate the `Z` coordinate (the second coordinate) out of the `R`-tree box representing this point in line 25. Note, how handy the capture of lambdas has become: within a foreign query infrastructure given by the `R`-tree, we can write to local variables `N` and `sum`. As a last step in each loop, we store the computed mean value of the attributes at this location. Hence, the coordinate `pointcloud[i]` and the named attribute contain our results. Note that the estimation of the mean in this way is not optimal from a numerical perspective and can be replaced with an incremental form, but we aim for simplicity in this example.

Running the prepared query over the whole point cloud is not reasonable as spatiotemporal filtering is especially useful for standardizing geometry and reducing the amount of data. Therefore, we exploit the following design facts: (1) point cloud data is implicitly inside the `R`-tree including row number information to the original dataset when needed and (2) the point cloud can be overwritten with a different cloud.

For this approach, therefore, we create a spatiotemporal grid with 2D coordinates and a time coordinate and estimate the `Z` value of a 4D point cloud from the filtering result.

The query grid covers a bit more than the spatial region of the point cloud generated by a permanent laser scanner (PLS) station at the Kijkduin beach system in the Netherlands. For each timestep, this query point cloud contains 6.6 million points such that a query with 10 timesteps evaluates 66 million neighborhoods. Note that

this query grid has been designed to be larger than the original point cloud to see the behaviour under missing data.

The original dataset (Vos et al. 2022) contains more than 4000 timesteps, often spaced in one hour, taken from the rooftop of a nearby hotel. The total number of valid points sums up to 6.038 billion 4D points. In order to cope with such a massive point cloud in an in-memory setting, we used a moderate server that we expect to be available to most scientists working with permanent laser scanning data. The Kijkduin data has been put into an HDF5 container with a flat storage layout to speed up reading through memory mapping. The file contains 5 columns of data each with 6 billion double values summing up to a raw size of 121 GB.

With a query grid with a 50cm point distance and 10 selected timestamps, we run a filter with a 50cm radius and 1 week of temporal range as well and average the Z coordinate and a second query with a 2m spatial range and, again, one week of temporal range. Due to differences in the spatial extent of the different query times and our choice of a larger grid, some of these neighborhoods turn out to be empty generating a Not-a-Number (NaN) value implied by a zero division. We easily drop those NaN values by slicing in Python.

Parts of the results are visualized in Fig. 7. The original measured point cloud is compared to the spatiotemporally filtered point cloud leading to a change map as depicted in the Figure. One can clearly see stronger (yellow) changes between a measured point cloud and the spatiotemporally filtered point cloud.

The overall performance of the query processing for this massive point cloud has been measured by wallclock time. Reading the dataset took half a minute (30 s, 6 GB/s). This number is based on holding the data in a disk array with decent hardware and in a single HDF5 container with contiguous dataset layout. After reading the data, we transfer ownership from numpy to our library instantiating geometry points. This takes another 38 s. Note that due to the design of the system, we could have streamed the points in smaller batches from disk to memory avoiding this duplication, but with 1 TB of main memory, it was an unneeded complication. However, when using a chunked storage layout, of course, the HDF5 data should be streamed in the chunks that have been used to organize the file on the disk. The spatial indexing now converts all points to boxes and organizes them with a sort-tile-recurse strategy. This takes 464 s, about 7 min. Afterward, we drop the point cloud data and add our query point cloud into the library with negligible time cost. The filtering query with 50 cm neighborhoods with a time extent of one week and 10 timesteps covering the whole observation duration then, was filtered to the average Z coordinate in another 945 s, which is 15 min. Hence, the whole procedure took 1478 s or less than 25 min. The throughput of the approach in this case is an impressive number of nearly 45,000 neighborhoods per second.

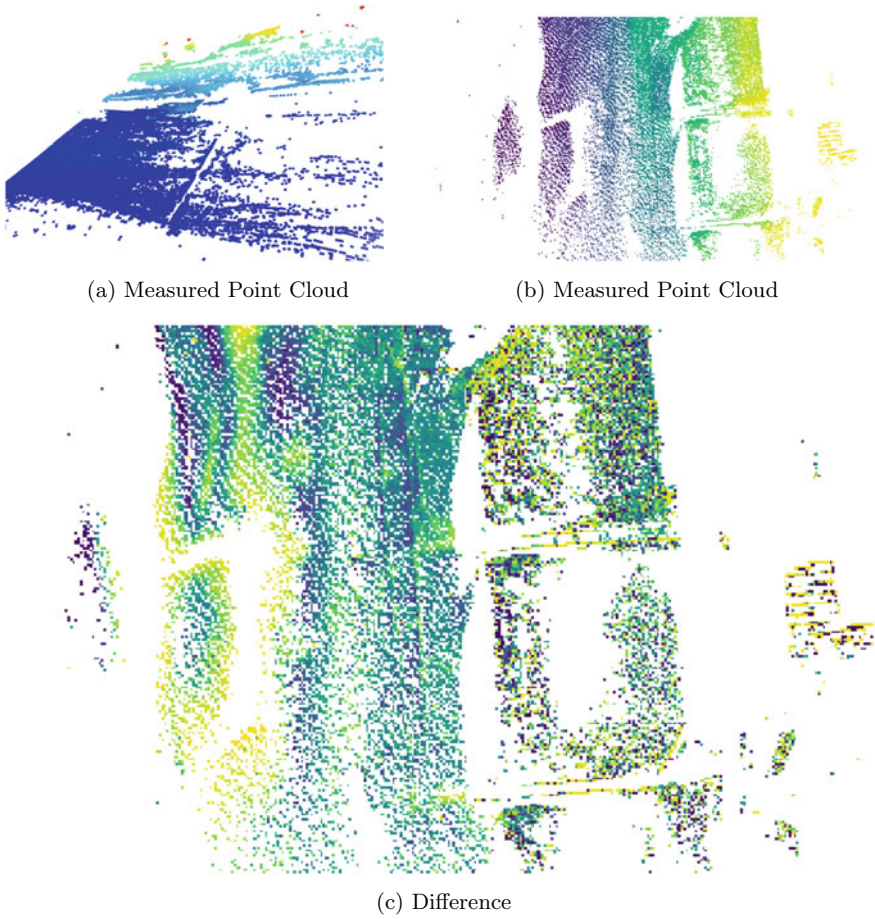


Fig. 7 The measured point cloud and the smooth result as well as the deviation map computed from the difference of the spatiotemporally filtered cloud and the measured cloud

6 Future Work

With this approach, we have shown that current hardware combined with a hardware-friendly programming environment like C++ is able to cope with the scale of modern point clouds without any programmatic handling of the big data nature. We can treat even the largest point cloud with more than 6 billion points as a simple in-memory dataset using a numpy array and work on it including spatial indexing, spatiotemporal filtering, and arbitrary user-defined filters. The system can be steered from Python in an intuitive and efficient way. However, successful big point cloud processing is not feasible in Python itself due to the need for parallelization and excessive loops when looping over all points of the data.

6.1 Interactive Analysis

Clearly, 25 min of work on a medium-performance server is not a problem for a point cloud acquired with months of measurements. However, we believe that speeding up the data science cycle for point cloud analysis with interactive performance of experiments can still speed up the research cycle as well.

Two aspects of future work are planned: First, to build a simple *distributed data infrastructure* on top of this paper’s computational infrastructure such that cloud computing machines or compute clusters commonly available in universities can be used to further reduce the analysis time.

Second, we want to investigate a view-based computation model such that the computation is constrained by the requirements of a camera. That is, the query point cloud is actually generated by view culling of a camera and only “visible” query points are processed. In this way, we hope to interactively work across even larger point clouds like the AHN3 point cloud covering the whole of the Netherlands.

6.2 Desktop and Web Application

For many point cloud processing tasks, desktop applications are being used and are quite popular. The desktop application paradigm is important, especially for field work where an Internet connection is not available and powerful server-scale compute systems are not commonly used.

At the same time, the community starts collecting, integrating, visualizing and analyzing an ever-growing amount of spatiotemporal point cloud data. Many of these point clouds are not compatible with the assumptions made for desktop applications.

In this context we aim to extend the point cloud infrastructure proposed in this paper with both: a web-based interface for browser-based access, processing and visualization of tera- or even petascale point cloud collections and the ability to use the very same techniques, algorithms, and sketches during development and fieldwork on a local machine (with significantly smaller datasets, of course).

7 Conclusion

From a technical point of view, the proposed library is built around the Python buffer protocol which allows the communication of data structures between different Python libraries and the Python core without copying or transforming data. Thus, a point cloud starts to exist in Python, for example, as a NumPy array with three columns (3D point cloud) or even 4 columns (4D point clouds including time). Such a point cloud is communicated to our C++ library in terms of a memory pointer and some descriptive aspects for efficient slicing support. In this library, the point cloud is indexed as an R*-

tree using the Sort-Tile-Recurse bulk loading algorithm. The resulting indexing tree consists of a well-balanced arrangement of possibly overlapping rectangles which is known to be extraordinarily performant for neighborhood queries such as small range queries or nearest-neighbor queries. Furthermore, we support additional query predicates of spatial nature like `ST_Within`, their negations, an ordering constraint, or whether the candidate points satisfy a user-implemented unary predicate.

With such an efficient access structure for neighborhoods of points (both spatial neighborhoods like within 30cm as well as topological neighborhoods like the ten nearest neighbors or temporal neighborhoods like within 100 s), we implement a fast singular value decomposition optimized for small environments with not more than a few hundred points. From this singular value decomposition, we can read the three real-valued eigenvalues (for 3D) and extract the structure tensor features such as linearity, planarity, scattering, omnivariance, anisotropy, eigenentropy, normalized trace, and change of curvature. The library is publicly available in Github.¹

To summarize, given the fact that there are nearly no overheads between a point cloud in numpy and processing in our library except for spatial indexing, we expect that this work will have a strong impact on point cloud processing research as the approach helps to avoid typical memory amplifications and scales to datasets that have been difficult to process before. Moreover, we aim to provide the 3D research community with a simple, extendable, performant, portable environment to formulate even complex computing tasks based on a mixture of OGC geometry, linear algebra, artificial intelligence libraries such as PyTorch and TensorFlow, and HPC platform for visualization, essentially without data transformation cost. This paper presents the first step into this direction.

References

- Anders K, Winiwarter L, Lindenbergh R, Williams JG, Vos SE, Höfle B (2020) 4d objects-by-change: spatiotemporal segmentation of geomorphic surface change from lidar time series. *ISPRS J Photogram Remote Sens* 159:352–363. <https://www.sciencedirect.com/science/article/pii/S0924271619302850>
- Besl PJ, McKay ND (1992) Method for registration of 3-D shapes. In: Schenker PS (ed) *Sensor fusion IV: control paradigms and data structures*, vol 1611. International society for optics and photonics, SPIE, pp 586 – 606. <https://doi.org/10.1117/12.57955>
- Blomley R, Weinmann M, Leitloff J, Jutzi B (2014) Shape distribution features for point cloud analysis and a geometric histogram approach on multiple scales. *ISPRS Ann Photogram Remote Sens Spatial Inf Sci II(3)*:9–16. <https://isprs-annals.copernicus.org/articles/II-3/9/2014/>
- Brenner C (2016) Scalable estimation of precision maps in a mapreduce framework. In: *Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems*. SIGSPACIAL '16, Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/2996913.2996990>
- Bueno M, Martínez-Sánchez J, González-Jorge H, Lorenzo H (2016) Detection of geometric key-points and its application to point cloud coarse registration. *Int Arch Photogram Remote Sens Spatial Inf Sci* 41:187–194

¹ <https://github.com/tum-bgd/pointcloudqueries>.

- Chen Y, Medioni G (1991) Object modeling by registration of multiple range images. In: Proceedings 1991 IEEE international conference on robotics and automation, vol 3. pp 2724–2729
- Choy CB, Dong W, Koltun V (2020) Deep global registration. CoRR abs/2004.11540. <https://arxiv.org/abs/2004.11540>
- Dean J, Ghemawat S (2008) Mapreduce: simplified data processing on large clusters. *Commun ACM* 51(1):107–113
- Eitel JU, Höfle B, Vierling LA, Abellán A, Asner GP, Deems JS, Glennie CL, Joerg PC, LeWinter AL, Magney TS et al (2016) Beyond 3-d: the new spectrum of lidar applications for earth and ecological sciences. *Remote Sens Environ* 186:372–392
- Elseberg J, Borrmann D, Nüchter A (2011) Efficient processing of large 3d point clouds. In: 2011 XXIII International symposium on information, communication and automation technologies, pp 1–7
- Fischer K, Simon M, Olsner F, Milz S, Gross HM, Mader P (2021) Stickypillars: robust and efficient feature matching on point clouds using graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 313–323
- Fürlinger K, Fuchs T, Kowalewski R (2016) DASH: A C++ PGAS library for distributed data structures and parallel algorithms. In: Proceedings of the 18th IEEE international conference on high performance computing and communications (HPCC 2016). Sydney, Australia, pp 983–990
- Gojcic Z, Zhou C, Wegner JD, Wieser A (2018) The perfect match: 3d point cloud matching with smoothed densities. CoRR abs/1811.06879. <http://arxiv.org/abs/1811.06879>
- Gumhold S, Wang X, MacLeod R (2001) Feature extraction from point clouds. In: Proceedings of 10th international meshing roundtable 2001
- Huang R, Xu Y, Hoegner L, Stilla U (2022) Semantics-aided 3d change detection on construction sites using uav-based photogrammetric point clouds. *Autom Construct* 134:104057
- Ioannou Y, Taati B, Harrap R, Greenspan M (2012) Difference of normals as a multi-scale operator in unorganized point clouds. In: 2012 Second international conference on 3D imaging, modeling, processing, visualization and transmission, pp 501–508
- Kammerl J, Blodow N, Rusu RB, Gedikli S, Beetz M, Steinbach E (2012) Real-time compression of point cloud streams. In: 2012 IEEE international conference on robotics and automation, pp 778–785
- Kromer RA, Abellán A, Hutchinson DJ, Lato M, Edwards T, Jaboyedoff M (2015) A 4d filtering and calibration technique for small-scale point cloud change detection with a terrestrial laser scanner. *Remote Sens* 7(10):13029–13052. <https://www.mdpi.com/2072-4292/7/10/13029>
- Lee KH, Lee YJ, Choi H, Chung YD, Moon B (2012) Parallel data processing with mapreduce: a survey. *ACM SIGMoD Record* 40(4):11–20
- Lokugam Hewage CN, Laefer DF, Vo AV, Le-Khac NA, Bertolotto M (2022) Scalability and performance of LiDAR point cloud data management systems: a state-of-the-art review. *Remote Sens* 14(20):5277. <https://www.mdpi.com/2072-4292/14/20/5277>
- Munoz D, Bagnell JA, Vandapel N, Hebert M (2009) Contextual classification with functional max-margin markov networks. In: IEEE conference on computer vision and pattern recognition (CVPR)
- Osada R, Funkhouser T, Chazelle B, Dobkin D (2002) Shape distributions. *ACM Trans Graph* 21(4):807–832
- Pajić V, Govedarica M, Amović M (2018) Model of point cloud data management system in big data paradigm. *ISPRS Int J Geo-Inf* 7(7):265. <http://www.mdpi.com/2220-9964/7/7/265>
- Polewski P, Yao W, Heurich M, Krzystek P, Stilla U (2018) Learning a constrained conditional random field for enhanced segmentation of fallen trees in als point clouds. *ISPRS J Photogram Remote Sens* 140:33–44. *geospatial Computer Vision*
- Pomerleau F, Liu M, Colas F, Siegwart R (2012) Challenging data sets for point cloud registration algorithms. *Int J Robot Res* 31(14):1705–1711
- Qi CR, Su H, Mo K, Guibas LJ (2017) Pointnet: deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)

- Qi CR, Yi L, Su H, Guibas LJ (2017) Pointnet++: deep hierarchical feature learning on point sets in a metric space. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) *Advances in neural information processing systems*. vol 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf
- Rusu RB, Blodow N, Beetz M (2009) Fast point feature histograms (fpfh) for 3d registration. In: 2009 IEEE international conference on robotics and automation, pp 3212–3217
- Tobler WR (1970) A computer movie simulating urban growth in the detroit region. *Econ Geogr* 46:234–240. <http://www.jstor.org/stable/143141>
- Vetrivel A, Gerke M, Kerle N, Nex F, Vosselman G (2018) Disaster damage detection through synergistic use of deep learning and 3d point cloud features derived from very high resolution oblique aerial images, and multiple-kernel-learning. *ISPRS J Photogram Remote Sens* 140:45–59. <https://www.sciencedirect.com/science/article/pii/S0924271616305913>, *geospatial Computer Vision*
- Vos S, Anders K, Kuschnerus M, Lindenbergh R, Höfle B, Aarninkhof S, de Vries S (2022) A high-resolution 4d terrestrial laser scan dataset of the kjkduin beach-dune system, the netherlands. *Sci Data* 9(1):191
- Weis M, Gutjahr C, Rueda Ayala V, Gerhards R, Ritter C, Schölderle F (2008) Precision farming for weed management: techniques. *Gesunde Pflanzen* 60(4):171–181
- Xie L, Furuhashi T, Shimada K (2020) Multi-resolution graph neural network for large-scale point-cloud segmentation. *CoRR* abs/2009.08924. <https://arxiv.org/abs/2009.08924>
- Xu Q, Xu Z, Philip J, Bi S, Shu Z, Sunkavalli K, Neumann U (2022) Point-nerf: point-based neural radiance fields. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 5438–5448
- Xu Y, Boerner R, Yao W, Hoegner L, Stilla U (2019) Pairwise coarse registration of point clouds in urban scenes using voxel-based 4-planes congruent sets. *ISPRS J Photogram Remote Sens* 151:106–123
- Yue X, Wu B, Seshia SA, Keutzer K, Sangiovanni-Vincentelli AL (2018) A lidar point cloud generator: from a virtual world to autonomous driving. In: *Proceedings of the 2018 ACM on international conference on multimedia retrieval*, pp 458–464
- Zhang R, Li G, Wiedemann W, Holst C (2022) Kdo-net: towards improving the efficiency of deep convolutional neural networks applied in the 3d pairwise point feature matching. *Remote Sens* 14(12). <https://www.mdpi.com/2072-4292/14/12/2883>
- Zwally HJ, Schutz B, Abdalati W, Abshire J, Bentley C, Brenner A, Bufton J, Dezio J, Hancock D, Harding D et al (2002) Icesat’s laser measurements of polar ice, atmosphere, ocean, and land. *J Geodyn* 34(3–4):405–445

Transferring Façade Labels Between Point Clouds with Semantic Octrees While Considering Change Detection



Sophia Schwarz, Tanja Pilz, Olaf Wysocki, Ludwig Hoegner, and Uwe Stilla

Abstract Point clouds and high-resolution 3D data have become increasingly important in various fields, including surveying, construction, and virtual reality. However, simply having this data is not enough; to extract useful information, semantic labeling is crucial. In this context, we propose a method to transfer annotations from a labeled to an unlabeled point cloud using an octree structure. The structure also analyses changes between the point clouds. Our experiments confirm that our method effectively transfers annotations while addressing changes. The primary contribution of this project is the development of the method for automatic label transfer between two different point clouds that represent the same real-world object. The proposed method can be of great importance for data-driven deep learning algorithms as it can also allow circumventing stochastic transfer learning by deterministic label transfer between datasets depicting the same objects.

Keywords Octree · Semantic information · Change detection · Urban point clouds · Label transferring

This article was selected based on the results of a double-blind review of an extended abstract.

S. Schwarz · T. Pilz (✉) · O. Wysocki · L. Hoegner · U. Stilla
Photogrammetry and Remote Sensing, TUM School of Engineering and Design,
Technical University of Munich (TUM), Munich, Germany
e-mail: tanja.pilz@tum.de

S. Schwarz
e-mail: sophiamaria.schwarz@tum.de

O. Wysocki
e-mail: olaf.wysocki@tum.de

L. Hoegner
e-mail: ludwig.hoegner@tum.de

U. Stilla
e-mail: stilla@tum.de

L. Hoegner
Department of Geoinformatics, University of Applied Science (HM), Munich, Germany

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
T. H. Kolbe et al. (eds.), *Recent Advances in 3D Geoinformation Science*, Lecture Notes
in Geoinformation and Cartography, https://doi.org/10.1007/978-3-031-43699-4_17

1 Introduction

Point cloud data has gained popularity in recent years for its ability to provide high-resolution 3D data of real-world scenes. Point clouds are used in a variety of applications, including aiding navigation of autonomous cars (Wilbers et al. 2019), preserving cultural heritage (Grilli and Remondino 2020), and reconstructing 3D building models (Haala and Kada 2010). Semantically enriched point clouds are particularly interesting for these applications since they not only provide geometric information but also add contextual information about real-world objects. Recent advancements in deep learning have proven effective for automatic semantic point cloud labeling (Wysocki et al. 2022), which assigns labels to individual points. However, the performance of neural networks depends on large, manually labeled ground-truth datasets that are tedious and time-consuming to acquire. Providing point clouds with accurate labels is so far only possible by manual annotation. This creates further problems in lacking ground-truth data for other research topics as well.

To tackle this issue, we introduce a method for automatic façade-semantic transfer between labeled and unlabeled point clouds representing the same building. We propose to achieve this goal by a semantic octree data structure preceded by a co-registration using the Iterative Closest Point concept. Additionally, the changes between the two point clouds are identified by utilizing the octree to describe the occupancy of the octree leaves.

2 Related Work

Generalized Iterative Closest Point (GICP) is a variant of the Iterative Closest Point (ICP) algorithm frequently used for point cloud co-registration (Xu and Stilla 2021; Segal et al. 2009). The ICP algorithm iteratively aligns two point clouds by minimizing the distance between corresponding point pairs (Segal et al. 2009). It iterates until the distance threshold is satisfied or a maximum number of iterations is reached. Instead of relying solely on the distance between points, GICP minimizes the distance between geometric primitives (e.g., planes) found in point clouds. This proves especially useful in an urban environment where regular, primitive structures are prevalent (Xu and Stilla 2021).

Another related topic is the approach of utilizing octrees, and their 3D voxel leaves for change detection (Kharroubi et al. 2022). For instance, such a concept is employed by Gehrung et al., who utilize ray tracing on a highly-efficient octree structure and Bayesian reasoning to remove dynamic objects from point clouds based on three states: *occupied*, *empty*, and *unknown* (Gehrung et al. 2017).

Gehrung et al. (2018) also proposed a fast voxel-based indicator for change detection using low-resolution octrees. This approach involves dividing the point cloud data into octree nodes and computing a histogram of voxel attributes, such as point density. Changes between two point cloud datasets can then be detected by compar-

ing the histograms of corresponding nodes. The approach was shown to be effective in detecting changes in large-scale urban areas with low-resolution point cloud data (Gehring et al. 2018).

3 Methodology

The proposed method for the semantic label transfer between two urban point clouds uses an octree-based data structure that considers semantic information as a leaf criterion (Fig. 1). With the same octree-based data structure, coarse change detection is achieved. Preceding this is a plane-based co-registration of the two point clouds that uses the Generalized Iterative Closest Point algorithm. We developed the method based on work in the module Photogrammetry Project at Technical University Munich (Dragicevic et al. 2023). The implementation is available in the repository.¹

3.1 Co-registration Using Generalized Iterative Closest Point

As point clouds stemming from different campaigns can have global registration deviations (Fig. 2a), we first ensure that they are co-registered. We propose a method that leverages the planar-like structure of buildings by using plane-to-plane point cloud co-registration. To additionally increase the robustness and simultaneously decrease computing time, a voxelizing precedes the GICP (Koide et al. 2021). For this, we compute the estimated normals for the downsampled source and target points. Next, we calculate the registration using the two input point clouds, the initial transformation matrix, the maximal corresponding distance between a plane-pair λ , and the convergence criteria χ (Fig. 1). After calculating the transformation matrix based on the downsampled points, we apply it to the complete dataset (Fig. 2b). As presented in Fig. 2b, the registered point clouds do not show the previously noted deviations.

3.2 Semantic Octree

The data structure of the semantic octree performs two tasks simultaneously: transferring labels and detecting changes.

We introduce two attributes to the octree leaves, a semantic label and a label describing whether a region has been removed. Considering a dynamic depth, the octree is set up only using the semantically labeled point cloud. While building up the octree, a node is defined as a leaf when it meets one of three leaf-criteria *empty leaf*, *one label* and *max depth*, which are checked in this order.

¹ https://github.com/SchwarzSophia/Transferring_urban_labels_between_pointclouds/.

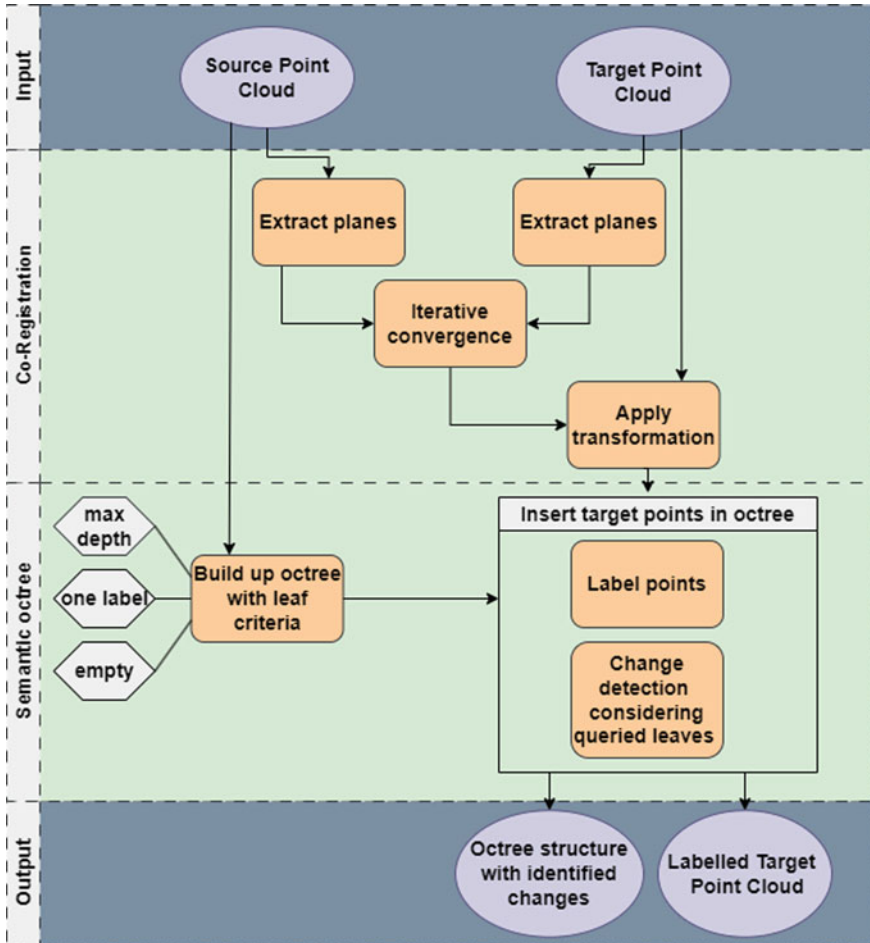


Fig. 1 Proposed workflow

A leaf is considered as *empty leaf* if no points are contained in its volume. We define an empty leaf as a *new* label, indicating new points compared to the target cloud, whereas *unchanged* are the matching areas. In this case, the semantic label indicates an unknown class, which later describes newly captured façade points. This is used to characterize added building parts in the change detection.

The criterion *one label* is fulfilled if all points in a node have the same semantic class. This class is defined as the semantic leaf label. The last criterion to be checked is *max depth*, referring to the maximum octree depth a leaf might reach. This prevents over-fitting of the data and, at the same time, shortens the computation time. If the maximal depth is reached and the labels in the leaf container are heterogeneous, the prevalent label in the container is identified as the semantic leaf label. The *removed*

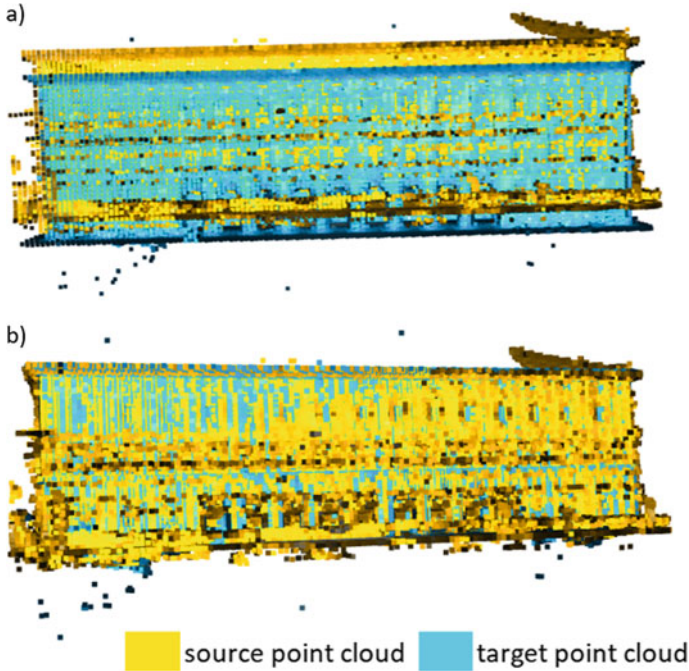


Fig. 2 Source and target point cloud **a** before registration and **b** after registration

label is set to true at this stage. The maximum depth is controlled by *max lat*, the largest permitted side length of the smallest leaf container: Formula 1 specifies the relationship between the lateral point cloud length, the maximal lateral length of the smallest octree leaf, and the depth of the octree.

$$depth = \log_2 \left(\frac{lateral\ point\ cloud\ length}{maxLat} \right) \tag{1}$$

With these criteria, we design the data structure to adapt to the density of the point cloud and the complexity of the depicted building. To label the points of the target point cloud, they are iteratively inserted into the octree. During this process, two determinations are made: the point is assigned the label of the leaf it falls into, and the *removed* label of the queried leaf is set to false. Finally, the target point cloud is labeled, and the octree leaves represent a coarse change detection.

4 Experiments

We used the open TUM-FAÇADE (Wysocki et al. 2022) dataset and the proprietary MF point cloud for the experiments. The TUM-FAÇADE segmentation benchmark served as a labeled point cloud, whereas MF was the unlabeled one. We manually labeled the MF point cloud to validate our method and evaluate our generated results. We determined optimal values for the co-registration experimentally. We achieved the best performance with a voxel size ν of 10 cm and a maximal correspondence distance of $\lambda = 10$ m. For the convergence criteria, we set the number of maximal iterations to 50 and both the relative fitness and relative root mean square error to $1 \mu\text{m}$. The parameters for the co-registration are dependent on the datasets in use. However, we can assume that similar values are reasonable for all point clouds representing urban buildings.

The datasets were pre-processed with FME (Feature Manipulation Engine) and Python using the Open3D library (Zhou et al. 2018) to ensure uniformity in the coordinate system across all input datasets. Additionally, statistical outlier removal techniques were applied to the point clouds, again utilizing Open3D functions (Zhou et al. 2018). The statistical outlier removal process involved evaluating the neighborhood density and the statistical distribution of densities for each point. A threshold was established to allow for permissible deviations from the mean density. As indicated by the threshold, points within sparser neighborhoods were subsequently removed from the analysis. Our study employed a threshold of 1.7σ , considering 20 neighboring points for density calculations and outlier detection. This specific parameter configuration allowed for effectively removing points exhibiting significantly deviant density values. By implementing these pre-processing steps, including coordinate system transformation and statistical outlier removal, we ensured a consistent and refined dataset for subsequent analysis and interpretation (Table 1).

Table 1 Confusion matrix of the example building calculated with a maximal lateral length of 10 cm

| Classes | wall | window | door | molding | deco | terrain | blinds | interior | other |
|----------|---------|--------|-------|---------|-------|---------|--------|----------|--------|
| wall | 1813248 | 2190 | 13 | 42085 | 1250 | 277 | 0 | 15443 | 9801 |
| window | 7592 | 250327 | 0 | 71662 | 0 | 7 | 7130 | 11308 | 9927 |
| door | 9 | 532 | 25869 | 4425 | 0 | 0 | 0 | 140 | 1329 |
| molding | 256994 | 32820 | 2071 | 774608 | 2445 | 0 | 748 | 3276 | 7508 |
| deco | 414 | 0 | 0 | 375 | 33691 | 0 | 0 | 0 | 17 |
| terrain | 5861 | 70 | 501 | 62 | 0 | 148116 | 0 | 2 | 9581 |
| blinds | 24 | 9482 | 0 | 2995 | 0 | 0 | 15755 | 0 | 0 |
| interior | 4716 | 26128 | 300 | 2559 | 0 | 9 | 277 | 162752 | 163 |
| other | 120476 | 8326 | 298 | 62538 | 10 | 5556 | 0 | 107 | 131681 |

4.1 Quantitative Analysis

For evaluation, we compared the computed labels to the labels of the manually classified MF. For this, we calculated a confusion matrix for different octree depths, the overall accuracy, and the Cohen Kappa coefficient.

We observe an overall accuracy of about 81% and a kappa coefficient of around 0.72 (Table 2). The three different lateral lengths shown have deviations in the third decimal, which is an insignificant difference. These results indicate a good matching within the classification.

The TUM-FAÇADE point cloud has a higher point density than the manually labeled cloud (reflected by Table 3). This deviation can be one of the reasons for inaccuracies in the method.

The identified changes are depicted in Fig. 7. The new points make up around 7.9 %.

4.2 Qualitative Analysis

Figure 3 provides a visualization of the result of our proposed method with the point cloud changes. As indicated in the results of the quantitative analysis, the method works well. The new class *empty*, generated by the method, is colorized in pink.

During our visual evaluation, we detected several inaccuracies in the complex features of the point cloud, e.g., the windows or the blinds as well in the areas that do not show holes in the source point cloud. The comparison in Fig. 5 illustrates the method's influence on holes within the source point cloud. In areas where the point cloud exhibited sufficient density, accurate label transfer was achieved, ensuring reliable results. However, in regions where the point cloud lacked density, inaccuracies surfaced. Specifically, rather than uniformly designating the entire hole with class 18, a combination of diverse classes became evident within the hole. For instance, the

Table 2 Overall accuracy and kappa coefficient for three different lateral lengths

| Lateral lengths (cm) | Overall accuracy | Kappa coefficient |
|----------------------|------------------|-------------------|
| 5 | 0.818 | 0.725 |
| 10 | 0.817 | 0.723 |
| 20 | 0.815 | 0.722 |

Table 3 Total number of points in TUM-FAÇADE and manually labeled point cloud

| Point cloud | Total number of points |
|-------------|------------------------|
| TUM-FAÇADE | 35,837 MLN |
| MF | 4461 MLN |

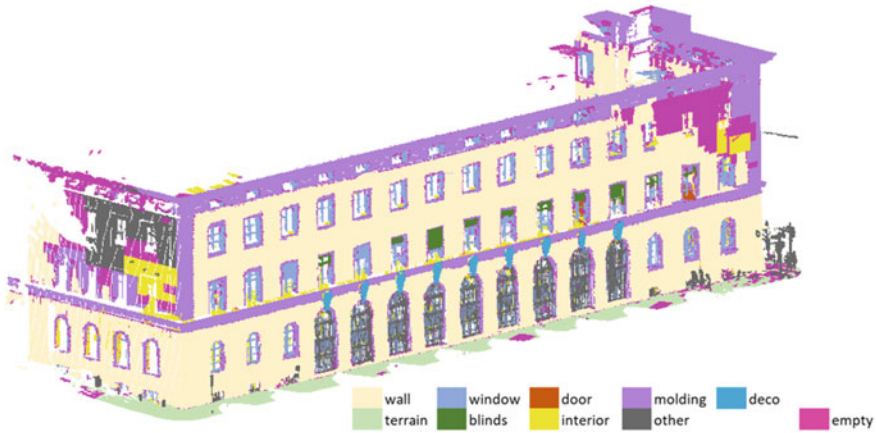


Fig. 3 MF point cloud with automatically transferred labels at a lateral length of 10 cm

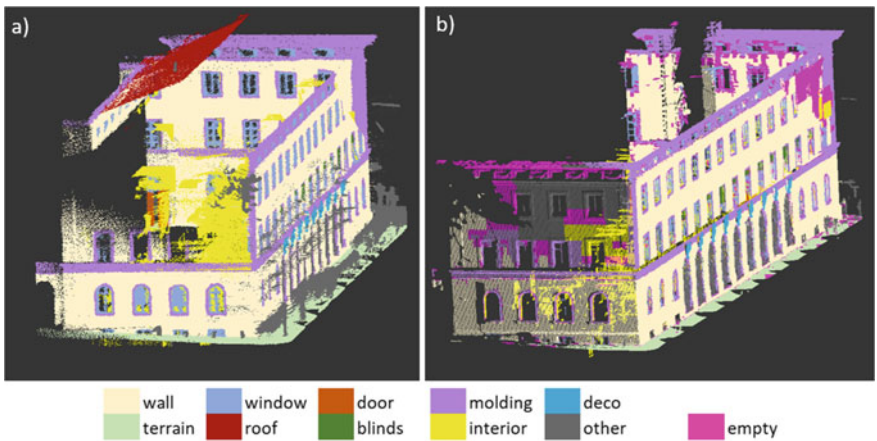


Fig. 4 South corner of the example building of **a** the source point cloud TUM-FAÇADE (Wysocki et al. 2022) and **b** the resulting point cloud

distinct features of the building’s corner molding appeared blended into the hole’s context, contributing to the observed inaccuracies.

The most noticeable error occurred at a hole on the south side of the building. Instead of assigning the class *empty*, most of the points in the concerned area were assigned the class *other*.

With a more detailed view, the quality of the labeled point cloud can be better assessed. Minor inaccuracies around complex structures are evident in the close-up of window structures, Fig. 6. While the wall was consistently labeled and the location of the window contours are transferred correctly, they are not precisely defined at the edges.

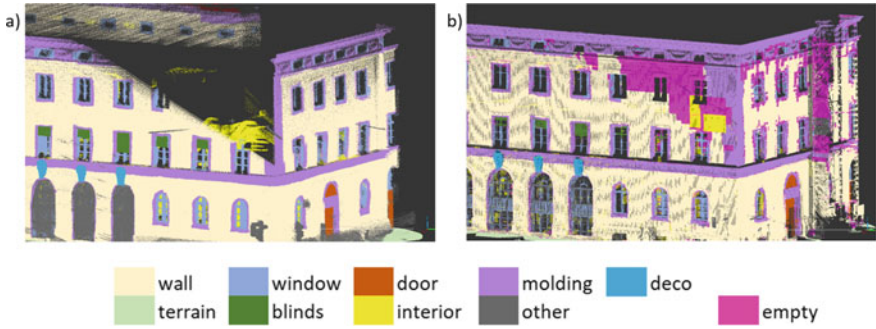


Fig. 5 North-east corner of the example building of **a** the source TUM-FAÇADE (Wysocki et al. 2022) and **b** target point cloud MF



Fig. 6 Close-up of exemplary windows

In Fig. 7, we see the algorithm’s performance with changes in the point clouds by comparing the *new* and the *unchanged* points. Our method detects rough changes in the building effectively. Fine changes, however, that do not change the geometric structure of the point cloud are not registered.

4.3 Discussion

Experiments revealed promising results for transferring labels between clouds captured from the same building façade using the proposed method. For the majority of the points, the method assigned the correct labels. Subsequently, we discuss the reasons for the remaining inaccuracies.



Fig. 7 Change detection with a lateral length of 10 cm

In our analysis, several inaccuracies occurred in the representation of complex features, such as windows or blinds (see Fig. 5b). These deviations can be attributed to the parameter *max depth* limitations. If structures are smaller than this parameter, they cannot be represented accurately.

Further inaccuracies arose with round façade elements. They were not transferred precisely because the algorithm works with cubic volumes. Therefore, the rounding can only be approximated, the accuracy of which depends on the *max depth* parameter. Nonetheless, defining the parameter is necessary to avoid over-fitting the octree to the source point cloud, which would cause too many small leaves with the label *new*, leading to wrongly labeled *new* points. We found an optimal *max depth* of 10 cm lateral length for our data in experiments.

Additionally, different point densities in point clouds (as shown in Table 3) pose a significant challenge. If the source point cloud has a lower point density than the target point cloud, the achievable accuracies are limited from the start. Since the octree structure considers the local density of the points, large cubes with few points may not be further subdivided.

This can lead to new features being incorrectly assigned a class, as illustrated in more detail in Fig. 4. We had larger holes in the source point cloud in our example building. Our proposed method handles this well. As shown in Fig. 5, all added points are labeled as *empty*. Nevertheless, there can be inaccuracies at the edge of the holes. Again, the volume is not further divided because only a few homogeneous points fall into the edge area. At this point, the limitations of change detection become apparent. Rough changes are well detected by the method. However, subtle changes that do not alter the structure of the building are not registered. The reason for this is the limitation of the method, as it cannot deal with the change of objects in the same place, for instance, recording open versus closed window blinds. In both cases, there are points in the cube, so no empty areas are detected. The points are not identified as changed but are assigned the wrong label.

Lastly, we acknowledge that some mislabeled points can stem from discrepancies in the manually labeled test data sets. These are due to different interpretations of points, leading to varied classifications or simply small errors in labeling. While these biases do not necessarily distort the test results gravely, they can still introduce inconsistencies in the final results.

5 Conclusion

In this paper, we present a method that utilizes generalized Iterative Closed Point algorithm for point cloud registration and an octree map to transfer annotations from a labeled point cloud to an unlabeled one. Additionally, we detect changes by utilizing the semantic octree. Our experiments corroborate the effectiveness of the proposed method in accurately transferring labels between point clouds (approx. 82 %) with similar geometric and semantic structures. However, we observe a few inaccuracies in the method, especially when looking at complex or round features.

As a next step, the edge blurring could be improved by introducing a minimal octree depth and dividing the nodes into containers according to their local density. Lastly, the method can be compared to other state-of-the-art methods for semantic information transfer to determine its relative strengths and weaknesses. Further research can help to refine and improve this for practical use. This method can be of great importance for data-driven deep learning algorithms as it can also allow circumventing stochastic transfer learning by deterministic label transfer between datasets depicting the same objects.

Acknowledgements This work was supported by the Bavarian State Ministry for Economic Affairs, Regional Development, and Energy within the framework of the IuK Bayern project *MoFa3D—Mobile Erfassung von Fassaden mittels 3D Punktwolken*, Grant No. IUK643/001. Moreover, the work was conducted within the framework of the Leonhard Obermeyer Center at the Technical University of Munich (TUM). We gratefully acknowledge the Geoinformatics team at the TUM for the valuable insights and for providing the CityGML datasets. This method was developed in the module Photogrammetry Project of the Master's study program Geodesy and Geoinformation at the TUM. We thankfully acknowledge the supervisor team for their constructive insights and support.

References

- Dragicevic S, Pilz T, Schwarz S (2023) Transferring urban labels between point clouds. Technical University of Munich, Tech. rep
- Gehring J, Hebel M, Arens M, Stilla U (2017) An approach to extract moving objects from MLS data using a volumetric background representation. *ISPRS Ann Photogram Remote Sens Spatial Inf Sci IV-1/W1:107–114*
- Gehring J, Hebel M, Arens M, Stilla U (2018) A voxel-based metadata structure for change detection in point clouds of large-scale urban areas. *ISPRS Ann Photogram Remote Sens Spatial Inf Sci IV(2):97–104*

- Grilli E, Remondino F (2020) Machine learning generalisation across different 3d architectural heritage. *ISPRS Int J Geo-Inf* 9(6):379
- Haala N, Kada M (2010) An update on automatic 3D building reconstruction. *ISPRS J Photogram Remote Sens* 65(6):570–580
- Kharroubi A, Poux F, Ballouch Z, Hajji R, Billen R (2022) Three dimensional change detection using point clouds: a review. *Geomatics* 2(4):457–485
- Koide K, Yokozuka M, Oishi S, Banno A (2021) Voxelized gicp for fast and accurate 3d point cloud registration. In: 2021 IEEE international conference on robotics and automation (ICRA), pp 11054–11059. <https://doi.org/10.1109/ICRA48506.2021.9560835>
- Segal A, Haehnel D, Thrun S (2009) Generalized-icp. In: *Robotics: science and systems*, vol 2. Seattle, WA, p 435
- Wilbers D, Merfels C, Stachniss C (2019) Localization with sliding window factor graphs on third-party maps for automated driving. In: 2019 International conference on robotics and automation (ICRA). IEEE, pp 5951–5957
- Wysocki O, Hoegner L, Stilla U (2022) TUM-FAÇADE: Reviewing and enriching point cloud benchmarks for façade segmentation. *Int Arch Photogram Remote Sens Spatial Inf Sci XLVI-2/W1-2022:529–536*
- Xu Y, Stilla U (2021) Towards building and civil infrastructure reconstruction from point clouds: a review on data and key techniques. *IEEE J Sel Top Appl Earth Obs Remote Sens* 14:2857–2885
- Zhou QY, Park J, Koltun V (2018) Open3D: a modern library for 3D data processing. [arXiv:1801.09847](https://arxiv.org/abs/1801.09847)

Investigating Data Fusion from Three Different Point Cloud Datasets Using Iterative Closest Point (ICP) Registration



Wahyu Marta Mutiarasari, Muhammad Zulkarnain Abd Rahman,
and Alias Abdul Rahman

Abstract A dataset from a single survey technique, such as terrestrial laser scanning, hardly generates a complete 3D model. Therefore, data fusion is usually used to integrate various data to overcome the technique limitations. For multisource point cloud data, Iterative Closest Point (ICP) registration can be used to combine the data. However, at the end of the ICP registration process, there are gaps between the point clouds. This paper contains an initial work to investigate the fusion quality of three point clouds (LiDAR point by drone, TLS point, and image-based point) using ICP registration. In this case, the TLS point is used as reference and target as well, while the other point clouds are the source data. This experiment observed the fused model of the ICP and inspected the roughness of each data for further assessment. Then, the M3C2 distance tool in the CloudCompare software was used to show the existence of distance between the point cloud models and the voids as well. Based on the results, the fusion of the three datasets completes the 3D model, except for a few holes caused by obstructions, such as vegetation and roof shadow. It is noted that the image-based point provides a more substitute data, while the LiDAR point by drone has a minor role to complete the model. Hence, for future work, the use of data from another technique can be considered for data fusion and distance refinement is important in this case.

Keywords Data fusion · Point cloud · Iterative closest point (ICP)

This article was selected based on the results of a double-blind review of an extended abstract

W. M. Mutiarasari (✉) · M. Z. Abd Rahman · A. Abdul Rahman
3D GIS Research Lab, Department of Geoinformation, Faculty of Built Environment and
Surveying, Universiti Teknologi Malaysia, Johor Bahru, Johor, Malaysia
e-mail: wahyu@graduate.utm.my

M. Z. Abd Rahman
e-mail: mdzulkarnain@utm.my

A. Abdul Rahman
e-mail: alias@utm.my

1 Introduction

Data fusion is the integration of various datasets (either multisensor or multiscale). It facilitates incomplete data by combining single-survey data with other acquisition techniques. Incomplete data, such as that from terrestrial laser scanning (TLS) represent imperfect 3D model objects because of self-occlusion problems or other severe occlusions (Li et al. 2018). In 3D model generation, data fusion aims to produce a more complete 3D building model. Currently, multisource data integration can be conducted at the point cloud level. For instance, image-based and laser-scanned points can be integrated into a dense 3D model. Several approaches have been used to combine such data with diverse characteristics, for example the iterative closest point (ICP) algorithm. This is a fusion approach for multisource point clouds as reported by Adamopoulos and Rinaudo (2021).

The ICP algorithm consists of two main processes: matching the corresponding points and transformation. This algorithm is simple to apply because it is also available in open source software, such as CloudCompare. In addition, Ren et al. (2019) suggested using the ICP because of highly accurate produced data. This has also been suggested by several previous studies for data fusion, especially for 3D building model generation (see Altuntas et al. (2016) and Shao et al. (2019)) in their works for heritage building documentation. However, the ICP process is limited in completely producing accurate combined data—i.e., the gaps after co-registration (Li et al. 2021).

This paper evaluated the results of the amalgamated data to determine the distances among the data. It fuses three datasets from three different techniques: LiDAR point by drone, TLS point, and image-based point. The next Sect. 2 describes related works that investigated combining data using the ICP algorithm. Section 3 explains the data used in this experiment, Sect. 4 describes the experiments, and Sect. 5 covers the results and discussions. Finally, the conclusions of this paper are in Sect. 6.

2 Related Works

Two datasets from two different techniques, such as TLS point and dense point from images, were integrated using the ICP algorithm to complement each other. The work by Li et al. (2021) applied it to two objects that had different conditions. The TLS point of the first model had large occlusions coming from the ground, while another one was recorded using a limited vertical field-of-view (FoV) laser scanner. Therefore, image data was obtained to overcome each condition. After data integration and detailed enhancement of the result, both models were completed up to 82%. Other data fusion methods of two datasets were performed by Castilla et al. (2021) and Hua et al. (2020), both utilizing TLS point and image-based point to generate 3D models of heritage buildings in different challenging conditions.

The completeness of the 3D model based on the fusion work above is not yet 100%. This means that some parts of the model have no data. As such, the idea to

add more data entails the possibility to make the fusion model more complete. Some data, as stated in the work by Bouziani et al. (2021), such as drone-based LiDAR and drone photogrammetry data, has potential for high-precision models. There have been only a few studies on the fusion of three datasets into one object, such as the works by Luhmann et al. (2022) and Maset et al. (2022). Their similarity is that they used two datasets of images (aerial and terrestrial based) plus a laser scanning point set. Both works also confirmed the completeness of data fusion. The first work stated that the structure-from-motion points covered the tower and roof of the object, which cannot be obtained by TLS. The other work said that the image-based point and mobile mapping system (MMS) point cloud complement each other. Instead of the completeness percentage of the 3D models, they evaluated the accuracy (approx. 5 mm) of the image combination, which was in the laser scanning range and the geometric detail precision of the MMS points, which was lower than the image data.

After ICP registration, it was confirmed that the integrated data still has gaps in the connected regions. The latest work above confirmed that the average value of cloud-to-cloud (C2C) absolute distance in the CloudCompare software was 2.7 cm. This distance (gap) clearly appeared in the work of Jaber and Abed (2020). Their work also investigated the incompleteness of fusion data using multiscale model-to-model cloud comparison (M3C2), and the result showed a standard deviation (STD) of 0.02 m. An effort to improve this data fusion quality by Li et al. (2021) was limited by the complex geometric forms. One can say that complete data without holes has not been achieved thus far.

This paper contains an initial work to investigate the fusion quality of three datasets—namely, LiDAR point by drone, TLS point, and image-based point, a different combination compared to those in previous works for 3D building modeling. Using the M3C2 distance, this paper shows the existing distance between point clouds and the voids that need to be fixed. In terms of completeness, the fusion of two datasets was claimed to lead to complete 3D building models (up to 82%). Therefore, further experiments are expected to improve the fusion data (distances and voids) and produce more complete 3D building models (more than 82% completeness) from the fusion of three datasets.

3 The Data

The three datasets used were obtained via different methods and at different times for this experiment, and they depict the same object. The object is an old library building. The first dataset is a LiDAR point that was taken by drone in 2021, sourced from Mutiarasari et al. (2023), and described mostly the building rooftop (see Fig. 1). The next dataset, obtained from Riyadi and Prasideya (2019), was recorded by TLS Topcon GLS-200 in 2019. It is a georeferenced laser scanning point that serves as an integration reference (see Fig. 2). The third dataset is the latest image data of the building environment, including 185 nadir and 237 oblique that was obtained by drone (DJI Phantom Pro 4) with an altitude of 80 m. Image processing was performed

to attain the dense point from the image. The high-accuracy process was done using the Agisoft software, and it produced a dense model, as shown in Fig. 3.

According to the figures above, the TLS point-based model depicts the whole building but lacks some rooftop data, while the first dataset offers the full rooftop data. Meanwhile, the image-based model shows some holes (no data) on the building's side that the TLS point-based model represents better. It can be said that each technique

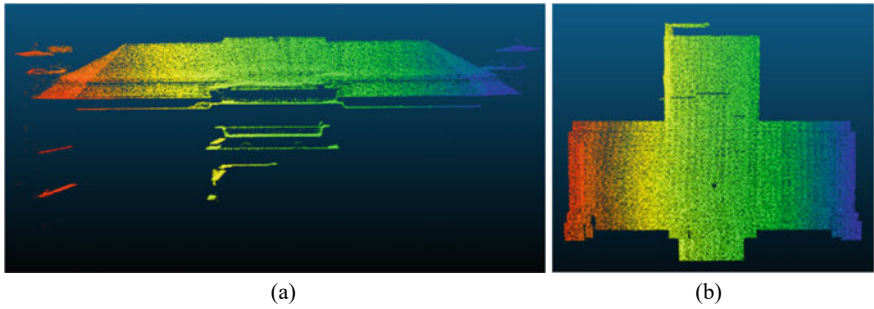


Fig. 1 LiDAR point by drone: a front view and b top view



Fig. 2 TLS point-based model: a front view and b top view

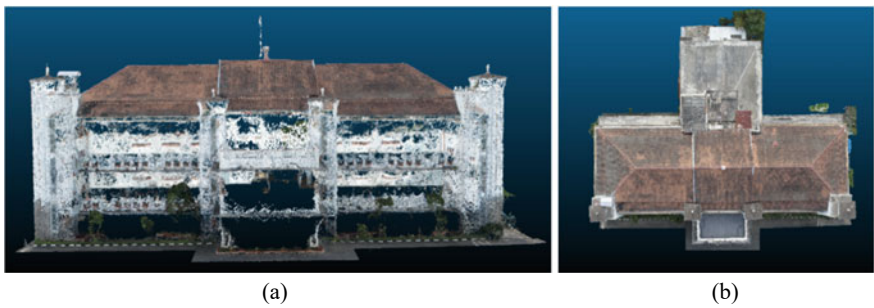


Fig. 3 Image-based point model: a front view and b top view

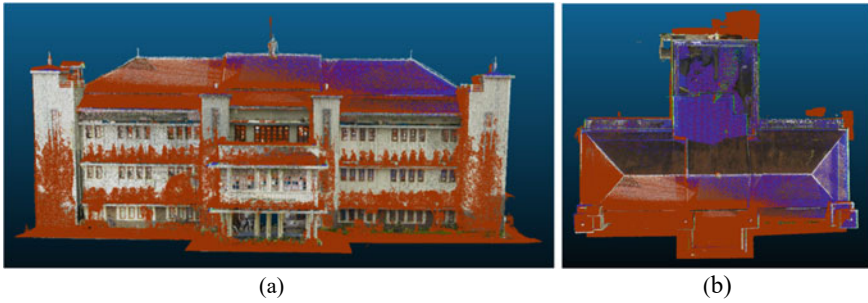


Fig. 4 3D dense model of building from ICP registration: **a** front view and **b** top view

has limitations to generate a complete 3D building model. Therefore, data fusion is applied to combine all the data so as to overcome the lack of data. The next section shows the integration process of these datasets and the evaluation of the results.

4 Experiments

4.1 ICP Registration

Before performing ICP registration, the data source (LiDAR point by drone and image-based point) were manually shifted closer to the target (TLS point) for coarse alignment. The integration data experiment applied ICP registration twice using TLS points as a reference. This resulted in joined data with the registration errors 0.927655 m for the LiDAR point by drone and 1.14988 m for the image-based point. In this case, the value shows that the LiDAR point by drone is closer to the reference than the image-based point. Thus, the joined data, as seen in Fig. 4a and b, presents the 3D dense model of the building from the TLS point (natural color), LiDAR point by drone (blue) and image-based point (brown). From the front and top views, the building looks like a complete 3D dense model, especially the rooftop part.

4.2 Surface Density and Roughness Estimation

Quantitative comparisons are presented in Table 1 to show the data quality. The number of points among the three datasets is significantly different both before and after noise removal (NR) at a radius of 0.05 m. TLS can produce the largest point data, LiDAR point by drone the smallest. This is because apart from the tool specifications, TLS can be arranged to record several scans of the object.

Table 1 Data comparison in terms of number, surface density, and roughness

| Data | Number of points | | Average surface density (points/m ²) | | Average roughness (m) | |
|----------------------|------------------|------------|--|--------|-----------------------|--------|
| | Before NR | After NR | I | II | I | II |
| LiDAR point by drone | 209,553 | 205,231 | 114 | – | 0.0046 | – |
| TLS point | 34,161,349 | 24,181,170 | 13,658 | 14,987 | 0.0022 | 0.0020 |
| Image-based point | 3,290,444 | 1,567,728 | 297 | 308 | 0.0012 | 0.0012 |

It is noted that the first dataset mostly depicts the rooftop part. For further assessment, surface density and roughness of the rooftop part (I) were calculated to compare the three datasets. However, calculations for the entire building (II) were also performed for comparison of the TLS and image-based points. Surface density value is used to observe how dense the points are in an area, whether the data provided is plentiful or vice versa. Based on Table 1, the second dataset has the largest value of average surface density (both the rooftop and the entire building), while the first dataset has the smallest value. This means that in an area of 1 m², the TLS data has the densest points for the rooftop part and the entire building as well. In terms of the average roughness, this was calculated for a 0.05 m radius, the same radius for the NR process. This estimates the “roughness” point cloud and can indicate the local measure of the noise of each data. Based on the results, the first dataset has the highest value among all the datasets. Thus, more noise (in a 0.05 m radius) is denoted in the LiDAR point by drone. Meanwhile, the image-based point has less noise for the rooftop part and the entire building.

4.3 M3C2 Distance

ICP registration is basically a process to bring data closer to the reference data. Therefore, integrated data still has distance. In this experiment, the distance was evaluated using M3C2; the results are presented in Table 2. The STD value depicts the distances between the point clouds, which are TLS and the data source. Based on Table 2, the first data pair has a larger distance than the second data pair. The M3C2 distance tool is also used to investigate the completeness of the 3D building model after data integration.

Table 2 STD of M3C2 distance

| Data pair | STD (m) |
|--------------------------------|---------|
| TLS point–LiDAR point by drone | 0.8978 |
| TLS point–image-based point | 0.7758 |

5 Results and Discussions

In this study, ICP registration was performed to integrate three datasets (LiDAR point by drone, TLS point, and image-based point) and then to investigate the results of data fusion. After the ICP registration process, the fusion model visually depicts a more complete building object. The LiDAR point by drone and the image-based point can cover the missing data (the holes) on the roof and the wall in the 3D building model of single data (TLS point). These data sources allow to complement the TLS data (both voids) at the top and on the side of the building. However, the LiDAR point by drone has a minor role to complete the model. It is not as significant as the image-based point as it provides only the top of the building with less surface density. Meanwhile, the image-based point has more surface density and offers data that includes the top and side of the building. As seen in Fig. 5a, the hole (in yellow polygon) at the rooftop is completed more by the red point (image-based point) than by the blue point (LiDAR by drone point). Also, Fig. 5b shows that the void at the wall is filled only by the image-based point.

In addition, as seen in Table 1, the LiDAR point by drone is indicated to have the most local noise and the image-based point the least local noise. This shows that

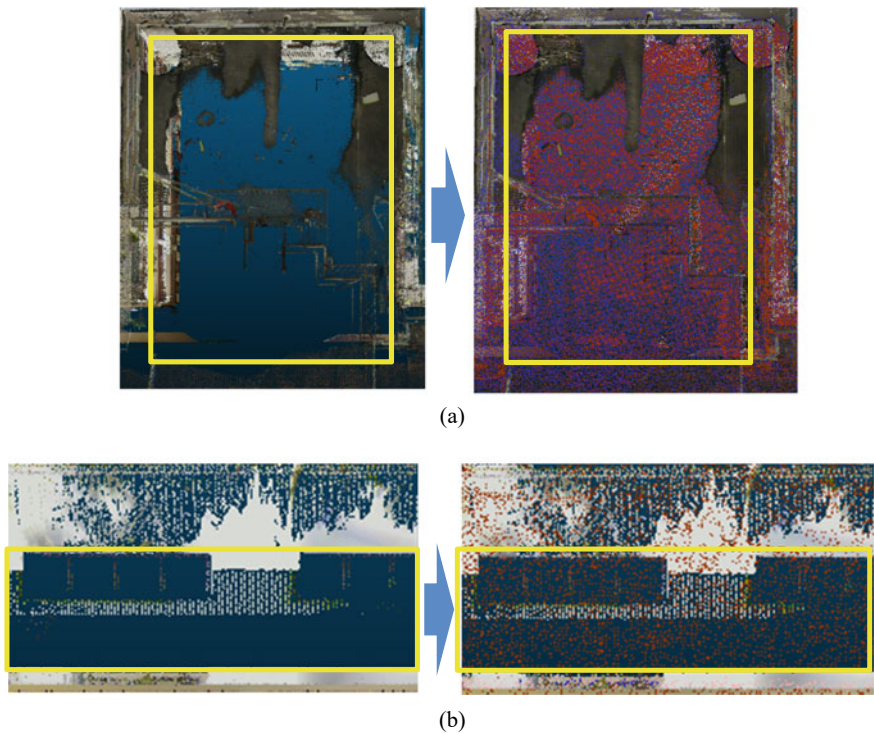


Fig. 5 Missing data on the roof (a) and the wall (b) completed by the data source

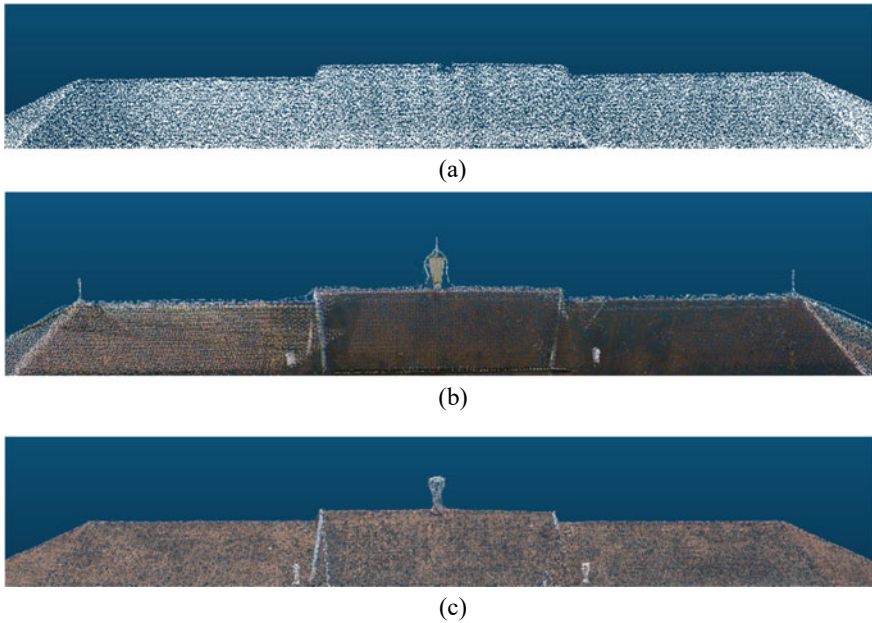


Fig. 6 Comparison of the details and sharpness of **a** LiDAR point by drone, **b** TLS point, and **c** image-based point

the image-based point provides better detail precision among all the datasets. For example, the image-based point in Fig. 6c provides rooftop detail that is missing in Fig. 6a, which is from the LiDAR point by drone. Moreover, the rooftop shape from the image-based point is sharper than the TLS point in Fig. 6b. In this case, though the TLS data has a denser point, the image-based point has better detail precision. This might be the impact of performing a high-accuracy process (image processing) to generate the dense point.

For further observation, the M3C2 distance shows the existing distances between point clouds, as indicated by the STD figure in Table 2. Furthermore, the voids in the TLS point-based model are observed to determine the completeness by data fusion. The rooftop part in Fig. 7a shows several voids on the TLS point-based model; thus, the voids can be completed by the substitute data, which are indicated by gray points in Fig. 7b from the LiDAR point by drone and in Fig. 7c from the image-based point. These gray points of M3C2 distance are the out-of-range values that can be assumed as missing data on the point cloud model.

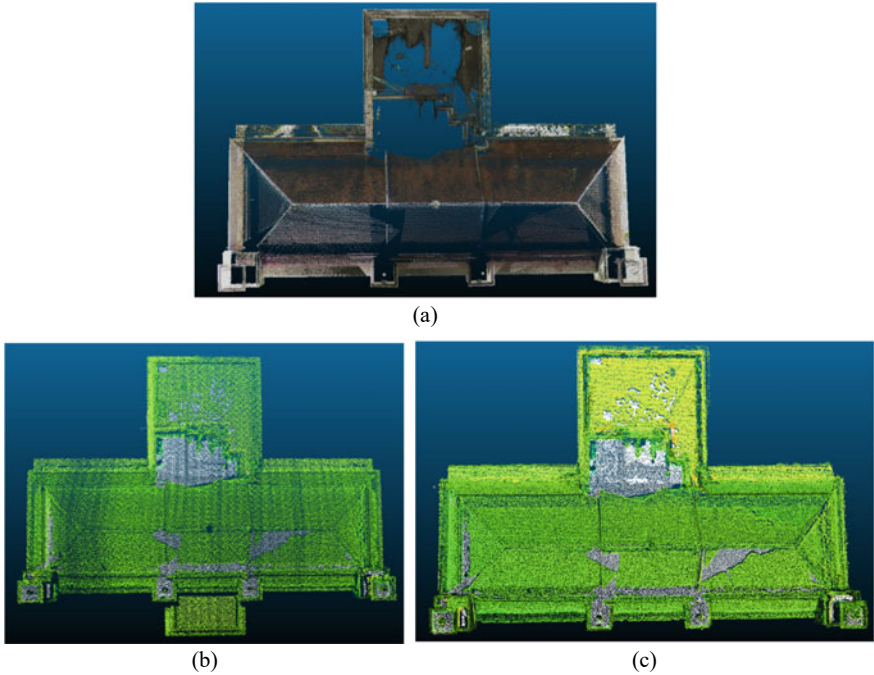


Fig. 7 The voids in the TLS point-based model (a) covered by the source data (b), (c) in gray

6 Conclusions

Three datasets (LiDAR point by drone, TLS point, and image-based point) have been fused using ICP registration in CloudCompare. This fusion shows the completeness of the 3D model. Based on the results, the source data can cover many holes in the TLS point-based model, but a few holes caused by obstructions such as vegetation and roof shadows cannot be resolved. Moreover, the LiDAR point by drone supplies less data and thus has minor role to complete the model. Hence, for future work, data from another technique, such as the MMS can be considered for data fusion. In terms of distances, the evaluation has proven that there are existing distances between point clouds. This means that to fix the holes with substitute data, the distances need to be minimized or even eliminated. Therefore, refinement to this difference is important in this case. Finally, this initial work could be extended further to generate a better fused 3D point cloud datasets and thus accurate 3D building models.

References

- Adamopoulos E, Rinaudo F (2021) Close-range sensing and data fusion for built heritage inspection and monitoring—a review. *Remote Sens* 13(3936):1–32
- Altuntas C, Yildiz F, Scaioni M (2016) Laser scanning and data integration for three-dimensional digital recording of complex historical structures: the case of Mevlana Museum. *ISPRS Int J Geo-Inf* 5(18):1–16
- Bouziani M, Amraoui M, Kellouch S (2021) Comparison assessment of digital 3D models obtained by drone-based lidar and drone imagery. In: *The international archives of the photogrammetry, remote sensing and spatial information sciences, the 6th international conference on smart city applications*, Karabuk University, Virtual Safranbolu, Turkey, vol XLVI-4/W5-2021, pp 113–118
- Castilla FJ, Ramón A, Adán A, Trenado A, Fuentes D (2021) 3D sensor-fusion for the documentation of rural heritage buildings. *Remote Sens* 13(1337):1–15
- Hua W, Qiao Y, Hou M (2020) The great wall 3D documentation and application based on multi-source data fusion—a case study of no. 15 enemy tower of the new Guangwu great wall. In: *The international archives of photogrammetry, remote sensing and spatial information sciences*, vol XLIII-B2-2020. XXIV ISPRS Congress, pp 1465–1470
- Jaber AS, Abed FM (2020) Revealing the potentials of 3D modelling techniques; a comparison study towards data fusion from hybrid sensors. In: *IOP conference series: materials science and engineering*, vol 737, issue 1. IOP Publishing, pp 012230-1–012230-10
- Li W, Wang C, Zai D, Huang P, Liu W, Wen C, Li J (2018) A volumetric fusing method for TLS and SFM point clouds. *IEEE J Sel Top Appl Earth Obs Remote Sens* 11(9):3349–3357
- Li S, Ge X, Hu H, Zhu Q (2021) Laplacian fusion approach of multi-source point clouds for detail enhancement. *ISPRS J Photogramm Remote Sens* 171:385–396
- Luhmann T, Chizhova M, Gorkovchuk D, Hastedt H, Chachava N, Lekveishvili N (2022) Combination of terrestrial laserscanning, UAV and close-range photogrammetry for 3D reconstruction of complex churches in Georgia. In: *The international archives of the photogrammetry, remote sensing and spatial information sciences*, vol XLII-2/W11, pp 753–761
- Maset E, Valente R, Iamoni M, Haider M, Fusiello A (2022) Integration of photogrammetry and portable mobile mapping technology for 3D modeling of cultural heritage sites: the case study of the Bziza temple. In: *The international archives of the photogrammetry, remote sensing and spatial information sciences, XXIV ISPRS congress (2022 edition)*, Volume XLIII-B2-2022, Nice, France, pp 831–837
- Mutiarasari WM, Chyntia NPP, Kurniawan AA, Pangestu A, Maghfiroh L, Vaherera P, Sa'id ABN (2023) LiDAR data extraction results for multi-object 3D models. *J Geoid* 19(1):41. <https://doi.org/10.12962/j24423998.v19i1.11367>
- Ren X, Zhang M, Wang W, Mao X, Ren J (2019) Fast and coarse registration of point cloud in ICP three-dimensional space. In: *Journal of physics: conference series, CISAI 2019*, vol 1453, issue 2020. IOP Publishing, pp 1–6
- Riyadi G and Prasyda AS (2019) Measurement study using terrestrial laser scanning (TLS) for 3D documentation of the UGM vocational college library building. In: *research report, Vocational College UGM, Yogyakarta, Indonesia*
- Shao J, Zhang W, Mellado N, Grussenmeyer P, Li R, Chen Y, Cai S et al (2019) Automated markerless registration of point clouds from TLS and structured light scanner for heritage documentation. *J Cult Herit* 35:16–24

Sensing Heathland Vegetation Structure from Unmanned Aircraft System Laser Scanner: Comparing Sensors and Flying Heights



Nina Homainejad , Lukas Winiwarter , Markus Hollaus ,
Sisi Zlatanova , and Norbert Pfeifer 

Abstract Low-cost lidar sensors mounted on unmanned aircraft systems (UAS) can be applied for the acquisition of small-scale forestry applications providing many advantages over airborne laser scanning (ALS), such as flexibility, low flight altitude and small laser footprint. Compared to 3D data generated from dense image matching using photogrammetry, lidar has the advantage of penetration through the canopy gaps, resulting in a better representation of the vertical structure of the vegetation. We analyse the effect of different flight altitudes on the penetration rate of heathland vegetation in the Blue Mountains, Australia using a Phoenix system based on a Velodyne Puck 16 scanner and a GreenValley LiAir X3-H system based on a Livox scanner. The different sensors achieve quite different performances, especially for the mid-vegetation layer between the canopy and the ground layer. Representation of this layer is especially important when investigating fuel availability for bushfire analyses. In this layer, the LiAir system achieves a sufficient picture at an altitude of 65 m above ground, whereas the Phoenix system needs to be flown as low as 40 m to get a comparable result.

Keywords Lidar · Point density · Penetration rate · Heathland

This article was selected based on the results of a double-blind review of the full paper.

Supported by University of New South Wales.

N. Homainejad (✉) · S. Zlatanova
University of New South Wales, School of Built Environment, Sydney 2052, Australia
e-mail: n.homainejad@unsw.edu.au

L. Winiwarter · M. Hollaus · N. Pfeifer
Technische Universität Wien, Department of Geodesy and Geoinformatics (E120), Wiedner
Hauptstraße 8–10, 1120 Wien, Austria

L. Winiwarter
University of British Columbia, Integrated Remote Sensing Studio (IRSS), 2424 Main Mall,
V6T 1Z4 Vancouver, B.C., Canada

19.1 Introduction

Fuel, defined as flammable live and dead vegetation, significantly influences the development and propagation of bushfires through its arrangement, structure and chemical composition. Fuel characteristics, such as the vegetation type and moisture content, fuel size, quantity and arrangement, can affect the fire spread, flame structure, duration, and intensity of bushfires. Therefore, the quantification and characterisation of vegetation as fuel is essential for the understanding of fire behaviour especially as fuel is the only influencing factor that can be manipulated through methods such as back burning and trimming of tree branches. These steps are important controls set for reducing the risk and the likelihood of the ignition of bushfires.

In Australia, fuel hazard assessment guides are applied for the conversion of visual vegetation assessments to indicative fuel loads, which is essential for some rate of spread models and for the calculation of fire line intensity or emissions. In the eastern states of Australia, the Victorian Government Department of Sustainability and Environment (DSE) Overall Fuel Hazard Assessment Guide (OFHAG) (of Environment 2012) and the interim Fuel Assessment and Fire Behaviour Prediction in Dry Eucalypt Forest Field Guide (Gould et al. 2008), also known colloquially as the “Project Vesta” field guide, are the two most widely applied fuel hazard assessment guides in dry eucalypt forest and shrub heath fuel types. The OFHAG and Project Vesta guide provide the necessary guidelines for the visual assessment and rating or scoring of fuel that is applied in predicting the rate of spread of fire based on other fire behaviour parameters.

The OFHAG categorises fuel structures into five layers based on their horizontal arrangement and vertical positions in forest, woodland and shrubland (of Environment 2012). These five layers include bark, next to the canopy, canopy fuel, elevated fuel (shrub fuels), near-surface fuel (suspended litter or aerated fuel) and surface fuels (litter fuel) (see Fig. 19.1). To develop maps of bushfire fuel hazards for fire risk analysis, the amount of fuel in these different layers is quantified. Such maps allow prioritising hazard reduction operations (such as prescribed burning), conducting post-burn assessments to determine effectiveness, and providing inputs into fire behaviour prediction models such as predicting flame depth and height, surface fire, rate of spread, spotting, and crown fire. However, visual fuel assessment, rating, and scoring methods are vulnerable to being subjective, inconsistent, and restricted by complex terrain (Hines et al. 2010; Sikkink and Keane 2008; Spits et al. 2017; Kelly et al. 2011). Hence, there is a need for an automated and impartial approach to the classification of fuel layers.

Lidar, both ALS and ULS, is a prominent technique for data acquisition and the characterisation and estimation of 3D vegetation elements within forested and city environments (Xu et al. 2021, 2022). Lidar point clouds acquired by Airborne Laser Scanners (ALS) and Unmanned Aircraft System Laser Scanners (ULS) are a reliable means of describing fuel structure. These techniques provide three-dimensional coordinates (x, y, z) of surface geometries, and the beams have the ability to penetrate through canopy gaps thus providing a more complete representation of the vertical structure of vegetation while allowing for the production of high-resolution

topographic maps that enable highly accurate estimation of terrain and vegetation height, cover, and canopy structure. Furthermore, this method can be used to extract different fuel layers.

ULS offers several advantages compared to ALS. Generally, ALS missions are applied on large-scale areas therefore this can hinder the cost related to ALS missions and make it more feasible to apply ULS on small-scale missions. Other key advantages of ULS over ALS include flexibility, low flight altitude and a small laser footprint. However, there are disadvantages related to ULS missions that should be taken into consideration, such as the limited battery capacity, restricted area size, and additional legal requirements for the operation of ULS, which can overall limit its usability in large-scale missions.

In this study, the use of lidar on heathland fuel properties is investigated. Heathland occupy around 2% of the Australian continent. They are mostly localised, but scattered widely throughout the southeast and southwest of the continent, with outliers on tropical sandstone plateaus and sandplains. The vegetation in this environment is structurally complex, vertically non-uniform, and discontinuous and occurs in patches of three to four hectares (Williams et al. 2012; Hammill et al. 2010). Heath shrubs ranging between 0.5 m and 2 m predominately occupy this environment. The presence of trees is typically absent or present as mallee – a multi-stemmed eucalyptus. Emergent trees are no taller than 12 m while mallee range between 5 m and 10 m. The near-surface stratum is comprised of tussock and hummock grasses, ephemeral herbs, low sedges, low shrubs and dead suspended material. The near-surface stratum in this vegetation ranges between 0.1 m and 0.3 m (Williams et al. 2012).

This vegetation group is considered one of the most fire-prone vegetation groups in the world (of New South Wales 2022). The presence of flammable terpenes and waxes in the foliage of some shrubs in the elevated fuel layer and the mallee in the canopy fuel layer promotes the combustion of live fuel components making heathland highly flammable and fire-prone throughout much of the year. The near-surface fuel layer in this environment also plays a critical role in fire spread. Bushfires in heathland will not spread without the presence of near-surface fuel layer unless the bushfire is burning under very severe fire weather conditions. This is because the bulk density of the elevated fuel layer which is composed of heath shrubs is low. In the event of a bushfire, the transition of fire from a surface fire to a crown fire is facilitated – making fire behaviour in these environments complicated (Williams et al. 2012).

Most often, detailed information about heathland fuel layers and vegetation structure is missing. Therefore, there is a great need for more precise data as input for bushfire behaviour models (Anderson et al. 2015). This study aims to assess how ULS flying altitude of the Phoenix Puck Velodyne 16 and GreenValley LiAir X3-H ULS sensor systems impact the representation of heathland structure in three different fuel layer structures in the GBMWA (see Fig. 19.1). To achieve this objective, a point density and a voxel-grid based occupancy mapping methodology is conducted for a comparative analysis of the Phoenix Puck Velodyne 16 and GreenValley LiAir X3-H ULS systems flown at varying heights (40 m, 50 m, 65 m, and 80 m) above ground level. Through this analysis, we investigated the changes in point density and its impact on the representation of three distinct fuel layers, namely, the near-

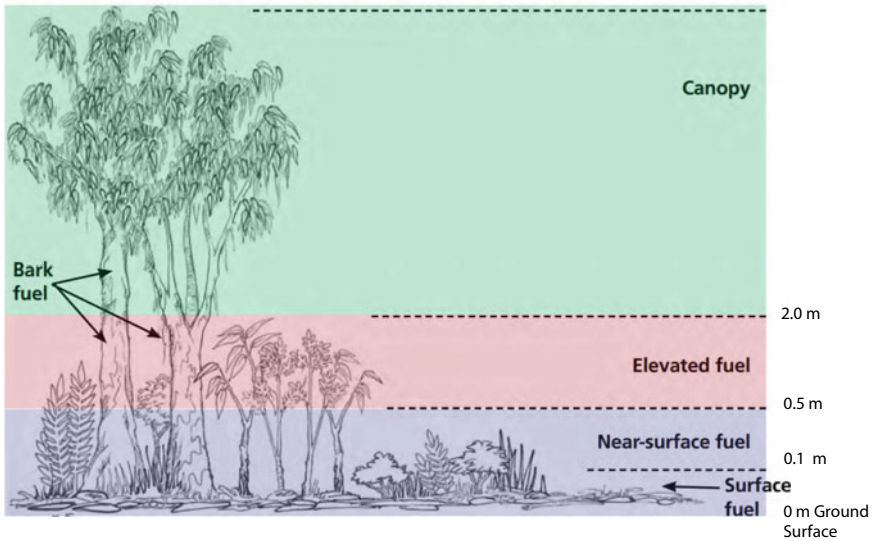


Fig. 19.1 Fuel layers and bark division for forests, woodlands and shrublands as per the Overall Fuel Hazard Assessment Guide. Figure modified from (of Environment 2012)

surface, shrub, and tree fuel layers (coloured in Fig. 19.1 in blue, red, and green, respectively). Furthermore, the data is compared with reference data acquired by a handheld mobile laser scanning system using a voxel-based approach. The findings from this study reveal the most appropriate flight altitude for the 3D modelling of the different heathland fuel layers in the GBMWA.

19.2 Methodology

The purpose of this analysis is to examine how flight altitude can impact the point density of the lidar point clouds in the three different heathland fuel layers, namely the near-surface, elevated and canopy fuel layers. Therefore, it is necessary to identify the most suitable flight height for acquiring the structures of the heathland for 3D modelling based on their point density distribution and count within the three fuel layers.

The method for analysing the vegetation structure of heathland vegetation group, outlined in Fig. 19.2, using ULS first requires the normalisation of height data. This step is required to accurately extract point density information in different height layers above the digital terrain model (DTM) (Hodgson and Bresnahan 2004). As the two datasets are not in the same height system, the preprocessing involves the conversion of ellipsoidal heights (Z) to orthometric heights (H). The correct estimation of the DTM is essential to ensure an unbiased estimation of vegetation height. Failure to accurately estimate the DTM can lead to an underestimation of canopy height or false representation of vegetation elements to different vertical layers prohibiting a direct comparison of their point cloud representation and thus resulting in errors of several decimetres.

Following the normalisation of the height, the fuel layers are vertically split into three different bins based on their height above the ground. The OFHAG is used as a reference source for this work (see Fig. 19.1). The near-surface layer for heathland is considered as all fuel between 0.1 m and 0.5 m above ground, the elevated fuel layer is considered as fuel between 0.5 m and 2 m, and the canopy fuel layer is considered as all fuel above 2 m.

For this approach, the data is horizontally sliced into the three different fuel layers by specifying the fuel heights. After separating the three fuel layers, the point density in each fuel layer is compared to the sum of the point densities. A raster-based approach is applied to analyse the point density of the lidar point clouds within the different fuel layers. Analysing the data in a regular raster structure is a convenient way to retrieve summary information about the data on a per-cell basis.

Following the extraction of the point densities for the different flight altitudes collected by the two different systems, the relative proportion of returns per vegetation layer (in relation to the full vertical column) is calculated. These relative proportions exhibit information on the representation of vegetation in the different fuel layers, represented in the dataset based on the different systems and the flight altitude, independent of the total point densities. The different relative proportions of the three extracted fuel layer point densities are compared visually. Through this analysis, an optimal configuration for 3D heathland modelling in support of bushfire behaviour modelling is recommended.

Moreover, the point distribution observed in the different flights are compared with ground reference data in a systematic manner. Voxel-based modelling is a widely utilised approach in various applications involving point clouds (Xu et al. 2021), demonstrating significant potential and value, particularly in vegetation modelling (Gorte and Pfeifer 2004; Barton et al. 2020; Eusuf et al. 2020; Homainejad et al. 2022), Homainejad et al. (2022a, b). Voxels provides a structured and efficient method for representing three-dimensional objects in a three-dimensional array. This approach leads to reduced processing time and memory usage. However, it's important to note that the utilisation of voxels can generate extensive datasets, necessitating the implementation of a specialised data structure (Aleksandrov et al. 2021, 2022; Gorte 2023). This methodology combines qualitative information, such as discrete vegetation layers, with quantitative information, such as volume, facilitating comprehensive analysis and representation of complex three-dimensional data. A voxel-based approach similar to Homainejad et al. (2022a, b) is applied to facilitate this comparison, enabling the structured representation of three-dimensional objects within a topologically explicit three-dimensional array. This approach offers advantages such as enhanced computational efficiency and optimised memory utilisation. In order to conduct the comparison, a voxel size is defined, that can be applied to all the different datasets, allowing for the voxelisation of the point cloud data. The optimal voxel size is selected based on the average data point density among the datasets. Subsequently, the focus of the analysis is narrowed down to the voxel occupancy within the near-surface and elevated fuel layers. The ground reference data is acquired using a terrestrial mobile lidar hand scanner, resulting in comparatively

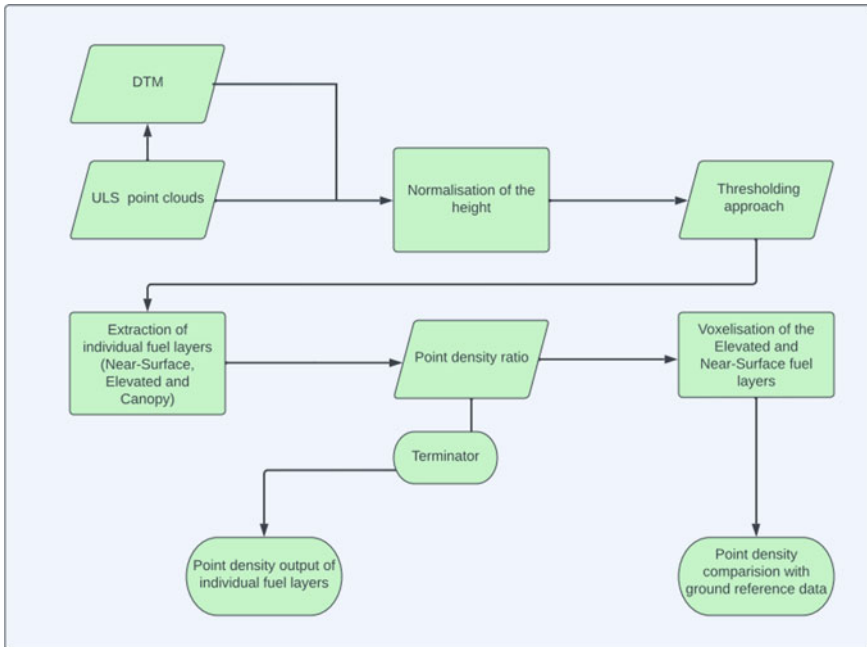


Fig. 19.2 Flowchart displays the workflow of processing lidar point densities in different fuel layers

fewer returns within the canopy layer when compared to the point cloud densities obtained from the ULS.

The presence or absence of voxels is determined by evaluating each layer and the corresponding occupancy in relation to the ground reference data. This assessment involves the quantification of true positives (TP), false negatives (FN), false positives (FP), and true negatives (TN) which are subsequently utilised to calculate measures of accuracy, precision, recall, and F1-score for each layer. By following this approach, a thorough understanding of the congruence between the ULS point cloud densities and the ground reference data is achieved, enabling valuable insights into the accuracy and reliability of the collected information within each respective layer.

19.3 Study Area and Datasets

19.3.1 Description of the Study Area

The Greater Blue Mountains World Heritage Area (GBMWhA) was established in 2000 by integrating eight national parks in New South Wales (NSW), Australia. This

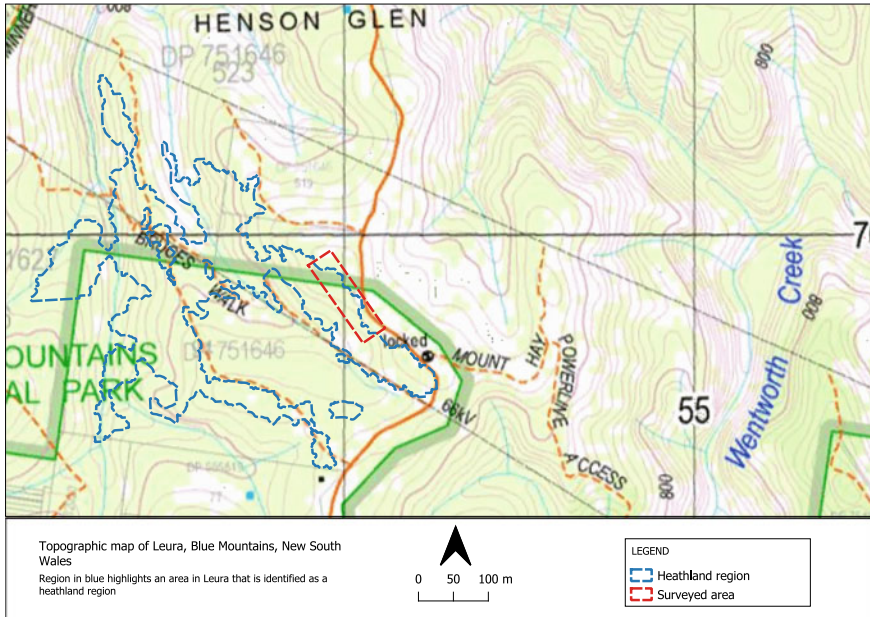


Fig. 19.3 Topographic map of Mount Hays, Leura acquired from (of New South Wales 2023)

vast area of approximately one million hectares is the largest integrated system of protected areas in NSW. It boasts a rich diversity of vegetation, including a remarkable 14 % of all eucalypt taxa identified in Australia (850 species) are found in the GBMWA, as documented by (Chapple et al. 2011). Heathland is also present, covering less than 2 % of the area.

Considering that heathland only occupy a small region of the GBMWA, correctly identifying heathland regions in the Blue Mountains is crucial for the study. To achieve this, the NSW Government's Department of Planning and Environment State Vegetation Type Map (SVTM) (of New South Wales 2023), which classifies vegetation in NSW into three hierarchy levels and maps the distribution of each plant community type, vegetation class, and vegetation formation across all tenures in NSW, was used to select the study region for the ULS flights.

A region along Mount Hays Road in Leura, NSW, was chosen for the survey flights. This region has an area of 32 ha, which is dominated by continuous Montane heathland (see Fig. 19.4). The terrain in this region is steep but accessible via Mount Hays Rd (see Fig. 19.3). The canopy is open, with the lower canopy occupied by heath shrubs that are around 2 m tall, while the upper canopy cover is populated with trees or mallee. Thus, this region was selected as a suitable study region for the ULS flights.

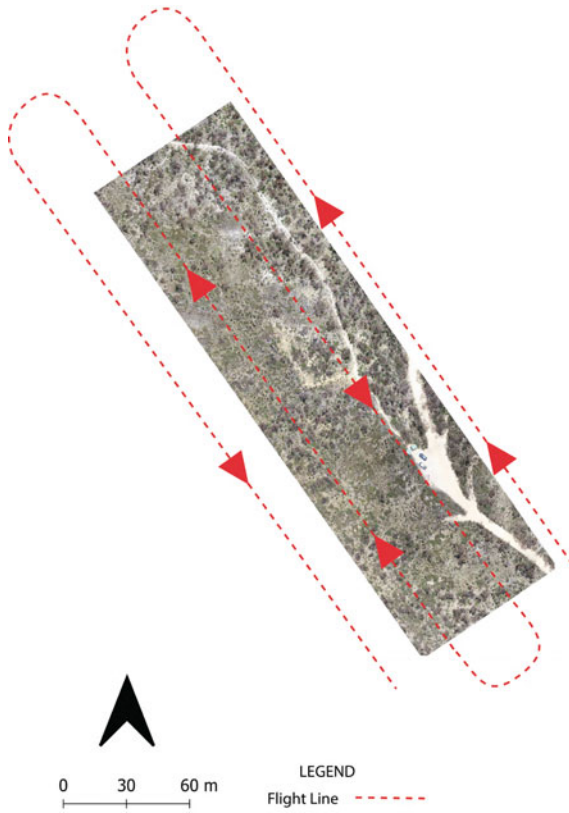


Fig. 19.4 Schematic flight plan trajectory of the study region in Mount Hays Rd, Leura

A series of ULS flights were carried out in this area on the 24th of November 2022 at a latitude of 33.6833°S and longitude of 150.3469°E, using two different UAS: 1) a Phoenix Scout using a Velodyne Puck 16 and 2) a Green Valley LiAir X3-H using a Livox Avia sensor. They were used for a series of flights at four different altitudes: 40 m, 50 m, 65 m, and 80 m above ground level (AGL), each flight covering an area of 500 m by 100 m at speeds of 10 m/s for the Phoenix Scout and 5 m/s for the Green Valley system. The weather conditions were clear, with temperatures of 15° - 25° C, and westerly and easterly wind at speeds varying from 13 km/h to 33 km/h (Bureau of Meteorology (BoM) 2020).

19.3.2 Sensors

19.3.2.1 Phoenix Scout with Velodyne Puck 16

The Velodyne VLP-16 Puck sensor is a dual return sensor with 16 laser channels. The 16 laser channels are aligned between -15° and $+15^\circ$ in 2° intervals from the scanner’s horizontal (or xy-) plane, resulting in a 30° vertical field of view (FOV) and a 360° horizontal field of view. The laser channels are stacked vertically with each laser emitting every $2.304 \mu\text{s}$. After the emission of all 16 lasers, there is a recharge period of $18.43 \mu\text{s}$. This sensor is mounted downwards looking (i.r rotated by 90° on a Phoenix rotary UAS). Thus the track of each laser on the ground follows the path of a hyperbola, and the forward motion of the platform adds a slight affine distortion to each hyperbola. In consequence, the resulting point distribution is not homogeneous. The Velodyne Puck 16 scanner produces point clouds with a higher density towards the centre and less dense towards the edge of the flight lines scans (see Fig. 19.5). Additional sensor specifications are listed in Table 19.1. The scanner operates in the near-infrared, at a laser wavelength of 905 nm.

19.3.2.2 GreenValley LiAir X3-H

The GreenValley LiAir X3-H system uses a triple return Livox Avia laser scanner. The Livox Avia supports a mode of non-repetitive scanning and a mode of repetitive scanning. The non-repetitive scanning provides a denser scan for the centre of the

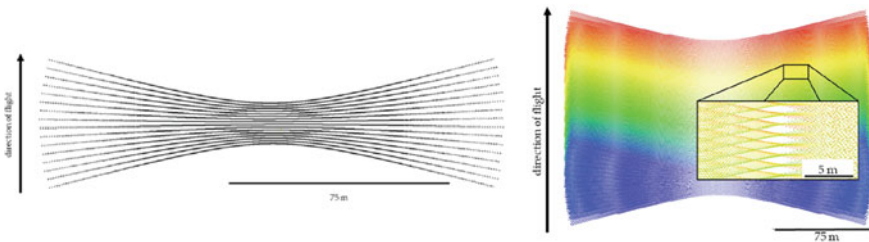


Fig. 19.5 Scan pattern of a Velodyne Puck sensor with 16 lasers over a target above the surface. Image obtained from (Lassiter et al. 2020)

Table 19.1 Livox avia and Velodyne puck 16 sensor specification

| Sensor specs | Livox avia | Velodyne puck 16 |
|------------------|--------------------------------|----------------------------------|
| Echo returns | Triple return | Dual returns |
| Range | 450 m | 100 m |
| Accuracy | 2 cm | 3 cm |
| Footprint | $0.28^\circ \times 0.03^\circ$ | $0.172^\circ \times 0.086^\circ$ |
| Weight | 498 g (without cables) | 830 g (without cables) |
| Laser wavelength | 905 nm | 905 nm |

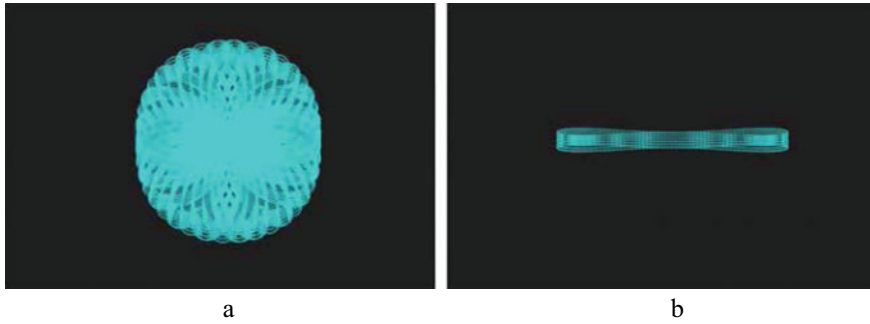


Fig. 19.6 Scan pattern of the Livox Avia from left to right, **a** non-repetitive circular scanning, **b** repetitive line scanning over a target above the surface. Figure reproduced from (LIVOX 2023)

FOV compared to the surrounding area. This system uses the repetitive scanning mode, with a horizontal FOV of 70.4° and a vertical FOV of 4.5° (see Fig. 19.6) and a scan pattern repeat cycle of 0.1 s (LIVOX 2023). The multi-line laser and high-speed scanning rate enable the Livox Avia scanner to achieve a point cloud acquisition rate of 240,000 points/s from first returns only. The Livox Avia laser scanner has been affixed to a rotary unmanned aerial system (UAS) integrated with a high-resolution camera featuring a 16 mm fixed focal length, thereby enabling the capture of red, green, and blue (RGB) images in addition to the infrared (IR) intensity data of the lidar system and operates at the same laser wavelength of 905 nm as the Velodyne sensor. Additional sensor specifications are tabulated in Table 19.1.

19.3.2.3 LiGrip

LiGrip is a rotating handheld Simultaneous Localisation And Mapping (SLAM) lidar System integrating a Velodyne VLP-16 Puck LiDAR sensor, similar to the Phoenix system, and an HD camera with a resolution of 4000×3000 . The lidar sensor and the camera are mounted on a rotating module. The handheld device weighs around 1.5 kg without a battery. The LiGrip handheld lidar sensor rotates 360° , revolving internally around the z-axis with a horizontal FOV of $0^\circ - 280^\circ$ and a vertical FOV of $0^\circ - 360^\circ$. One full revolution takes 0.1 s and the point clouds are acquired during constant motion of the sensor with a relative accuracy of ≤ 3 cm. The LiGrip handheld lidar sensor achieves a point cloud acquisition rate of 300,000 points/s.

19.3.3 Data Properties

The lidar point cloud attributes recorded by the sensors in this study include the x-, y-, and z-Coordinates, the echo number (within each laser shot), the number of echoes (total for each laser shot), the intensity, RGB colour (if applicable), and a time stamp.

Additional attributes can be calculated from the point coordinates. Of the many point cloud attributes, point density is an important lidar mapping aspect. Point density is impacted by a number of factors that include the scan pattern and the variations along the scan line, flight speed, influences referred to as banding, variation due to aircraft altitude, including effects from a constant flight height for uneven, rough surfaces and steep surfaces; swath overlap, the surface material properties and obstructions during the data acquisition as well as variation due to vegetation, which can all result in changes in the point density. The determination of point density is contingent upon the sensor utilised and specifically, the pulse rate and geometry of the emitted pulses from said sensor (Otepka et al. 2013).

The altitude at which an aircraft conducts a lidar point cloud acquisition is a crucial determinant of the point density distribution within the resulting dataset. In cases where the altitude is excessively low, a narrow swath may ensue, precluding overlap with or contact with adjacent swaths. Moreover, in rugged and uneven terrain, such as regions with steep ridges, a uniform flight height (set above sea level) can result in variable densities between higher and lower regions. Additionally, steep slopes may result in inadequate surface point capture due to the flat incidence angles of the laser on the surface. Despite the equivalent horizontal density, sloped areas contain fewer points covering a given surface area compared to flat terrain.

The echo number provides information on the return number of a certain point (first, last, intermediate) while the number of echos provides information on the total number of returns for the respective laser pulse.

The intensity or amplitude of point clouds is an attribute that provides valuable information on the backscattering characteristics and signal return strength of the data. When calibrated appropriately, this attribute can yield quantitative data on the backscattering properties of objects. Objects with low backscattering properties, such as tall vegetation (trees and shrubs), can be distinguished from those with strong backscattering properties, such as buildings, roads, and grassland. It is worth noting that grassland, due to its height, exhibits a unique behaviour where multiple returns are uncommon. The complex three-dimensional structure of tree canopies typically results in multiple surfaces being hit by the laser pulses, leading to more lidar returns and a higher point density in contrast to contrast grassland. Consequently, this attribute can be employed in the classification of urban areas such as road surfaces, buildings, and grass surfaces, as well as distinguishing them from vegetation. However, when a laser beam only partially hits a target, as is often the case in vegetated areas due to the small size of objects (i.e., branches, leaves) with respect to the laser beam footprint, the backscattered energy and, consequently, the intensity measure is decreased. However, partial hits of the beam allow for multiple echoes in the first place, which give a much clearer picture of the vertical distribution of vegetation.

19.3.3.1 Phoenix Using Velodyne Puck

The properties of the different datasets acquired by the Velodyne Puck sensor such as the point cloud density, intensity, echo number and number of echo are examined.

The point cloud density varies between the flights at different altitudes. The 40 m flight height results in a high point density, with lower point densities as the flight height increases. Table 19.2 presents an overview of the point cloud density counts for different flight altitudes, indicating that the point cloud count and point cloud density pts/m² are highest at a 40 m flight altitude. However, as the flight altitude

Table 19.2 GreenValley LiAir X3-H and Velodyne Puck 16 total point count and point cloud density distribution pts/m²

| Sensor | Flight Altitude(m) | Total point count (million points) | Point density (pts/m ²) |
|------------------------|--------------------|------------------------------------|-------------------------------------|
| Velodyne Puck 16 | 40 | 7 | 279 |
| | 50 | 4 | 166 |
| | 65 | 2 | 53 |
| | 80 | 0.53 | 12 |
| GreenValley LiAir X3-H | 40 | 120 | 4754 |
| | 50 | 77 | 3059 |
| | 65 | 70 | 2876 |
| | 80 | 35 | 1399 |

Table 19.3 Percentage of echo numbers for each flight

| Sensor | Flight altitude (m) | Echo number | Density of echo returns (percentage) |
|------------------------|---------------------|-------------|--------------------------------------|
| Velodyne Puck 16 | 40 | 1 | 96.1 |
| | | 2 | 3.9 |
| | 50 | 1 | 99 |
| | | 2 | 1 |
| | 65 | 1 | 99.9 |
| | | 2 | 0.1 |
| | 80 | 1 | 100 |
| | | 2 | 0 |
| GreenValley LiAir X3-H | 40 | 1 | 75.6 |
| | | 2 | 21.8 |
| | | 3 | 2.5 |
| | 50 | 1 | 89.1 |
| | | 2 | 10.6 |
| | | 3 | 0.3 |
| | 65 | 1 | 87.1 |
| | | 2 | 12.6 |
| | | 3 | 0.3 |
| | 80 | 1 | 85.2 |
| | | 2 | 14.4 |
| | | 3 | 0.4 |

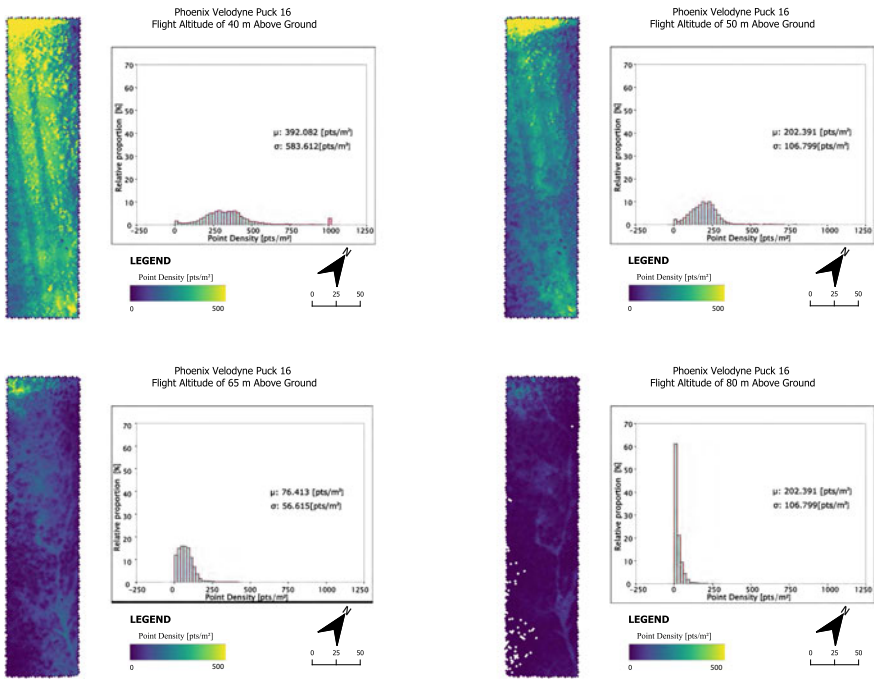


Fig. 19.7 Point density distribution of the Velodyne Puck 16 point clouds using a 2×2 m cell size. Regions with lower point density exhibit a darker hue, while the colour gradually lightens as the density of points increases

increases, both the point count and point density pts/m² decrease. The majority of the echo returns for all four flights are single returns (see Table 19.3). The point density distribution in point clouds provides valuable information on the spatial arrangement of the data. The histograms generated for each of three of the flights, 40m, 50m and 80m flight altitudes, are more right-skewed for the exception of the 65 m flight altitude that displays a uniform distribution (see Fig. 19.7).

The Velodyne Puck sensor provides intensity information but due to calibration issues, it was unable to provide calibrated intensity information. The Velodyne Puck is also not equipped with an R,G,B camera and therefore no colour information was available for the lidar point clouds.

19.3.3.2 Green Valley

Despite being a triple return sensor, the Livox Avia sensor’s third echo return accounts for less than 4 % of the total returns at a flight height of 40 m and less than 1 % for all other flights. In contrast, the second echo return is visibly higher at a flight height of

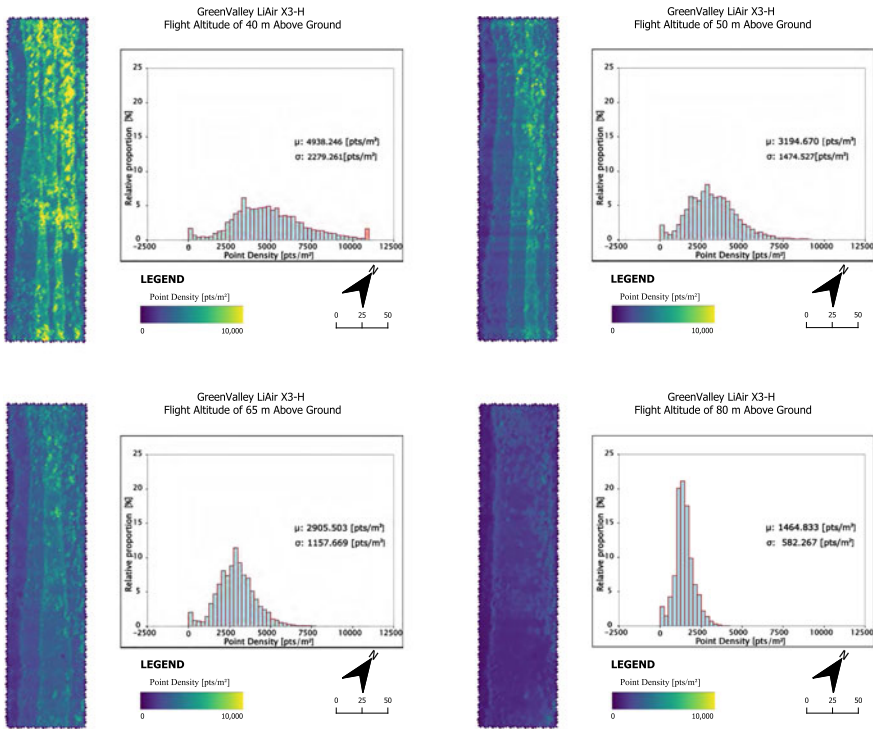


Fig. 19.8 Point density distribution of the GreenValley LiAir X3-H point clouds using a 2×2 cell size. Regions with lower point density exhibit a darker hue, while the colour gradually lightens as the density of points increases

40 m, contributing to 25 % of the echo returns, but is comparatively lower for other flights (see Table 19.3).

The analysis of point cloud density captured by the Livox Avia lidar sensor reveals noticeable variations among the four conducted flights. Table 19.2 provides a representation of the point counts and the point cloud distribution pts/m². The 40 m flight height displays a higher point cloud density compared to the higher altitude flights, especially when contrasted against the 80 m flight height and the point count and the point cloud density decreases as the flight altitude increases.

The histograms for the point densities resulting from the LiAir X3-H sensor display a bell shape curve. The distribution is symmetric around its mean, with the majority of values falling close to the mean and gradually decreasing in frequency towards the tails of the distribution. With a mean value of 4938 pts/m² for the 40 m flight altitude, 3194 pts/m² for the 50 m flight altitude, 2905 pts/m² for the 65 m flight altitude and 1464 pts/m² for the 80 m flight altitude (see Fig. 19.8). These findings provide valuable insights into the point cloud characteristics and demonstrate the variation in point densities across different flight altitudes.

19.4 Results and Discussion

Figure 19.9 displays the proportion of returns in the three different fuel layers of the heathland vegetation group, the near-surface, the elevated and the canopy fuel layer of the Phoenix Velodyne Puck 16 and the GreenValley LiAir systems. The figure shows the point densities for the flights at different altitudes for both sensors, separated by the different vegetation layers and normalised by the total point density in each vertical column. Overall, the Phoenix Velodyne system (see Fig. 19.9 top) shows a higher return rate for lower height levels, especially with increasing flight altitudes. This especially shows with a high number of pixels where no points are recorded in the canopy- and the elevated layer for 65 m and 80 m flight altitude (represented in green and red colours, respectively). Consequently, the ground layer of the heathland region is well represented and contains most of the points from these altitudes.

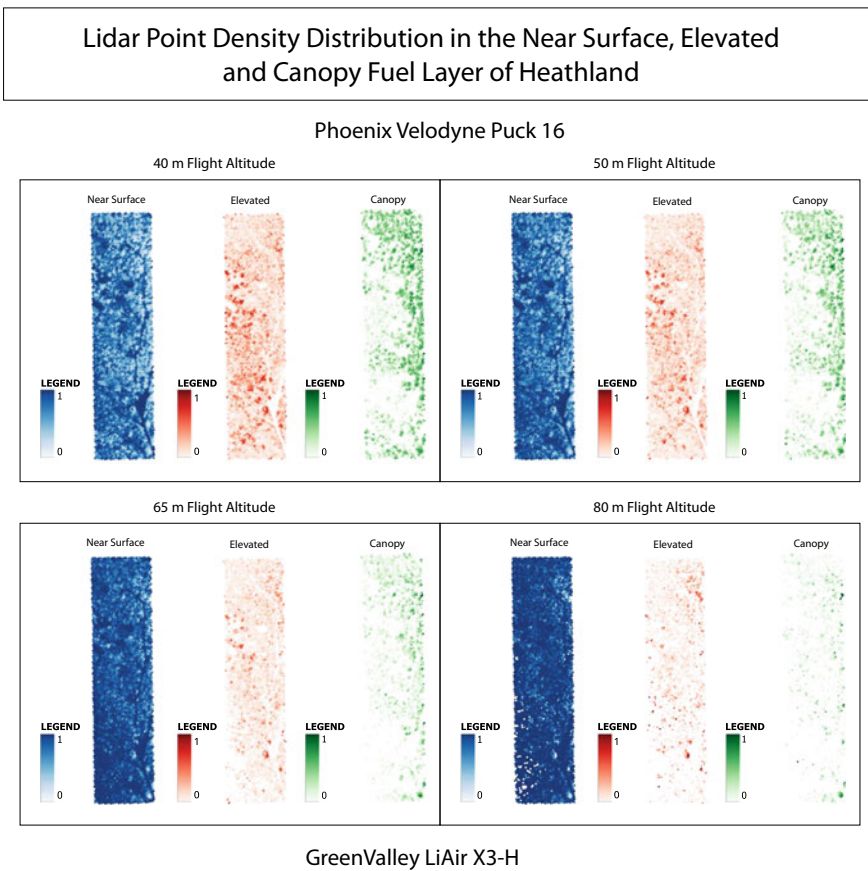


Fig. 19.9 Proportion of returns in the near-surface, elevated and canopy fuel layer at a 2 × 2 m cell size

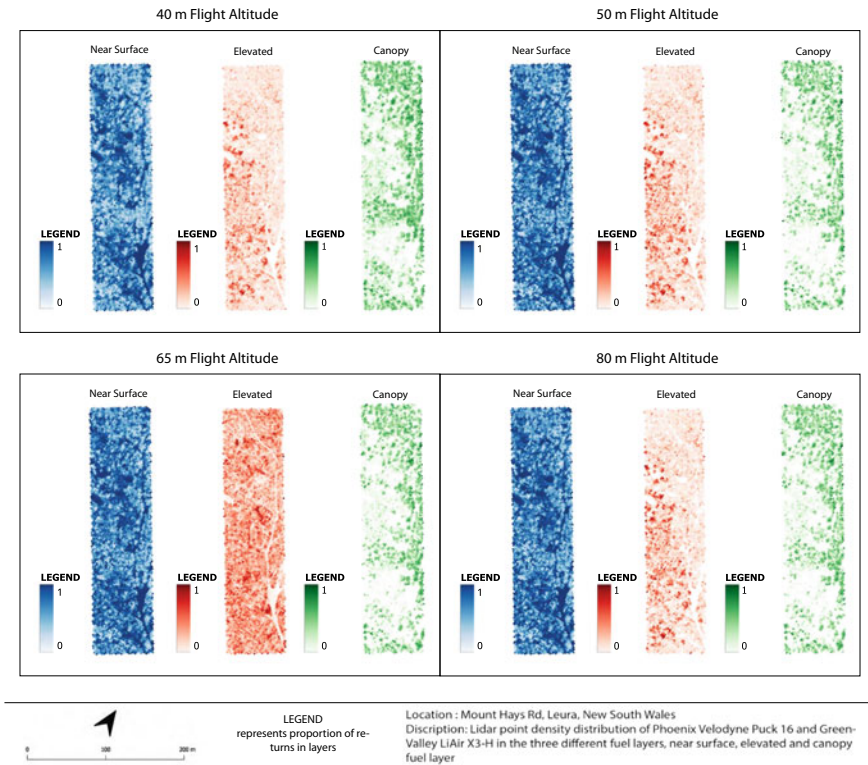


Fig. 19.9 (continued)

For the GreenValley LiAir system, a different pattern is observed: at a flight altitude of 65 m above ground, the elevated layer exhibits the highest return counts, corresponding to a relative increase of representation in this layer. Interestingly, the roads show up as low values in this plot, as the elevated layer is typically kept free of any branches or vegetation, while the canopy layer can overlap with the roads (cf. Figure 19.4). This pattern, while present for other altitudes, is observed most prominently at this altitude.

Much of the difference between the two sensors can be explained by the overall average point densities, e.g., the Phoenix Velodyne system achieves a value of 46 pts/m² at 65 m altitude, whereas the GreenValley LiAir system records 2876 pts/m². This difference is mainly caused by the different field of view—for the Velodyne system, more than 50% of the laser shots cannot be used as they are sent above the horizon due to the 360° vertical field of view. In the case of the Phoenix, the near-surface layer consistently exhibits significantly higher prominence compared to the elevated and canopy fuel layers. This distinctive behaviour can likely be attributed to dual return sensor. The pulse repetition rate, however, is similar between the two systems (240,000 pulses/s for the LiAir system and 300,000 pulses/s for the Phoenix

| | Overall Acc. | Recall | Precision | F1-Score |
|----------------------------------|--------------|--------|-----------|----------|
| Phoenix 40 m Near Surface Fuel | 0.91 | 0.71 | 0.87 | 0.78 |
| Phoenix 40 m Elevated Fuel | 0.86 | 0.39 | 0.94 | 0.55 |
| Phoenix 50 m Near Surface Fuel | 0.91 | 0.69 | 0.93 | 0.80 |
| Phoenix 50 m Elevated Fuel | 0.83 | 0.24 | 0.95 | 0.39 |
| Phoenix 65 m Near Surface Fuel | 0.90 | 0.64 | 0.98 | 0.77 |
| Phoenix 65 m Elevated Fuel | 0.80 | 0.06 | 0.93 | 0.11 |
| Phoenix 80 m Near Surface Fuel | 0.80 | 0.29 | 0.75 | 0.42 |
| Phoenix 80 m Elevated Fuel | 0.78 | 0.00 | 0.57 | 0.01 |
| GreenValley 40 Near Surface Fuel | 0.93 | 0.77 | 0.95 | 0.85 |
| GreenValley 40 Elevated Fuel | 0.91 | 0.58 | 0.90 | 0.71 |
| GreenValley 50 Near Surface Fuel | 0.93 | 0.72 | 0.99 | 0.83 |
| GreenValley 50 Elevated Fuel | 0.88 | 0.40 | 0.98 | 0.56 |
| GreenValley 65 Near Surface Fuel | 0.93 | 0.71 | 1.00 | 0.83 |
| GreenValley 65 Elevated Fuel | 0.86 | 0.37 | 0.97 | 0.54 |
| GreenValley 80 Near Surface Fuel | 0.91 | 0.65 | 0.99 | 0.78 |
| GreenValley 80 Elevated Fuel | 0.85 | 0.29 | 0.98 | 0.44 |

Fig. 19.10 The results from the ground reference comparison displaying the overall accuracy (OA), the recall, precision and the F1 scores

system). Additionally, the GreenValley system seems to be more sensitive to the partial—potentially weak—returns caused by vegetation layers above the ground, effectively recording a large number of points there.

Comparison of the voxel occupancy of the ULS layers to the ground reference data in this study reveals comparable results between the Phoenix 50 m flight altitude and the GreenValley lidar systems in the context of voxel-based analysis for vegetation fuel characterisation. The findings reveal that the Overall Accuracy (OA) and F1 of the Phoenix 50 m is closely aligned with the results obtained from the GreenValley system (Fig. 19.10). Notably, the performance of the Phoenix system appears to be significantly influenced by flight altitude, in contrast to the GreenValley system, which exhibits more consistent results across different flight altitudes. Analysis of elevated fuel layers highlight the advantage of flying at lower flight altitudes for both lidar sensors. The recall values for the elevated fuel, indicating the proportion of occupied voxels in the reference data that are successfully identified in the ULS data, do not exceed 40% for the Phoenix system. This suggests the presence of a potential issue, possibly attributed to the dual return sensor. While the performance of the GreenValley system in this regard is relatively better, it is not without its limitations. Particularly, flying at excessively low altitudes (e.g., 40 m) with the GreenValley system leads to a decline in precision, particularly for the elevated fuel. This suggests

a higher number of filled voxels in the ULS data compared to the ground reference, potentially hinting at incomplete representation in the reference data rather than a direct correlation with flight altitude.

Overall, these findings suggest that the choice of flight altitude and sensor system has a significant influence on the accuracy of the data obtained for different fuel layers in the Blue Mountains' heathland vegetation groups. Researchers and practitioners studying or monitoring these vegetation groups should carefully consider these factors to ensure reliable and accurate data collection and analysis.

19.5 Conclusion

This paper compared two sensors at four different flight altitudes each, to investigate the penetration capabilities and the ability to map the elevated vegetation structure for heathland vegetation group in the Greater Blue Mountains World Heritage Area. The investigation found that a high point density typically correlated with good penetration and that a focused FOV performed better than a 360° FOV. Additionally, it is concluded that partial returns from the canopy get too weak for the sensors to robustly detect for too high flight altitudes, resulting in an over-representation of the ground layer.

Flight altitude emerged as a critical variable influencing data quality. The study reveals that excessively high altitudes can result in weakened returns from the elevated and canopy fuel layers of heathland vegetation group. Hence, careful consideration of flight altitude is paramount when aiming to acquire ULS point clouds of heathland and similar environments, as improper altitude selection can compromise the accuracy and completeness of the captured vegetation structure.

In light of these findings, this paper provides practical recommendations for researchers and practitioners involved in mapping heathland vegetation groups. Specifically, the Phoenix system at a flight altitude of 40m is recommended for acquiring precise near-surface vegetation data, while the GreenValley LiAir system demonstrates superior accuracy in capturing information pertaining to the elevated fuel layers.

Overall, the results of this study underscore the importance of tailoring flight altitudes and sensor selection to the specific characteristics of heathland vegetation in order to achieve reliable and accurate data collection.

References

- Aleksandrov M, Zlatanova S, Heslop DJ, Voxelisation and voxel management options in unity 3D X-4/W2-2022:13–20. <https://doi.org/10.5194/isprs-annals-x-4-w2-2022-13-2022>
- Aleksandrov M, Zlatanova S, Heslop D (2021) Voxelisation algorithms and data structures: a review. *Sensors* 21(24):8241

- Anderson WR, Cruz MG, Fernandes PM, McCaw L, Vega JA, Bradstock RA, Fogarty L, Gould J, McCarthy G, Marsden-Smedley JB (2015) A generic, empirical-based model for predicting rate of fire spread in shrublands. *Int J Wildland Fire* 24(4):443–460
- Barton J, Gorte B, Eusuf MSRS, Zlatanova S (2020) A voxel-based method to estimate near-surface and elevated fuel from dense lidar point cloud for hazard reduction burning. *ISPRS Ann Photogram Remote Sens Spatial Inf Sci VI-3/W1-2020*:3–10. <https://doi.org/10.5194/isprs-annals-vi-3-w1-2020-3-2020>
- Chapple RS, Ramp D, Bradstock RA, Kingsford RT, Merson JA, Auld TD, Fleming PJS, Mulley RC, Integrating science into management of ecosystems in the greater blue mountains 48(4):659–674. <https://doi.org/10.1007/s00267-011-9721-5>
- Environment, D., Resources, N.: Overall Fuel Hazard Guide for South Australia, second edn
- Eusuf, M.S.R.S., Barton, J., Gorte, B., Zlatanova, S.: Volume estimation of fuel load for hazard reduction burning: First results to a voxel approach. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIII-B3-2020*, 1199–1206 (aug 2020). 10.5194/isprs-archives-xliii-b3-2020-1199-2020
- Gorte, B.: Analysis of very large voxel datasets **119**, 103316. 10.1016/j.jag.2023.103316
- Gorte, B., Pfeifer, N.: Structuring laser-scanned trees using 3d mathematical morphology. vol. 35, pp. 929–933. Citeseer (2004)
- Gould, J.S., McCaw, W., Cheney, N., Ellis, P., Knight, I., Sullivan, A.: Project Vesta: fire in dry eucalypt forest: fuel structure, fuel dynamics and fire behaviour. CSIRO Publishing (2008)
- Hammill, K., Tasker, E.: vegetation, fire and climate change in the greater blue mountains world heritage area. techreport, <https://www.environment.nsw.gov.au/-/media/OEH/Corporate-Site/Documents/Parks-reserves-and-protected-areas/Types-of-protected-areas/vegetation-fire-climate-change-greater-blue-mountains-world-heritage-area-100941.pdf>
- Hines, F., Hines, F., Tolhurst, K.G., Wilson, A.A., McCarthy, G.J.: Overall fuel hazard assessment guide, 4th edition. No. Report no. 82, Victorian Government, Department of Sustainability and Environment Melbourne (2010)
- Hodgson ME, Bresnahan P (2004) Accuracy of airborne lidar-derived elevation. *Photogrammetric Engineering Remote Sensing of Environment* 70(3):331–339
- Homajnejad, N., Zlatanova, S., Pfeifer, N.: A voxel-based method for the three-dimensional modelling of heathland from lidar point clouds: First results **V-3-2022**, 697–704 (2022). 10.5194/isprs-annals-v-3-2022-697-2022
- Homajnejad, N., Zlatanova, S., Sepasgozar, S.M.E., Pfeifer, N.: Influence of voxel size and voxel connectivity on the 3D modelling of australian heathland parameters **X-4/W2-2022**, 113–119 (2022). 10.5194/isprs-annals-x-4-w2-2022-113-2022
- Homajnejad, N., Zlatanova, S., Pfeifer, N., Sepasgozar, S.M.E.: A methodology for an automated three-dimensional heathland assessment workflow in support of bushfire behaviour modelling. In: The 3rd Built Environment Research Forum. MDPI (mar 2022). 10.3390/enviro-sciproc2021012024
- Kelly, A.L., Franks, A.J., Eyre, T.J.: Assessing the assessors: Quantifying observer variation in vegetation and habitat assessment **12**(2), 144–148. 10.1111/j.1442-8903.2011.00597.x
- Lassiter, H.A., Whitley, T., Wilkinson, B., Abd-Elrahman, A.: Scan pattern characterization of velodyne VLP-16 lidar sensor for UAS laser scanning **20**(24), 7351. 10.3390/s20247351
- LIVOX: Specs, <https://www.livoxtech.com/avia/specs>
- Meteorology, B.: New south wales in november 2020: very warm and dry, <http://www.bom.gov.au>
- New South Wales, S., Department of Planning, I., Environment: Nsw state vegetation type map, <https://datasets.seed.nsw.gov.au/dataset/nsw-state-vegetation-type-map>
- Otepka J, Ghuffar S, Waldhauser C, Hochreiter R, Pfeifer N (2013) Georeferenced point clouds: A survey of features and point cloud management. *ISPRS International Journal of Geo-Information* 2(4):1038–1065
- New South Wales (Department of Planning, S., Environment): Heathlands | nsw environment, energy and science

- Sikkink, P.G., Keane, R.E.: A comparison of five sampling techniques to estimate surface fuel loading in montane forests **17**(3), 363. 10.1071/wf07003
- Spits C, Wallace L, Reinke K (2017) Investigating surface and near-surface bushfire fuel attributes: A comparison between visual assessments and image-based point clouds. *Journal of Sensors* 17(4):910
- Williams, R.J., Gill, A.M., Bradstock, R.A.: *Flammable Australia Fire Regimes, Biodiversity and Ecosystems in a Changing World*. CSIRO Publishing
- Xu, H., Wang, C.C., Shen, X., Zlatanova, S.: Evaluating the performance of high level-of-detail tree models in microclimate simulation **X-4/W3-2022**, 277–284 (2022). 10.5194/isprs-annals-x-4-w3-2022-277-2022
- Xu H, Wang CC, Shen X, Zlatanova S (2021) 3d tree reconstruction in support of urban microclimate simulation: A comprehensive literature review. *Buildings* 11(9):417

Comparison of Cloud-to-Cloud Distance Calculation Methods - Is the Most Complex Always the Most Suitable?



Vitali Diaz , Peter van Oosterom , Martijn Meijers , Edward Verbree , Nauman Ahmed, and Thijs van Lankveld

Abstract Cloud-to-cloud (C2C) distance calculations are frequently performed as an initial stage in change detection and spatiotemporal analysis with point clouds. There are various methods for calculating C2C distance, also called inter-point distance, which refers to the distance between two corresponding point clouds captured at different epochs. These methods can be classified from simple to complex, with more steps and calculations required for the latter. Generally, it is assumed that a more complex method will result in a more precise calculation of inter-point distance, but this assumption is rarely evaluated. This paper compares eight commonly used methods for calculating the inter-point distance. The results indicate that the accuracy of distance calculations depends on the chosen method and a characteristic related to the point density, the intra-point distance, which refers to the distance between points within the same point cloud. The results are helpful for applications that analyze spatiotemporal point clouds for change detection. The findings will be helpful in future applications, including analyzing spatiotemporal point clouds for change detection.

Keywords Cloud-to-cloud distance calculation · Change detection · Spatiotemporal analysis

This article was selected based on the results of a double-blind review of an extended abstract

V. Diaz (✉) · P. van Oosterom · M. Meijers · E. Verbree
GIS Technology, Faculty of Architecture and the Built Environment, Delft University of
Technology, Delft 2628 BL, The Netherlands
e-mail: v.diazmercado@tudelft.nl; vitalidime@gmail.com

P. van Oosterom
e-mail: P.J.M.vanOosterom@tudelft.nl

N. Ahmed · T. van Lankveld
Netherlands eScience Center, Amsterdam 1098 XH, The Netherlands

1 Introduction

Accurate calculation of the inter-point distance, which refers to the distance between two pairs of point clouds each taken at distinct epochs, is important for different applications involving point clouds, such as change detection (Stilla and Xu 2023) and spatiotemporal analysis (Anders et al. 2021). The selection of the method for calculating inter-point distance depends on several factors, such as the nature and size of the clouds, presence of noise or outliers, distance definition (e.g., Euclidean, geodesic, or Hausdorff distance), method's computational efficiency, method's time cost and memory usage, and availability of the method's implementation.

There are currently various tools with diverse implemented methods for calculating inter-point distance (e.g., CloudCompare (GPL software. 2023)). These implementations often allow the user to specify the distance type, which is usually Euclidean, and how the data is navigated for the subsample of points (including the querying of a specific number of neighbor points). In practice, this enables the selection of any implemented method. In general, it is assumed that a more complex method will result in a more accurate inter-point distance calculation, but this assumption is not always verified.

In this research, we compare eight methods for calculating inter-point distance following a controlled displacement test to analyze the accuracy of distance computations. The association of the inter-point distance with the point density is also investigated through the analysis of the intra-point distance, i.e., the distance between individual points within the same point cloud.

2 Methods and Data

Two concepts related to point distance are discussed in this paper, which are defined for a better understanding of the methodology. On the one hand, intra-distance refers to the distance between points within the same point cloud. This calculation is carried out by determining the median of the Euclidean distance from a specific point to its k nearest neighbors, providing a local measure related to the point density. This process is applied individually to each single point within the same cloud, offering valuable insight into the internal relationship of the data points. On the other hand, inter-distance, often called C2C (Cloud-to-Cloud) distance, measures the spatial separation between two corresponding point clouds taken at different epochs (time steps). These clouds are commonly identified as the reference cloud and the compared cloud. Inter-distance calculation is performed individually for each point of the compared cloud, enabling an assessment of the relationship between these two clouds. Various methods exist for calculating inter-distance, each with distinct assumptions about distance calculation. Inter-distance calculation yields one value for each point in the compared cloud, providing a comprehensive understanding of the spatial dissimilarities/similarities between the two point clouds.

Controlled displacement test

In this methodology, the process unfolds as follows. (1) Firstly, a specific point cloud is designated as the “reference cloud.” (2) Subsequently, the intra-distance is calculated for every point within the reference cloud, as well as the average intra-distance. (3) To explore a range of scenarios, artificial displacements are proposed based on the average intra-distance (as detailed below). (4) These proposed displacements are systematically applied to all points within the reference cloud, creating a “compared cloud” for each displacement scenario. (5) Moving forward, the calculation of inter-point distance between the compared and the reference cloud takes place. Eight different methods were tested (as outlined below). (6) Finally, each method is evaluated to determine its accuracy in capturing the applied artificial displacement.

The following methods were applied for the inter-distance calculation: (1) The nearest neighbor, (2) Least squares plane, (3) Linear interpolation, (4) Quadratic (height function), (5) 2.5D triangulation, (6) Natural Neighbor Interpolation (NNI), (7) Inverse Distance Weight (IDW), and (8) Multiscale Model to Model Cloud Comparison (M3C2). Most of the methods are available in CloudCompare, but we implemented them in Matlab for batch processing and further investigation.

The batch processing of inter-point distance involves the following steps, which are applied to each point in the compared cloud. (1) For every point in the compared cloud, the k nearest neighbors are retrieved from the reference cloud, which are identified as “selected points”. (2) Subsequently, the inter-distance method is applied to these selected points, resulting in a distance vector (dx , dy , dz). It is important to note that for Natural Neighbor Interpolation and Inverse Distance Weight, only the dz component can be calculated.

The deviation between the applied and calculated displacements was calculated to assess the ability to capture the correct artificial displacement for each method. Deviation was determined as the absolute difference between the applied and calculated displacements divided by the applied displacement, and it is expressed as a percentage. Six intervals were considered for assessing the deviation: 0 to 10, 10 to 20, 20 to 30, 30 to 40, 40 to 50, and greater than 50%. We focused on the dz component to evaluate the inter-distance methods because it is the value that all methods can return.

The artificial displacements were carefully chosen to analyze three specific scenarios:

- (a) When the displacements are smaller than the average intra-point distance.
- (b) When the displacements are equal to the average intra-point distance.
- (c) When the displacements are greater than the average intra-point distance.

For both the calculation of intra-distance and inter-distance, the eight nearest neighbors were used, i.e., $k = 8$. This number of neighbors allows the application of all the methods. For example, in the Quadratic (height function) case, at least six points are required. Using a higher number of points is not advisable due to the detrimental impact on points selection times. The methodology was applied in three

databases: (1) bunny, (2) lake, and (3) coast scan. The first two are available with the LAStools (Isenbur 2023), and the last from Vos et al. (2022).

3 Results and Discussion

In the case of the object (bunny) (Fig. 1), the methods that best capture the applied vertical displacement (dz) are nearest neighbor, NNI, and IDW. However, when for horizontal displacements, the NNI method performs better.

Interestingly, the results exhibit sensitivity to the direction of displacement. When displacement is applied vertically (in the z -axis), the calculated distances align closely with the applied displacement, particularly for points near the top of the object. In contrast, for horizontal displacements, the points located on the sides show better accuracy in the calculated displacement.

In the case of the lake database (terrain with hills and trees) (Fig. 2), most methods effectively capture the vertical displacement on the terrain. Notably, the nearest neighbor, NNI, and IDW methods stand out for accurately representing displacement on trees. However, regarding horizontal displacement, most methods fail to provide accurate results, with only the nearest neighbor and NNI methods presented notable performance. Specifically, in horizontal offsets, the results indicate that the points close to the sides of objects (trees) capture the displacement more accurately. For terrains, only the nearest neighbor method appears to yield improved results.

On the beach, most methods exhibit similar performance in capturing vertical displacement (dz). This is primarily because, for most points in the compared cloud, the neighbors from the reference cloud effectively represent the same section of the beach. However, when a horizontal offset is applied, most methods fall short in accuracy, and only the nearest neighbor method performs relatively better.

4 Conclusion and Future Work

This paper has presented the results of comparing eight commonly used inter-point distance calculation methods in point cloud analysis. The findings show that the accuracy of the inter-point distance calculation depends on the intra-point distance. Specifically, it has been observed that better accuracy is achieved when the inter-point distance is smaller than the intra-point distance.

Furthermore, the results of this study challenge the assumption that a method with more steps always yields superior performance. The comparative analysis has revealed that a more complex method is not necessarily the most suitable for inter-point distance calculation. These insights are invaluable for future applications involving the analysis of spatiotemporal point clouds for change detection. Researchers and practitioners can leverage these findings to make informed decisions when selecting suitable methods tailored to their analysis requirements.

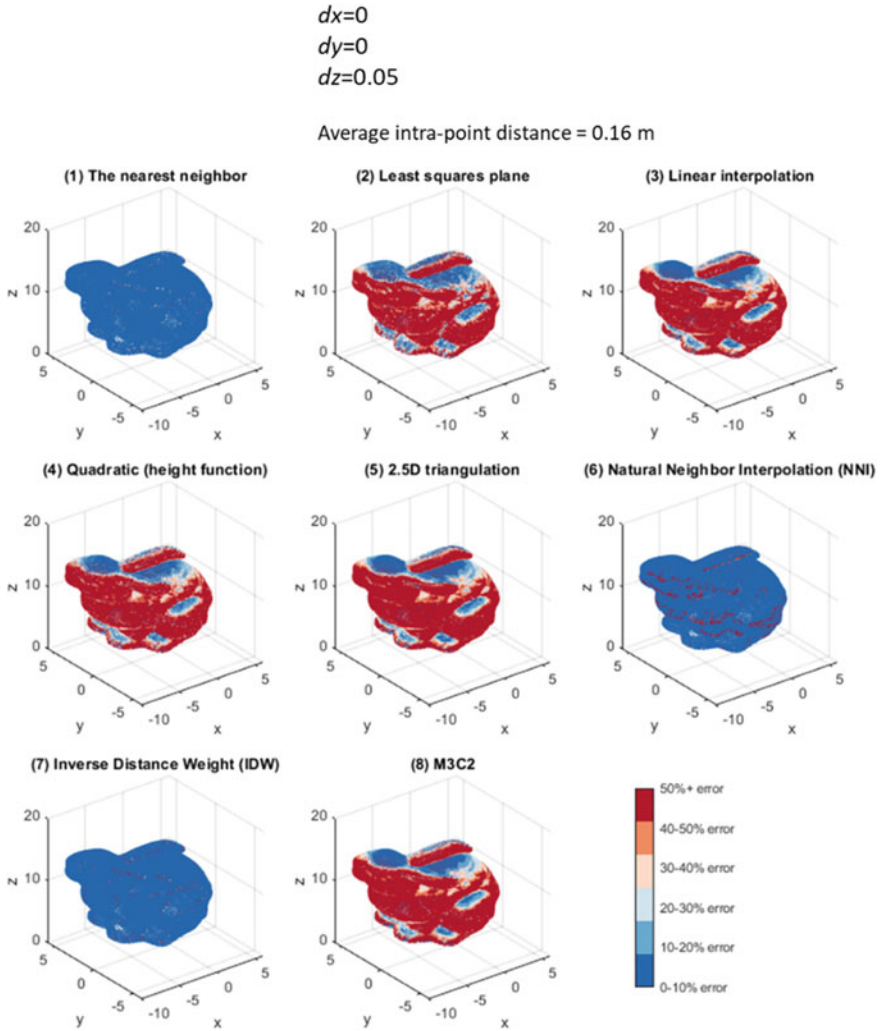


Fig. 1 Results for the bunny database

In our future work, we aim to explore the performance of these methods with larger and more complex point clouds while investigating the impact of various parameters, including point density, sampling techniques, and noise levels. Also, we will expand the scope of our study to include other transformations (e.g., rotation), beyond the artificial displacement (dx , dy , dz).

Finally, we also plan to analyze the AHN (*Actueel Hoogtebestand Nederland*, in Dutch) database, which covers lidar data for the entire Netherlands. We will select the most suitable method and implement it efficiently within a Database Management System (DBMS) using Space Filling Curve (SFC) techniques, particularly for the

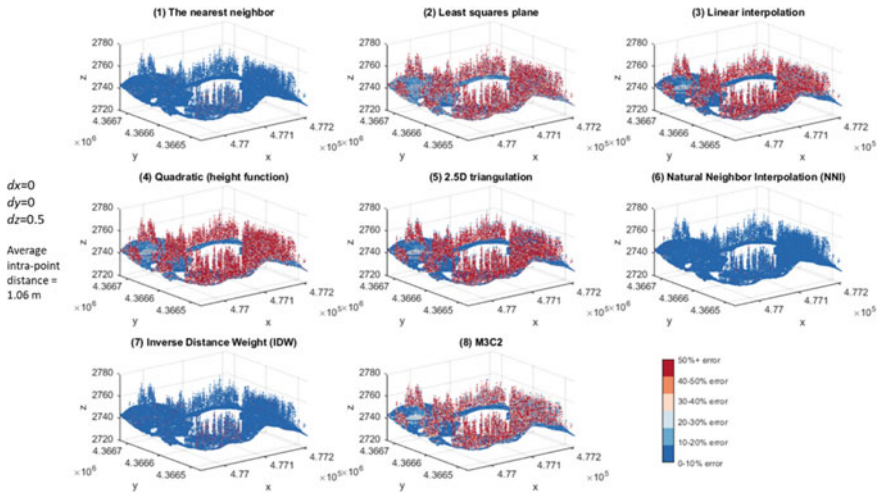


Fig. 2 Results for the lake database

comprehensive change detection analysis of the AHN-2, -3, and -4 datasets. Outputs project can be consulted at nd-pc.org and www.gdmc.nl/publications.

References

- Anders K, Winiwarter L, Mara H, Lindenberg R, Vos SE, Höfle B (2021) Fully automatic spatiotemporal segmentation of 3D LiDAR time series for the extraction of natural surface changes. *ISPRS J Photogramm Remote Sens* 173:297–308. <https://doi.org/10.1016/j.isprsjprs.2021.01.015>
- GPL software (2023) CloudCompare (version 2.13.alpha). Retrieved from <http://www.cloudcompare.org/>. Accessed on 22 May 2023
- Isenbur M (2023) LAStools—efficient LiDAR processing software. Retrieved from <http://rapidlasso.com/LAStools>. Accessed on 22 May 2023
- Stilla U, Xu Y (2023) Change detection of urban objects using 3D point clouds: a review. *ISPRS J Photogramm Remote Sens* 197:228–255. <https://doi.org/10.1016/j.isprsjprs.2023.01.010>
- Vos S, Anders K, Kuschnerus M, Lindenberg R, Höfle B, Aarninkhof S, de Vries S (2022) A high-resolution 4D terrestrial laser scan dataset of the Kijkduin beach-dune system, The Netherlands. *Sci Data* 9(191):1–11. <https://doi.org/10.1038/s41597-022-01291-9>

Deriving 3D Models from Point Clouds

Reconstructing Façade Details Using MLS Point Clouds and Bag-of-Words Approach



Thomas Froech, Olaf Wysocki, Ludwig Hoegner, and Uwe Stilla

Abstract In the reconstruction of façade elements, the identification of specific object types remains challenging and is often circumvented by rectangularity assumptions or the use of bounding boxes. We propose a new approach for the reconstruction of 3D façade details. We combine mobile laser scanning (MLS) point clouds and a pre-defined 3D model library using a Bag of words (BoW) concept, which we augment by incorporating semi-global features. We conduct experiments on the models superimposed with random noise and on the TUM-FAÇADE dataset (Wysocki et al., 2023). Our method demonstrates promising results, improving the conventional BoW approach. It holds the potential to be utilized for more realistic facade reconstruction without rectangularity assumptions, which can be used in applications such as testing automated driving functions or estimating façade solar potential.

Keywords Point clouds · Façade reconstruction · Bag-of-Words Approach · mobile laser scanning

This article was selected based on the results of a double-blind review of an extended abstract.

T. Froech (✉) · O. Wysocki · L. Hoegner · U. Stilla
Photogrammetry and Remote Sensing, TUM School of Engineering and Design,
Technical University of Munich (TUM), Munich, Germany
e-mail: thomas.froech@tum.de

O. Wysocki
e-mail: olaf.wysocki@tum.de

L. Hoegner
e-mail: ludwig.hoegner@tum.de; ludwig.hoegner@hm.edu

U. Stilla
e-mail: stilla@tum.de

L. Hoegner
Department of Geoinformatics, University of Applied Science (HM), Munich, Germany

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
T. H. Kolbe et al. (eds.), *Recent Advances in 3D Geoinformation Science*, Lecture Notes
in Geoinformation and Cartography, https://doi.org/10.1007/978-3-031-43699-4_21

1 Introduction

Semantic 3D building models up to level of detail (LoD)2 level are widely used and available today (Biljecki et al., 2015). LoD3 models characterized by a higher level of detail of their façade representation are scarce¹. These are necessary for various applications, such as testing automated driving functions or estimating façade solar potential (Wysocki et al., 2023). The primary challenges in developing 3D façade reconstruction methods lie in the availability of street-level measurements and complexity of detailed façade element reconstruction; which is often circumvented by reconstructing the elements as bounding boxes (Hensel et al., 2019). Recently, however, there has been an increase in the availability of high-accuracy, street-level MLS point cloud data (Wysocki et al., 2023). In addition, databases that contain high-quality hand-modeled 3D façade details already exist (Trimble Inc., 2021). These possess the potential to bridge the data-gap and move beyond the assumption of rectangularity and the use of bounding boxes.

In this paper, we propose an approach for the reconstruction of façade details leveraging the accuracy of MLS point clouds and the ubiquity of high quality 3D façade elements' libraries. Specifically, we employ an enhanced bag-of-words (BoW) approach (Csurka et al., 2004) to match measured façade elements with those from the library, without the rectangular assumptions.

2 Related Work

2.1 LoD3 Building Model Reconstruction

The reconstruction of LoD3 building models has attracted attention over an extended period of time (Huang et al., 2020; Tuttas and Stilla, 2013; Pantoja-Rosero et al., 2022; Ripperda, 2010; Wysocki et al., 2023). Recent advancements have shown that façade elements reconstruction can be robustly performed, and yet identifying a specific object type remains challenging, for example, distinguishing between rectangular and oval window types (Wysocki et al., 2023). An example is the study of Hoegner and Gleixner, which aims at the extraction of a 3D model of façades and windows from a MLS point cloud (Hoegner and Gleixner, 2022). Their approach is based on a voxel octree structure and visibility analysis. While they report a detection rate of 86%, they simplify windows and façades by representing them solely as rectangular shapes.

Other studies, such as that of Stilla and Tuttas, are devoted to the use of airborne laser scanning (ALS) point clouds for the reconstruction of 3D building models (Tuttas and Stilla, 2013). They introduce an approach for the generation of façade planes with windows and the enrichment of a semantic city model with windows from a multi-aspect oblique view ALS point cloud.

¹ <https://github.com/OloOcki/awesome-citygml>.

Following a different, image and deep learning-based approach for 3D model reconstruction, Fan et al. propose VGI3D, an interactive platform for low-cost 3D building modeling from volunteered geographic information (VGI) data using convolutional neural network (CNN)s in 2021 (Fan et al., 2021). Their easy to use, lightweight, and quick application takes a small number images and a user's sketch of the façade boundary as an input (Fan et al., 2021). For the automatic detection of façade elements, the object detection CNN YOLO v3 (Redmon and Farhadi, 2018) is utilized (Fan et al., 2021).

2.2 *Bag of Words Approach*

We use an adapted variant of the BoW concept in our study. The original BoW approach is introduced by Salton and McGill in 1986 (Salton and McGill, 1986). Csurka et al. apply the BoW concept to images, where the detection and description of keypoints is one of the foundations of their approach: Bag-of-Visual-Words (BoVW) (Csurka et al., 2004). This concept is also applied to point cloud data. Based on the Work of Xu et al. on object classification of aerial images (Xu et al., 2010), Kang and Yang make use of a bag-of-visual-words (BoVW) approach to construct a supervoxel representation of raw point cloud data (Kang and Yang, 2018). The BoVW concept is further developed and adapted. Zhu et al. introduce the local-global feature BoVW (LGFBoVW). They combine local and global features on histogram level. Such a combination increases the robustness compared to the traditional BoVW-approach (Zhu et al., 2016).

3 Methodology

3.1 *Overview*

Figure 1 provides an overview of our proposed method. The training starts with 3D model preparation and sampling, where we create binary images from the sampled point clouds. From these images, we extract and describe features, which are then clustered to obtain a visual dictionary. By quantizing with the Euclidean distance, we assign the closest codeword to every feature vector. Next, we count each codewords occurrences, providing the representations of the models as bags of codewords. During the inference, we represent the target point clouds by the codebook. To compare both representations, we employ histogram distances, where the model with the closest histogram distance to a target point cloud is selected as the best match.

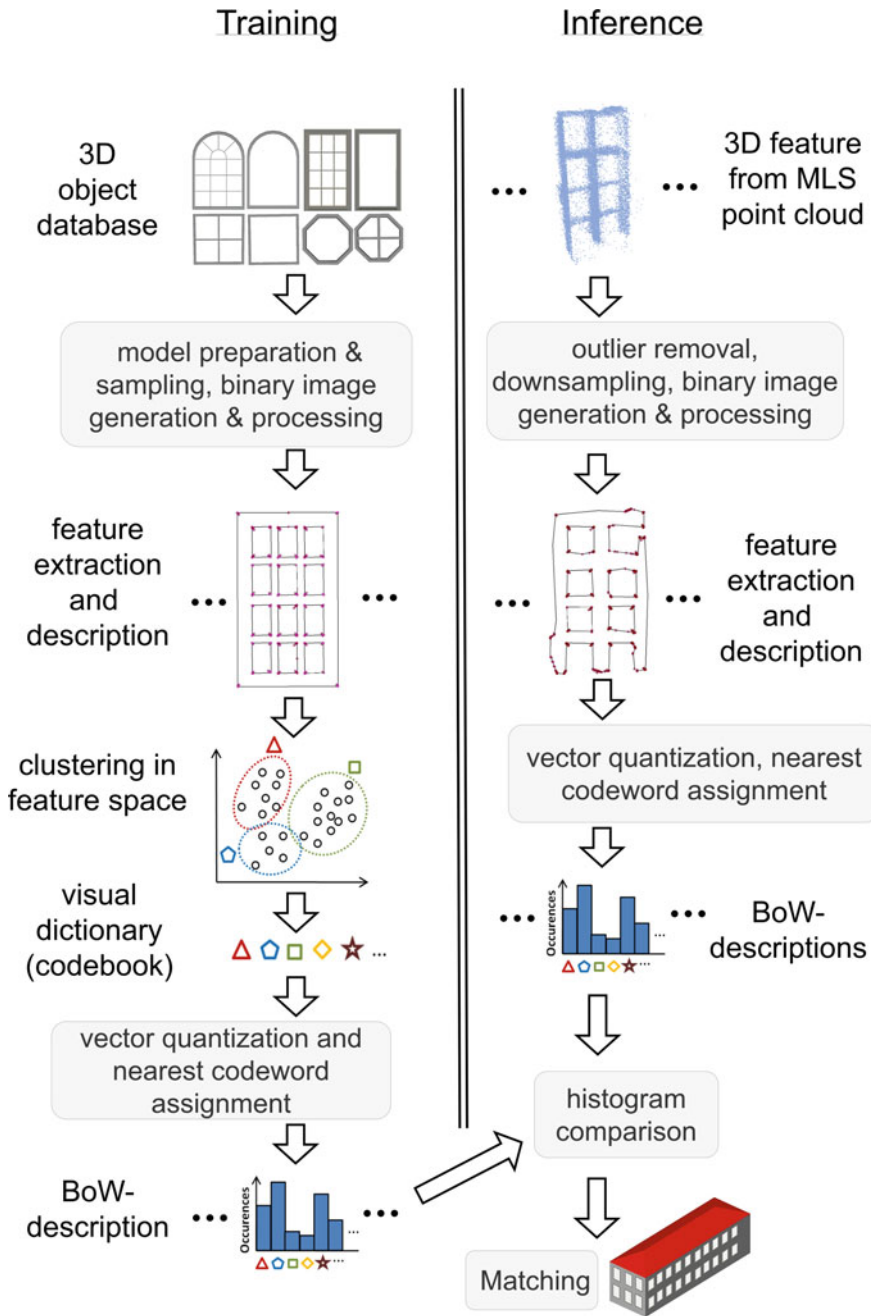
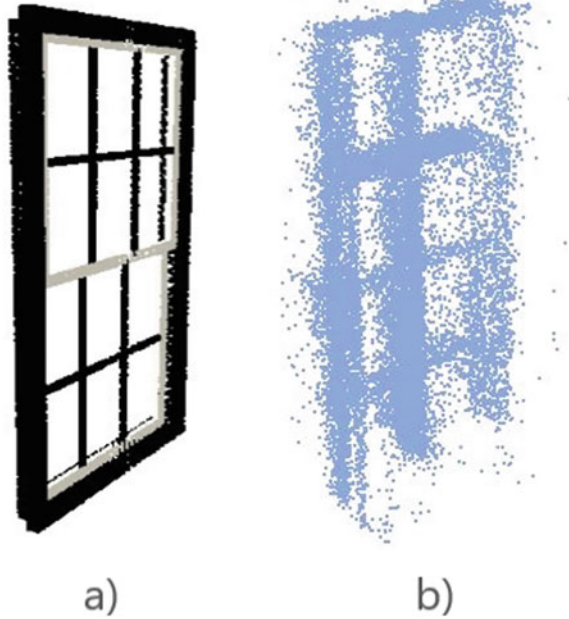


Fig. 1 Overview of the approach, left side: training process, right side inference (illustration based on Memon et al. (2019))

Fig. 2 a Point cloud sampled from CAD b point cloud extracted from the TUM-FAÇADE dataset



3.2 Sampling of the CAD Models

A key element of the method we propose is the correspondence between features extracted from the MLS point clouds and the CAD models. In order to improve the comparability, we sample the CAD models with a sampling distance d to obtain a point cloud. Figure 2a gives an example of such a point cloud. To enable comparison to point clouds from MLS data, Fig. 2b shows an exemplary window that has been extracted from the TUM-FAÇADE dataset.

The diversity of the CAD models does not allow for the same sampling distance to be used for all models. The consequence of this could be sparse point clouds or point clouds with too many points. Therefore, we choose d individually for the CAD models to obtain a suitable point cloud. An alternative approach could be to normalize the CAD models before the sampling. Figure 2a gives an example of a point cloud sampled from a CAD model.

3.3 Binary Image Generation and Processing

We normalize the point clouds after outlier removal and downsampling. To account for the decreasing point density with increasing height, we use an outlier removal that depends on the average height of the windows from the MLS point cloud.

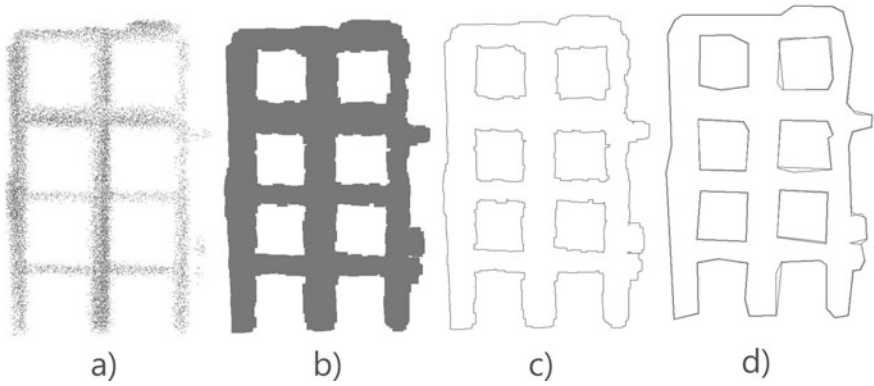


Fig. 3 Exemplary image processing on a projected point cloud: **a** projected image of a point cloud, **b** dilated image, **c** edge detection (Laplace), **d** line simplification (Douglas-Peucker)

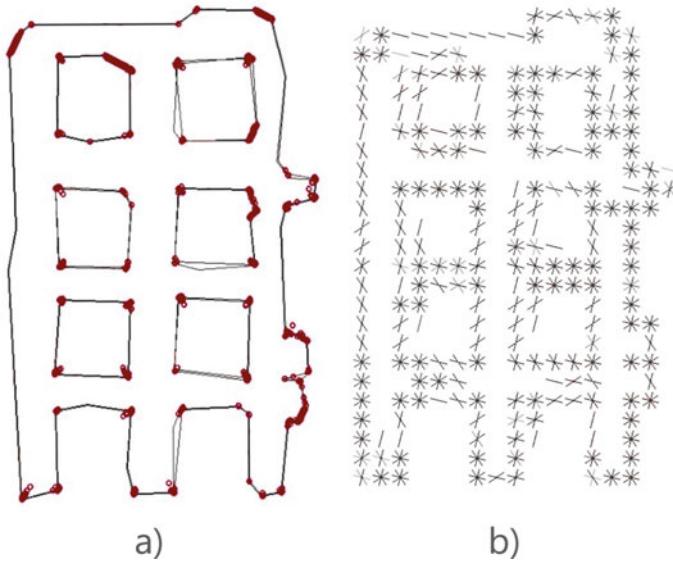


Fig. 4 Examples for feature extraction: **a** ORB-keypoints **b** HOG-image

Next, the point clouds are ortho-projected to a binary image ensuring its frontal view. As illustrated in Fig. 3, to enhance the extraction of meaningful keypoints, we apply the standard image processing techniques. Figure 4a shows that the majority of the identified keypoints are located at semantically meaningful positions.

3.4 *Feature Extraction*

We use the Oriented FAST and Rotated BRIEF (ORB) (Rublee et al., 2011) descriptor as a keypoint point detector. This binary descriptor is based on the Binary Robust Independent Features (BRIEF) descriptor (Rublee et al., 2011; Calonder et al., 2012). We choose this descriptor due to its resistance to noise and higher computational speed compared to other descriptors such as Scale Invariant Feature Transform (SIFT) (Rublee et al., 2011; Lowe, 1999).

In our study, we use dense feature sampling as an alternative approach to the interest point detection. This method is opposed to the concept of identifying and describing key points. In dense feature sampling, descriptors are sampled at points on a dense grid over the image, hence the name. We sample the ORB descriptor for each of the points in the dense grid. This approach allows the extraction of a large amount of information at the cost of higher computational intensity (Nowak et al., 2008).

We use the Histogram of Oriented Gradients (HOG) feature descriptor (Dalal and Triggs, 2005) to incorporate semi-global information into the BoW approach. The fundamental concept of this descriptor is the investigation of the gradients and their orientation within the image on a dense grid (Dalal and Triggs, 2005). Normalized histograms of these gradients are established for each of these grid cells (Dalal and Triggs, 2005). The resulting one-dimensional vector characterizes the structure of the objects in the image (Dalal and Triggs, 2005; Mallick, 2016).

3.5 *Incorporation of Semi-Global Features*

Generally, shapes possess a limited number of features (Bronstein et al., 2011). This poses difficulties in extracting large numbers of distinct features. Semi-global information can be used to mitigate the effects of this issue. However, semi-global features, as well as global features, cannot be directly integrated into the standard BoW method. Their (semi-) global uniqueness prevents the establishment of a frequency of occurrence. We propose a method to incorporate HOG descriptors as semi-global information into the BoW-approach to overcome this issue. Figure 4b gives an example of the information obtained with HOG. A 2D diagram that displays the distribution of the gradients in the respective cell is shown for every cell in the image, in which a gradient is present. Figure 5 gives an overview of our concept. We establish a structure similar to a histogram by considering each HOG descriptor variate as a separate histogram bin, with the value of the bin equaling the value of the corresponding variate. This structure comprises as many bins as HOG variates. We concatenate the occurrence histogram that we obtain from the BoW approach with the histogram-similar structure constructed from the HOG-variates to a combined histogram.

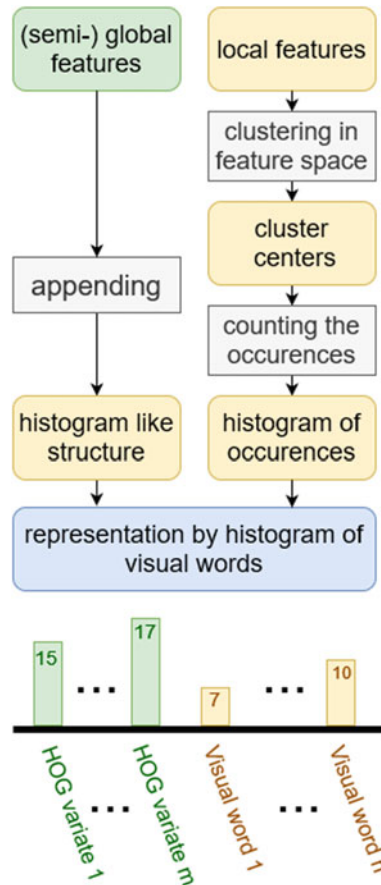


Fig. 5 Incorporation of semi-global features

3.6 Clustering

For simplicity, we make use of the standard K-Means clustering algorithm in our BoW approach. We treat every descriptor variate as an axis in this clustering. Our clustering problem thus has the same dimensionality as the extracted feature descriptor. With K-Means clustering, the number of clusters n has to be determined in advance. The setting of this hyper-parameter is critical for the performance of the BoW method since the meaningful assignment of data points to cluster centers depends on it (Kang and Yang, 2018). There appear only a few empty or overcrowded cluster centers for this n . We acknowledge that there might be better configurations for n , and that there are more sophisticated clustering algorithms available that might lead to improved results.

3.7 Histogram Comparison

We use histogram distances to assess the similarity of the histograms within our BoW approach. There is a large number of different histogram distances available (Cha, 2008). The specific choice of histogram distance employed holds limited significance regarding the general approach that we introduce, thereby allowing for arbitrary selection in our methodology. We make use of the histogram distances that are introduced in the following.

The Minkowski distance is given as (Cha, 2008):

$$D = \left(\sum_{i=1}^n |q_i - p_i|^p \right)^{\frac{1}{p}} \quad (1)$$

The Minkowski distance is characterized by the parameter p . It can be interpreted as a more general form of the Manhattan distance ($p = 1$), the euclidean distance ($p = 2$) and the Chebyshev distance ($p = \infty$) (Cha, 2008). It can be used to evaluate the similarity of two histograms by pairwise comparison of the individual bins and subsequent accumulation of the obtained distances.

The Jensen-Shannon divergence is given as (Bamler and Shi, 2022):

$$\text{JSD}(P, Q) = \frac{1}{2} (\text{KL}(P \| M) + \text{KL}(Q \| M)) \quad (2)$$

where $\text{KL}(P \| Q)$ is the Kullback-Leibler-Divergence between the probability distributions P and Q . The Kullback-Leibler-Divergence given as (Bamler and Shi, 2022):

$$\text{KL}(P \| Q) = \sum_i P(i) \ln \left(\frac{P(i)}{Q(i)} \right) \quad (3)$$

The Jensen-Shannon divergence, like the Kullback-Leibler divergence, is based on the concept of entropy (Cha, 2008) and is used to quantify the similarity of two probability distributions. The Jensen-Shannon divergence can be interpreted as a symmetric version of the Kullback-Leibler divergence (Cha, 2008).

The Pearson Chi-Square-Distance between the probability distributions P and Q is given as (Cha, 2008):

$$D\chi^2(P, Q) = \sum_i \frac{(P(i) - Q(i))^2}{P(i)} \quad (4)$$

4 Experiments

We inferred our method on the models, superimposed with random noise and on a building from the labeled MLS point cloud of the TUM-FAÇADE data set (Wysocki et al., 2023). To ensure the comparability of our experiments, we consistently used the same number of clusters n in all of them.

We found a suitable n in a heuristic way. We performed experimental clustering and feature descriptor quantization for different values of n . We then evaluated the frequency of occurrence of the respective cluster centers. We find that in most of our cases 25 clusters give satisfactory results.

We acquired four CAD models from the SketchUp 3D Warehouse library (Trimble Inc., 2021) (Fig. 6). We chose the models so that each window that is present in the TUM-FAÇADE dataset at least matches one of the models. We additionally chose one window that is of octagon like shape as a model without a matching window in the dataset. We manually edited some of the CAD models to add or remove window bars.

When sampling the CAD-models to a point cloud, transparency properties have to be taken into account. As shown in Fig. 7, we removed glass from all models to exploit the presence or absence of window bars. In Fig. 7b detailed window bars

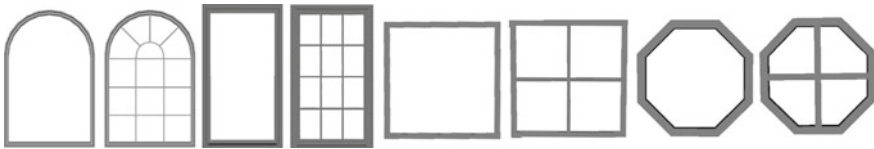
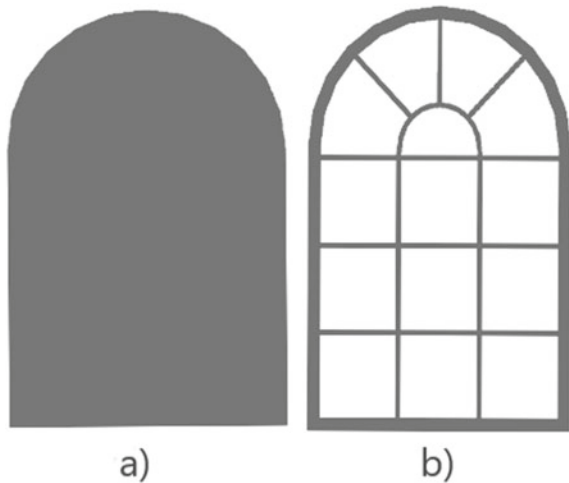


Fig. 6 3D window CAD models

Fig. 7 Point clouds, sampled from CAD model: **a** without removal of window glass **b** with removal of window glass



become apparent due to glass components not being sampled. In contrast, Fig. 7a illustrates that when the glass components are sampled using the same method as the rest of the window, it results in the window appearing as an opaque object. With MLS scans, the laser beams usually penetrate the glass and thus penetrate into the interior of the building. This justifies the removal of glass from the CAD models.

We evaluate our results by determining the overall accuracy, the user's accuracy the producer's accuracy, and Cohen's kappa coefficient.

The overall accuracy (OA) is calculated by dividing the number of correctly classified pixels by the total number of pixels (Congalton and Green, 2019):

$$OA = \frac{\text{number of correctly classified samples}}{\text{total sample number}} \quad (5)$$

The producer's accuracy (PA) is used to determine the percentage of the ground truth samples that is classified correctly (Congalton and Green, 2019):

$$PA = \frac{\text{number of correctly classified samples of a class X}}{\text{total number of ground truth samples in class X}} \quad (6)$$

The user's accuracy (UA) is used to calculate the percentage of the classification result of a class that is classified correctly (Congalton and Green, 2019):

$$UA = \frac{\text{number of correctly classified samples of a class X}}{\text{total number of samples classified as class X}} \quad (7)$$

Cohen's kappa coefficient represents an alternative to using OA. It has a value range from 0 to 1, where 0 represents complete randomness while 1 would represent a perfect classifier (Congalton and Green, 2019). It can be interpreted as a measure for the concordance between the predicted class assignments and the class assignment of the ground truth data (Grandini et al., 2020). It can be calculated from the confusion matrix according to the following formula:

$$\kappa = \frac{OA - RM}{1 - RM} \quad (8)$$

with the random match (RM):

$$RM = \frac{\sum(\text{product of row and column sums})}{(\text{total sum})^2} \quad (9)$$

The implementation is available in the repository (Fröch, 2023).

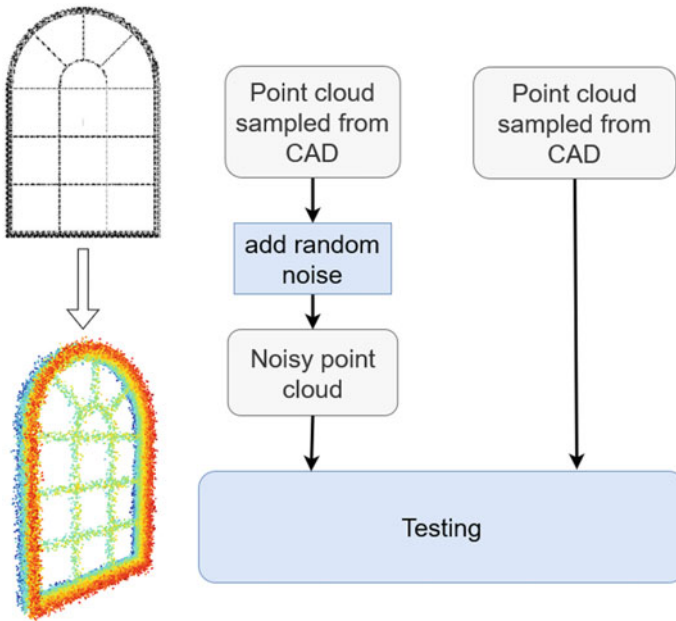


Fig. 8 Schema: addition of random noise to the point clouds sampled from CAD

4.1 Experiments on Models Superimposed with Random Noise

We added random noise to the point clouds that we sample from the pre-processed CAD models, as schematically described in Fig. 8. We used the Chi-Square histogram distance for these experiments. Results from these experiments are summarized in Figs. 9, 10 and Table 1.

We find that incorporating HOG descriptors improves the matching quality. Table 1 shows that the OA is improved from 0.47 to 0.69 this way. The kappa-coefficient is improved from 0.36 to 0.65. We observe sensitivity towards noise. By doubling the noise level, the kappa coefficient is diminished to 0.43, while the overall OA drops to 0.5.

In addition, we identify a dependency of the matching quality on the type of CAD model. Figures 9 and 10 show that the matching is more stable for the arched and octagon shaped windows than for the rectangular and quadratic windows.

We conducted further experiments with different combinations of features and histogram distances. More information on the experiments can be found here. Figure 11 summarizes the standard deviations and the variances of the user's and producer's accuracies for six of these experiments with different sets of hyper parameters. We find that the arched window with no bars and the two octagon-shaped windows are matched in the most stable manner compared to the other window types.

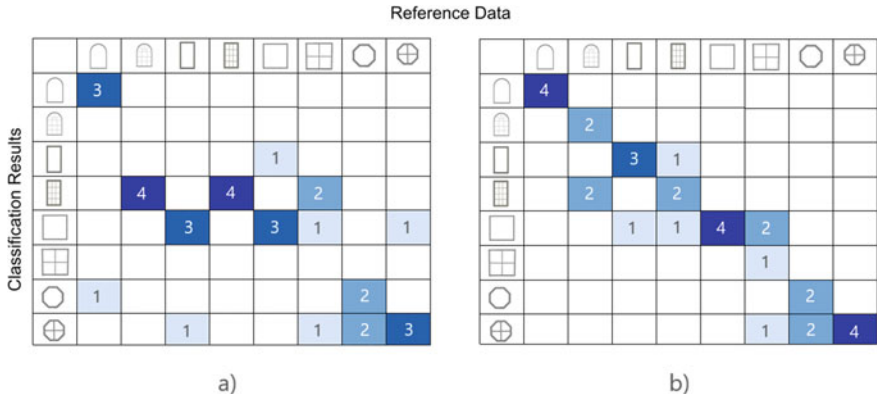


Fig. 9 Confusion matrices: a) ORB, b) ORB and HOG

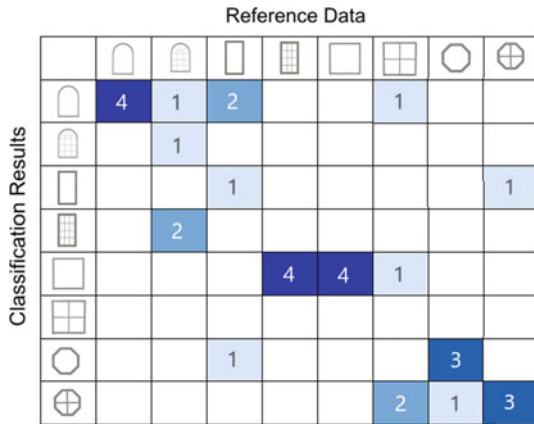


Fig. 10 Confusion matrix: ORB and HOG, noise level with factor 2

4.2 Experiments on the TUM-FAÇADE dataset

The tested façade only comprises rectangular and arched windows with window bars (Fig. 13). We observe an improvement of the matching quality with the incorporation of HOG-descriptors, as the OA increases to 0.57. Also, the dependence on the histogram distance is shown in Table 2 and Fig. 12.

We observe a correlation of the matching quality with the building height in Fig. 13. This can presumably be attributed to the point density of the MLS point cloud, which decreases with increasing altitude.

Table 1 Results of the experiments with the models superimposed with random noise

| Experiment | Overall accuracy | Kappa coefficient |
|---|------------------|-------------------|
| ORB, Chi-square dist. | 0.47 | 0.36 |
| ORB and HOG, Chi-square dist. | 0.69 | 0.65 |
| ORB and HOG, Chi-square dist., noise *2 | 0.50 | 0.43 |









| |  |  |  |  |  |  |  |  |
|----------------------------|---|---|---|---|---|---|---|---|
| Users Acc. - Variance | 0.07 | 0.160 | 0.21 | 0.07 | 0.08 | 0.18 | 0.02 | 0.02 |
| User's Acc. - Std.Dev. | 0.26 | 0.40 | 0.46 | 0.26 | 0.28 | 0.42 | 0.15 | 0.14 |
| Producer's Acc. - Variance | 0.010 | 0.12 | 0.14 | 0.19 | 0.10 | 0.09 | 0.06 | 0.02 |
| Producer's Acc. - Std.Dev. | 0.102 | 0.35 | 0.38 | 0.44 | 0.32 | 0.03 | 0.24 | 0.14 |

Fig. 11 Standard deviation and variance of user’s and producer’s accuracies from experiments with six different combinations of hyper parameters

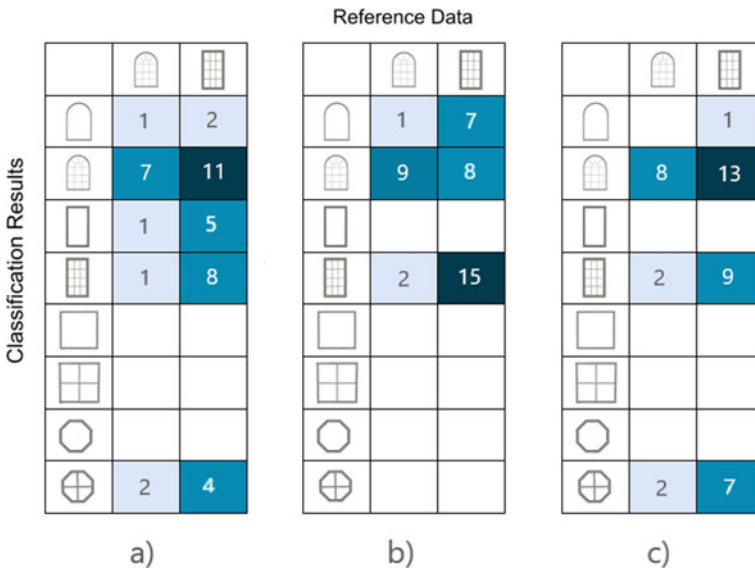


Fig. 12 **a** ORB, Jensen-Shannon divergence, **b** ORB and HOG, Jensen-Shannon divergence, **c** ORB and HOG, Minkowski-distance

Table 2 Results of the experiments on the TUM-FAÇADE dataset (Wysocki et al., 2023)

| Experiment | Overall accuracy |
|--|------------------|
| ORB, Jensen-Shannon divergence | 0.36 |
| ORB and HOG, Jensen-Shannon divergence | 0.57 |
| ORB and HOG, Minkowski-distance | 0.41 |



Fig. 13 Correctly (green) and falsely (red) matched windows
Reference Data

| | | |
|--|---|----|
| | | |
| | 2 | 6 |
| | 7 | 8 |
| | | 2 |
| | 3 | 12 |
| | | 2 |
| | | |
| | | |
| | | |

Fig. 14 Results of the experiments on the TUM-FAÇADE dataset with dense feature sampling

The experiment with ORB, HOG, and the Jensen-Shannon Divergence was also conducted using dense feature sampling instead of extracting key points. Figure 14 summarizes the results from this experiment. However, with an OA of 0.45, we observe a decrease in accuracy.

5 Discussion

5.1 3D Interest Point Detection

In our study, we employ 2D projections of point clouds to generate binary images and extract different types of features from them. We adopt this approach based on the intuition that the front view contains most of the information that characterizes the window.

The point cloud data as well as the CAD models are three-dimensional. Therefore, instead of a projection and subsequent extraction of 2D feature descriptors, it seems obvious to use 3D interest point detection directly. There is a number of such operators available. Examples are unsupervised stable interest point detection (USIP) for points clouds (Li and Lee, 2019) or Harris 3D for meshes (Sipiran and Bustos, 2011).

Assuming that the front view of a window contains the majority of information that characterizes a specific window type, the potential benefit derived from incorporating 3D information is expected to be marginal. The desired correspondence between the point cloud that is sampled from the CAD model and the MLS point cloud represents an obstacle when using such detectors. The often irregular 3D shape of the windows from the MLS point clouds does not necessarily correspond to the very regular 3D-shape of the windows that are sampled from the CAD models. Therefore, the extracted keypoints are most probably not going to correspond to the keypoints found in the model in most cases.

5.2 Influences of the Histogram Distance

We observe, that the use of the chi-square histogram distance yields the best results in the set of experiments on the models superimposed with random noise (see Sect. 4.1). However, due to the asymmetry of the particular histogram distance used, these results should be viewed with caution. The asymmetry could lead to unbalanced assessments of the similarity of histograms. Using the symmetrical form of the chi-square histogram distance (Cha, 2008) may yield more meaningful results in the context of this study. In the experiments with the TUM-FAÇADE dataset, the use of the Jensen-Shannon divergence leads to the best results. This histogram distance is, in contrast to the Chi-Square distance that we use, symmetric.

5.3 Dense Feature Sampling

We find that using dense feature samples does not improve the performance of our method compared to the other methods. We attribute this to the binary properties of the images that we generate from the projected point clouds. Figure 4a demonstrates

that almost all important interest points that characterize the object in the image are identified by the ORB keypoint detector. When using dense feature sampling, most points for which the descriptors are determined are located in empty regions of our binary images or in the vicinity of straight lines. Therefore the information gain of using dense feature sampling is not particularly large in context of our method.

6 Conclusion

In this paper we present a method for reconstructing façade details by using MLS point clouds and the BoW concept. We incorporate semi-global features to address the issue of insufficient distinct features in shapes. In our two sets of experiments, we demonstrate improved performance compared to the conventional BoW approach. Our method seems to be sensitive to noise and point cloud sparsity. Future work could focus at increasing the accuracy of the method as well as its computational efficiency. In the Future, our method could be applied in façade reconstruction pipelines that aim at a more realistic reconstruction without assumptions of rectangularity or the use of bounding boxes.

Acknowledgements This work was supported by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy within the framework of the IuK Bayern project *MoFa3D - Mobile Erfassung von Fassaden mittels 3D Punktwolken*, Grant No. IUK643/001. Moreover, the work was conducted within the framework of the Leonhard Obermeyer Center at the Technical University of Munich (TUM). We gratefully acknowledge the Geoinformatics team at the TUM for the valuable insights and for providing the CityGML datasets.

References

- Bamler R, Shi Y (2022) Estimation theory: 2—recap of basic propability theory -part 4. 13-5-2022, Chair of remote sensing technology technical university of Munich. Lecture of Estimation Theory
- Biljecki F, Stoter J, Ledoux H, Zlatanova S, Çöltekin A (2015) Applications of 3D city models: state of the art review. *ISPRS Int J Geo-Inf* 4(4):2842–2889
- Bronstein AM, Bronstein MM, Guibas LJ, Ovsjanikov M (2011) Shape google. *ACM Trans Graph* 30(1):1–20. <https://doi.org/10.1145/1899404.1899405>
- Calonder M, Lepetit V, Ozuysal M, Trzcinski T, Strecha C, Fua P (2012) BRIEF: computing a local binary descriptor very fast. *IEEE Trans Patt Anal Mach Intell* 34(7):1281–1298. <https://doi.org/10.1109/tpami.2011.222>
- Cha SH (2008) Taxonomy of nominal type histogram distance measures. In: Long C, Sohrab SH, Bognar G, Perlovsky L (eds) *MATH'08: proceedings of the American conference on applied mathematics*. World Scientific and Engineering Academy and Society, pp 325–330
- Congalton RG, Green K (2019) *Assessing the accuracy of remotely sensed data: principles and practices*. Taylor and Francis, 3 edn
- Csurka G, Dance CR, Fan L, Willamowski J, Bray C (2004) Visual categorization with bags of keypoints. In: *Proceedings of the ECCV international workshop on statistical learning in computer vision* (2004)

- Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). IEEE. <https://doi.org/10.1109/cvpr.2005.177>
- Fan H, Kong G, Zhang C (2021) An interactive platform for low-cost 3d building modeling from VGI data using convolutional neural network. *Big Earth Data* 5(1):49–65. <https://doi.org/10.1080/20964471.2021.1886391>
- Fröch T (2023) Project photogrammetry—reconstructing façade details using MLS point clouds and bag-of-words approach Tech rep, Technical University Munich. <https://github.com/ThomasFroech/ReconstructingFacadeDetailsBoW>
- Grandini M, Bagli E, Visani G (2020) Metrics for multi-class classification: an overview. <https://doi.org/10.48550/ARXIV.2008.05756>
- Hensel S, Goebbls S, Kada M (2019) Facade reconstruction for textured LOD2 CityGML models based on deep learning and mixed integer linear programming. *ISPRS Annals Photogram Remote Sens Spatial Inf Sci IV-2/W5:37–44*. <https://doi.org/10.5194/isprs-annals-iv-2-w5-37-2019>
- Hoegner L, Gleixner G (2022) Automatic extraction of facades and windows from MLS point clouds using voxelspace and visibility analysis. *Int Arch Photogram Remote Sens Spatial Inf Sci XLIII-B2-2022:387–394*. <https://doi.org/10.5194/isprs-archives-xxliii-b2-2022-387-2022>
- Huang H, Michelini M, Schmitz M, Roth L, Mayer H (2020) LoD3 building reconstruction from multi-source images. *Int Arch Photogram Remote Sens Spatial Inf Sci XLIII-B2-2020:427–434*. <https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-427-2020>, <https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLIII-B2-2020/427/2020/>
- Kang Z, Yang J (2018) A probabilistic graphical model for the classification of mobile LiDAR point clouds. *ISPRS J Photogram Remote Sens* 143:108–123. <https://doi.org/10.1016/j.isprsjprs.2018.04.018>
- Li J, Lee GH (2019) USIP: unsupervised stable interest point detection from 3D point clouds. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 361–370
- Lowe D (1999) Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE international conference on computer vision. IEEE. <https://doi.org/10.1109/iccv.1999.790410>
- Mallick S (2016) Histogram of oriented gradients explained using OpenCV. learnopencv.com/histogram-of-oriented-gradients/
- Memon SA, Akhtar F, Mahmood T, Azeem M, Shaukat Z (2019) 3d shape retrieval using bag of word approaches. In: 2019 2nd international conference on computing, mathematics and engineering technologies (iCoMET). IEEE. <https://doi.org/10.1109/icomet.2019.8673397>
- Nowak E, Jurie F, Triggs B (2008) Sampling strategies for bag-of-features image classification. In: Leonardis A, Bischof H, Prinz A (eds) European conference on computer vision 2006, Part IV, vol 3954. Lecture notes in computer science. Springer, Berlin Heidelberg, pp 490–503
- Pantoja-Rosero BG, Achanta R, Kozinski M, Fua P, Perez-Cruz F, Beyer K (2022) Generating LoD3 building models from structure-from-motion and semantic segmentation. *Autom Construct* 141:104430
- Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. <https://doi.org/10.48550/ARXIV.1804.02767>
- Ripperda N (2010) Rekonstruktion von Fassadenstrukturen mittels formaler Grammatiken und Reversible Jump Markov Chain Monte Carlo Sampling. Ph.D. thesis, Leibniz Universität Hannover, Hannover (2010), wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Leibniz Universität Hannover
- Rublee E, Rabaud V, Konolige K, Bradski G (2011) ORB: An efficient alternative to SIFT or SURF. In: 2011 international conference on computer vision. IEEE. <https://doi.org/10.1109/iccv.2011.6126544>
- Salton G, McGill MJ (1986) Introduction to modern information retrieval. McGraw-Hill Inc, New York, NY, USA

- Sipiran I, Bustos B (2011) Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Comput* 27(11):963–976. <https://doi.org/10.1007/s00371-011-0610-y>
- Trimble Inc.: SketchUp 3d warehouse (2021). <https://3dwarehouse.sketchup.com>
- Tuttas S, Stilla U (2013) Reconstruction of façades in point clouds from multi aspect oblique ALS. *ISPRS Ann Photogram Remote Sens Spatial Inf Sci* II-3/W3:91–96. <https://doi.org/10.5194/isprsannals-II-3-W3-91-2013>, <http://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/II-3-W3/91/2013/>
- Wysocki O, Grilli E, Hoegner L, Stilla U (2023) Combining visibility analysis and deep learning for refinement of semantic 3d building models by conflict classification. *ISPRS Ann Photogram Remote Sens Spatial Inf Sci* X-4/W2:289–296. <https://doi.org/10.5194/isprs-annals-x-4-w2-2022-289-2022>
- Wysocki O, Hoegner L, Stilla U (2023) TUM-FAÇADE: reviewing and enriching point cloud benchmarks for façade segmentation. arXiv preprint [arXiv:2304.07140](https://arxiv.org/abs/2304.07140)
- Xu S, Fang T, Li D, Wang S (2010) Object classification of aerial images with bag-of-visual words. *IEEE Geosci Remote Sens Lett* 7(2):366–370. <https://doi.org/10.1109/lgrs.2009.2035644>
- Zhu Q, Zhong Y, Zhao B, Xia GS, Zhang L (2016) Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery. *IEEE Geosci Remote Sens Lett* 13(6):747–751. <https://doi.org/10.1109/lgrs.2015.2513443>

Generating 3D Roof Models from ALS Point Clouds Using Roof Line Topologies



Gefei Kong and Hongchao Fan

Abstract The automation of 3D roof reconstruction has become a critical research topic in the field of GIScience. Existing roof plane-based methods for this purpose need to segment roof planes and further extract roof vertices and edges after topology analysis. However, the roof plane-based primitive extraction and topology analysis may lead to additional errors for the next step's extraction result of roof vertices and edges. In this study, based on segmented roof plane point clouds, roof edges parallel to the x–y plane are extracted at first, and then the topology relationships of these special roof edges are analyzed and corrected by simple rules. This new approach simplifies the extraction of basic roof primitives and analyzes the roof structures and extracts roof vertices and edges at the same time, which reduce the accumulated errors by the processes of “multi-step primitive extraction—topology analysis—extraction of roof vertices and edges”. The qualitative and the preliminary quantitative experiment results indicate that the proposed approach can achieve the 3D roof reconstruction well.

Keywords 3D roof reconstruction · Point cloud data · Rule-based

1 Introduction

As global urbanization continues to grow and smart cities become more established, the demand for 3D building models increases (Peters et al. 2022). According to CityGML2.0, the famous standard of 3D building models, the 3D building model can be divided into 5 levels of detail (LoDs) ranging from LoD0 to LoD4 (Gröger

This article was selected based on the results of a double-blind review of an extended abstract.

G. Kong (✉) · H. Fan

Department of Civil and Environmental Engineering, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway

e-mail: gefei.kong@ntnu.no

H. Fan

e-mail: hongchao.fan@ntnu.no

et al. 2012). With the advancements in photogrammetry and computer science, it has become easier to acquire the information of building footprints and heights, and it enables the automation of 3D building reconstruction in LoD0 and LoD1. However, automating the 3D building reconstruction in higher LoDs, from LoD2 to LoD4, remains a challenge and continually attracts researchers. In higher-LoD 3D building models, the 3D roof structure is one of the most important components and plays a crucial role in accurately representing the geometry and appearance of buildings.

Existing methods of 3D roof reconstruction from point cloud data usually extract roof primitives (e.g., planes and their groups) at first with the assistance of pre-defined primitives (Huang et al. 2013; Li and Shan 2022) and machine learning methods (Li and Shan 2022; Chen et al. 2017; Nan and Wonka 2017; Wang et al. 2020; Huang et al. 2022). Then the 3D roof structure is reconstructed from these primitives by topology analysis (Xiong et al. 2014) or global optimization (Nan and Wonka 2017; Wang et al. 2020; Huang et al. 2022). After that, the roof vertices and edges are extracted to digitalize the roof structure. The methods based on pre-defined primitives perform stably and can maintain the correct topology of a 3D roof structure, but they are also limited by the inflexible pre-defined primitives. The data-driven methods without predefinition of primitives are more flexible because they directly extract planes from point cloud data by using computer-vision and machine-learning methods. However, additional errors of extracted roof vertices and edges may be introduced by the independent steps of primitive extraction and topology analysis. This is because that the extracted primitives (usually are extracted and regularized plane polygons or fitted roof plane functions) are obtained by multiple sub-step calculation and abstraction on the segmented point cloud data. These additional errors in these sub-steps would carry over to the topology analysis and ultimately lead to a reduction in the accuracy of the digital 3D roof model.

In this study, to avoid the issue of additional errors caused by separated steps of topology analysis and the multi-step generation of plane polygons and functions, a new integrated, unsupervised, and rule-based method is proposed. In the proposed method, firstly, the roof edges that are parallel to the x - y plane (i.e., roof lines) are extracted from segmented roof plane point clouds. These roof lines are then corrected and merged using simple rules with the assistance of building footprint data, and the resulting optimized roof lines are used to generate the final 3D roof structure. The advantages of the proposed method are as follows.

1. Instead of using roof plane polygons or fitted plane functions for topology analysis and structure reconstruction, this new proposed method utilizes the roof edges that are parallel to the x - y plane as the input of these steps. This change simplifies the extraction of basic primitives and allows that the proposed method analyzes topology and further extract roof faces and vertices at the same time with less constraints, which reduces the additional errors.
2. In addition to point cloud data, the proposed method also uses the building footprint data, which can be easily obtained by official or open sources. The additional building footprint data can help to improve the accuracy of extracted roof vertices, which implicitly constrains the adjacent (topological) relationship

between roof lines and ultimately enhances the overall quality of the 3D roof structure.

2 Methodology

The proposed approach starts from the segmented roof planes and consists of three main modules: (1) the extraction of roof edges that are parallel to the x–y plane (i.e., roof lines), (2) the correction and merging of roof lines with the assistance of building footprints, and (3) the generation of roof structure by corrected roof lines.

2.1 Extraction of Roof Lines

Roof lines contain all roof vertices of the 3D structure for a roof. The extraction of roof lines contributes to the location of roof vertices and key roof edges. Hence, this module serves as the foundation for the following roof structure generation. Moreover, the roof lines are easier to be located and extracted since they are parallel to the x–y plane and only the z-direction should be considered, while the extraction of all roof edges should consider the features in three directions (x-, y- and z-) which makes them more challenging to be extracted accurately.

As shown in the Fig. 1, for a roof, we define that its each roof plane contains two significant roof lines: lower and upper roof lines. They can be located and extracted easily by using the point sets around them. The workflow of this process is described as follows.

1. For the point clouds of a roof plane $rp = \{pts = (x, y, z)\}$ in a roof $R = \{rp\}$, , firstly, get its top-10 minimum (min) and maximum (max) z-values and calculate the means of them as the min and max z-values ($z_{rp,min}, z_{rp,max}$) of the roof plane. Then, select point sets whose z-values fall within the range $[z_{rp,min} \pm (thrs_z/2)]$

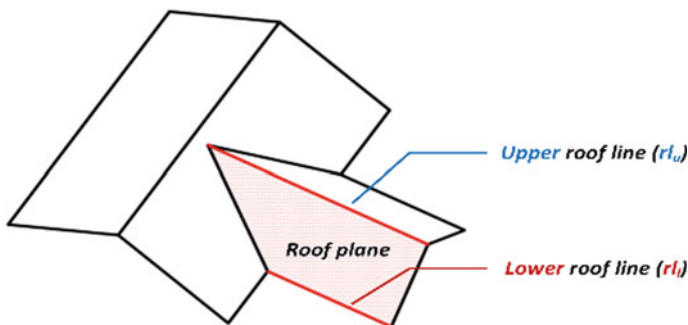


Fig. 1 The illustration of lower and upper roof lines

and $[z_{rp,max} \pm (thrs_z/2)]$ respectively, where $thrs_z$ is the threshold of limiting the range of lower and upper point sets. The selected point set around $z_{rp,min}$ is the lower point set $S_{rp,l}$, and the point set around $z_{rp,max}$ is the upper point set $S_{rp,u}$.

2. For the point sets $S_{rp,l}$ and $S_{rp,u}$, calculate their minimum bounding rectangles (MBRs) respectively. The edge with lowest z-values for the MBR of $S_{rp,l}$ is regarded as the lower roof line $rl_{rp,l}$, while the edge with highest z-values for $S_{rp,u}$ MBR is regarded as the upper roof line $rl_{rp,u}$.
3. Repeat previous two steps for each roof plane. Once this is completed, the lower and upper roof lines of all roof planes, i.e., $RL = \{\{rl_{rp,l}, rl_{rp,u}\} | rp \in R\}$, are obtained.

2.2 Correction and Merging of Roof Lines

This module can be divided into two sub-modules: (1) correction module and (2) merging module of roof lines.

In the correction module, the endpoints of roof lines which are close to its corresponding building footprint (FP_R), are corrected at first: they are aligned to the nearest vertices or the nearest points on the edges of FP_R . The rule defining that an endpoint is “close” to FP_R is: the Euclid distance between the endpoint and its closest FP_R vertex or edge is smaller than a specific threshold $thrs_{dp}$. This correction module is necessary because that the coordinates of extracted roof lines from the previous module are based on the point coordinates of roof plane, which may be affected by the occlusion or point density and cannot be aligned with the actual building footprint. The FP_R provides a series of accurate coordinates of some key points with a shape reference, and hence can improve the accuracy of the roof lines.

After the preliminary aligning step, the merging module is implemented to merge these roof lines and their endpoints, to avoid the topological gap between connected roof planes. In the merging module,

1. Separate all roof lines into several z-clusters by their z-values. The z-clusters are obtained by the algorithm Density-Based Spatial Clustering of Applications with Noise (DBSCAN), where the epsilon of this DBSCAN is set as $thrs_{dbz}$. The z-values of points in a z-cluster will be updated to the mean z-value of these points.
2. In each z-cluster, its contained roof lines are sorted by their lengths in descending order and then are aligned (merged) iteratively. In each iteration, the buffer of a roof line in this z-cluster is created with the buffer radius set as $thrs_{bf}$. The merging rule between this buffered roof line and other roof lines is defined as: the endpoints of other roof lines located within the buffer of this line are aligned to the nearest points on the current buffered roof line. In the next iteration, the buffer of the next roof line is created based on the corrected endpoints, and the moved endpoints are tagged and will not be considered in the following iterations. This process will repeat until all roof lines in the z-cluster have been considered.

3. After the moving step (step (2)), in each z-cluster, cluster endpoints by DBSCAN with the epsilon set as $thrs_{dp}$, and then further merge the endpoints in a cluster to get fewer endpoints. The merging rule of endpoints in a cluster is defined as: for the clusters that contain endpoints at FP_R vertices, the other endpoints in the cluster are aligned to their nearest endpoints at FP_R vertices included in this cluster; for the clusters that do not contain this type's endpoints, all endpoints in the cluster are merged into one new endpoint. The coordinate of the new endpoint is the mean of all endpoints in this cluster.
4. Repeat steps (2) and (3), until all z-clusters have been considered.

After the merging module, the correction module is repeated to further correct the merged roof lines. These steps in this module eliminate topology gaps between neighbored roof lines which are not close to the boundary of the building footprint and simplify the final 3D representation of a roof structure.

2.3 Generation of 3D Roof Structures

After the two modules introduced in Sects. 2.1 and 2.2, the simplified and more accurate roof lines of all roof planes are obtained. The 3D structure of this roof can be generated based on these roof lines. The workflow of this module is described as follows.

1. Analyze and record the intersection relationship between any two roof lines in 2D. If two roof lines intersect, identify whether they are lower or upper line respectively and record their corresponding roof faces (planes) which they belong.
2. For the endpoints of roof lines for each roof plane, order them by convex hull algorithm (Barber et al. 1996) to generate the preliminary face of this roof plane. The repeated endpoints are removed to simplify the face and support the representation of triangular roof planes. Faces of all roof planes $\{f\}$ can be obtained after repeating this process.
3. Execute the difference operation of intersected 2D faces. The intersection information between two faces is from step (1) and the face information is from step (2). For each face f and its related intersected faces f_{inst} , the following rule from (a) to (c) is defined to clean f_{inst} and remove the faces in f_{inst} which are not need to be considered:
 - a. These faces will be considered: the faces in f_{inst} whose upper roof lines are intersected with the lower roof line of f .
 - b. These faces will be considered: the faces in f_{inst} whose areas are smaller than the area of f . At the same time, their lower/upper roof lines are intersected with the lower/upper roof line of f .
 - c. The faces in f_{inst} not meeting the above two rules will be removed and will be not considered in difference operation.

After the cleaning, the f is used as the original face in the difference operation and the union of its cleaned f_{inst} is used as the face to be removed from the f . The 2D face after difference operation f_{diff} of f is regarded as one of the final roof face sets.

4. Add z-values for the new 2D face f_{diff} after difference operation by the following rules (d) and (e), to get the new 3D face f_{diff3D} .
 - d. For the face vertices in f_{diff} whose x-y coordinates on the boundary of the original face f , their z-values are defined as their nearest points' z-values in the vertex set of f .
 - e. For the face vertices in f_{diff} whose x-y coordinates not on the boundary of f : if they can be founded in the x-y coordinates of the vertex set for all faces $\{f\}$, their z-values are defined as the z-values of the matched vertices. Otherwise, their z-values are defined as the z-values of their nearest points in the vertex set of all faces $\{f\}$.

In addition, for the face vertices in f_{diff} whose distances to their nearest points in f or $\{f\}$ are smaller than the half of the min length of their corresponding footprint boundary but not at any vertex of f , they will be merged to their corresponding nearest vertices to simplify the roof structure.

5. Repeat steps (3) and (4) to obtain the new 3D faces $\{f_{diff3D}\}$ of all faces $\{f\}$.

In this module, by the topology analysis between these roof line-based faces, new roof vertices and edges presenting the right roof topology structure are extracted. These new 3D roof faces construct the final structure of this roof. We can further re-organize these faces to the corresponding output format (e.g., OBJ format) and show them in the related software.

3 Result and Discussion

The proposed method is implemented and tested on the point cloud data of 50 roofs in Trondheim, Norway. The point cloud data was provided by the mapping authority of Trondheim Municipality, with the point density of 12–20 points/m². The building footprint data was the national open geographical data and provided by the Norwegian Mapping Authority (FKB-Buildings Dataset 2023). Ground truth data of 3D roof structures in this experiment is manually extracted. Both (simple) primary (e.g., gabled and hipped) and (complex) combined (e.g., T-shape and L-shape) roof types are considered in the experiment.

The parameters mentioned in the proposed method are set as follows: $thrs_z = 0.7$ m, $thrs_{dp} = 1$ m, $thrs_{dbz} = 0.2$ m, and $thrs_{bf} = 0.45$ m. These settings are based on the input point cloud data and experience. The settings of $thrs_z$ and $thrs_{dbz}$ refer to the point spacing (point density) of the point cloud data: $thrs_z$ is set as a value larger than the double of max point spacing and $thrs_{dbz}$ is set as a value slightly smaller

than the min point spacing of the used dataset. The setting of $thrs_{dp}$ refers to the setting of the min length constraint of a building side in research (Awrangjeb 2016). For $thrs_{bf}$, considering it is similar to $thrs_{dp}$ and also related to the matching and merging of the endpoints of roof lines, we set this buffer radius parameter as a value around the half of $thrs_{dp}$ to provide a stricter constraint and avoid the over-merging.

To quantitatively evaluate the proposed method, the metrics including the vertex distance errors in x-, y-, and z- directions (vd_x , vd_y and vd_z) and the precision and recall of roof vertices (VP , VR) and edges (EP , ER) are calculated. The true positive vertices are defined as the predicted vertices whose minimum distances to ground truth vertices are lower than a specified threshold. The true positive edges are defined as the same way. In this paper, this threshold is set as 1 m. The quantitative evaluation result is 0.15 m of vd_x , 0.20 m of vd_y , 0.10 m of vd_z , 82.85% of VP , 94.06% of VR , 76.70% of EP , and 84.38% of ER . This result demonstrates that the proposed method can effectively and accurately reconstruct 3D roof structures.

The qualitative evaluation result is shown in Fig. 2. The 3D roof reconstruction result of different roof types is included in Fig. 2. As shown in Fig. 2, when reconstructing the roofs with the primary roof type (R1) and basic combined roof types (R2: T-shape, R3: L-shape), the proposed method performs well and accurately reconstruct 3D roof models. Furthermore, when reconstructing the roofs with more complex connection (topology) relationships (R4 to R6), the 3D roof models can still be reconstructed accurately by our method. The topologies of these reconstruction results are also correct. Overall, this evaluation result further demonstrates that the proposed method can successfully reconstruct the 3D roof structure with correct topologies in most cases in face of different roof types.

However, the proposed method is not perfect. The support of reconstructing roof models with concave planes is achieved based on the spatial topology operations between the overlapped convex roof planes. When facing the concave part of a roof plane which is not contained by the topology analysis, our method cannot present it properly. Hence, the extraction of roof lines for concave roof planes should be considered in the future. Moreover, in the current version of the proposed method, there is no explicit rule or constraint for triangular and over-small roof faces. It may lead to the topology error of the reconstruction result of some roof structures, and an example is shown by the red ellipse in Fig. 2. This problem is particularly acute when the point clouds of these roofs' planes are sparse or inaccurately segmented. Hence, this issue should be addressed in the future research.

4 Conclusion

The reconstruction of 3D roof structure is a key step in achieving 3D building modelling in LoD2 and higher LoDs. In this study, we introduce a novel integrated, unsupervised, and rule-based method to achieve 3D roof reconstruction. In this new method, the easily extracted lower and upper roof lines replace the roof plane polygon/boundary/function as the basic primitive. The 3D roof structure is

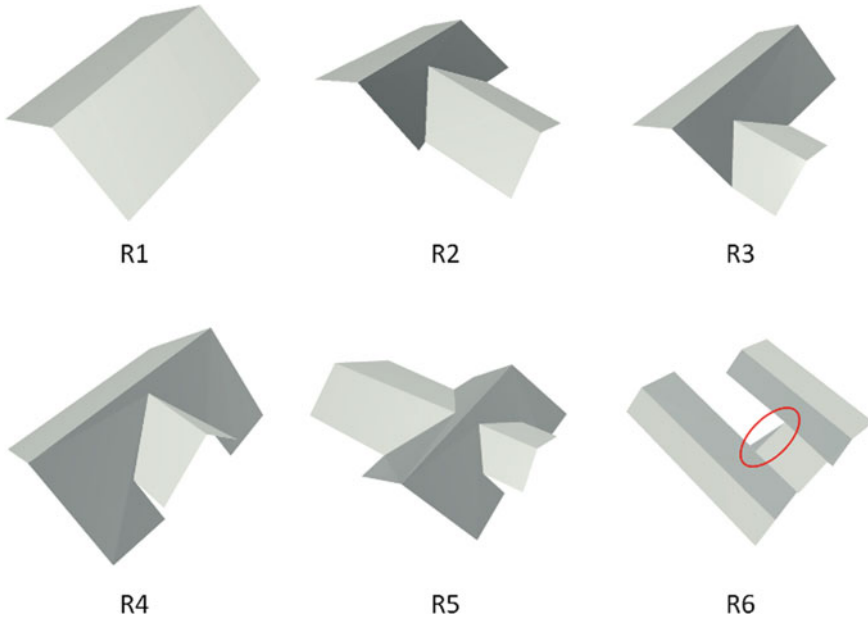


Fig. 2 Qualitative evaluation result of the proposed method

reconstructed based on the extracted and corrected roof lines of roof planes, with the support of the corresponding building footprints and several simple rules. The experiment result indicates that the proposed method is effective in reconstructing 3D roofs of various types.

In the future, the extraction of additional roof lines for concave roof planes and the explicit rules designed for triangular and over-small planes should be proposed, to further improve the generalization of the proposed method in face of more roof plane shapes and inaccurate roof segmentation results. Moreover, more detailed comparison experiments are necessary to be conducted to evaluate the performance of the proposed method more comprehensively.

Acknowledgements The work in this paper is supported by NTNU Digital project (project No. 81771593).

References

- Awrangjeb M (2016) Using point cloud data to identify, trace, and regularize the outlines of buildings. *Int J Remote Sens* 37(3):551–579
- Barber CB, Dobkin DP, Huhdanpaa H (1996) The quickhull algorithm for convex hulls. *ACM Trans Math Softw* 22(4):469–483

- Chen D, Wang R, Peethambaran J (2017) Topologically aware building rooftop reconstruction from airborne laser scanning point clouds. *IEEE Trans Geosci Remote Sens* 55(12):7032–7052
- FKB-Buildings Dataset, <https://kartkatalog.geonorge.no/metadata/fkb-bygning/8b4304ea-4fb0-479c-a24d-fa225e2c6e97>. Last accessed 21 Apr 2023 (download in 2021).
- Gröger G, Kolbe TH, Nagel C, Häfele K-H (2012) OGC city geography markup language (CityGML) encoding standard
- Huang H, Brenner C, Sester M (2013) A generative statistical approach to automatic 3D building roof reconstruction from laser scanning data. *ISPRS J Photogramm Remote Sens* 79:29–43
- Huang J, Stoter J, Peters R, Nan L (2022) City3D: Large-scale building reconstruction from airborne LiDAR point clouds. *Remote Sens* 14(9):2254
- Li Z, Shan J (2022) RANSAC-based multi primitive building reconstruction from 3D point clouds. *ISPRS J Photogramm Remote Sens* 185:247–260
- Nan L, Wonka P (2017) Polyfit: polygonal surface reconstruction from point clouds. In: *Proceedings of the IEEE international conference on computer vision*. IEEE: Piscataway, NJ, pp 2353–2361
- Peters R, Dukai B, Vitalis S, van Liempt J, Stoter J (2022) Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the Netherlands. *Photogramm Eng Remote Sens* 88(3):165–170
- Wang S, Cai G, Cheng M, Marcato Junior J, Huang S, Wang Z, Su S, Li J (2020) Robust 3D reconstruction of building surfaces from point clouds based on structural and closed constraints. *ISPRS J Photogramm Remote Sens* 170:29–44
- Xiong B, Oude Elberink S, Vosselman G (2014) A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds. *ISPRS J Photogramm Remote Sens* 93:227–242

MLS2LoD3: Refining Low LoDs Building Models with MLS Point Clouds to Reconstruct Semantic LoD3 Building Models



Olaf Wysocki, Ludwig Hoegner, and Uwe Stilla

Abstract Although highly-detailed LoD3 building models reveal great potential in various applications, they have yet to be available. The primary challenges in creating such models concern not only automatic detection and reconstruction but also standard-consistent modeling. In this paper, we introduce a novel refinement strategy enabling LoD3 reconstruction by leveraging the ubiquity of lower LoD building models and the accuracy of MLS point clouds. Such a strategy promises at-scale LoD3 reconstruction and unlocks LoD3 applications, which we also describe and illustrate in this paper. Additionally, we present guidelines for reconstructing LoD3 facade elements and their embedding into the CityGML standard model, disseminating gained knowledge to academics and professionals. We believe that our method can foster development of LoD3 reconstruction algorithms and subsequently enable their wider adoption.

Keywords LoD3 reconstruction · LoD3 applications · Refinement strategy · CityGML · Semantic 3D city models

This article was selected based on the results of a double-blind review of an extended abstract.

O. Wysocki (✉) · L. Hoegner · U. Stilla
Photogrammetry and Remote Sensing, TUM School of Engineering and Design,
Technical University of Munich (TUM), Munich, Germany
e-mail: olaf.wysocki@tum.de

L. Hoegner
e-mail: ludwig.hoegner@tum.de; ludwig.hoegner@hm.edu

U. Stilla
e-mail: stilla@tum.de

L. Hoegner
Department of Geoinformatics, University of Applied Science (HM), Munich, Germany

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
T. H. Kolbe et al. (eds.), *Recent Advances in 3D Geoinformation Science*, Lecture Notes
in Geoinformation and Cartography, https://doi.org/10.1007/978-3-031-43699-4_23

1 Introduction

Reconstructing semantic 3D building models is a long-standing challenge in geoinformatics and photogrammetry. Recent open source and proprietary solutions have proved effective for country-scale level of detail (LoD) 1 and 2 reconstructions,¹ given that aerial observations and footprints are provided (Roschlaub and Batscheider 2016). Such 3D models find their applications in various fields ranging from analysing roof solar potential to estimating building-energy demand (Biljecki et al. 2015).

However, at-scale reconstruction of semantic 3D building models at LoD3 remains in its infancy (Biljecki et al. 2014). The primary obstacle has been the lack of highly-detailed, street-level data capturing the distinctive feature of the LoD3: facade elements. Numerous initiatives stemming from academia and industry promise to close this data-gap by conducting large-scale mobile laser scanning (MLS) campaigns acquiring high-density, street-level point clouds and images (Wysocki et al. 2022c).

Much research has been devoted to reconstructing facade elements (Szeliski 2010), yet methods embedding these elements directly into the semantic 3D building models standards' structure are scarce. The principal challenges lie in the incompleteness of acquired MLS data, where often only the frontal facade is measured. This feature hinders from-scratch reconstruction, which assumes very high point cloud coverage (Pantoja-Rosero et al. 2022; Huang et al. 2020; Nan and Wonka 2017). Another challenge concerns single-object semantics, typically derived in the semantic segmentation process. Such an approach implicitly discards the hierarchical complexity of the semantic data model, rendering it insufficient for directly creating semantic 3D data models, for instance, embedding 3D window objects into a wall surface belonging to a building entity within a city model (Beil et al. 2021).

In this paper, as opposed to traditional from-scratch reconstruction, we propose the refinement strategy to create LoD3 building models. The strategy minimizes the influence of data incompleteness by using low LoD solid geometry as prior and adding new geometry only where required. The method also reduces the complexity of maintaining hierarchical reconstruction by embedding new objects into existing model structures; We list our contributions as follows:

- Conflict-driven refinement strategy of LoD 1 and 2 to LoD3 building models while considering uncertainty
- Introducing guidelines of semantic LoD3-level facade elements reconstruction
- Presenting applications of LoD3 building models.

2 LoD3 Applications

There are multiple studies elaborating on applications of semantic 3D building models, which, however, concentrate on LoD 1 and 2 models (Biljecki et al. 2015;

¹ <https://github.com/OloOcki/awesome-citygml>.

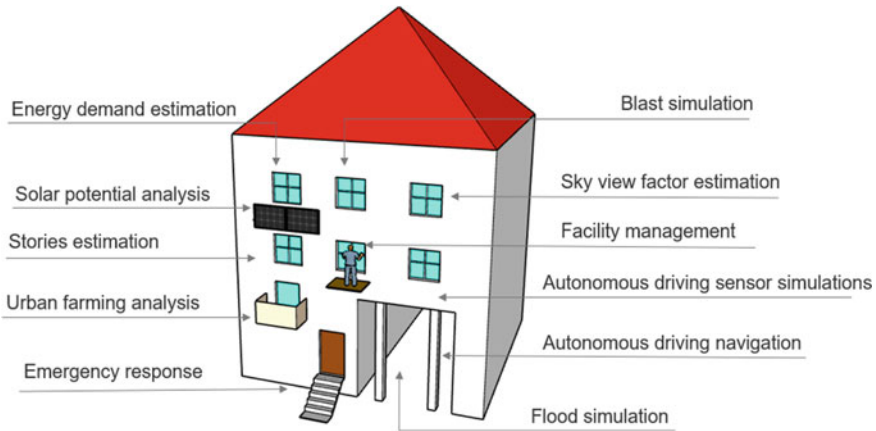


Fig. 1 Applications of semantic 3D building models at LoD3

Willenborg et al. 2018). This phenomenon primarily owes to the ubiquity of LoD 1 and 2 models; for example, there are approximately 140 million freely available semantic 3D building models in the US, Switzerland, and Poland.²

However, many professionals and academics have already expressed the need for LoD3 building models in various applications. As illustrated in Fig. 1, the applications concern simulating flood damage (Amirebrahimi et al. 2016), estimating heating demand (Nouvel et al. 2013), calculating façade solar potential (Willenborg et al. 2018), testing automated driving functions (Schwab and Kolbe 2019), supporting autonomous driving navigation (Wysocki et al. 2022b), managing building facilities (Moshrefzadeh et al. 2015), measuring sky-view factor (Lee et al. 2013), analysing building potential for vertical farming (Palliwal et al. 2021), investigating blasts impact (Willenborg et al. 2016), navigating for emergency units (Kolbe et al. 2008), and inferring number of stories (Biljecki and Ito 2021).

3 Refining LoD 1 and 2 to Obtain LoD3 Building Models

As shown in the strategy overview in Fig. 2, the measurements and 3D building models are analyzed in the visibility analysis part, which estimates absent objects in LoD 1 or 2 (conflicts). The conflict semantic is derived in the following step. The semantic 3D reconstruction part not only reconstructs geometries but also assigns them semantics and embeds into the standardized, CityGML 2.0 building model representation (Gröger et al. 2012); yielding refined, LoD3 building model. Since the visibility analysis and conflict understanding parts are at length presented in publications such as Wysocki et al. (2022a, 2022b, 2023), Tuttas and Stilla (2013),

² <https://github.com/OloOcki/awesome-citygml>.

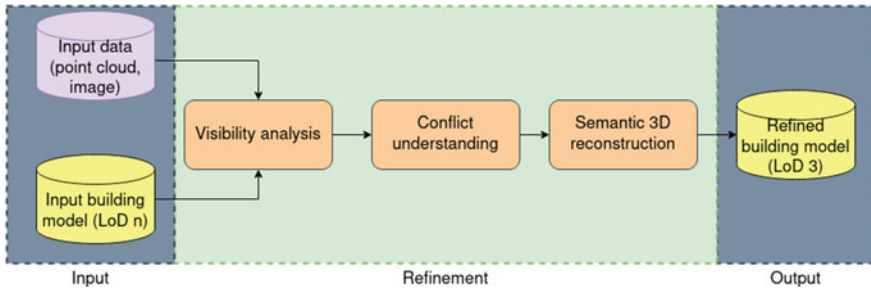


Fig. 2 Refinement strategy workflow

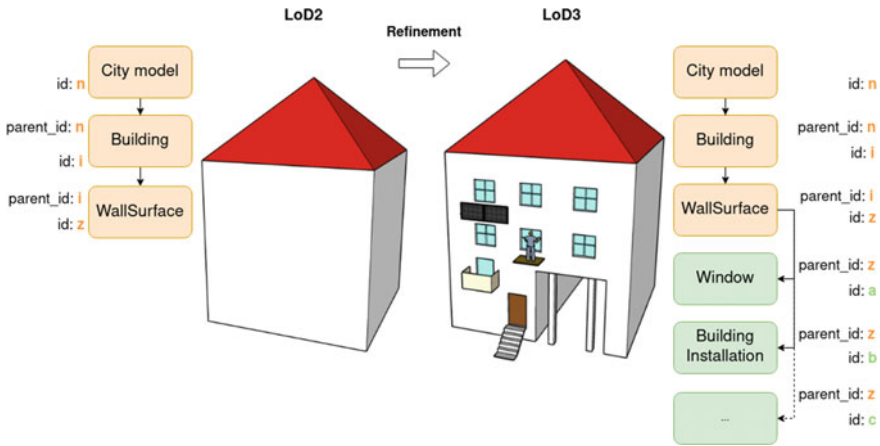


Fig. 3 Concept of the refinement: wall surface serves as a projection plane for facade elements and as a link to building entity and subsequently city model

Tuttas et al. (2015). We present merely an excerpt about them in Sect. 3.1, and elaborate on embedding reconstructed objects into the widely-adopted CityGML standard of semantic 3D building models in Sect. 3.2. The concept of the refinement exploits planar surfaces present in the semantic 3D city models both for detection and reconstruction as well as for the semantic embedding into the model, as illustrated in Fig. 3.

3.1 Visibility Analysis and Conflict Understanding

The method comprises three main input blocks, as shown in the overview in Fig. 4. The backbone of our method is the visibility analysis, where conflicts between laser rays and vectors are analyzed and projected as *conflict probability map* to the wall, yielding geometric cues about missing LoD3 facade elements (Fig. 5). Both modal-

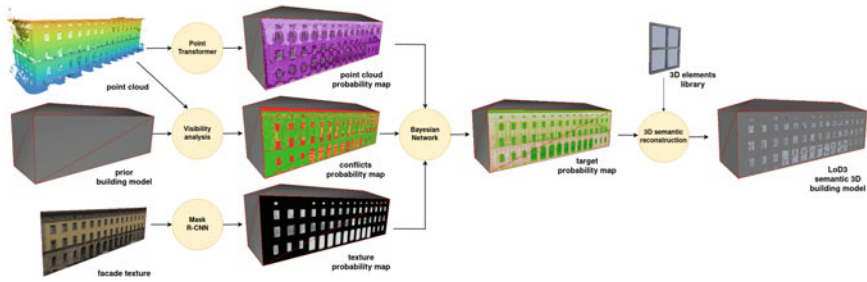


Fig. 4 Conflicts understanding performed using combination of visibility analysis, semantically segmented point clouds and images. Adapted from Wysocki et al. (2023)

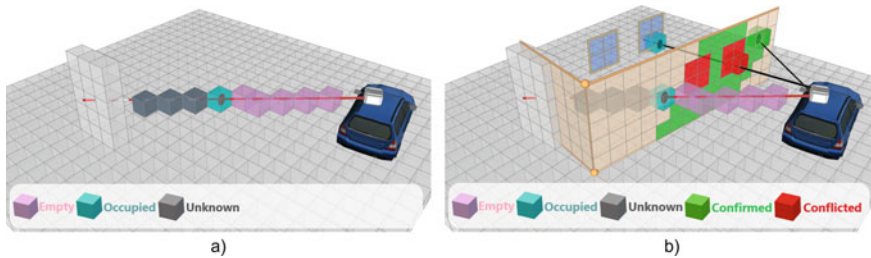


Fig. 5 Visibility analysis employed on a voxel grid to identify LoD3 objects, absent in LoD 1 and 2 building models (*conflicts*). **a** Ray casting from sensor origin provides voxel state *empty* if the observation ray traverses it; *occupied* when it contains hit point; *unknown* if unmeasured; **b** Joint analysis of rays and vector models provides another set of states: *confirmed* when *occupied* voxel intersects with vector plane; and *conflicted* when the plane intersects with an *empty* voxel. Adapted from Wysocki et al. (2023)

ities have different uncertainties of global positioning accuracy of building model surfaces and of point clouds along the ray. We assume the probability distribution of global positioning accuracy of a building surface $P(A)$ is described by the Gaussian distribution $\mathcal{N}(\mu_1, \sigma_1)$, where μ_1 and σ_1 are the mean and standard deviation of the Gaussian distribution. Analogically, we describe the probability distribution of global positioning accuracy of a single point in point cloud $P(B)$ by the Gaussian distribution $\mathcal{N}(\mu_2, \sigma_2)$. We use the joint probability distribution of two independent events $P(A)$ and $P(B)$ to obtain final probability scores for the $P_{confirmed}$ and conflicted $P_{conflicted}$ states of the voxel V_n (Wysocki et al. 2023):

$$V_n = \left\{ \begin{array}{l} P_{confirmed}(A, B) = P(A) * P(B) \\ P_{conflicted}(A, B) = 1 - P_{confirmed}(A, B) \end{array} \right\} \quad (1)$$

We derive the semantic information about conflicts from two data sources: 3D point clouds and images projected to the wall. The 3D point clouds are semantically segmented using the enhanced Point Transformer network (Wysocki et al. 2023). Their final class scores are projected onto the wall surface, forming a *point cloud*

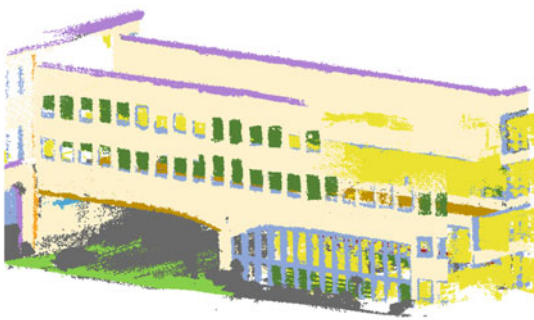
probability map. Analogically, *texture probability map* is created from the inferred classes of the Mask-RCNN network (Wysocki et al. 2023) applied to the building wall texture. The probability maps are combined using the Bayesian network, which estimates geometry and semantics for the facade elements reconstruction. We automatically reconstruct facade elements by fitting a pre-defined library of objects into their identified respective instances. The fitting comprises a selection of a corresponding 3D model matched with the detected class and rigid scaling to the identified extent.

3.2 Embedding 3D Reconstructed Objects into the CityGML Standard

For semantic segmentation of facade-level MLS point clouds, up to 17 classes can be recognized (Fig. 6). As we show in Table 1, these are pertinent to the reconstruction adhering to the CityGML 2.0 standard (Gröger et al. 2012; Special Interest Group 3D 2020; Biljecki et al. 2014).

However, the refinement necessitates addressing application-specific requirements, prior model’s LoD, and characteristics of street-level measurements (Table 1).

Ground Surface, Roof Surface, and Wall Surface positions shall not be altered, as it will corrupt the overall geometry of the prior model since the street-level measurement typically covers only one or two sides of the building. Such change shall be only undertaken if the point cloud significantly covers the building’s outer shell (e.g., see Nan and Wonka 2017) and is of higher global accuracy than usually cadastre-derived Wall Surface (Roschlaub and Batscheider 2016). Yet, even when the geometry is altered, the entity’s identifiers shall not be changed, as there exist external and



| No | Class | HEX |
|----|-----------------------|---------|
| 1 | wall | #FFF2CC |
| 2 | window | #8EA9DB |
| 3 | door | #C65911 |
| 4 | balcony | #008080 |
| 5 | molding | #B482DA |
| 6 | deco | #00B0F0 |
| 7 | column | #FF0000 |
| 8 | arch | #305496 |
| 9 | drainpipe | #FFA500 |
| 10 | stairs | #BC8F8F |
| 11 | ground surface | #7CFC00 |
| 12 | terrain | #C6E0B4 |
| 13 | roof | #A80000 |
| 14 | blinds | #548235 |
| 15 | outer ceiling surface | #BF8F00 |
| 16 | interior | #FFFF00 |
| 17 | other | #696969 |

Fig. 6 Facade-level classes available in point cloud datasets on an example of the TUM-FAÇADE dataset (Wysocki et al. 2022c)

Table 1 Analyzed point cloud and CityGML classes relevant for the LoD3 reconstruction with new, proposed functions (green) for absent ones (Special Interest Group 3D 2020)

| Point cloud class | CityGML class | LoD | Function | Refinable | Confidence score |
|-------------------|-----------------------|------------|----------------|-----------|------------------|
| Ground surface | Ground surface | 1, 2, 3, 4 | – | ~ | ~ |
| Roof surface | Roof surface | 1, 2, 3, 4 | – | ~ | ~ |
| Wall | Wall surface | 1, 2, 3, 4 | – | ~ | ✓ |
| Window | Window | 3, 4 | – | ✓ | ✓ |
| Door | Door | 3, 4 | – | ✓ | ✓ |
| Underpass | Building installation | 3, 4 | 1002 underpass | ✓ | ✓ |
| Balcony | Building installation | 3, 4 | 1000 balcony | ✓ | ✓ |
| Molding | Building installation | 3, 4 | 1016 molding | ✓ | ✓ |
| Deco | Building installation | 3, 4 | 1017 deco | ✓ | ✓ |
| Column | Building installation | 3, 4 | 1011 column | ✓ | ✓ |
| Arch | Building installation | 3, 4 | 1008 arch | ✓ | ✓ |
| Drainpipe | Building installation | 3, 4 | 1018 drainpipe | ✓ | ✓ |
| Stairs | Building installation | 3, 4 | 1060 stairs | ✓ | ✓ |
| Blinds | Building installation | 3, 4 | 1019 blinds | ✓ | ✓ |

internal links associated with city model objects. For example, an identifier of a wall can be linked to a report about its solar potential; in case the identifier is changed, this information will be lost.

Regardless of the total building coverage, identified openings, such as underpasses, windows, and doors, shall cut out the Wall Surface geometry where required, and 3D geometries should be fitted into this empty space (Fig. 7).

In this way, the highly-detailed representation, watertightness, and planarity of the prior model are maintained.

The notion of detection and reconstruction confidence is required since many applications nowadays rely not only on geometric accuracy but also on the associated confidence of the object's semantics. For example, map-based navigation of cars uses a voting process fusing multiple sensor detections to decide upon the next manoeuvre (Wilbers et al. 2019). Therefore, it is of great value to retain this information and add it as Generic Attribute called *confidence* $\in [0, 1]$ to the specific object. This attribute also allows for an update of the object's position and semantics when

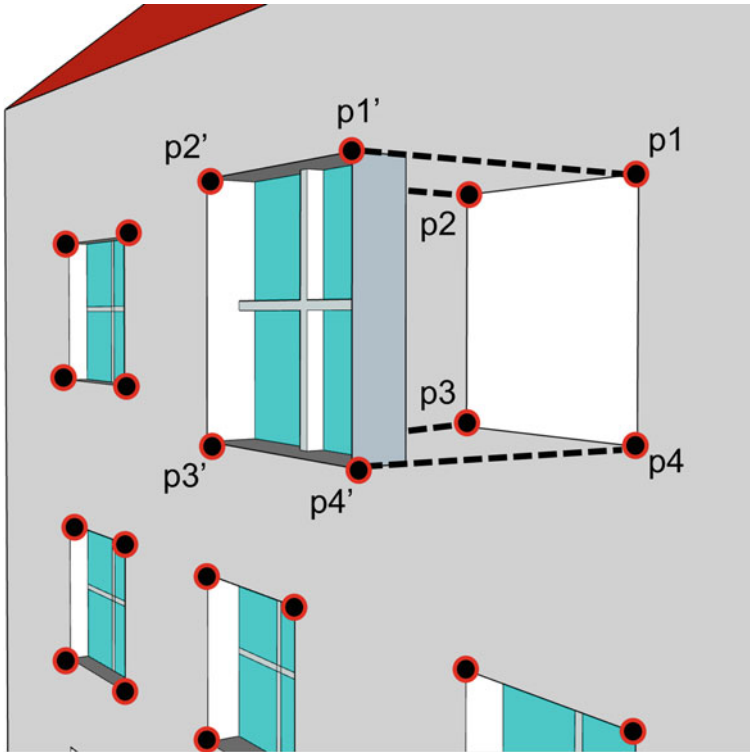


Fig. 7 Exemplary junction points (red-black) of cut shapes as fitting points on an example of pre-defined 3D window models

the next measurement epoch exceeds the confidence score (Zou and Sester 2022). Here, of great value is the timestamp of acquisition, too Gröger et al. (2012).

Although it is allowed to model an underpass at LoD2, in practice, it is rarely the case², primarily owing to the aerial perspective of data acquisition for LoD2 models (Dukai et al. 2020). We deem underpasses as facade openings that substantially impact facade semantics and geometry; as such classify them as LoD3 features.

4 Examples

The TUM-FAÇADE (Wysocki et al. 2022c) dataset, TUM-MLS-2016 (Zhu et al. 2020), and openly available LoD2 and LoD3 building models³ (Bayerische Vermessungsverwaltung 2023) depicting the main campus of the Technical University of Munich, Germany, were employed for the experiments.

³ <https://github.com/tum-gis/tum2twin>.

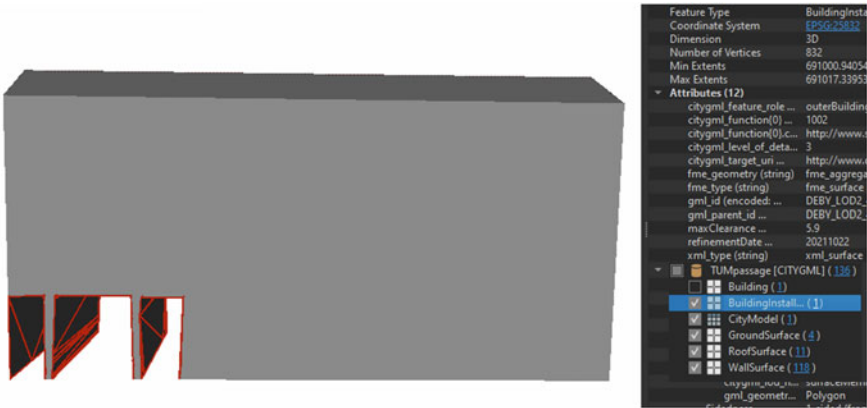


Fig. 8 Refinement result on an example of underpass embedded into the CityGML 2.0 model (right table)

The TUM-MLS-2016 was collected using the Mobile Distributed Situation Awareness (MODISSA) platform comprising two Velodyne HDL-64E LiDAR sensors. The TUM-FAÇADE data was derived from the TUM-MLS-2016 and enriched by facade-level classes. The point cloud dataset was complemented by a proprietary point cloud dataset MF, which was also acquired at the Technical University of Munich campus. The point cloud was geo-referenced using a mobile mapping platform supported by the German SAPOS RTK system (3D Mapping Solutions 2023).

The LoD2 building models were modeled by combining 2D footprints and aerial observations (Roschlaub and Batscheider 2016). We textured the models by manually acquiring terrestrial pictures and projecting them to respective faces; we used a 13MP rear camera of a Xiaomi Redmi Note 5A smartphone and applied no orthorectification to mimic a direct projection from a mobile mapping platform or street view imagery (Hou and Biljecki 2022). Based on LoD2 models and a combination of the TUM-FAÇADE and the proprietary MF point cloud, LoD3 models were manually created. Such models served as a comparison set (ground-truth) to automatically generated models. Additionally, they can be used as examples for practitioners. In this paper, we showcased the modeling paradigm of refinement; for the extensive discussion and comparison of the detection and reconstruction performance, see Wysocki et al. (2022a, 2022b, 2023).

We present examples of three refinement types: windows, doors, and underpasses (Figs. 8, 9, and 10); The code is freely available within the repository.⁴

⁴ <https://github.com/OloOcki/scan2lod3>.

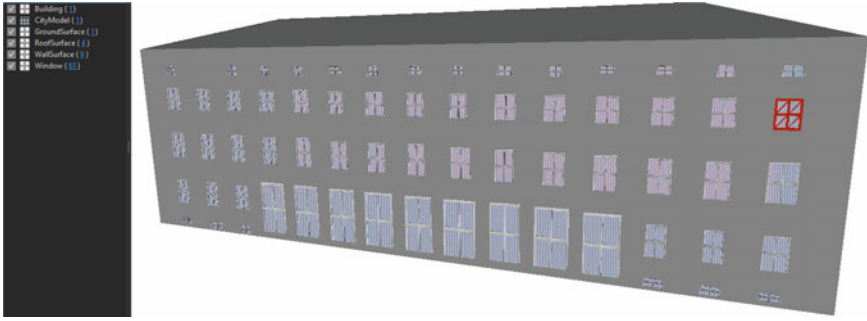


Fig. 9 Refinement result on an example of windows embedded into the CityGML 2.0 model (left table) with the highlighted one instance (red)

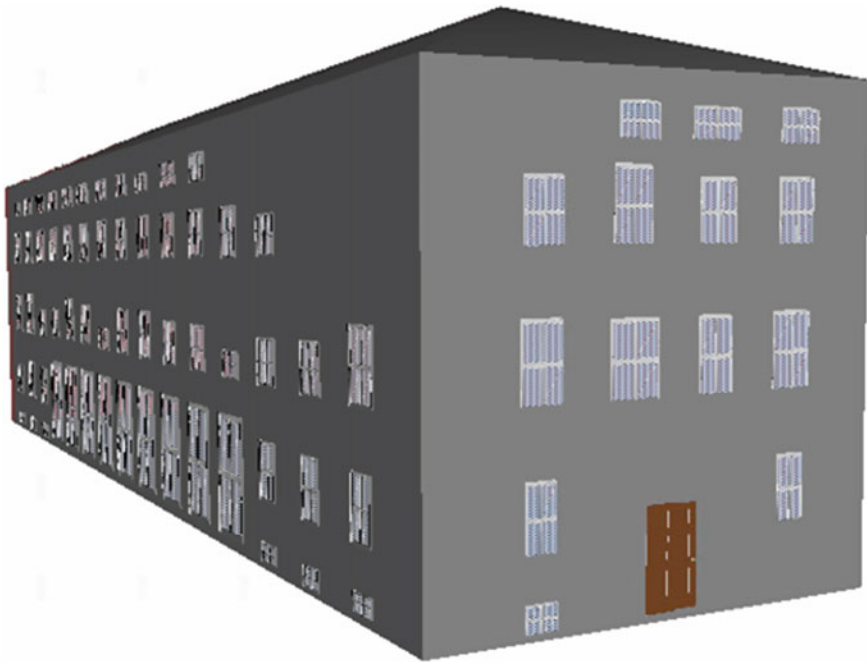


Fig. 10 Refinement result on an example of window and door types embedded into the CityGML 2.0 model. Note that here the TUM-MLS-2016 point cloud did not cover the whole building, and as such windows are partially missing, which can be complemented, for example, by an additional laser measurement or oblique images

5 Discussion

As shown in Figs. 8, 9, and 10, usage of planes of existing building models provides reliable projection targets for ray-to-model conflicts, semantic point clouds, and images; Moreover, it ensures homogeneous extent and pixel size. Such plane-like assumptions have also proven effective for the co-registration of point clouds in the urban environment (Schwarz et al. 2023; Dong et al. 2020). In our experiments, we investigated terrestrial images projected to the plane, yet using pre-textured models or available oblique photos is possible, too, owing to the exploitation of the target plane. More evidence shall improve performance as we base our inference on Bayesian reasoning. Therefore, combining other modalities is foreseeable and can mitigate occlusion issues as illustrated in Fig. 10.

Subsequent modeling also benefits from such a setup, as it is based on conventional constructive solid geometry operations, minimizing the reconstruction task complexity. However, reliance on existing models requires flawless models and assumes their adherence to the CityGML standard. On the other hand, in case the models do not exist for a particular region of interest, they can be generated using such established solutions as PolyFit (Nan and Wonka 2017) for reconstructing LoD2 without footprints; or 3Dfier for prismatic generation of LoD1 building models (Ledoux et al. 2021).

In this experiment, we created a library of objects with merely one type per each facade element. However, works exist that tackle more sophisticated matching of various object library types while assuming correct detection, e.g., employing the Bag of Words approach (Froech et al. 2023). Note that our training dataset for point cloud semantic segmentation consists of several building types, which are representative of a central European architecture style and for such setup yields promising results (Tan et al. 2023; Wysocki et al. 2023). The scalability for other architectural is foreseeable yet needs to be explored in future work.

6 Conclusion

According to the Cambridge Dictionary, the word refinement means “a small change that improves something” or “the process of making a substance pure” (Cambridge Dictionary 2023). In the case of our proposed strategy, both definitions are valid. On the one hand, we perform relatively minor changes to the 3D building model, improving its accuracy and semantic completeness significantly. On the other hand, we preserve valid and remove undesired information from LoD1 and 2 while reconstructing LoD3 building models.

We conclude that our refinement strategy allows for at-scale LoD3 building reconstruction by leveraging the ubiquity of existing semantic 3D building models and MLS accuracy. The experiments corroborate that the approach minimizes the reconstruction complexity and maintains 3D cadastre records by retaining 3D building

database identifiers. This paper also presents guidelines for researchers and practitioners who investigate the reconstruction of LoD3 CityGML building models. We firmly believe that the presented research can facilitate further development of algorithms for LoD3 reconstruction and unlock shown LoD3 building applications.

Acknowledgements This work was supported by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy within the framework of the IuK Bayern project *MoFa3D - Mobile Erfassung von Fassaden mittels 3D Punktwolken*, Grant No. IUK643/001. Moreover, the work was conducted within the framework of the Leonhard Obermeyer Center at the Technical University of Munich (TUM). We gratefully acknowledge the Geoinformatics team at the TUM for the valuable insights and for providing the CityGML datasets.

References

- 3D Mapping Solutions: MoSES mobile mapping platform—technical details (2023). <https://www.3d-mapping.de/ueber-uns/unternehmensbereiche/data-acquisition/unser-vermessungssystem/>. Accessed 30 Jan 2023
- Amirebrahimi S, Rajabifard A, Mendis P, Ngo T (2016) A BIM-GIS integration method in support of the assessment and 3D visualisation of flood damage to a building. *J Spat Sci* 61(2):317–350
- Bayerische Vermessungsverwaltung: 3D-Gebäudemodelle (LoD2) (2023) <https://geodaten.bayern.de/opengeodata/OpenDataDetail.html?pn=lod2>. Accessed 30 Jan 2023
- Beil C, Kutzner T, Schwab B, Willenborg B, Gawronski A, Kolbe TH (2021) Integration of 3D point clouds with semantic 3D city models—providing semantic information beyond classification. *ISPRS Ann Photogrammetry Rem Sens Spat Inf Sci* 8
- Biljecki F, Ito K (2021) Street view imagery in urban analytics and GIS: a review. *Landsc Urban Plan* 215:104217
- Biljecki F, Ledoux H, Stoter J, Zhao J (2014) Formalisation of the level of detail in 3D city modelling. *Comput Environ Urban Syst* 48:1–15
- Biljecki F, Stoter J, Ledoux H, Zlatanova S, Çöltekin A (2015) Applications of 3D city models: state of the art review. *ISPRS Int J Geo-Inf* 4(4):2842–2889
- Cambridge Dictionary (2023) Refinement definition. Cambridge University Press & Assessment. <https://dictionary.cambridge.org/dictionary/english/refinement>. Accessed 30 Jan 2023
- Dong Z, Liang F, Yang B, Xu Y, Zang Y, Li J, Wang Y, Dai W, Fan H, Hyyppä J et al (2020) Registration of large-scale terrestrial laser scanner point clouds: a review and benchmark. *ISPRS J Photogrammetry Remote Sens* 163:327–342
- Dukai B, Peters R, Wu T, Commandeur T, Ledoux H, Baving T, Post M, van Altena V, van Hinsbergh W, Stoter J (2020) Generating, storing, updating and disseminating a countrywide 3D model. *Int Arch Photogrammetry Remote Sens Spat Inf Sci* 44:27–32
- Froech T, Wysocki O, Hoegner L, Stilla U (2023) Reconstructing facade details using MLS point clouds and Bag-of-Words approach. In: Accepted to proceedings of 3D GeoInfo 2023, Lecture notes in geoinformation and cartography
- Gröger G, Kolbe TH, Nagel C, Häfele KH (2012) OGC city geography markup language CityGML encoding standard. Open Geospatial Consortium, Wayland, MA, USA
- Hou Y, Biljecki F (2022) A comprehensive framework for evaluating the quality of street view imagery. *Int J Appl Earth Observ Geoinf* 115:103094
- Huang H, Michelini M, Schmitz M, Roth L, Mayer H (2020) LoD3 building reconstruction from multi-source images. *Int Arch Photogrammetry Remote Sens Spat Inf Sci XLIII-B2-2020:427–434*. <https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-427-2020>, <https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLIII-B2-2020/427/2020/>

- Kolbe TH, Gröger G, Plümer L (2008) CityGML-3D city models and their potential for emergency response, pp 273–290 (2008)
- Ledoux H, Biljecki F, Dukai B, Kumar K, Peters R, Stoter J, Commandeur T (2021) 3dfier: automatic reconstruction of 3D city models. *J Open Source Softw* 6(57):2866
- Lee D, Pietrzyk P, Donkers S, Liem V, van Oostveen J, Montazeri S, Boeters R, Colin J, Kastendeuch P, Nerry F et al (2013) Modelling and observation of heat losses from buildings: the impact of geometric detail on 3D heat flux modelling. In: Proceedings 33rd AERSEL symposium “Towards horizon 2020: earth observation and social perspectives”, Matera, Italy, 3–6 June 2013-Geomatics Synthesis Project
- Moshrefzadeh M, Donaubaauer A, Kolbe TH (2015) A CityGML-based façade information model for computer aided facility management. In: Bridging Scales-Skalenübergreifende Nah-und Fernerkundungsmethoden, vol 35. Wissenschaftlich-Technische Jahrestagung der DGPF (2015)
- Nan L, Wonka P (2017) PolyFit: polygonal surface reconstruction from point clouds. In: Proceedings of the IEEE international conference on computer vision, pp 2353–2361
- Nouvel R, Schulte C, Eicker U, Pietruschka D, Coors V (2013) CityGML-based 3D city model for energy diagnostics and urban energy policy support. In: IBPSA world, pp 1–7
- Palliwal A, Song S, Tan HTW, Biljecki F (2021) 3D city models for urban farming site identification in buildings. *Comput. Environ. Urban Syst.* 86:101584
- Pantoja-Rosero BG, Achanta R, Kozinski M, Fua P, Perez-Cruz F, Beyer K (2022) Generating LoD3 building models from structure-from-motion and semantic segmentation. *Autom. Constr* 141:104430
- Roschlaub R, Batscheider J (2016) An INSPIRE-conform 3D building model of Bavaria using cadastre information, LiDAR and image matching. *Int Arch Photogram Remote Sens Spat Inf Sci XLI-B4:747–754*
- Schwab B, Kolbe TH (2019) Requirement analysis of 3D road space models for automated driving. *ISPRS Ann Photogram Remote Sens Spat Inf Sci IV-4/W8:99–106*
- Schwarz S, Pilz T, Wysocki O, Hoegner L, Stilla U (2023) Transferring facade labels between point clouds with semantic octrees while considering change detection. In: Accepted to proceedings of 3D GeoInfo 2023, Lecture Notes in Geoinformation and Cartography
- Special Interest Group 3D (2020) Modeling guide for 3D objects—part 2: modeling of buildings (LoD1, LoD2, LoD3)—SIG3D quality wiki EN. <https://en.wiki.quality.sig3d.org/>. Accessed 30 Jan 2023
- Szeliski R (2010) Computer vision: algorithms and applications. Springer Science & Business Media
- Tan Y, Wysocki O, Hoegner L, Stilla U (2023) Classifying point clouds at the facade-level using geometric features and deep learning networks. In: Accepted to proceedings of 3D GeoInfo 2023, Lecture Notes in Geoinformation and Cartography
- Tuttas S, Stilla U (2013) Reconstruction of façades in point clouds from multi aspect oblique ALS. *ISPRS Ann Photogram Remote Sens Spat Inf Sci II-3/W3:91–96*
- Tuttas S, Stilla U, Braun A, Borrmann A (2015) Validation of BIM components by photogrammetric point clouds for construction site monitoring. *ISPRS Ann Photogram Remote Sens Spat Inf Sci II-3/W4:231–237*
- Wilbers D, Merfels C, Stachniss C (2019) Localization with sliding window factor graphs on third-party maps for automated driving. In: 2019 International conference on robotics and automation (ICRA), pp 5951–5957. IEEE
- Willenborg B, Pültz M, Kolbe TH (2018) Integration of semantic 3D city models and 3D mesh models for accuracy improvements of solar potential analyses. *Int Arch Photogram Remote Sens Spat Inf Sci XLII-4/W10:223–230*
- Willenborg B, Sindram M, Kolbe TH (2016) Semantic 3D city models serving as information hub for 3D field based simulations. *Lösungen für eine Welt im Wandel*, pp 54–65
- Willenborg B, Sindram M, Kolbe TH (2018) Applications of 3D city models for a better understanding of the built environment. In: Trends in spatial analysis and modelling: decision-support and planning strategies, pp 167–191

- Wysocki O, Grilli E, Hoegner L, Stilla U (2022a) Combining visibility analysis and deep learning for refinement of semantic 3D building models by conflict classification. *ISPRS Ann Photogram Remote Sens Spat Inf Sci X-4/W2-2022*: 289–296
- Wysocki O, Hoegner L, Stilla U (2022b) Refinement of semantic 3D building models by reconstructing underpasses from MLS point clouds. *Int J Appl Earth Observ Geoinf* 111:102841
- Wysocki O, Hoegner L, Stilla U (2022c) TUM-FAÇADE: reviewing and enriching point cloud benchmarks for façade segmentation. *Int Arch Photogram Remote Sens Spat Inf Sci XLVI-2/W1-2022*: 529–536
- Wysocki O, Xia Y, Wysocki M, Grilli E, Hoegner L, Cremers D, Stilla U (2023) Scan2LoD3: reconstructing semantic 3D building models at LoD3 using ray casting and Bayesian networks, pp 6547–6557
- Zhu J, Gehring J, Huang R, Borgmann B, Sun Z, Hoegner L, Hebel M, Xu Y, Stilla U (2020) TUM-MLS-2016: an annotated mobile LiDAR dataset of the TUM City Campus for semantic point cloud interpretation in urban areas. *Remote Sens* 12(11):1875
- Zou Q, Sester M (2022) Uncertainty representation and quantification of 3D building models. *Int Arch Photogram Remote Sens Spat Inf Sci XLIII-B2-2022*:335–341

Semantic Segmentation of Buildings Using Multisource ALS Data



Agata Walicka and Norbert Pfeifer

Abstract In this paper we propose to utilize a SparseCNN network and national ALS datasets of Vienna and Zurich to achieve generalization of a classifier by including both datasets simultaneously in the training phase. The data was classified into ground and water, vegetation, building and bridges, and ‘other’. The results were evaluated using median IoU. The classifier trained with both datasets performed only slightly worse on ground and water and on vegetation in comparison to the classifiers trained and tested using dataset from the same city (maximum drop of 0.3 pp from a value above 94%). For building and bridges the accuracy slightly improves (at least 0.6 pp), whereas for ‘other’ results are inconsistent. The classifier trained using both datasets performed substantially better than the classifiers trained using one dataset and tested on the other. Thus, training using multiple datasets leads to a more general classifier while maintaining accuracy.

Keywords LiDAR · ALS · Deep learning · Semantic segmentation · Buildings

1 Introduction

Semantic segmentation is a first step of ALS (Airborne Laser Scanning) point cloud processing algorithms that are used for many city management applications, such as detection of building footprints, creation of digital twins and city models, preparation of deformation maps and many others. As a result, its accuracy highly influences the results of further processing.

This article was selected based on the results of a double-blind review of an extended abstract.

A. Walicka (✉)

Institute of Geodesy and Geoinformatics, Wrocław University of Environmental and Life Sciences, Grunwaldzka 53, 50-357 Wrocław, Poland
e-mail: agata.walicka@upwr.edu.pl

N. Pfeifer

Department of Geodesy and Geoinformation, Technische Universität Wien,
1040 Vienna, Austria

Recently, deep learning approaches for semantic segmentation of ALS data gained the attention of the community as they enable high semantic segmentation accuracy with relatively fast processing after the network training phase. However, deep learning requires a large amount of high quality training data that is not easily available. Because of this reason, most of the deep learning models are trained and tested on available benchmarking data sets, such as ISPRS Vaihingen (Niemeyer et al. 2014), DALES (Varney et al. 2020), or Dublin (Laefer and Abuwarda 2017; Zolanvari et al. 2019) data sets. For instance, in Wen et al. (2021) a global-local graph attention convolutional neural network was presented and its performance was tested using Vaihingen and IEEE GRSS Data Fusion Contest 2019 (DFC2019). Similarly, in Zeng et al. (2023) a deep learning model was trained and tested on three benchmarking data sets: Vaihingen, DFC2019 and LASDU. However, while the benchmarking data sets are characterized by high accuracy of labels, they usually cover a relatively small area. As a result, the possibilities of training the model and testing its accuracy are very limited.

Therefore, lately the ALS data collected and published by different European countries becomes more and more commonly used for this purpose. For instance, in Widyaningrum et al. (2021) the AHN data set was employed to test a semantic segmentation framework based on the DGCNN (Dynamic Graph CNN) method. In Soilán et al. (2019) tested the ability of PointNet architecture to classify point clouds between the different acquisitions of AHN data. In Nurunnabi et al. (2021) the AHN data and Luxembourg data was used for testing the accuracy of a deep learning based binary classification for ground and non-ground surfaces.

For both benchmarking and national data set oriented investigations, the network is trained and tested individually for each data set that is processed. As a result, the possible generalization of the trained models to other data sets is not investigated. Therefore, in Xie et al. (2021) tested the ability of the deep learning model to accurately classify point clouds collected for different areas and using different sensors. The performed experiments included two models trained separately for Vaihingen and LASDU data sets. Then the performance of each model was evaluated in two scenarios: (1) when the classifier is trained and tested on data from the same benchmarking data set and (2) when the model is trained on one data set and tested on the other. The tests included the analysis of the influence of input point density, data augmentation and intensity feature on the accuracy of generalization. However, in Xie et al. (2021) the possibility of a joint training using two different datasets was not taken into consideration.

Therefore, in this paper, we would like to show the advantages of using two data sets simultaneously for training a deep network, especially in the context of semantic segmentation of buildings. To achieve this goal, we propose to utilize the SparseCNN network (Graham et al. 2018; Graham and Van Der Maaten 2017) and ALS national data sets collected for Vienna and Zurich.

2 Test Sites and Data

In recent years, many European countries decided to publish their national airborne laser scanning data sets. The free access to spatial data is provided by, for instance, the Netherlands, Poland, Denmark, Switzerland and many others. Since these data sets were collected independently by various countries, their characteristics and the form in which they are made available differ substantially between data sets. Some of these differences can potentially be exploited in order to create more robust deep learning based classifiers. However, some of them cause it is complicated to use them to their full potential. One of the limiting factors are differences between the classification schemes and the classification accuracy.

Since nowadays simple point cloud semantic segmentation algorithms are commonly available, most of the available data sets are classified before they are published. However, in most cases, the manual correction of the classification is not provided (e.g. Denmark). This leads to many errors and lack of consistency in the classification of the data set. Because of this problem, many of the available data sets cannot be used for training of deep learning networks. On the other hand, several countries decided to manually correct and/or provide very accurate automatic classification (e.g., Switzerland and the Netherlands). These data sets are characterized by small amount of errors and high consistency of the classification. However, these data sets often present different approaches for data classification schemes. For instance, in the Netherlands the vegetation points are included in the 'other' class whereas in, for instance, the Swiss, Danish and Polish data sets vegetation constitutes a separate class. Because of this difference it is not always possible to make full use of Dutch data without further preprocessing that aims for unification of classification.

As a result of the analysis of the available European data sets, their accuracy and classification scheme, the Zurich¹ and Vienna data sets were selected for the first experiments. For both of these datasets, the manual correction of automatic classification was provided. As a result, the classification accuracy of both of them is extremely high. Moreover, these data sets are characterized by consistent classification schemes. Both of them were classified into ground, buildings, vegetation, water, bridges and 'other'. Additionally, the Vienna data set includes the noise class that consists of high and low points. The Vienna data set consists of 12 rectangular tiles that were divided equally for training and testing. The tiles have the size of 1270 m by 1020 m and include 20 m of overlap. The point density in the tiles varies between 20 pts/m² and 140 pts/m². The data was collected in October and November 2015. The Zurich data set consists of 31 square, non-overlapping tiles that have size of 1000 m by 1000 m. 14 of these tiles were used for training and 17 for testing purposes. The point density in the training tiles varies between 8 pts/m² and 20 pts/m². The data was collected between 2017 and 2018. The documentation of the Zurich data set states that the data was collected during low vegetation and snow coverage season.

¹ Data source: Federal Office of Topography swisstopo.

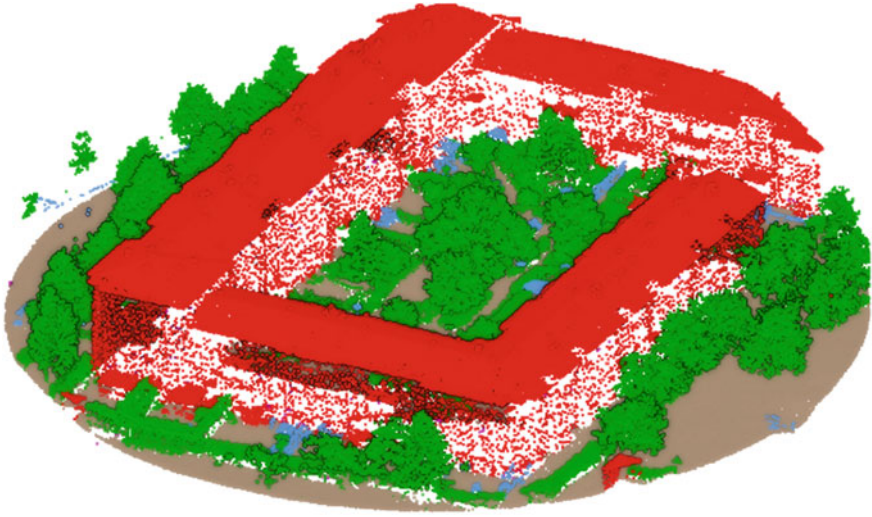


Fig. 1 The example of data patch. Colors represent data classification (red—buildings, green—vegetation, brown—ground, blue—other)

For the training purposes, the training data sets were divided into a set of patches according to the procedure described in Winiwarter et al. (2019). Each patch consists of 600 thousand points and in most cases has a shape of a circle (Fig. 1). In order to fit the test data into graphic card memory, each tile of Vienna data was divided into a set of smaller tiles.

3 Methods

The semantic segmentation was performed using the SparseCNN implementation of neural networks. In the case of point cloud processing, SparseCNN is a voxel-based network that performs three dimensional convolution operation. To limit the amount of operations, the network restricts processing and results to the set of non-empty voxels. This also prevents the convolutional operations from increasing the amount of active voxels. The network takes the coordinates of each voxel and its feature values as an input. The features are processed as tensors while the coordinates are stored in a hash table. Hash tables are updated only when the size of the point cloud changes as a result of the neural network operation. SparseCNN enables definition of typical neural network layers, such as convolution, deconvolution, maxpooling, dropout etc. However, most of them are adjusted for the processing of sparse 3D data as described above.

For the point cloud semantic segmentation we implemented a typical U-net style network with the architecture tested in Li et al. (2021). We defined a voxel size of

8 cm and filter size of 3, the encoder stage consists of 7 levels with additional residual connections. Each convolution operation was producing additional 16 features. The initial value of learning rate was equal to 0.001 and it was adjusted in each epoch according to the cosine decay learning rate schedule. The network was trained over 18 epochs. The training was performed using a batch size of 6. For class balancing, we used a weighted cross-entropy loss function with weights described in Wen et al. (2020). Number of returns and return number were used as input features for the neural network.

The point cloud was classified into four classes: ground and water, buildings and bridges, vegetation and ‘other’. The experiments aimed at understanding how introduction of multi source training data can influence the results of semantic segmentation of buildings. As a result, the three types of experiments were performed:

- (1) The classifier was trained using training data from Vienna and its accuracy was evaluated based on the test data set of Vienna. Similarly, the classifier was trained using training data from Zurich and evaluated on Zurich data.
- (2) The classifier was trained using training data from both Vienna and Zurich data sets simultaneously and its accuracy was evaluated independently using test data of both data sets.
- (3) The classifier was trained using the training data set from Vienna and tested using the Zurich test data set. Similarly, the classifier was trained using training data from Zurich and tested using the Vienna data set.

The amount of training data for Zurich and Vienna stayed the same for each experiment.

The accuracy of semantic segmentation was evaluated for each tile of both data sets based on Intersection over Union (IoU) metric. The final accuracy was established by calculating the median value of the statistic for the data set of each city.

4 Results and Discussion

The performed experiments revealed that it is possible to achieve generally high accuracy of semantic segmentation in the case of each experiment. The accuracy of the results is presented in Table 1.

In the case of Experiment 1, the achieved median IoU of semantic segmentation was between 97.3 and 53.5% for the test data of Vienna and between 97.8 and 50.8% for the test set of Zurich. The lower accuracy of ‘other’ class was caused by several reasons including underrepresentation of the class in the data (both training and testing), large differences in the geometry of objects that are included in this class and the smallest classification accuracy of this class in the training data.

Experiment 2 revealed that including additional training data from another source enabled to keep the accuracy of ground and vegetation classes while enabling to increase the accuracy of building and bridges class by almost 1 percent point and

Table 1 Median values of IoU achieved for different combinations of training and testing data

| Type of dataset | | Median IoU (%) | | | |
|-----------------|--------------|------------------|-------------|-----------------------|-------------|
| Training data | Testing data | Ground and water | Vegetation | Buildings and bridges | Other |
| Vienna | Vienna | 97.3 | 94.8 | 92.4 | 53.5 |
| Zurich | Vienna | 95.9 | 91.6 | 89.7 | 35.7 |
| Vienna & Zurich | Vienna | 97.2 | 94.7 | 93.3 | 55.1 |
| Zurich | Zurich | 97.8 | 96.6 | 93.6 | 50.8 |
| Vienna | Zurich | 96.6 | 95.2 | 90.5 | 41.6 |
| Vienna & Zurich | Zurich | 97.5 | 96.4 | 94.2 | 50.0 |

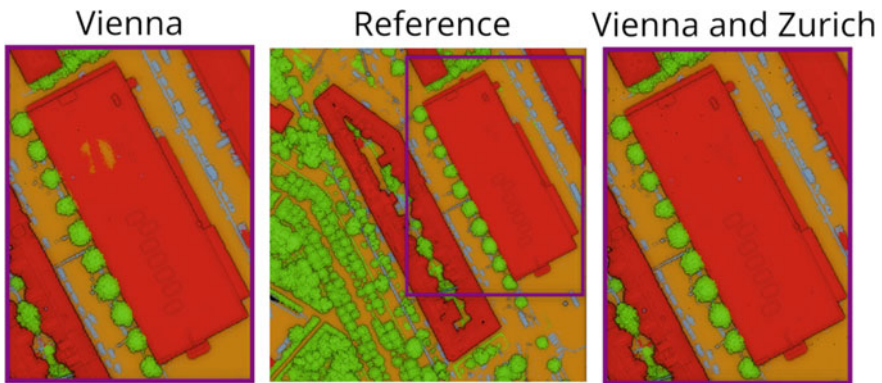


Fig. 2 The example of the improvement of semantic segmentation for the Vienna dataset. The semantic segmentation result achieved when the training was performed using: only Vienna data (left) and Vienna and Zurich data (right). The reference data is visualized in the middle figure. Colors represent data classification (red—buildings and bridges, green—vegetation, brown—ground and water, blue—‘other’)

‘other’ class by 1.5 percent point in the case of Vienna data. A similar result can be observed for the Zurich data set. In this case the accuracy of ground and water and vegetation classes were kept whereas the buildings and bridges accuracy is slightly improved. However, we can observe a decrease in accuracy of ‘other’ class. The example of the improvement of the semantic segmentation of Vienna data achieved by introducing Zurich data to the classifier is presented in Fig. 2. When the classifier was trained using only Vienna data, there were two clusters of points on the building roof that were classified as ground. When the Zurich data was introduced to the classifier, these clusters were correctly classified as building. Figure 3 presents the improvement of the semantic segmentation of Zurich data. The classifier trained only using Zurich data was not able to correctly classify the roof of the train station. When the data from Vienna was introduced into the training process, the classifier was able to reduce the misclassification area to several small clusters.

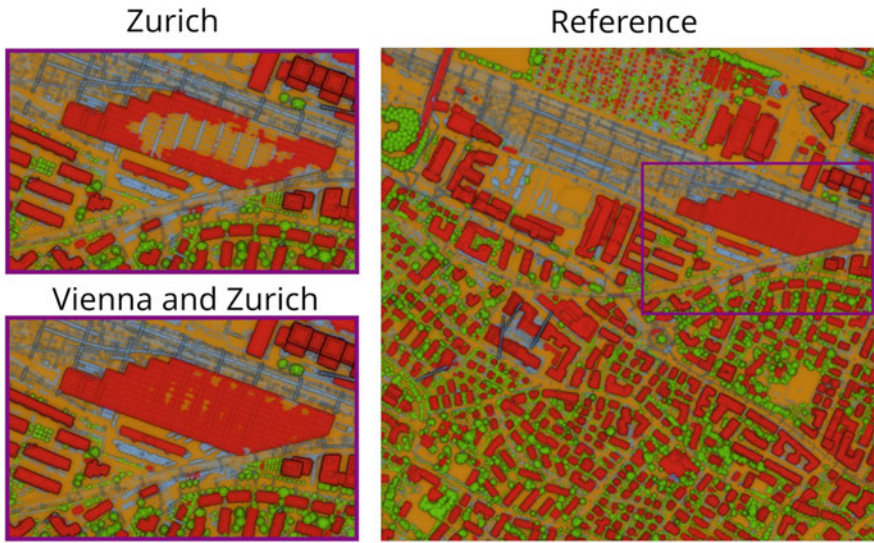


Fig. 3 The example of the improvement of semantic segmentation for the Zurich dataset. The semantic segmentation result achieved when the training was performed using: only Zurich data (top left) and Vienna and Zurich data (bottom left). The reference data is visualized in the right part of the figure. Colors represent data classification (red—buildings and bridges, green—vegetation, brown—ground and water, blue—‘other’)

On the other hand, Experiment 3 shows that the classifier trained on both data sets achieves substantially higher accuracy than the classifier trained on Zurich data and tested on Vienna data. The difference varies between 1.3 percent point and 19.4 percent point. A similar effect can be observed for a classifier trained on Vienna data and tested on Zurich data. In this case the difference varies between 0.9 percent point to 8.4 percent point. The highest improvements are achieved by ‘other’ class in both cases. However, the second biggest improvement for both data sets is achieved for buildings and bridges class and is equal to around 3.5 percent point. Figure 4 presents the improvements of the semantic segmentation of Vienna data that can be achieved by introducing the Vienna dataset to the training data instead of using only the Zurich dataset. When the classifier was trained only using the Zurich data set a whole building was classified as vegetation while after introducing the Vienna data set most of the building was correctly classified. However, there are also smaller improvements that lead to the better median IoU accuracy. An example of several small improvements is presented in Fig. 5. The figure shows the semantic segmentation result achieved by using a classifier trained using the Vienna dataset and tested on Zurich data in comparison with the classifier that was trained on both datasets. While the Zurich classifier was able to more accurately classify the train station roof, it also tended to overestimate the building range leading to “leaking” of the buildings points to the terrain points. This is clearly visible in the purple rectangle area. It also misclassified cars as buildings (yellow rectangle) and classified part of the viaduct as ground (pink

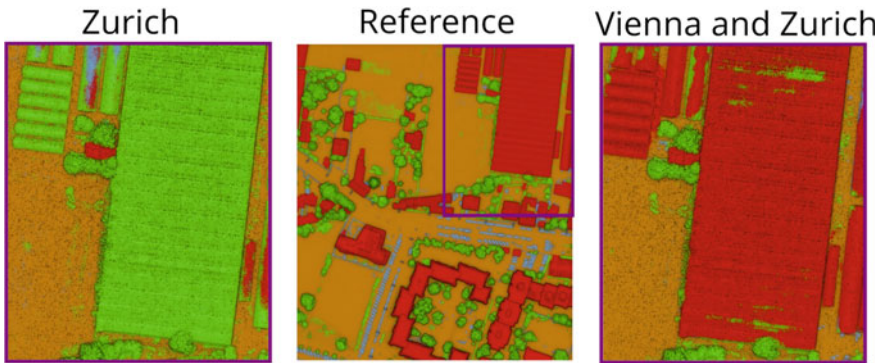


Fig. 4 The example of the improvement of semantic segmentation for the Zurich dataset. The semantic segmentation result achieved when the training was performed using: only Vienna data (left) and Vienna and Zurich data (right). The reference data is visualized in the middle of the figure. Colors represent data classification (red—buildings and bridges, green—vegetation, brown—ground and water, blue—‘other’)

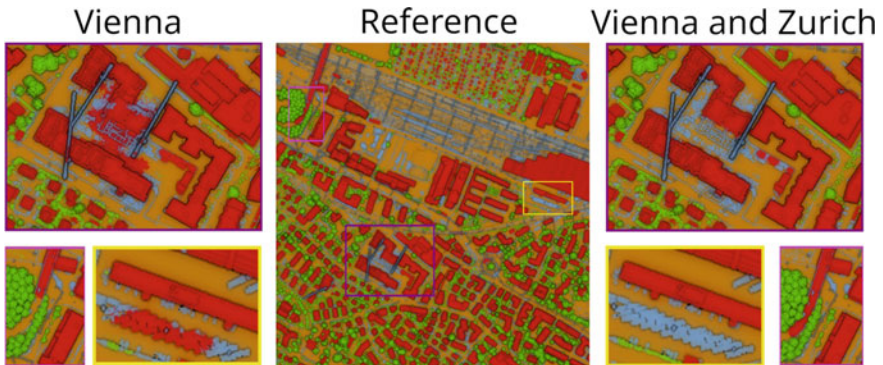


Fig. 5 The example of the improvement of semantic segmentation for the Vienna dataset. The semantic segmentation result achieved when the training was performed using: only Zurich data (left) and Vienna and Zurich data (right). The reference data is visualized in the middle of the figure. Colors represent data classification (red—buildings and bridges, green—vegetation, brown—ground and water, blue—‘other’)

rectangle). All of these errors were avoided when the Vienna data set was included in the training process.

The performed experiments proved that the classifier resulting from mixing multi source data sets is more general than the classifier that is trained using only one source of data. This approach limits the probability of overfitting the model to the common errors typical to the used data set. Moreover, the Experiment 3 reveals that it is not the exceptional similarity of Vienna and Zurich data sets that leads to the high accuracy of a joint classifier because the accuracy of a classifier trained on Zurich

data and tested on Vienna data and the other way around is substantially lower than in the case of a joint classifier.

5 Conclusions

The performed experiments show that including additional data source in training data enables to achieve similar or higher accuracy of semantic segmentation. Moreover, it allows us to achieve a more general classifier that is able to accurately classify both data sets. This approach may be especially useful when the data set that we aim at classifying has lower semantic segmentation accuracy of training data than the other accessible data sets. In this case, instead of including more of lower accuracy training data and overfitting to the common classification errors in this data set, data from additional sources can be added.

6 Future Work

Since the current experiments provide promising results, further work will be carried out in order to achieve the universal classifier. The aim will be to achieve a classifier that will be able to accurately classify various ALS datasets without the need for additional training. As a part of future work, the classifier will be trained and tested using various available ALS data sets. However, this will require additional work related to unification of the existing classification schemes for the data sets which classification schemes do not match the adopted scheme of classification into buildings and bridges, vegetation, ground and water, and ‘other’.

Acknowledgements This research was funded by National Science Center, Poland in the frame of a grant number UMO-2021/41/N/ST10/02996.

References

- Graham B, Engelcke M, Van Der Maaten L (2018). 3d semantic segmentation with submanifold sparse convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 9224–9232
- Graham B, Van der Maaten L (2017). Submanifold sparse convolutional networks. arXiv preprint [arXiv:1706.01307](https://arxiv.org/abs/1706.01307)
- Laefer DF, Abuwarda S, Vo AV, Truong-Hong L, Gharibi H (2017). 2015 aerial laser and photogrammetry survey of Dublin city collection record
- Li N, Kähler O, Pfeifer N (2021). A comparison of deep learning methods for airborne lidar point clouds classification. *IEEE J Sel Top Appl Earth Obs Remote Sens* 14:6467
- Niemeyer J, Rottensteiner F, Soergel U (2014). Contextual classification of lidar data and building object detection in urban areas. *ISPRS J Photogram Remote Sens* 87:152

- Nurunnabi A, Teferle F, Li J, Lindenbergh R, Hunegnaw A (2021). An efficient deep learning approach for ground point filtering in aerial laser scanning point clouds. *Int Arch Photogram Remote Sens Spat Inf Sci* 43:31–38
- M. Soilán, R. Lindenbergh, B. Riveiro, A. Sánchez-Rodríguez (2019). PointNet for the automatic classification of aerial point clouds, *ISPRS Ann. Photogramm Remote Sens Spat Inf Sci IV-2/W5*, pp 445–452
- Varney N, Asari VK, Graehling Q (2020). DALES: a large-scale aerial LiDAR data set for semantic segmentation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp 186–187 (2020)
- Wen C, Yang L, Li X, Peng L, Chi T (2020). Directionally constrained fully convolutional neural network for airborne LiDAR point cloud classification. *ISPRS J Photogram Remote Sens* 162:50
- Wen C, Li X, Yao X, Peng L, Chi T (2021). Airborne LiDAR point cloud classification with global-local graph attention convolution neural network. *ISPRS J Photogram Remote Sens* 173:181
- Widyaningrum E, Bai Q, Fajari MK, Lindenbergh RC (2021). Airborne laser scanning point cloud classification using the DGCNN deep learning method. *Remote Sens* 13(5):859
- Winiwarter L, Mandlbürger G, Schmohl S, Pfeifer N (2019). Classification of ALS point clouds using end-to-end deep learning. *PFG–J Photogram Remote Sens Geoinf Sci* 87:75
- Xie Y, Schindler K, Tian J, Zhu XX (2021). Exploring cross-city semantic segmentation of ALS point clouds. *Int Arch Photogram Remote Sens Spat Inf Sci* 43:247
- Zeng T, Luo F, Guo T, Gong X, Xue J, Li H (2023). Recurrent residual dual attention network for airborne laser scanning point cloud semantic segmentation. *IEEE Trans Geosci Remote Sens* 61
- Zolanvari S, Ruano S, Rana S, Cummins A, da Silva RE, Rahbar M, Smolic A (2019). DublinCity: annotated LiDAR point cloud and its applications. arXiv preprint [arXiv:1909.03613](https://arxiv.org/abs/1909.03613)

Classifying Point Clouds at the Facade-Level Using Geometric Features and Deep Learning Networks



Yue Tan, Olaf Wysocki, Ludwig Hoegner, and Uwe Stilla

Abstract 3D building models with facade details are playing an important role in many applications now. Classifying point clouds at facade-level is key to create such digital replicas of the real world. However, few studies have focused on such detailed classification with deep neural networks. We propose a method fusing geometric features with deep learning networks for point cloud classification at facade-level. Our experiments conclude that such early-fused features improve deep learning methods' performance. This method can be applied for compensating deep learning networks' ability in capturing local geometric information and promoting the advancement of semantic segmentation.

Keywords Geometric features · Point cloud classification · Deep learning

1 Introduction

Nowadays, semantic 3D building models are widely used in many fields, such as architecture, engineering, construction, and facilities management (Biljecki et al. 2015). Meanwhile, the widespread use of light detection and ranging (LiDAR) data provides the possibility of at-scale reconstruction of 3D building models up to Level

This article was selected based on the results of a double-blind review of an extended abstract.

Y. Tan (✉) · O. Wysocki · L. Hoegner · U. Stilla
Photogrammetry and Remote Sensing, TUM School of Engineering and Design, Technical
University of Munich (TUM), Munich, Germany
e-mail: yue.tan@tum.de

O. Wysocki
e-mail: olaf.wysocki@tum.de

L. Hoegner
e-mail: ludwig.hoegner@tum.de; ludwig.hoegner@hm.edu

U. Stilla
e-mail: stilla@tum.de

L. Hoegner
Department of Geoinformatics, University of Applied Science (HM), Munich, Germany

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
T. H. Kolbe et al. (eds.), *Recent Advances in 3D Geoinformation Science*, Lecture Notes
in Geoinformation and Cartography, https://doi.org/10.1007/978-3-031-43699-4_25

of Detail (LoD)² (Haala and Kada 2010).¹ Street-level point clouds with rich and detailed facade-level semantic can enable the reconstruction of highly detailed LoD3 building models. In addition, the reliable and detailed segmentation of point clouds can also provide improvement on the achievable LoD. Such point clouds stem from mobile laser scanning (MLS) units, whose availability has increased in recent years (Wysocki et al. 2022). With the wide use of low-cost laser scanning system, it is easy and cheaper to acquire point clouds now. However, the automatic interpretation of 3D point clouds by semantic segmentation for high LoD reconstruction represents a very challenging task (Grilli and Remondino 2020). It leads to a growing need of innovative point cloud classification methods to extract highly-detailed semantic information (Grilli and Remondino 2020).

In this study, we propose a method combining the deep learning networks and geometric features together for improving networks' performance in point cloud classification at facade-level. Our contributions are as follows:

- Comparison of deep learning (DL) and machine learning (ML) approaches on classifying building point clouds at facade-level
- A method improving the performance of DL networks by adding geometric features into DL approaches
- Analysis on impact from selection of geometric features upon the deep learning networks.

2 Related Works

Recently, machine- and its subset deep-learning algorithms (ML/DL) have become the state-of-the-art approach to classify point clouds (Grilli and Remondino 2020). For example, the traditional machine-learning-based approaches that apply geometric features for point cloud classification, such as Random Forest, have been used in the cultural heritage domain (Grilli and Remondino 2020). Another example comes from Weimann et al., who employ urban scenario point cloud with traditional ML models including k-Nearest Neighbor (KNN) and support vector machines (SVM) using geometric features (Weimann et al. 2013). In addition to these mentioned ML algorithms, 3D neural networks also have exhibited formidable capabilities in point cloud segmentation. PointNet (Qi et al. 2017) and its successor PointNet++ (Qi et al. 2017) have achieved promising results in the field of point cloud classification, especially for man-made objects. Besides, the utilization of the self-attention mechanism in point cloud classification also has yielded remarkable classification outcomes, and the Point Transformer algorithm emerges as an illustrious paradigm of its application (Zhao et al. 2021). Moreover, the graph-based method Dynamic Graph CNN (DGCNN) has been proven to attain the new state of the art performance in roadway target detection for point cloud dataset (Simonovsky and Komodakis 2017).

¹ <https://github.com/OloOcki/awesome-citygml>.

In general, the difference between DL approaches and other machine learning approaches is that features are learned from the training process. This kind of representation learning can provide access to capture undefined structure information of point cloud in training, which is often seen as the reason for the rapid development in 2D and 3D understanding tasks (Griffiths and Boehm 2019).

Increasing number of deep learning methods are proposed to process point cloud data, however, only few studies are devoted to point cloud segmentation at facade-level for LoD3 reconstruction. The complexity of the architecture scene makes it a challenging task to classify buildings' point clouds with rich semantics information, especially for distinguishing facade components that are translucent (e.g., windows) or with low ratio to the overall point cloud (e.g., doors) (Wysocki et al. 2022).

3 Methodology

In this paper, we focus on improving the neural networks' classification performance for point cloud using geometric features. For this purpose, as shown in Fig. 1, both point coordinates and geometric features are prepared for training the selected DL models. With the classifier, performance on unseen scenarios can be validated.

3.1 Geometric Features Extraction

The careful selection for compact and robust geometric features subset is based on previous studies. There have been many studies demonstrating the powerful capacity of geometric features in point cloud classification, which can be viewed as a detailed semantic interpretation of local 3D structures (Grilli and Remondino 2020; Weinmann et al. 2013). According to the study of Grill and Remondino (2020), *planarity*(p), *omnivariance*(o), and *surface variation*(c) are efficient in classification of building point cloud data applying Random Forest. In this paper, except for the mentioned geometric features, we also include principal component analysis (PCA) variables and its eigenvector for each point in classifying the points. The selected covariance geometric features are used for describing local structure's dimension and shape. Here, *planarity* can provide information about the presence of planar 3D structure, while *surface variation* measures the change of curvature for the local structure (Weinmann et al. 2013). Besides, PCA components and its further measurement *omnivariance* obtained by the covariance matrix can summarize the distribution of surrounding points within the local window, and eigenvectors are able to show the direction of such a cluster's distribution. These geometric features exhibit strong responsiveness to the buildings' detailed facade in Grilli's study (Grilli

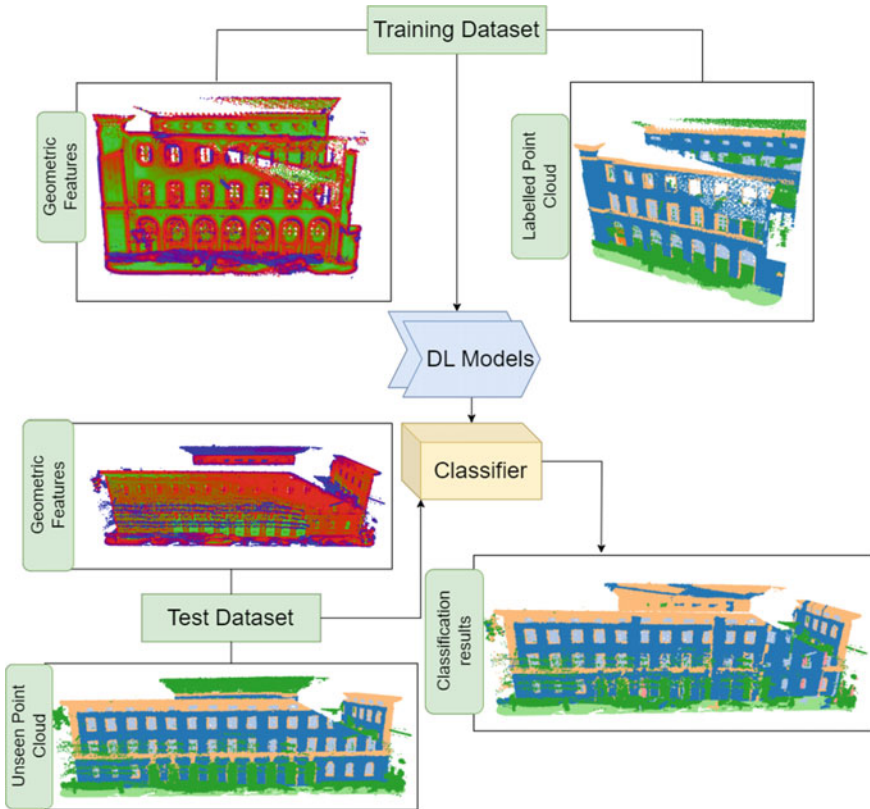


Fig. 1 Point cloud classification with geometric features process for unseen datasets

and Remondino 2020), such as protruding moulding and open windows, showing their potential in enhancement for DL models.

In our method, geometric features are calculated based on surrounding points that locate in a spherical space with a fixed radius. To find such surrounding points for each point, we apply a K-Dimension (KD) tree for nearest neighbor queries. Then, we calculate the respective structure tensor based on Singular Value Decomposition (SVD), which can directly provide information of surrounding points' distribution and structure. In this way, the eigenvalues $\lambda_1, \lambda_2, \lambda_3$, as well as the corresponding eigenvectors e_1, e_2, e_3 are derived from the covariance matrix. In addition, the eigenvalues with $\lambda_1 > \lambda_2 > \lambda_3$, known as PCA components, are used for further measures of other covariance features (Weinmann et al. 2013):

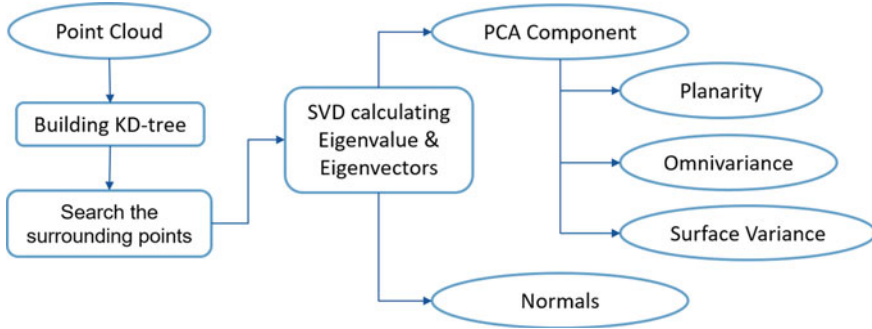


Fig. 2 Workflow of geometric features extraction

$$p = \frac{\lambda_2 - \lambda_1}{\lambda_1}, \tag{1}$$

$$o = (\lambda_1 \times \lambda_2 \times \lambda_3)^{\frac{1}{3}}, \tag{2}$$

$$c = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \tag{3}$$

Besides, the second vector of eigenvectors, which can describe the general direction perpendicular to the curve defined by surrounding points, is also included. Figure 2 represents how the geometric features are calculated.

3.2 Classification

Our method employs two deep learning models PointNet and PointNet++, in which the geometric features are early-fused. In addition, result of Random Forest is also evaluated as reference not only for the deep neural networks but also for the selection of geometric features. Point-based networks are the most frequently investigated methods, among which PointNet is considered as pioneer work. It learns point-wise features separately for each point using multilayer perceptron (MLP) layers and aggregates a global representation for the overall batch of point cloud with a max-pooling layer (Guo et al. 2021). While PointNet++ is an improved version of PointNet, it improves the limitation that PointNet cannot capture local structures in different metric space where points exist (Qi et al. 2017). With a hierarchical structure designed inside the networks, PointNet++ has the capability to extract information regarding local geometric features.

In order to add geometric features into networks, internal structure of the neural networks needs to be updated. For PointNet, features learned in this network are calculated by convolution on coordinate of each point and then gathered globally into a general single layer. Hence, we expand the input dimension of coordinate into

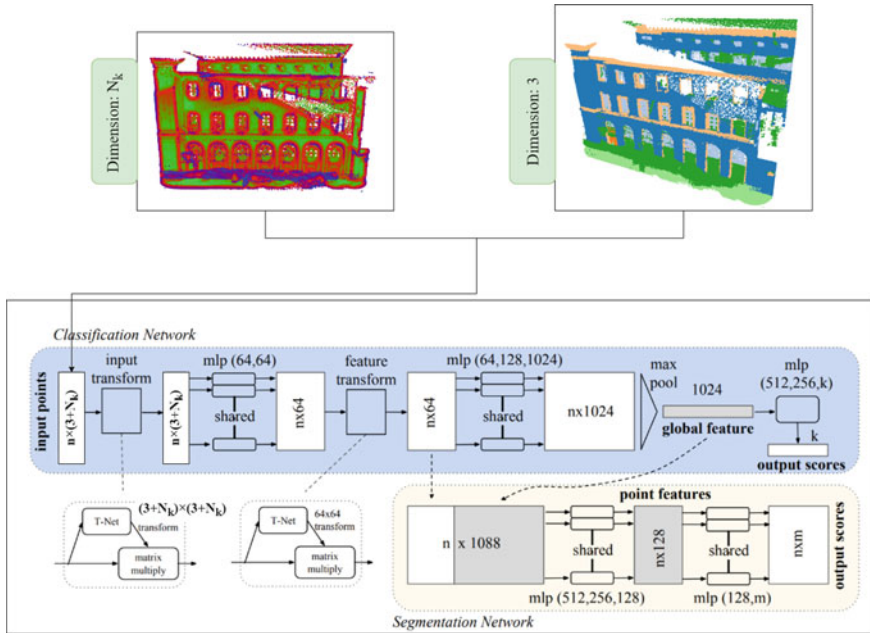


Fig. 3 Proposed early-fusion in PointNet networks. Created based on Qi et al. (2017)

$3 + N_f$, where N_f is the number of pre-computed geometric features. As for PointNet++, features from different scale of metric spaces are collected using hierarchical structure so-called set abstraction layers, consisting of sample layer, grouping layer and PointNet layer. The former two layers generate regions in different scales using point cloud coordinates, while the PointNet layer is used for gathering features from group of points in different regions created in the former two layers. For this reason, we increase the input dimension for PointNet layer to $N \times K \times (d + N_f)$, where N is the number of points, K is the number of regions after grouping, d is the dimension of coordinates, N_f is the dimension of external geometric features. We demonstrate such an early-fusion in Fig. 3.

Apart from experiment with these DL models, performance of the ML model Random Forest is also tested, and importance of all dimensions from geometric features as well as coordinates is determined. With this result as a prior knowledge, different combinations of geometric features are selected and applied for improving the deep learning networks' performance.

4 Experiments

4.1 Datasets

The TUM-FAÇADE dataset was used in this experiment comprising MLS point clouds representing the main campus of the Technical University of Munich, Munich, Germany (Wysocki et al. 2023). We tested the performance on two buildings: No. 4959323 and No. 4959459, and different training datasets were separately applied for the two unseen buildings. Here, unseen building refers to dataset that would not participate in training the DL models but shall be used for testing the performance of the classifiers. For the point cloud of building No. 4959323, four buildings including No. 4906981, No. 4959322, No. 4959460 and No. 4959462 were gathered as the training data. While for the building No. 4959459, the building No. 4959322 was set as the training data. For pre-processing, we downsampled and merged 17 classes into 7 representative facade classes considering the complexity of the classification, unbalanced contribution of points in classes, following the approach of Wysocki et al. (2022). The original manually labelled classes include *wall*, *window*, *door*, *balcony*, *molding*, *deco*(decoration), *column*, *arch*, *stair*, *ground surface*, *terrain*, *roof*, *blinds*, *outer ceiling surface*, *interior* and *other*, after pre-processing they were merged into *wall*, *molding*, *door*, *column*, *arch*, *terrian*, *roof* and *other*. For validating the building No. 4959323, redundant points in training data were decreased significantly after downsampling within a distance of 0.1 m, from 81 million to 4 million, while for building No. 4959459 the sampling distance was set as 0.05 m, number of points was reduced from 5 million to 1 million. Experiment set-up is shown in Fig.4.

4.2 Geometric Features

Using the method presented in Fig. 2, we extracted six covariance geometric features (see Fig. 6) as well as the desired eigenvector. To obtain local geometric features, the radius of spherical space for each point was set as 0.8m. In addition, Random Forest also measured the importance of each feature component, and the result was used for selection of different geometric features combinations. For analysis on different combination of geometric features, we calculated the importance of different features with Random Forest (Fig. 5). As shown in Fig. 5, coordinate components x , y , z were the most powerful factors in Random Forest classification. Features with a score over 0.05 were also considered to be influential in this part of experiment, including *surface variation*, *planarity*, PCA components and the second dimension of second eigenvector. Based on this result, we selected two different feature combinations, one selection included nine kinds of features as input: *planarity*, *surface variation*, *omnivarience*, three PCA components, and three dimensions of second eigenvector, while for the other, *omnivarience*, the first dimension and

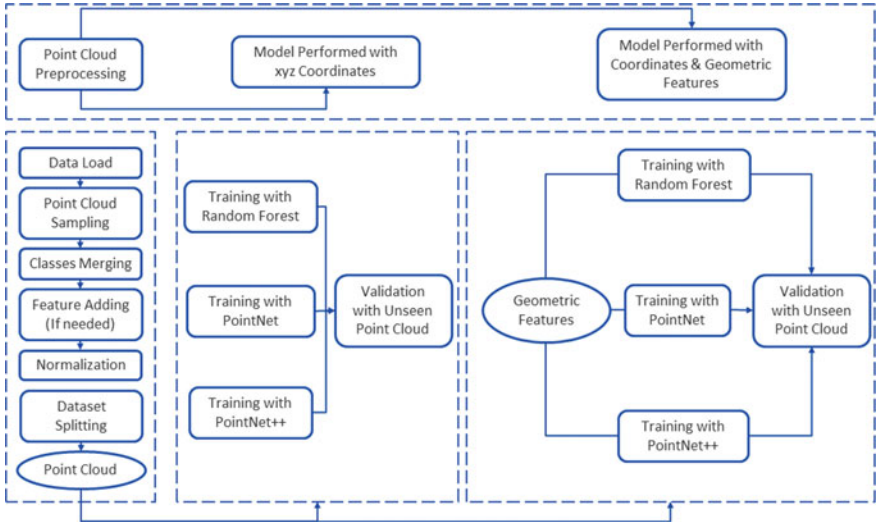


Fig. 4 Experiments set-up

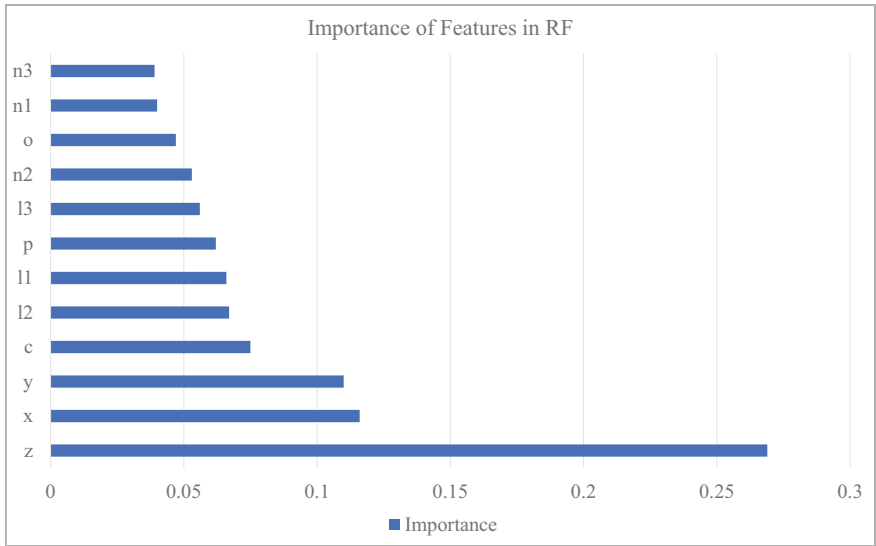


Fig. 5 Feature importance ranking in Random Forest

Table 1 Overall accuracy for different methods

| Datasets | RF (%) | PointNet (%) | PointNet++ (%) |
|-------------|--------|--------------|----------------|
| 59_XYZ | 33.2 | 39.6 | 54.8 |
| 59_XYZ + 9F | 49.7 | 55.8 | 39.2 |
| 59_XYZ + 6F | 49.1 | 52.1 | 62.1 |
| 23_XYZ | 68.4 | 69.1 | 83.1 |
| 23_XYZ + 9F | 66.5 | 78.5 | 84.7 |
| 23_XYZ + 6F | 64.2 | 85.5 | 87.5 |

third dimension of second eigenvector were removed due to their insignificant performance in Random Forest classification, only top six features in terms of importance including *surface variation*, *planarity*, PCA components and the second dimension of second eigenvector were kept for comparison. These two selections of geometric features served as an additional input for DL models PointNet and PointNet++.

4.3 Validation of Improved Models

Validation of improved models for facade-level classification was based on the overall accuracy of different approaches. Except for comparison between with and without geometric features, different combinations of geometric features were also tested in this study. As shown in Table 1, the dataset labelled with 59 refers to performance on the building No. 4959459, and 23 means validation on building No. 4959323. In addition to dataset with only coordinates as input (XYZ), dataset with nine kinds of geometric features (XYZ+9F) and six kinds of geometric features (XYZ+6F) were also evaluated. From statistical results in Table 1, we can see that among all the various combinations, coordinates with six kinds of geometric features (XYZ+6F) as input performs better for PointNet++. Therefore, we took the result of XYZ and XYZ+6f for visualization analysis on influence of geometric features. Confusion matrix in Fig. 7 represents comparison of classification results on different classes using PointNet++ with and without geometric features. Moreover, a visual examination (Figs. 8 and 9) also shows us how PointNet++ performs with or without geometric features on the two unseen buildings. The implementation is available in the repository.²

² <https://github.com/yue-t99/PointNet2-GeometricFeatures-Facade>.

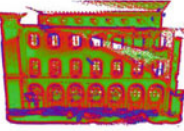

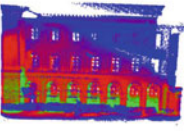



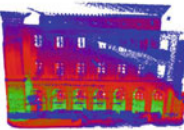

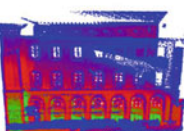



| Geometric features | Training dataset | Test Dataset |
|--------------------|---|---|
| Planarity |  |  |
| Omnivariance |  |  |
| Surface variation |  |  |
| PCA_1 |  |  |
| PCA_2 |  |  |
| PCA_3 |  |  |

Fig. 6 Covariance geometric features used to train the ML and support DL classifiers

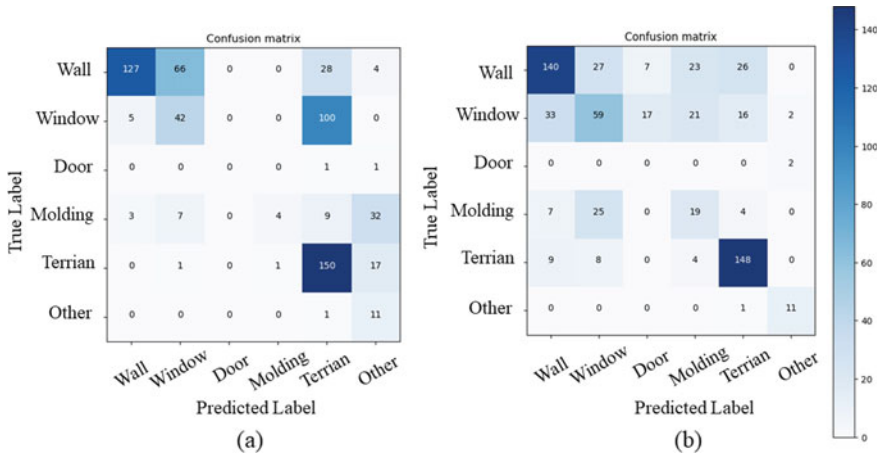


Fig. 7 PointNet++ classification result on building No. 59 with and without geometric features, **a** PointNet++ with xyz only (54.8%), **b** PointNet++ with xyz and 6 geometric features (62.1%)

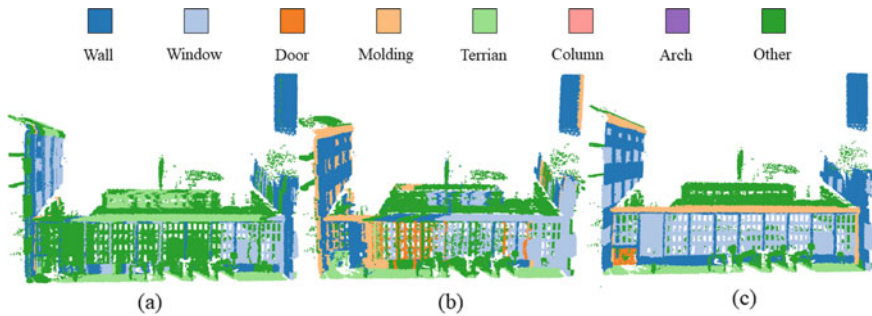


Fig. 8 PointNet++ classification result on building 59 with and without geometric features, **a** PointNet++ with xyz only (54.8%), **b** PointNet++ with xyz and 6 geometric features (62.1%), **c** manually labelled result

5 Discussion

The accuracy results of different models corroborate that adding geometric features do improve the performance of deep neural network. As presented in Table 1, overall accuracy with PointNet tends to be divergent when there is only coordinates as input data and one building set as training dataset, and adding geometric features can help to avoid this problem, since more local structure information is introduced to this DL model. Classification results in PointNet++ represent the same enhancement with six geometric features, respectively from 54.8 to 62.1% for the building No. 59 and 83.1 to 87.5% for building No. 23. Observation from visualized classification results also verifies this statement. Comparing to PointNet++ trained with coordinates

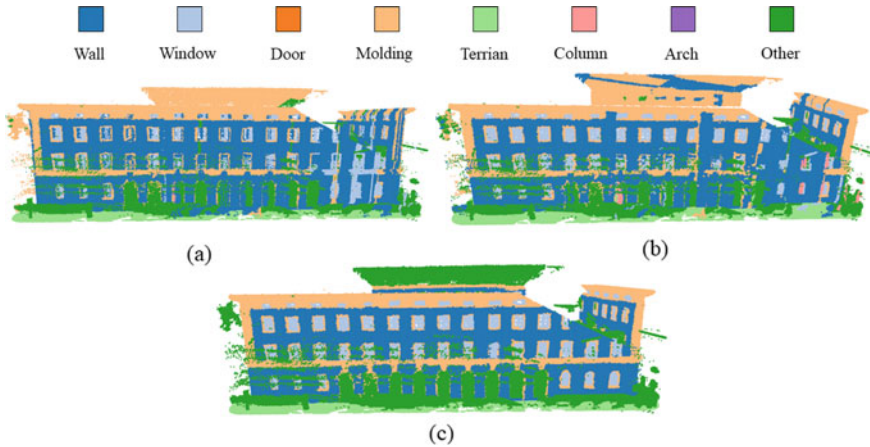


Fig. 9 PointNet++ classification result on building 23 with and without geometric features, **a** PointNet++ with xyz only (83.1%), **b** PointNet++ with xyz and 6 geometric features (87.5%), **c** manually labelled result

only, the features-extended version in Figs. 8 and 9 presents to be more accurate in distinguishing molding and window.

From the accuracy result in Table 1, we also observe that when applying the same dataset for training, among all the approaches tested, PointNet++ with both coordinates and geometric features gives the most accurate result, with an accuracy of 87.5%. Comparing different selection of geometric features, for the deep learning model PointNet++, *surface variation*, *planarity*, PCA components and the second dimension of second eigenvector can improve the performance.

From the accuracy result tested with different combination of geometric features, we can also reasonably infer that selection of geometric features is important, otherwise accuracy might be decreased. In Table 1, when applying combination of nine geometric features for testing on building No. 59, instead of improvement in overall accuracy, the performance of PointNet++ is decreased.

In our method, we choose PointNet and PointNet++ due to their different sensitivity in capturing local structures, and improvement for PointNet is more significant than PointNet++. In previous studies, PointNet shows limits in recognizing local structure. By comparing the results of these two DL models, we can have a deeper insight of the improvement from geometric features. In this experiment, when applying xyz coordinates as input data only, this weakness of PointNet remains noticeable comparing to its' modified version PointNet++.

6 Conclusion

Our work demonstrates the capability of geometric features in enhancing deep learning networks' accuracy for facade-level point cloud classification. The method proposed in this paper can compensate for the inadequacy of deep learning networks in capturing local structure information to a certain extent, providing more sufficient solution for automatic interpretation of 3D buildings.

The validation presents that comparing to the traditional DL and ML method, our early-fused solution reaches a increased accuracy by approximately 10% for the DL model PointNet and 5% for PointNet++, leading us to the conclusion that our method could be a alternative supplement for the point-based neural networks. Such enhancement possesses the potential for further expansion in employing deep learning networks for generalization across large 3D building scenario.

The assessed facades exhibited intricate and diverse measuring conditions, posing a challenge for classifying and testing (Wysocki et al. 2022), leading to different levels of sensitivity in various features. Furthermore, the deep learning method we used in our experiments relies on a point-based approach that requires a computationally intensive neighborhood search mechanism (Guo et al. 2021), which duplicates the pre-computed geometric features, leading to a decrease in computational efficiency. Therefore, employing more different kinds of DL models represents an alternative approach for improvement that can be considered. With more geometric features applied in, optimized combination for different DL models could be utilized for detailed segmentation of point cloud. In future, our work will focus on generalizing such experiments for high LoD reconstruction.

Our work provides an opportunity for a better understanding of the detailed facade in building point cloud with deep neural networks, which could be used for many scene understanding related application, such as visualization for navigation purposes (Biljecki et al. 2015).

Acknowledgements This work was supported by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy within the framework of the IuK Bayern project *MoFa3D—Mobile Erfassung von Fassaden mittels 3D Punktwolken*, Grant No. IUK643/001. Moreover, the work was conducted within the framework of the Leonhard Obermeyer Center at the Technical University of Munich (TUM).

References

- Biljecki F, Stoter J, Ledoux H, Zlatanova S, Çöltekin A (2015) Applications of 3d city models: state of the art review. *ISPRS Int Jo Geo-Inf* 4(4):2842–2889
- Griffiths D, Boehm J (2019) A review on deep learning techniques for 3d sensed data classification. *Remote Sens* 11(12):1499
- Grilli E, Remondino F (2020) Machine learning generalisation across different 3d architectural heritage. *ISPRS Int J Geo-Inf* 9

- Guo Y, Wang H, Hu Q, Liu H, Liu L, Bennamoun M (2021) Deep learning for 3d point clouds: a survey. *IEEE Trans Pattern Anal Mach Intell* 43(12):4338–4364. <https://doi.org/10.1109/TPAMI.2020.3005434>
- Haala N, Kada M (2010) An update on automatic 3d building reconstruction. *ISPRS J Photogram Remote Sens* 65(6):570–580
- Qi CR, Su H, Mo K, Guibas LJ (2017) Pointnet: deep learning on point sets for 3d classification and segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*
- Qi CR, Yi L, Su H, Guibas LJ (2017) Pointnet++: deep hierarchical feature learning on point sets in a metric space. In: *Advances in neural information processing systems*, vol 30
- Simonovsky M, Komodakis N (2017) Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*
- Weinmann M, Jutzi B, Mallet C (2013) Feature relevance assessment for the semantic interpretation of 3d point cloud data. *ISPRS Ann Photogram Remote Sens Spat Inf Sci* 2:313–318
- Wysocki O, Grilli E, Hoegner L, Stilla U (2022) Combining visibility analysis and deep learning for refinement of semantic 3d building models by conflict classification. *ISPRS Ann Photogram Remote Sens Spat Inf Sci* 10:289–296
- Wysocki O, Hoegner L, Stilla U (2023) TUM-FAÇADE: reviewing and enriching point cloud benchmarks for façade segmentation. *arXiv preprint [arXiv:2304.07140](https://arxiv.org/abs/2304.07140)*
- Zhao H, Jiang L, Jia J, Torr PH, Koltun V (2021) Point transformer. In: *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pp 16259–16268

3D Data Acquisition, Analysis and Simulation for Urban Digital Twins

Enriched Semantic 3D Point Clouds: An Alternative to 3D City Models for Digital Twin for Cities?



Imane Jeddoub, Zouhair Ballouch, Rafika Hajji, and Roland Billen

Abstract Digital Twins (DTs) for cities represent a new trend for city planning and management, enhancing three-dimensional modeling and simulation of cities. While progress has been made in this research field, the current scientific literature mainly focuses on the use of semantically segmented point clouds to develop 3D city models for DTs. However, this study discusses a new reflection that argues on directly integrating the results of semantic segmentation to create the skeleton of the DTs and uses enriched semantically segmented point clouds to perform targeted simulations without generating 3D models. The paper discusses to what extent enriched semantic 3D point clouds can replace semantic 3D city models in the DTs scope. Ultimately, this research aims to reduce the cost and complexity of 3D modeling to fit some DTs requirements and address its specific needs. New perspectives are set to tackle the challenges of using semantic 3D point clouds to implement DTs for cities.

Keywords Digital twin · Semantic point cloud · Semantic segmentation · 3D city model · Urban simulations

Imane Jeddoub and Zouhair Ballouch—The authors contributed equally to the work.

This article was selected based on the results of a double-blind review of the full paper.

I. Jeddoub (✉) · Z. Ballouch · R. Billen

Geomatics Unit, UR SPHERES, University of Liège, 4000 Liège, Belgium

e-mail: i.jeddoub@uliege.be

Z. Ballouch

e-mail: Zouhair.Ballouch@student.uliege.be

R. Billen

e-mail: rbillen@uliege.be

Z. Ballouch · R. Hajji

College of Geomatic Sciences and Surveying Engineering, Hassan II Institute of Agronomy and Veterinary Medicine, 10101 Rabat, Morocco

e-mail: r.hajji@iav.ac.ma

1 Introduction

3D city models (3DCMs) and Digital Twins (DTs) for cities have gained significant interest in the urban and geospatial fields (Ellul et al. 2022; Ferré-Bigorra et al. 2022; Mylonas et al. 2021). Both approaches are created based on the combination of various datasets and techniques, i.e., basically 3D reality capture and surveying technologies (Deng et al. 2021b; Ledoux et al. 2021; Lehner and Dorffner 2020). 3D point cloud data from laser scanning has proven its potential as an input layer to create 3D semantic city models and geospatial Digital Twins (Bacher 2022; Beil et al. 2021; Lehner and Dorffner 2020; Lehtola 2022; Lu 2019; Nys et al. 2020; Scalas et al. 2022; Xue et al. 2020). Indeed, point clouds have a simple and easy-to-handle structure that replicates all the physical city features based on point geometries. They are considered a point-based model, where entities are represented as a set of points. Furthermore, their discrete representation, as well as the lack of structure, topology, and connectivity, make them easy to handle, but at the same time, they require costly processing, especially to enrich them with semantic information by applying, for example effective knowledge-based approaches (i.e., Machine Learning (ML) and Deep Learning (DL) approaches) (Döllner 2020; Richter 2018). The current emergence of Artificial Intelligence (AI) is revolutionizing the field of 3D semantic segmentation and yielding highly satisfactory results (Su et al. 2022; Wilk et al. 2022). Nevertheless, the success of newly developed DL approaches relies heavily on the semantic richness of training data.

In the context of implementing semantic 3D city models, most recent works were carried out to optimize the automatic reconstruction of 3D city models. They essentially combine elevation data coming from LiDAR (airborne, terrestrial or mobile) or photogrammetry along with 2D building footprints to generate the city model (Dukai et al. 2019; Ledoux et al. 2021; Nys et al. 2020; Ortega et al. 2021; Pađen 2022; Peters 2022). For instance, 3dfier is an automatic framework that allows the reconstruction of a LoD1.2 model with respect to some specific set of rules (Ledoux et al. 2021). Another related work develops an automatic workflow that segments roof surfaces from point cloud data and generates buildings at LoD2.1 (Nys et al. 2020). Although various methods are in practice to generate accurate semantic 3D city models to perform various spatial and thematic analyses, the city modeling process is still tedious and time consuming (Girindran et al. 2020; Naserentin and Logg 2022).

Following the classic modeling pipeline to automatically generate a 3D city model from LiDAR point clouds, two phases are crucial (Ballouch and Hajji 2021). The first step requires a semantic segmentation of the point cloud to extract the semantic classes that will be used in the second phase (i.e., automatic modeling). However, it will be interesting to investigate if the initial phase in the processing pipeline already has something to offer to DTs for cities instead of going through the modeling process. In fact, an enriched 3D semantic point cloud would help to better manipulate and interpret the 3D data as well as fulfill the DT's needs (Lehtola 2022). First, the initial geometrical accuracy is maintained, which certainly opens new opportunities

to perform simulations directly on point clouds instead of creating surface models. Second, point cloud data should be seen as a preliminary level of DTs as they fulfill their minimum conditions (i.e., replicating all city entities such as buildings, roads, vegetations, terrain, etc.). Finally, it is important to highlight that point clouds can be easily updated over time to reflect changes in the urban environment, whereas updating a 3D city model can be more complicated due to its hierarchical structure.

In this paper, reflections about the use of semantic 3D point cloud data in urban DTs are presented. The questions that are raised here are: how DTs for cities can benefit from enriched semantically 3D point cloud data while meeting the requirements of urban simulations and analysis that goes beyond visual interpretations? And is semantic point cloud data sufficient to fit the DTs analysis instead of a semantic 3D city model?

The paper structure is as follows: Sect. 2 briefly introduces the mainstream uses of semantic point clouds in urban applications. Section 3 discusses to what extent the point cloud may be an alternative to the 3D city model in the context of creating DTs for cities. The same section reflects the requirements of DTs and compares them with the potentialities of semantic point clouds. The advantages and the limitations of both 3D city model and point cloud are highlighted. Then, we conclude the section by giving an overview of the main findings and introducing the future perspectives. Section 4 concludes this work.

2 Mainstream Uses of Point Clouds in Urban Applications

The use of 3D LiDAR point clouds is becoming increasingly relevant in various emerging urban applications, including urban simulations, Virtual and Augmented Reality (VR and AR), Building Information Modeling (BIM), 3D urban mapping, Smart Cities (SCs), Urban Digital Twins (UDTs), and many others. Point clouds can be collected faster than other surveyed data, so enabling regular updates for specific urban applications. They provide a detailed digital representation of urban settings with accurate spatial information and large-scale coverage, especially when acquired through airborne sensors. Besides, the rapid development of LiDAR acquisition techniques has made it possible to create high-precision 3D point cloud representations of urban environments at an affordable cost. These point clouds are capable of depicting objects of varying sizes, providing remarkably lifelike depictions of cities and other landscapes. Moreover, with the increased capacity of GPUs, high density 3D point clouds can be efficiently rendered and displayed instantaneously.

One of the mainstream uses of point clouds in urban applications is for autonomous driving. The recent advancements in DL techniques have enabled the reliable navigation and decision-making required in autonomous driving through the use of dense, geo-referenced, and accurate 3D point cloud data provided by LiDAR. This data provides real-time environment perception and allows for the creation of high-definition maps and urban models, making it an indispensable technology for autonomous vehicles (Li et al. 2021). Another application of point clouds in urban

environments is 3D change detection, which is made possible through the implementation of point clouds (Kharroubi et al. 2022). Recent advancements in computer vision and machine learning have further enhanced the automatic and intelligent detection of changes in urban settings. Moreover, point clouds are suitable for use in virtual and augmented reality applications due to their ability to provide a more immersive way of perceiving 3D digital objects (Alexiou et al. 2017). Furthermore, 3D point cloud data has been used as reference data for city modeling (Badenko et al. 2019; Huang et al. 2022; Nurunnabi et al. 2022; Nys et al. 2020; Wang et al. 2019; Yan et al. 2019). For example, 3D BAG¹ has multiple Level of Details (LoDs) of 3D buildings as an up-to-date data set for the whole city of the Netherlands. The datasets are generated based on the building footprints from the BAG and the height data from AHN acquired by airborne laser scanning (ALS) (Dukai et al. 2019, 2021; Dukai 2020; León-Sánchez et al. 2021). Additionally, several cities around the world have acquired 3D point cloud data to model their buildings. For instance, Helsinki used classified ALS point cloud data to determine the elevation position and the roof shapes of the buildings (Hämäläinen 2021). The city has also used classified point clouds to map, update, and maintain the City Information Model. Another deployment of point cloud data is in the context of creating UDTs for city-state Singapore called “Virtual Singapore”. The authors proposed an automatic tree modeling framework at multiple LoDs combining airborne and mobile LiDAR scanning datasets with different remote sensing data to address the limitations of each acquisition technique (Gobeawan et al. 2018). In addition, to create a CityGML model for the city, 3D building models were created using aerial images and airborne point cloud data (Soon and Khoo 2017).

In recent years, considering BIM models as one of the input layers to implement an UDTs (Deng et al. 2021a; Lehtola 2022; Stojanovic et al. 2018), many approaches in the Architecture Engineering Construction (AEC) field have been discussed to automate and support the process of creating a BIM model from a point cloud for several applications known as the Scan-to-BIM workflow (Hellmuth 2022). The use of Scan-to-BIM practices has led to highly accurate data and faster project delivery in the construction industry (Perez-Perez et al. 2021; Soilán et al. 2020). To further improve this process, the industry and academia are exploring ways to automate the segmentation of point clouds into individual building components and model them using continuous surfaces of solid geometries (Perez-Perez et al. 2021). However, the process is still facing challenges that are partially solved, and the approaches still require some manual modeling and are based on proprietary modeling software (Deng et al. 2021a). The LiDAR point clouds were used also to generate derived products such as DTM, DSM or mesh models which will be used in turn for 3D city modelling and visualizations purposes (Biljecki et al. 2015; Guth et al. 2021). Thus, several improvements were made to effectively render massive point cloud data through the web allowing seamless data access (Oosterom 2015; Richter 2018). In addition, some tools are in common use that directly work with point cloud data, bypassing the complex and expensive approaches of deriving 3D city models from point cloud. More and more point cloud data are available and relevant. However,

¹ <https://3dbag.nl/en/viewer>.

working with point cloud data in the 3D city modeling and UDTs scopes remain challenging. Although the new improvement of CityGML 3.0 allows the use of point cloud data to mimic the city objects, the semantic information is not handled, and some approaches are proposed to extend the semantic capabilities of 3D point cloud data (Kutzner et al. 2020).

The raw point clouds are widely used, their usefulness can be limited due to their unstructured nature. Semantic point clouds, on the other hand, provide a semantic label associated with each point, which allows for a better understanding of the scanned urban scene and opens new possibilities for a range of urban applications (Ballouch et al. 2022; Ballouch and Barramou 2022). Semantic point cloud plays a crucial role in creating 3D urban models that form the primary basis of DTs. It offers an accurate basis for the creation of semantic models in different formats such as CityGML and its encoding CityJSON, or (IFC) Industry Foundation Classes (Beil et al. 2021). The use of semantic point clouds enables precise extraction of urban objects, which is an essential step in the 3D modeling process of cities. With a semantic point cloud, automated object modeling is simplified. For instance, buildings can be extracted and matched with the corresponding building footprints to generate the corresponding 3D models (Kong et al. 2022). Furthermore, an enriched semantic point cloud enhances the enrichment of 3D models by providing richer and more detailed information about the urban environment. Besides, the semantic richness of semantic point clouds can be useful to quickly identify objects relevant to a specific task or application in the context of urban applications. Recent advances in 3D semantic segmentation allow for the extraction of maximum semantic information that comprises the urban environment, such as vegetation, roads, railways, etc. This semantic information can be used to create the basis for the DT of a given city, i.e., the geometric model onto which other data can be integrated. In addition, it is important to regularly update the digital model to accurately reflect real-time changes in the urban environment and keep urban applications up to date. Besides, the use of semantic point clouds is an interesting source of data for training DL models for semantic segmentation tasks. By using semantically segmented point clouds, precise datasets can be formed to achieve high performing pretrained models in different urban contexts to meet the requirements of plenty of urban applications. Additionally, semantic point clouds can be used to extract building footprints, which is crucial for the 3D modeling of buildings. Similarly, airborne semantic point clouds can be used to extract roofs, enabling the creation of accurate models of building roofs that can be used to meet the specific requirements of urban applications. In addition, incorporating structured knowledge and semantics into 3D point cloud (beyond semantic segmentation) are beneficial in meeting the needs of urban applications (Poux and Billen 2019).

To conclude, 3D LiDAR point clouds in urban applications have grown increasingly important due to their rapid and cost-effective means of gathering accurate spatial information of urban settings. These point clouds capture the real-time state of the city for almost all spatial entities at various scales, depending on the laser scanning survey methods employed, whether airborne, terrestrial, or mobile. Enriched 3D semantic point clouds play a crucial role in creating 3D urban models, automating

object modeling, extracting maximum semantic information, and updating urban application models.

Up to date, there are no initiatives that rely on enriched semantic 3D point cloud data to meet the DT requirements since they are primarily used as input data for urban modeling. Thus, deployed in the generation of semantic 3D city models. Indeed, using semantic point clouds as a fundamental input layer to build DTs deserves consideration. To better understand this, we must first identify the requirements of DTs for cities, the performance of semantic point clouds to address the limitations of 3D city models, and the possibilities of studying semantic point clouds as an alternative to perform simulations directly on point clouds without going to 3D models.

3 Is a Semantic 3D Point Cloud an Alternative to 3D City Model for DT Applications?

To address this key research question, we will split it into two sub-questions. Firstly, does the point cloud meet the DTs' requirements? (Sect. 3.1); Secondly, is the point cloud a good alternative to 3D city models? (Sect. 3.2). We finally give some research guidelines related to extending the use of semantic point clouds in DTs for cities (Sect. 3.3).

3.1 Semantic Point Cloud: An Input Layer to DTs for Cities

The presence discourse in the urban and geospatial context is predominantly about the relevance and the potentiality of considering semantic 3D city models as an input layer to create DTs for cities (Alva et al. 2022; Dimitrov and Petrova-Antonova 2021; Ketzler et al. 2020; Würstle et al. 2022). However, it is worth considering the potentialities and advantages of semantic point clouds to serve DTs needs as a fundamental input layer without going through the 3D city modeling process.

To tackle this research question, it is interesting to identify the requirements of DTs for cities. Indeed, DTs for cities are conceptualized as a risk-free, living virtual ecosystem that mimics all the city elements to generate knowledge, assist urban decision-making through the city lifecycle, and provide outcomes at the city level (Hristov et al. 2022; Nguyen and Kolbe 2022; Würstle et al. 2022). Furthermore, from technical perspective, most of the research led to a tacit agreement on what constitutes a DT for cities in the geospatial domain and the Smart Cities initiatives previously announced by Stoter et al. (2021). Thus, DTs for cities are based on (1) 3D city models enriched with geometrical and semantic information, (2) often incorporate heterogeneous data namely coupled with historical and sensor data in near or real time (at an appropriate rate of synchronization), thus enabling (3) a link (e.g., data flow

between the real counterpart and the virtual twin and vice versa), (4) allowing updates and analysis through a set of simulations, predictions, and visualization tools, and (5) providing an integrated view of the multiple datasets and models through their life cycle, enabling to manage and adapt cities' current and future states.

If we intend to unpack the DT definition, we will first start from the assumption that the DT for cities is a digital realistic city replica that incorporates all its city features. Thus, we can clearly validate this characteristic since a point cloud by nature is a high geometrically 3D representation of urban environments such as cities and other landscapes. However, back to definition, a DT must have semantic and geometrical information. This is completely accurate from geometrical dimension of a point cloud but is not applicable for semantics. In this regard, various approaches are proposed to enrich the point cloud and extend its semantic capabilities, whether through 3D semantic segmentation (Hu et al. 2021), or a conceptual data model called "Smart Point Cloud Infrastructure" (Poux 2019), or data integration (GIS data, 3D city models) (Beil et al. 2021).

Although possibilities exist to tackle the lack of semantics in point cloud data, the enrichment of such data remains critical and challenging. Indeed, the current advancement in the scope of building DTs for cities is more focused on data integration approaches, including the association and integration of both point cloud data and semantic 3D city models using for example the new "PointCloud" module of CityGML 3.0 (Beil et al. 2021). This module provides a new concept to bridge the gap between the geometrically detailed point cloud data and the enriched 3D semantic model. The integration of both datasets intuitively assigns sets of points to the corresponding objects. The existing approach in CityGML 3.0 provides an alternative for extending point cloud data to cover more semantic information beyond classification using various methods. Thus, integration of point cloud data with different data sets from GIS, BIM, and 3D city models helps to overcome the limitations of each approach and meet the DT requirements.

At the same time, a widespread algorithm and approaches have emerged to extract 3D objects automatically and effectively by semantically segmenting LiDAR point clouds using supervised learning methods, including Machine Learning-based segmentation, as well as Deep Learning-based segmentation such as multi-view-based methods, voxel-based methods, and direct methods that consume point clouds directly. Recent advances in semantic segmentation allow the extraction of the main urban features, such as buildings, vegetation, roads, railways (Zhou et al. 2023), and many more that are relevant for DT's applications (Döllner 2020; Lehtola 2022; Masoumi et al. 2023).

In another hand, 3D semantic segmentation is relevant to update DTs for cities and track the changes at city-scale. That said that 3D semantic point cloud data enable the identification of the changes as they appear in the real world and updating corresponding information. For example, point cloud data allows to have a realistic and big picture of the status of an urban object under construction, especially if the current project does not have the necessary elements to generate a 3D model (i.e., lack of definitive footprint that is mandatory to generate accurate model). This says that the semantic point cloud can help in urban planning and management which is

one of the common use cases of DTs for cities. In addition, the advantage of enriched semantic point cloud data is that almost all urban classes are extracted (i.e., static, and dynamic objects) and for specific applications, classes that are required or need to be updated are simply retained. Nevertheless, the classes that are not crucial are neglected. It is worth mentioning as well that for different use cases, different classes are deployed, which is completely in line with the DT's requirements that replicate all the city objects as one snapshot, and for each use case, the data will be derived. Hence, semantic 3D point cloud enables us to precisely define the urban classes, thus augmenting the performance of the semantic extendibility, improving modeling capabilities, giving new interpretability of the data from different perspectives, and opening new doors for various simulations and urban analysis.

Turning to one of the promising characteristics of a DT (i.e., the simulation feature), yet the available processes and simulation tools that involve the direct use of 3D point clouds are still limited. Few studies are conducted to explore the potential of this type of data. For example, the authors of Peters et al. (2015) introduced a new approach based on the medial axis transform to performing visibility analysis. The approach could be used for any typical airborne LiDAR data, which gives more realistic results and effectively handles the missing parts of the point cloud (e.g., walls and roofs). Furthermore, performing visibility analysis is more insightful when working with point cloud data, as vegetation is considered. Another study case on an urban scale performs the visibility analysis for both surface model and point cloud data and puts them together for in-depth analysis according to their efficiency and accuracy. To ensure intervisibility between the reference points (i.e., the observer and target points), the authors of Zhang et al. (2017) generate cubes for each point to block the sight lines. The study concludes that consistent input data (i.e., dense and classified point clouds) will certainly improve the findings.

On the other hand, solar radiation is a relevant use case in 3D urban modeling. Historically, solar irradiance was measured using DSM. However, 3D city models gained a significant amount of interest to improve the sun exposition estimations. In addition, the authors have developed a tool that uses point cloud data to model illumination and solar radiation (Pružinec and Ďuračiová 2022). The algorithm is based on voxels and has shown its capabilities for green areas as well as urban environments.

Figure 1 depicts an illustrative example from our research works, demonstrating the simulation of solar radiation performed on semantic point clouds. The point cloud data utilized in the study was acquired in the Flanders region of Belgium. The pre-trained RandLA-Net model (Hu et al. 2020) on the Semantic3d dataset (Hackel et al. 2017) was used for semantic segmentation of point clouds. The relevant semantic classes that have the potential to impact solar radiation were extracted, including high vegetation, low vegetation, buildings, and scanning artifacts. To perform the simulation, the "pcsr" function proposed by Pružinec and Ďuračiová (2022) was used. The source code for this tool was adapted from its publicly available version on GitHub as an open-access resource (<https://github.com/hblyp/pcsr>, accessed on August 2, 2022).

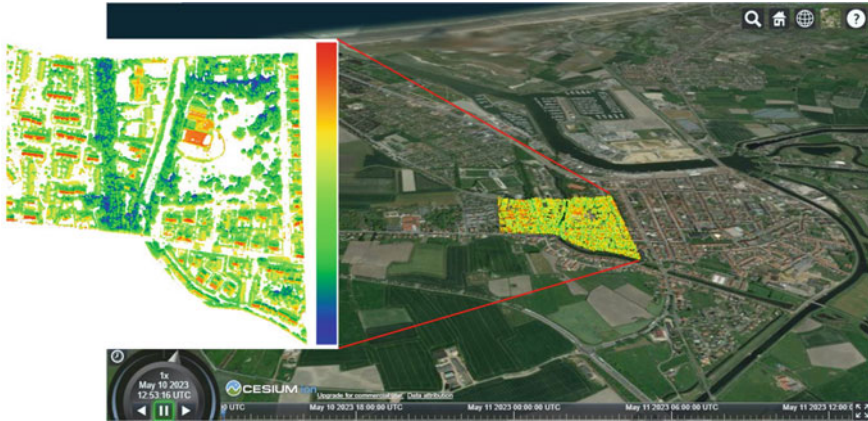


Fig. 1 Example of solar radiation performed on semantic point cloud

A further characteristic of DTs that is undoubtedly satisfied, is the visualization and the interactivity aspects. Point cloud data is supported through various visualization tools (i.e., web applications and game engines platforms). Additionally, point clouds are considered as a form of natural communication used as an input data to enhance immersivity and interactivity of Virtual Reality or Augmented Reality experiences. Moreover, for enhancing visualization, most of DT’s initiatives tend to foster the ability to process, store, handle, and disseminate massive point clouds through the web, namely using the CesiumJS WebGL virtual globe. For instance, the Digital Twin of the City of Zurich sets a research agenda where further developments of the DTs for city are required namely, how to benefit from the derived mobile mapping point cloud data to improve the façades of the buildings as well as how to incorporate vegetation acquired from point cloud into the DTs. It is worth pointing that some visualization applications do not demand rich semantics, while others need specific attributes to perform simulations (Schrotter and Hürzeler 2020).

While the state of the art is well developed regarding the applications of 3D city models, some urban applications do not necessarily need a semantic 3D city model. Hence, enriched 3D semantic point cloud will certainly give new opportunities to perform some sophisticated analysis for DTs instead of creating surface models.

3.2 Semantic Point Cloud and Semantic 3D City Models: Advantages and Limitations

While it is out of the scope of this article to compare 3D semantic model-based DTs and enriched semantic point cloud-based DTs, we will nevertheless highlight certain advantages and limitations of both semantic 3D city model and semantic 3D point clouds.

3D city models nowadays exhibit significant differences due to various factors including data acquisition, processing, storage, dissemination, use, and maintenance, as well as technical, socio-economic, political, and cultural variations. Consequently, it has become challenging to identify best practices, assess the quality of 3D city models, foster their appropriateness for specific use cases, and integrate effectively diverse datasets. Moreover, comparing multiple datasets present some difficulties, creating ambiguity in selecting the most suitable one. These concerns have implications for urban DTs, which rely on 3D city models as a key component (Lei et al. 2022). Despite the availability of advanced 3D representation techniques and methods for creating 3D city models (Toth and Józków 2016), significant challenges remain in achieving accurate and interactive 3D modeling of the urban environment. It is not just a matter of representing the environment in 3D, but also ensuring that the model is closer to the real world by attempting to represent as many urban objects of the physical world as possible without being restricted to a specific feature (i.e., buildings as they represent the identity of the city).

Research has identified several problems associated with 3D modeling (Stoter 2020), including limited data collection capabilities (Ledoux et al. 2021), reduced levels of automation (Park and Guldmann 2019), the lack of established modelling standards and rules (Eriksson et al. 2020), and limited applications for visualizing city models (Liamis and Mimis 2022). There are three types of modeling techniques: geometric modeling, mesh modeling, and hybrid modeling. Geometric modeling uses simple geometric primitives (planes, cylinders, lines, etc.) to represent objects, which reduces the volume of generated data and allows for semantic data to be embedded in the model. However, this method is dependent on the algorithms used and the resulting representation may lack fine details. Mesh modeling is useful for representing fine surface details, but the generated data remains voluminous, making interpretation and manipulation laborious for the user. Furthermore, 3D mesh models have limited analytic capabilities. However, few studies are conducted to improve the usability and applicability of mesh models by integrating semantic 3D city models with 3D mesh models (Willenborg et al. 2018). Another related work enhances semantic segmentation of urban mesh using a hybrid model and a feature-based approach for semantic mesh segmentation in an urban scenario using real-world training data (Tutzauer et al. 2019). While meshes alone do not inherently allow semantic data to be embedded in the model since no shape or element recognition is performed. Semantic information could be introduced by modifying them or storing them using specialized data formats such as CityJSON that support semantics.

3D city modeling has different challenges that limit their full automation and usage. Firstly, there is an inconsistency between models generated using heterogeneous dataset, reconstruction methods, and software, which affects geometry, appearance, and semantics. Standardization is the second challenge. Up to date, there are no common standards that are established to handle DTs for cities from a technical point of view. However, we should take advantage of the existing standards by enabling convergence between them in a meaningful way with respect to the discrepancies (different geometries, semantics, structures and various spatial scales). Data quality is a major obstacle to create 3D city models, with many existing models containing

errors that prevent their use in other software and applications. Data interoperability involves converting 3D models from one format to another. Language barriers may hinder understanding and interoperability. Indeed, public administrations often do not provide integrated and standardized 3D city models, making further analysis difficult. In addition, datasets may be managed in different standards and have different sets of information, making them unqualified for particular use cases. There is a lack of means to characterize data and their fit for purpose. In addition to the challenge posed by the heterogeneous nature of 3D city models in terms of making comparisons, another issue arises from the data integration approaches (Lei et al. 2022). Data maintenance/governance is also a challenge, with governmental organizations lacking strategies for updating and maintaining different versions of the data. Lastly, implementing 3D data in the real world requires more precise definitions of specifications, validation mechanisms, clear semantics to address knowledge and skills gaps and integration of public and private sector models (Stoter et al. 2021).

It is well known that in the scientific literature and in practice, the point cloud is considered a primary resource for reconstructing a semantic 3D city model. Indeed, 3D city models are by definition, a simplification of the real world (i.e., an abstraction at a certain LoD). With this in mind, 3D city models do not aim to represent all the features of the real world in the same detail as point cloud data. Thus, point cloud allows to avoid the abstraction needed for 3D city models, and objects such as trees are correctly rendered instead of being generalized according to city modeling standards. Furthermore, for a given point cloud, different 3D semantic model could be generated according to the use case, the standards and the quality of the acquired data. Moreover, recent advances in semantic segmentation and point cloud processing have made significant progress toward optimizing the algorithms and approaches.

Another particularity of point cloud data is the lack of a specific standard to generate and process them. However, there may be variations in format and representation (e.g., voxels). In contrast, for 3D city models, there are many standards deployed to generate a semantically rich 3D model, namely CityGML and its JSON encoding, CityJSON. These standards are recognized as the foundation of DTs for cities. The existence of a range city modeling standards raises data interoperability issues. This does not mean that the standardization efforts are irrelevant, but this standard heterogeneity makes data integration challenging especially in the context of creating DTs in practice. This is also justified by having several 3D city models for the same urban scale from different stakeholders, but there is usually a single national LiDAR acquisition. Of course, for some cities, we may find more than one acquisition, however they are captured at different timescale having overlapped regions. It is also sometimes collected to fill some missing information for large scale areas (i.e., urban land expansion). This extension of point cloud data to the temporal scale serves in the context of DTs given a 4D point cloud. However, this point cloud requires a high storage infrastructure, and detecting the changes is tricky since point-to-point corresponding is problematic.

Regarding the point cloud, another challenge that hinders its full potentials is the lack of topology, which can make simulating object behavior challenging.

For instance, connections between different urban objects are difficult to represent without topology, which is why 3D models with a surface model are preferred for such representations, which are relevant for simulations namely for Computational Fluid Dynamics (CFD). Furthermore, 3D city models offer the possibilities to store attributes for objects (e.g., buildings) but also for surfaces, to build hierarchy (Building + Building Part) and to store the type of surfaces (namely used for energy modelling). It is also worth mentioning that 3D city models are significantly taking less space (compared to a raw point cloud, which is more than 10pts/m² nowadays).

To conclude, semantic point clouds and semantic 3D city models are both great inputs to build DTs for cities. Both bring new opportunities but still have some weaknesses. However, all DT initiatives invest in hybrid models, enabling them to bridge the gap between different approaches and compensate for the limitations of the others.

3.3 Semantic Point Cloud: A New Research Field for DTs

The potential benefits of implementing this new research path include reducing the cost of modeling, computation time, to take advantage of the semantic richness of the semantic point cloud since frequently we make large-scale acquisitions and heavy processing operations to end up exploiting only the buildings class in 3D modeling without other details of the urban environment (i.e., vegetation, roads). This approach also avoids the complexities of 3D modeling, particularly for other urban objects than buildings like transportation infrastructure and vegetation. It's also advantageous for updating urban DTs and conducting specific simulations that require accurate and detailed information about the urban environment. The proposed reflection challenges the frequently used approach of relying solely on 3D modeling for DTs applications and suggests that semantic point clouds can be a viable alternative, particularly for addressing the limitations of 3D models and meeting the needs of DTs in an easy and effective way. However, it is important to note that while semantic 3D point clouds may be a useful input layer for some DT applications, they may not be a complete replacement for 3D city models in all cases. The choice between using semantic 3D point clouds or 3D city models as an input layer for DT applications will depend on the specific application purposes, the available resources, and the required level of accuracy and detail.

Further research is needed to explore the potential of semantic point clouds and develop new approaches for integrating them into DTs applications.

As a first step of our reflection, we investigated the feasibility of some simulations that can be performed directly on point clouds. In the next steps, we will evaluate and validate our approach by comparing it with 3D city models that utilize the same data, in order to affirm its effectiveness and accuracy.

This work also suggests some perspectives to meet the requirements of DTs:

- Future research should focus on exploring the potential of semantic point clouds and developing integration methods for their use in DTs applications.
- It is important to consider the specific requirements of the application, available resources, and desired level of accuracy and detail when choosing between semantic 3D point clouds and 3D city models as an input layer for DT applications.
- Establishing standards for DTs could bring several benefits. Firstly, it would enable increased interoperability among different systems applying this concept, thereby facilitating collaboration and data exchange. Additionally, clearly defined standards could help ensure the security and protection of data, as well as the quality of the created models.
- Defining a preliminary LoD for semantic point clouds can help ensure the quality and usability of data for specific DTs applications.
- Developing new approaches and algorithms that enable the direct simulation of urban environments using semantically rich point clouds instead of generating 3D model, more precisely for sophisticated simulations such as computational fluid dynamics.
- Studying change detection and updating of DTs with semantically rich point clouds.

4 . Conclusions

In this paper, we have proposed a research reflection on the use of semantic 3D point clouds as an alternative to 3D city models for DTs needs. We have introduced the limitations and performance of both 3D city models and semantic point clouds. Furthermore, we explain how semantic point clouds can overcome the limitations of 3D city models to create a DTs. We then presented the initial guidelines of the suggested reflection, which aims to answer the research question of whether a point cloud can meet the requirements of DTs by going beyond considering a semantic point cloud as input for modeling and performing simulations directly on it without resorting to 3D modeling. This research direction should be further explored to match point clouds to DTs' requirements and extend their urban applications. In short, semantic 3D point clouds appear as potential data that goes beyond the current deployment of creating 3D city models, which puts them at the forefront of new needs in urban simulations.

References

Alexiou E et al (2017) Towards subjective quality assessment of point cloud imaging in augmented reality. In: 2017 IEEE 19th international workshop on multimedia signal processing (MMSP), pp 1–6. <https://doi.org/10.1109/MMSP.2017.8122237>

- Alva P et al (2022) Use cases for district-scale urban digital twins. *Int Arch Photogramm Remote Sens Spatial Inf Sci XLVIII-4/W4-2022*:5–12. <https://doi.org/10.5194/isprs-archives-XLVIII-4-W4-2022-5-2022>.
- Bacher U (2022) Hybrid aerial sensor data as basis for a geospatial digital twin. *Int Arch Photogramm Remote Sens Spatial Inf Sci XLIII-B4-2022*:653–659. <https://doi.org/10.5194/isprs-archives-XLIII-B4-2022-653-2022>
- Badenko V et al (2020) Airborne LIDAR data processing for smart city modelling. In: Anatolijs B et al (eds) *Proceedings of ECEE 2019*. Springer International Publishing, Cham, pp 245–252. https://doi.org/10.1007/978-3-030-42351-3_21
- Ballouch Z et al (2022) Toward a deep learning approach for automatic semantic segmentation of 3D lidar point clouds in urban areas. In: Barramou F et al (eds) *Geospatial intelligence: applications and future trends*. Springer International Publishing, Cham, pp 67–77
- Ballouch Z, Hajji R (2021) Semantic segmentation of airborne LiDAR data for the development of an urban 3D model. In: *Building information modeling for a smart and sustainable urban space*. John Wiley & Sons, Ltd., pp 113–130. <https://doi.org/10.1002/9781119885474.ch7>
- Ballouch Z et al (2022) A prior level fusion approach for the semantic segmentation of 3D point clouds using deep learning. *Remote Sens* 14(14):3415. <https://doi.org/10.3390/rs14143415>
- Beil C et al (2021) Integration of 3D point clouds with semantic 3D city models—providing semantic information beyond classification. *ISPRS Ann Photogramm Remote Sens Spatial Inf Sci VIII-4/W2-2021*:105–112. <https://doi.org/10.5194/isprs-annals-VIII-4-W2-2021-105-2021>
- Biljecki F et al (2015) Applications of 3D city models: state of the art review. *IJGI* 4(4):2842–2889. <https://doi.org/10.3390/ijgi4042842>
- Deng T et al (2021b) A systematic review of a digital twin city: a new pattern of urban governance toward smart cities. *J Manage Sci Eng* 6(2):125–134. <https://doi.org/10.1016/j.jmse.2021.03.003>
- Deng M et al (2021a) From BIM to digital twins: a systematic review of the evolution of intelligent building representations in the AEC-FM industry. *J Inf Technol Constr* 26:58–83. <https://doi.org/10.36680/j.itcon.2021.005>
- Dimitrov H, Petrova-Antonova D (2021) 3D city model as a first step towards digital twin of Sofia City. Presented at the international archives of the photogrammetry, remote sensing and spatial information sciences—ISPRS archives. <https://doi.org/10.5194/isprs-archives-XLIII-B4-2021-23-2021>
- Döllner J (2020) Geospatial artificial intelligence: potentials of machine learning for 3D point clouds and geospatial digital twins. *PFG* 88(1):15–24. <https://doi.org/10.1007/s41064-020-00102-3>
- Dukai B et al (2020) Generating, storing, updating and disseminating a countrywide 3D model. *Int Arch Photogramm Remote Sens Spat Inf Sci XLIV-4/W1-2020*:27–32. <https://doi.org/10.5194/isprs-archives-XLIV-4-W1-2020-27-2020>
- Dukai B et al (2019) A multi-height lod1 model of all buildings in the Netherlands. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci IV-4/W8*:51–57. <https://doi.org/10.5194/isprs-annals-IV-4-W8-51-2019>
- Dukai B et al (2021) Quality assessment of a nationwide data set containing automatically reconstructed 3D building models. *Int Arch Photogram Remote Sens Spat Inf Sci XLVI-4-W4-2021*:17–24. <https://doi.org/10.5194/isprs-archives-XLVI-4-W4-2021-17-2021>
- Ellul C et al (2022) Location-enabled digital twins—understanding the role of NMCAS in a European context. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci X-4/W2-2022*:53–60. <https://doi.org/10.5194/isprs-annals-X-4-W2-2022-53-2022>
- Eriksson H et al (2020) Requirements, development, and evaluation of a national building standard—a Swedish case study. *ISPRS Int J Geo-Inf* 9(2):78. <https://doi.org/10.3390/ijgi9020078>
- Ferré-Bigorra J et al (2022) The adoption of urban digital twins. *Cities* 131:103905. <https://doi.org/10.1016/j.cities.2022.103905>
- Girindran R et al (2020) On the reliable generation of 3D city models from open data. *Urb Sci* 4(4):47. <https://doi.org/10.3390/urbansci4040047>

- Gobeawan L et al (2018) Modeling trees for virtual Singapore: from data acquisition to CITYGML models. *ISPRS Int Arch Photogramm Remote Sens Spat Inf Sci XLII-4/W10:55–62*. <https://doi.org/10.5194/isprs-archives-XLII-4-W10-55-2018>
- Guth PL et al (2021) Digital elevation models: terminology and definitions. *Remote Sens* 13(18):3581. <https://doi.org/10.3390/rs13183581>
- Hackel T et al (2017) Semantic3D.net: a new large-scale point cloud classification benchmark. [arXiv:1704.03847](https://arxiv.org/abs/1704.03847) [cs]
- Hämäläinen M (2021) Urban development with dynamic digital twins in Helsinki city. *IET Smart Cities* 3(4):201–210. <https://doi.org/10.1049/smc2.12015>
- Hellmuth R (2022) Update approaches and methods for digital building models—literature review. *ITcon* 27:191–222. <https://doi.org/10.36680/j.itcon.2022.010>
- Hristov PO et al (2022) Enabling city digital twins through urban living labs. *Int Arch Photogramm Remote Sens Spat Inf Sci XLIII-B1-2022:151–156*. <https://doi.org/10.5194/isprs-archives-XLIII-B1-2022-151-2022>
- Hu Q et al (2020) RandLA-Net: efficient semantic segmentation of large-scale point clouds. In: 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE, Seattle, WA, USA, pp 11105–11114. <https://doi.org/10.1109/CVPR42600.2020.01112>
- Hu Q et al (2021) Towards semantic segmentation of urban-scale 3D point clouds: a dataset, benchmarks and challenges. In: 2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE, Nashville, TN, USA, pp 4975–4985. <https://doi.org/10.1109/CVPR46437.2021.00494>
- Huang J et al (2022) City3D: large-scale building reconstruction from airborne LiDAR Point clouds. *Remote Sens* 14:9. <https://doi.org/10.3390/rs14092254>
- Ketzler B et al (2020) Digital twins for cities: a state of the art review. *Built Environ* 46:547–573. <https://doi.org/10.2148/benv.46.4.547>
- Kharroubi A et al (2022) Three dimensional change detection using point clouds: a review. *Geomatics* 2(4):457–485. <https://doi.org/10.3390/geomatics2040025>
- Kong G et al (2022) Automatic building outline extraction from ALS point cloud data using generative adversarial network. *Geocarto Int* 37(27):15964–15981. <https://doi.org/10.1080/10106049.2022.2102246>
- Kutzner T et al (2020) CityGML 3.0: new functions open up new applications. *PFG* 88(1):43–61. <https://doi.org/10.1007/s41064-020-00095-z>
- Ledoux H et al (2021) 3dfier: automatic reconstruction of 3D city models. *J Open Source Softw* 6(57):2866. <https://doi.org/10.21105/joss.02866>
- Lehner H, Dorffner L (2020) Digital geoTwin Vienna: towards a digital twin city as Geodata hub. *PFG J Photogramm Remote Sens Geoinf Sci* 88(1):63
- Lehtola VV et al (2022) Digital twin of a city: review of technology serving city needs. *Int J Appl Earth Observ Geoinf* 102915. <https://doi.org/10.1016/j.jag.2022.102915>
- Lei B et al (2022) Assessing and benchmarking 3D city models. *Int J Geogr Inf Sci*. <https://doi.org/10.1080/13658816.2022.2140808>
- León-Sánchez C et al (2021) Testing the new 3D bag dataset for energy demand estimation of residential buildings. *Int Arch Photogramm Remote Sens Spat Inf Sci XLVI-4/W1-2021:69–76*. <https://doi.org/10.5194/isprs-archives-XLVI-4-W1-2021-69-2021>
- Li Y et al (2021) Deep learning for LiDAR point clouds in autonomous driving: a review. *IEEE Trans Neural Netw Learn Syst* 32(8):3412–3432. <https://doi.org/10.1109/TNNLS.2020.3015992>
- Liamis T, Mimis A (2022) Establishing semantic 3D city models by GRextADE: the case of the Greece. *J Geovis Spat Anal* 6(1):15. <https://doi.org/10.1007/s41651-022-00114-0>
- Lu Q et al (2019) Developing a dynamic digital twin at building and city levels: A case study of the West Cambridge campus. *J Manage Eng* 36. [https://doi.org/10.1061/\(ASCE\)ME.1943-5479.0000763](https://doi.org/10.1061/(ASCE)ME.1943-5479.0000763)
- Masoumi H et al (2023) City digital twins: their maturity level and differentiation from 3D city models. *Big Earth Data* 0(0):1–46. <https://doi.org/10.1080/20964471.2022.2160156>

- Mylonas G et al (2021) Digital twins from smart manufacturing to smart cities: a survey. *IEEE Access*, 1–1. <https://doi.org/10.1109/ACCESS.2021.3120843>
- Naserentin V, Logg A (2022) Digital twins for city simulation: automatic, efficient, and robust mesh generation for large-scale city modeling and simulation. <http://arxiv.org/abs/2210.05250>
- Nguyen SH, Kolbe TH (2022) Path-tracing semantic networks to interpret changes in semantic 3D city models. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci X-4/W2-2022*:217–224. <https://doi.org/10.5194/isprs-annals-X-4-W2-2022-217-2022>
- Nurunnabi A et al (2022) Robust techniques for building footprint extraction in aerial laser scanning 3D point clouds. *Int Arch Photogramm Remote Sens Spat Inf Sci XLVIII-3-W2-2022*:43–50. <https://doi.org/10.5194/isprs-archives-XLVIII-3-W2-2022-43-2022>
- Nys G-A et al (2020) Automatic 3D buildings compact reconstruction from Lidar point clouds. In: Presented at the international archives of the photogrammetry, remote sensing and spatial information sciences—ISPRS archives. <https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-473-2020>
- Oosterom P et al (2015) Massive point cloud data management: Design, implementation and execution of a point cloud benchmark. *Comput Graph* 49. <https://doi.org/10.1016/j.cag.2015.01.007>
- Ortega S et al (2021) Generating 3D city models from open LiDAR point clouds: advancing towards smart city applications. In: Mobasheri A (ed) *Open source geospatial science for urban studies: the value of open geospatial data*. Springer International Publishing, Cham, pp 97–116. https://doi.org/10.1007/978-3-030-58232-6_6
- Paden I et al (2022) Towards automatic reconstruction of 3D city models tailored for urban flow simulations. *Front Built Environ* 8
- Park Y, Guldmann J-M (2019) Creating 3D city models with building footprints and LIDAR point cloud classification: a machine learning approach. *Comput Environ Urban Syst* 75:76–89. <https://doi.org/10.1016/j.compenvurbsys.2019.01.004>
- Perez-Perez Y et al (2021) Scan2BIM-NET: deep learning method for segmentation of point clouds for scan-to-BIM. *J Constr Eng Manage* 147(9):04021107. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0002132](https://doi.org/10.1061/(ASCE)CO.1943-7862.0002132)
- Peters R et al (2015) Visibility analysis in a point cloud based on the medial axis transform. In: *Eurographics workshop on urban data modelling and visualization*, 6 pages. <https://doi.org/10.2312/UDMV.20151342>
- Peters R et al (2022) Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the Netherlands. *Photogramm Eng Remote Sens* 88(3):165–170. <https://doi.org/10.14358/PERS.21-00032R2>
- Poux F (2019) The smart point cloud: structuring 3D intelligent point data. <https://doi.org/10.13140/RG.2.2.20457.75367>
- Poux F, Billen R (2019) Voxel-based 3D point cloud semantic segmentation: unsupervised geometric and relationship featuring vs deep learning methods. *ISPRS Int J Geo-Inf* 8(5):213. <https://doi.org/10.3390/ijgi8050213>
- Pružinec F, Ďuračiová R (2022) A point-cloud solar radiation tool. *Energies* 15:7018. <https://doi.org/10.3390/en15197018>
- Richter R (2018) Concepts and techniques for processing and rendering of massive 3D point clouds
- Scalas A et al (2022) Potential of the geometric layer in urban digital twins. *ISPRS Int J Geo-Inf* 11(6):343. <https://doi.org/10.3390/ijgi11060343>
- Schrotter G, Hürzeler C (2020) The digital twin of the city of Zurich for urban planning. *PFG J Photogramm Rem Sens Geoinf Sci* 88. <https://doi.org/10.1007/s41064-020-00092-2>
- Soilán M et al (2020) 3D point cloud to BIM: semi-automated framework to define IFC alignment entities from MLS-acquired LiDAR data of highway roads. *Remote Sens* 12(14):2301. <https://doi.org/10.3390/rs12142301>
- Soon KH, Khoo VHS (2017) CITYGML modelling for Singapore 3D national mapping. *Int Arch Photogramm Remote Sens Spat Inf Sci XLII-4/W7*:37–42. <https://doi.org/10.5194/isprs-archives-XLII-4-W7-37-2017>

- Stojanovic V et al (2018) Towards the generation of digital twins for facility management based on 3D point clouds. Presented at the (2018)
- Stoter JE et al (2020) State of the art in 3D city modelling: Six challenges facing 3D data as a platform. In: GIM international: the worldwide magazine for geomatics, vol 34
- Stoter JE et al (2021) Digital twins: a comprehensive solution or hopeful vision? In: GIM international: the worldwide magazine for geomatics
- Su Z et al (2022) Semantic segmentation of 3D point clouds based on high precision range search network. *Remote Sens* 14(22):5649. <https://doi.org/10.3390/rs14225649>
- Toth C, Józkóv G (2016) Remote sensing platforms and sensors: a survey. *ISPRS J Photogramm Remote Sens* 115:22–36. <https://doi.org/10.1016/j.isprsjprs.2015.10.004>
- Tutzauer P et al (2019) Semantic urban mesh enhancement utilizing a hybrid model. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci IV-2-W7*:175–182. <https://doi.org/10.5194/isprs-annals-IV-2-W7-175-2019>
- Wang Y et al (2019) A survey of mobile laser scanning applications and key techniques over urban areas. *Remote Sens* 11:13. <https://doi.org/10.3390/rs11131540>
- Wilk Ł et al (2022) Semantic urban mesh segmentation based on aerial oblique images and point clouds using deep learning. *Int Arch Photogramm Remote Sens Spat Inf Sci XLIII-B2-2022*:485–491. <https://doi.org/10.5194/isprs-archives-XLIII-B2-2022-485-2022>
- Willenborg B et al (2018) Integration of semantic 3D city models and 3D mesh models for accuracy improvements of solar potential analyses. *ISPRS Int Arch Photogramm Remote Sens Spat Inf Sci XLII-4/W10*:223–230. <https://doi.org/10.5194/isprs-archives-XLII-4-W10-223-2018>
- Würstle P et al (2022) Viability testing of game engine usage for visualization of 3D geospatial data with OGC standards. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci X-4/W2-2022*:281–288. <https://doi.org/10.5194/isprs-annals-X-4-W2-2022-281-2022>
- Xue F et al (2020) From LiDAR point cloud towards digital twin city: clustering city objects based on Gestalt principles. *ISPRS J Photogramm Remote Sens* 167:418–431. <https://doi.org/10.1016/j.isprsjprs.2020.07.020>
- Yan J et al (2019) Integration of 3D objects and terrain for 3D modelling supporting the digital twin. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci IV-4/W8*:147–154. <https://doi.org/10.5194/isprs-annals-IV-4-W8-147-2019>
- Zhang G et al (2017) Point cloud based visibility analysis: first experimental results. In: Proceedings of the 20th AGILE conference on geographic information science
- Zhou Y et al (2023) Sampling-attention deep learning network with transfer learning for large-scale urban point cloud semantic segmentation. *Eng Appl Artif Intell* 117:105554. <https://doi.org/10.1016/j.engappai.2022.105554>

Unsupervised Roofline Extraction from True Orthophotos for LoD2 Building Model Reconstruction



Weixiao Gao, Ravi Peters, and Jantien Stoter

Abstract This paper discusses the reconstruction of LoD2 building models from 2D and 3D data for large-scale urban environments. Traditional methods involve the use of LiDAR point clouds, but due to high costs and long intervals associated with acquiring such data for rapidly developing areas, researchers have started exploring the use of point clouds generated from (oblique) aerial images. However, using such point clouds for traditional plane detection-based methods can result in significant errors and introduce noise into the reconstructed building models. To address this, this paper presents a method for extracting rooflines from true orthophotos using line detection for the reconstruction of building models at the LoD2 level. The approach is able to extract relatively complete rooflines without the need for pre-labeled training data or pre-trained models. These lines can directly be used in the LoD2 building model reconstruction process. The method is superior to existing plane detection-based methods and state-of-the-art deep learning methods in terms of the accuracy and completeness of the reconstructed building. Our source code is available at <https://github.com/tudelft3d/Roofline-extraction-from-orthophotos>.

Keywords Building rooflines extraction · 3D building models · True orthophotos

This article was selected based on the results of a double-blind review of the full paper.

W. Gao (✉) · J. Stoter
Department Urbanism, Delft University of Technology, Delft, The Netherlands
e-mail: w.gao-1@tudelft.nl

J. Stoter
e-mail: j.e.stoter@tudelft.nl

R. Peters
3DGI, Zoetermeer, The Netherlands
e-mail: ravi.peters@3dgi.nl

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
T. H. Kolbe et al. (eds.), *Recent Advances in 3D Geoinformation Science*, Lecture Notes
in Geoinformation and Cartography, https://doi.org/10.1007/978-3-031-43699-4_27

425

1 Introduction

With the recent advancements in computer vision and photogrammetry technology, the acquisition of both 2D and 3D data for large-scale urban environments has become feasible. This has led to the derivation of urban 2D data from street view, aerial, and satellite images, while 3D data is derived from LiDAR point clouds, dense image matching generated point clouds, and textured meshes. Building models in 3D urban scenes are essential for various applications such as solar irradiation (Biljecki et al. 2015, 2018), photovoltaic analysis (Catita et al. 2014; Eerenstein et al. 2015), building illumination analysis (Saran et al. 2015), and urban planning (Chen and Ran 2011; Czyńska and Rubinowicz 2014). However, for large-scale urban scene analysis, it is often necessary to find the optimal balance between simplicity and preserving essential, detailed geometric features. To this end, LoD2 building models, conforming to the CityGML standard (Kolbe et al. 2021), have gained popularity.

The automatic reconstruction of LoD2 building models from 2D and 3D data has been a topic of active research. Traditionally, point clouds obtained from airborne LiDAR have been used to generate such models (Peters et al. 2022). However, due to the high costs and long intervals associated with acquiring and collecting LiDAR point clouds for large-scale urban scenes, it can be challenging to capture rapidly developing and changing areas in cities. Consequently, there may be missing data for these regions, which makes it difficult to update the models accurately. To address this issue, researchers have started exploring the use of point clouds generated from aerial images with the dense image matching pipelines for automatic LoD2 building model reconstruction (Wang et al. 2023). Dense image matching point clouds offer several advantages over LiDAR point clouds, including being easily obtainable, cost-effective, and having color information. Furthermore, they can be frequently updated to reflect changes in the urban environment, making them more suitable for vector building reconstruction tasks in large-scale, high-frequency urban scenes where the data must be updated frequently to reflect urban development. As a result, dense image matching point clouds have become an area of interest for researchers studying automated LoD2 building model reconstruction.

Generating dense point clouds from aerial oblique photogrammetry images can result in significant errors and introduce noise into the generated point clouds, compared to high-precision LiDAR point clouds. This can make originally planar regions, such as roof planes of buildings, become uneven, which creates challenges for reconstructing LoD2 building models. Traditional point cloud plane detection-based methods (Schnabel et al. 2007; Lafarge and Mallet 2012) can also exacerbate the issue caused by dense image matching point clouds by increasing complexity and reducing accuracy in the reconstructed models. However, the original images used to generate these point clouds and the synthesized orthoimagery typically contain less noise and fewer errors, providing an opportunity to extract roof structures directly from the images and combine them with height information from the point clouds extracted from the same source images for reconstructing high-precision LoD2 building models.

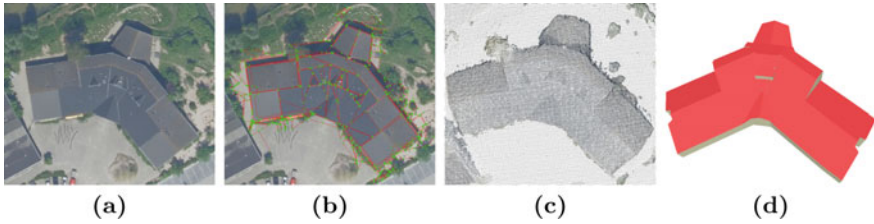


Fig. 1 The workflow of our method. First, we use a true orthophoto (a) as input. Next, we perform line extraction (b) to partition the building footprint, which generates separate roof parts. We then utilize a dense point cloud (c) to extrude the partition results and reconstruct a LoD2 building model (d)

In recent years, more and more research has focused on how to use deep learning methods to extract roof structures from images (Alidoost et al. 2019; Nauata et al. 2020; Zhang et al. 2020, 2021; Zhao et al. 2022; Chen et al. 2022). Although these methods have performed well on public datasets (Nauata et al. 2020), they still require a large amount of manually labeled training data and cannot guarantee good generalization ability. More importantly, they cannot ensure the geometric completeness of the extracted roof structures, especially for complex buildings.

This paper presents a method for extracting rooflines using line detection, which can be used for the reconstruction of building models at the LoD2 level (see Fig. 1). As our results will show, our approach has the advantage of being able to extract relatively complete roof lines without the need for any pre-labeled training data or pre-trained models. Moreover, our method surpasses both traditional plane detection-based methods and state-of-the-art deep learning methods based on transformers in terms of both the accuracy and completeness of the reconstructed building. Our method is applicable to large-scale urban scenes and can be used for extracting roof structures of complex buildings.

2 Recent Advances in Roofline Extraction

In recent years, deep learning methods have received increasing attention for extracting roof structures from images using neural networks. For instance, Alidoost et al. (2019) proposed a CNN-based approach for 3D building reconstruction from a single 2D image, which achieved accurate height prediction, roofline segmentation, and building boundary extraction. However, the accuracy of the method degraded for test data with different spatial-spectral characteristics and complicated buildings. Another pioneering work by Nauata et al. (2020) used CNNs to detect geometric primitives and infer their relationships, fusing all information into a planar graph through holistic geometric reasoning for reconstructing a building architecture. This work made significant improvements over the existing state-of-the-art but has limitations in handling missed corners, curved buildings, and weak image signals.

Furthermore, Zhang et al. (2020) proposed Conv-MPN, a message passing neural architecture, for structured outdoor architecture reconstruction, which achieved significant performance improvements over existing prior-free solutions but has the drawback of extensive memory consumption and limitations on the number of corner candidates. They further improved the method by presenting a novel explore-and-classify framework (Zhang et al. 2021) for structured outdoor architecture reconstruction. The method learns to classify the correctness of primitives while exploring the space of reconstructions via heuristic actions, and demonstrated significant improvements over all state-of-the-art methods, with a few limitations related to slow test-time inference and corner detection failures in extreme cases. Another similar work is presented by Zhao et al. (2022), who proposed RSGNN, an end-to-end learning framework for planar roof structure extraction in polygon format from VHR remote sensing images, which combines primitive detectors and GNN-based relationship inference and shows superior results in both qualitative and quantitative evaluations. However, the method has limitations in handling complex roof structures and labeling accuracy of reference data.

A recent work HEAT is proposed by Chen et al. (2022), which is an attention-based neural network for reconstructing a planar graph from 2D raster images. It uses transformer decoders and iterative inference at test time but still faces challenges in missing corners and rare structures. Despite their advantages, deep learning methods require a large amount of training data with manual labels and may not ensure good generalization and geometric completeness for extracting roof structures, especially for complex buildings. Even a single missing roof line can result in significant geometric errors in the reconstructed building models.

Lussange et al. (2023) recently proposed a novel approach for extracting roof sections from satellite images to reconstruct LoD2 building models. Their method employs deep learning with a Mask R-CNN model to segment roof sections in 2D satellite images and uses panoptic segmentation to infer heights-to-ground for full 3D reconstruction. While this method shows potential for large-scale LoD2 building reconstruction, it has limitations, as it heavily relies on hand-annotated training data and struggles with complex roof structures.

3 Methodology

Our method involves three main steps: (1) building image cropping; (2) extracting line segments from the building images; and (3) reconstructing LoD2 building models.

3.1 Building Image Cropping

Due to the large size of the original true orthophoto and the presence of other objects in addition to buildings, direct line detection would extract many redundant line

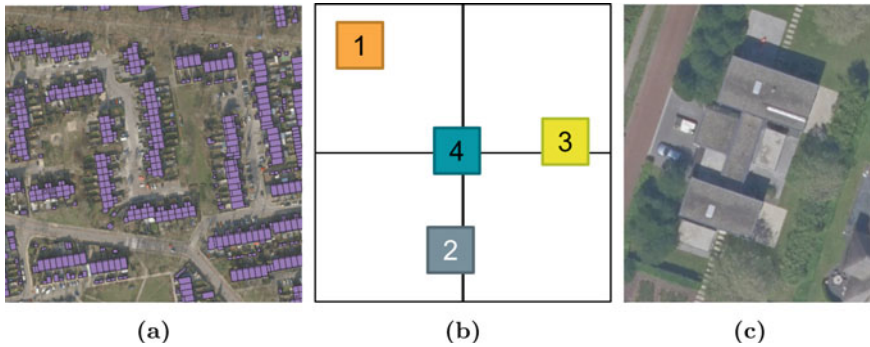


Fig. 2 The workflow of building image cropping. **(a)** Building footprints on the map. **(b)** Four types of correspondences that can exist between a bounding box of a building and true orthophotos. **(c)** Cropped building image

segments. Therefore, our first objective is to extract a single building image for each building from the true orthophoto.

We use GIS data, specifically the building footprints on the map (see Fig. 2a), to assist with the extraction process. First, we merge adjacent buildings according to their footprints. Next, we establish a bounding box based on the footprint of a single building and determine the corresponding region in the true orthophotos by using the coordinates of the four corners of the bounding box (as illustrated in Fig. 2b). The geographic coordinates of the top-left corner of the true orthophoto are already known. The correspondences between a specific bounding box and orthophotos can be divided into four cases: (1) a single image, (2) two images joined horizontally, (3) two images joined vertically, and (4) four images joined both horizontally and vertically. In the final step, we use the building boundary boxes to crop and stitch the corresponding images, generating a single composite image for each building (see Fig. 2b). These images will serve as input data for the next stage of line segment detection.

3.2 Line Extraction

Our objective is to detect line segments in the single building image obtained in the previous step and convert the detection results from image coordinates to geographic coordinates. We use the KIPPI (Bauchet et al. 2018) algorithm as our primary tool for line detection. Firstly, it employs a line segment detector to identify line segments in the image. These detected line segments are then globally regularized based on geometric properties like parallelism, perpendicularity, and collinearity. Finally, the line segments are bidirectionally extended using kinetic data structures to segment the image. As shown in Fig. 3b, rooflines represent a subset of these segmented lines.

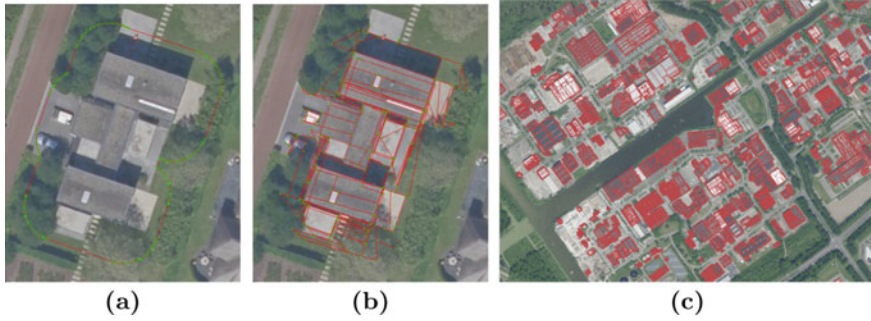


Fig. 3 Line extraction workflow on the single building image. (a) Buffered building footprints. (b) Cropped line segments. (c) Georeferenced rooflines

However, the single building image may include redundant information such as non-building features surrounding the building. To filter out such information, we use a buffer (of 60 pixels in all experiments) generated from the building footprint to refine the segmented lines (see Fig. 3a). Next, we convert the coordinates of each node of every line segment in the image to geographic coordinates using Eq. 1 as follows:

$$\begin{aligned} g_x &= t_x + (p_x + b_x) \times s \\ g_y &= t_y - (p_y + b_y) \times s \end{aligned} \quad (1)$$

where g_x and g_y represent the geographic coordinates of the line node, while p_x and p_y represent its pixel coordinates. The t_x and t_y denote the geographic coordinates of the top left corner of the corresponding true orthophoto. The s represents the pixel size. The results provide geometric information for the next step of building reconstruction (see Fig. 3c).

It is important to highlight that despite the removal of redundant lines, certain artifacts may remain in the roof structure. Nonetheless, in the context of 3D building model reconstruction, prioritizing roofline completeness takes precedence over eliminating redundancy. The presence of real roof lines is crucial for accurate reconstruction, as the absence of such lines can lead to inaccuracies in the final model. Any remaining redundant lines can be effectively addressed in the subsequent reconstruction step through cell selection with the aid of an elevation prior (Peters et al. 2022).

3.3 LoD2 Building Reconstruction

In this step, our goal is to reconstruct LoD2 building models using the set of line segments extracted in the previous step, the building footprint, and the point cloud generated by the dense image matching pipeline from the same source images. Our

approach is based on the method proposed by Peters et al. (2022), with the main difference being the replacement of the line segments generated through point cloud detection of planes with the set of line segments extracted from the image. We then use the set of line segments extracted from the image to subdivide the building footprint and obtain the roof plane structure. Finally, based on the height information provided by the point cloud, we extrude the roof parts to generate a watertight and 2D-manifold solid building model in LoD2. Since we rely on the building footprint to extract the building point clouds and filter rooflines, it is possible that the reconstructed models may not capture roof overhangs accurately, potentially leading to their omission.

4 Experimental Results

4.1 Dataset

We utilized an orthophoto dataset with a resolution of 8 cm (Dutch Kadaster 2023), covering three cities including Almere, Den Haag, and Leiden. The building footprint data was obtained from the Building and Address Register of the Netherlands (BAG). The dense point cloud was generated using oblique aerial images with Nframes SURE software by a workflow designed and executed by the Dutch Kadaster. Our experimental setup consisted of a computer equipped with an AMD Ryzen Threadripper 1920X 12-Core Processor and 32 GB of memory.

4.2 Evaluation and Comparison

Our method for line extraction was tested on three cities in the Netherlands, with extraction times of 1.62 h for 27,395 buildings in Almere, 3.22 h for 43,520 buildings in Den Haag, and 2.85 h for 53,502 buildings in Leiden. The time required for line extraction in these cities is mainly influenced by the number of buildings and the complexity of their external geometric structures.

Evaluation of rooflines. We selected twelve representative buildings from the three cities and we projected the rooflines extracted from the 3D BAG building models (Peters et al. 2022) onto the orthophotos as a reference. We compared our roof line detection method (i.e., KIPPI method) with the HEAT method (Chen et al. 2022), which currently holds the top position in public benchmark datasets. To predict the roof structure of the generated single-building images (as shown in Fig. 4), we used the trained model provided by the authors of HEAT (Chen et al. 2022). We calculated the completeness of our method by measuring the percentage of extracted lines that overlapped with the reference lines (within 25 pixels offset in all experiments). Our method achieved around 90% overlap with the reference lines, while the HEAT method only achieved about 44%. This comparison clearly shows that our

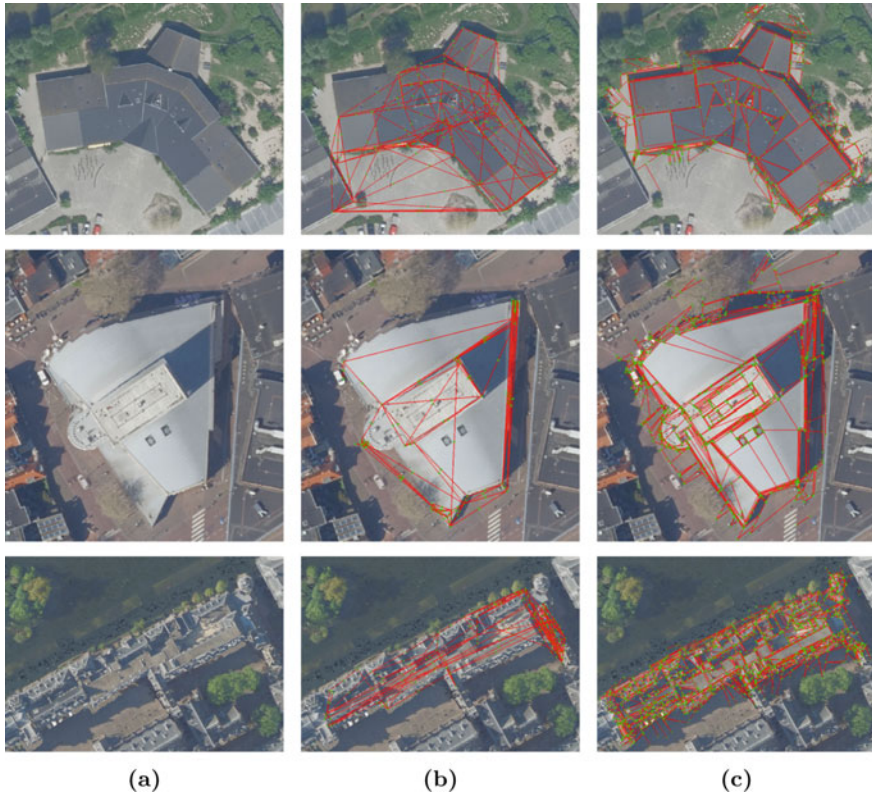


Fig. 4 Roofline comparison. (a) Input building image. (b) HEAT results. (c) KIPPI results

method produces more complete results, which is further evident in the subsequent 3D reconstruction outcomes. In addition, deep learning-based methods require fixed-size image inputs due to limited GPU memory, and HEAT (Chen et al. 2022) limits the size of each building image to 256×256 . This limitation results in a significant loss of image resolution when scaling building images, especially for individual large and complex buildings (see Fig. 4).

Evaluation of reconstructed buildings. We selected twelve buildings from the three cities that were previously identified and used them to reconstruct LoD2 models and evaluate our results. We used 3D BAG building models (Peters et al. 2022) reconstructed from LiDAR point clouds as the ground truth for evaluation. To compare the reconstructed building models, we utilized the same image-dense matching generated point clouds as input and applied three different approaches for roofline generation: (1) plane detection from the dense matching point cloud, (2) HEAT, and (3) our method (i.e., KIPPI). The evaluation criteria focused on model complexity, Hausdorff distance (Aspert et al. 2002), and RMSE. We measured the complexity of the models based on the number of reconstructed faces and calculated the Hausdorff

Table 1 A comprehensive quantitative evaluation of LoD2 building models using the averaged number of faces, mean Hausdorff distance, and RMSE metrics

| | Faces | Mean | RMSE |
|--------|---------|-------------|-------------|
| 3D BAG | 1762.5 | – | – |
| HEAT | 3150.75 | 0.67 | 1.16 |
| Planes | 5660.08 | 0.49 | 0.67 |
| KIPPI | 4356.33 | 0.38 | 0.65 |

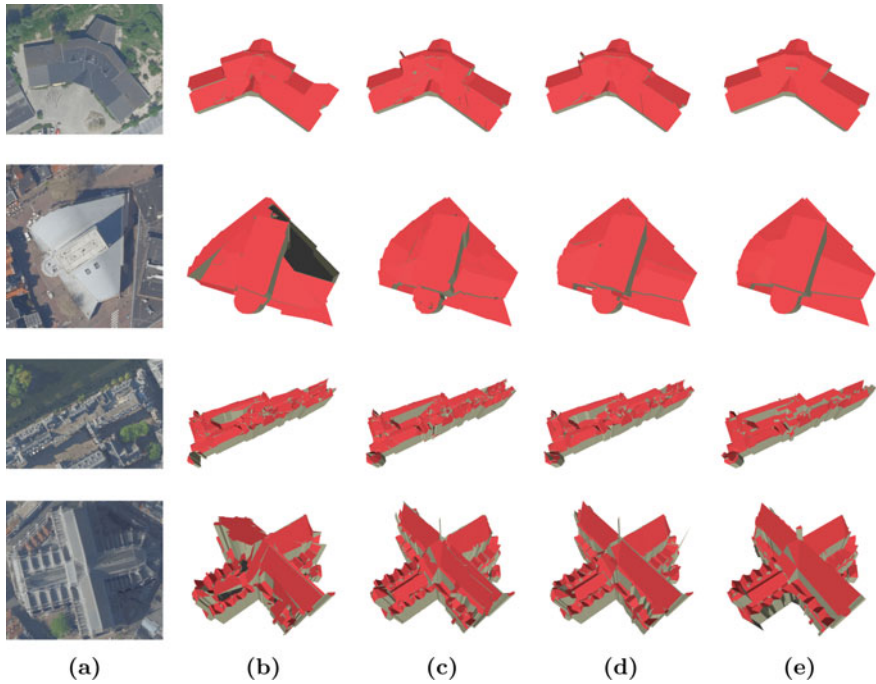


Fig. 5 Comparison of LoD2 Building Models: Input building images are shown in (a), while the reconstruction results using HEAT generated rooflines are shown in (b), plane detection method in (c), and our line segments in (d). Ground truth 3D BAG models reconstructed from LiDAR point clouds and plane detection are shown in (e)

distance using sampled points from the building models. We computed the average Hausdorff distance and root mean square error (RMSE) of all the selected buildings in comparison to their respective ground truth models. It should be noted that if a model contains more faces with larger errors, then it is likely to have redundant faces. Conversely, models with smaller errors and more faces tend to capture more details.

Table 1 and Fig. 5 show the quantitative and qualitative results of the different methods, respectively. Our method produced results with higher accuracy and completeness than the HEAT method (Chen et al. 2022), which had the lowest completeness

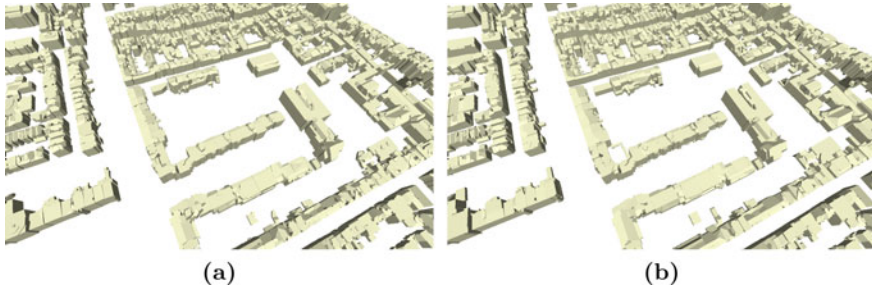


Fig. 6 Comparison of large-scale LoD2 building models reconstructed from extracted rooflines and image dense matching generated point clouds. **(a)** Building models reconstructed using rooflines extracted from plane detection. **(b)** Building models reconstructed using rooflines extracted from our method (i.e., KIPPI method)

plexity but the highest error due to missing line segment completeness. Furthermore, the reconstructed building models using our rooflines exhibited lower complexity and lower errors compared to the rooflines obtained from the plane detection method (Peters et al. 2022). Figure 5d demonstrate that even when orthophoto images are somewhat blurry, our method can maintain high accuracy and low complexity, indicating its high robustness. It is important to mention that in KIPPI (Bauchet et al. 2018), the gradient magnitude parameter can be adjusted to effectively handle low-intensity values in shadowed regions. Moreover, in the case of textureless regions lacking clear edge information, such as the example depicted in Fig. 4c, the curved surface nature of the roof can result in the loss of certain rooflines during the detection phase.

Figure 6 illustrates the outcomes of the LoD2 building model reconstruction for a large-scale urban scene. Notably, our method demonstrates its capacity to be seamlessly applied to such scenes, as depicted in Fig. 6b. This suggests its potential as a viable alternative to image-based approaches for reconstructing the complete 3D BAG building model. Additionally, we observe that our method produces building models with enhanced geometric regularity compared to the plane detection-based method (see Fig. 6b).

5 Conclusion

In conclusion, this paper presented a method for extracting rooflines using line detection, which is used for the improved reconstruction of building models at the LoD2 level from point clouds generated from images. The approach has the advantage of being able to extract relatively complete rooflines without the need for any pre-labeled training data or pre-trained models. Furthermore, the method surpassed both traditional plane detection-based methods and state-of-the-art deep learning methods based on transformers in terms of both the accuracy and completeness of the

reconstructed building. The method is applicable to large-scale urban scenes and can be used for extracting roof structures of complex buildings. Although deep learning methods have received increasing attention for extracting roof structures from images, they still require a large amount of manually labeled training data and cannot ensure the geometric completeness of the extracted roof structures, especially for buildings with complex structures. The proposed method has the potential to overcome these limitations and improve the LoD2 model reconstruction from dense image matching point clouds which can then become a promising alternative for automated LoD2 building model reconstruction from LiDAR. Further research could be directed toward developing more accurate algorithms for line detection from low-resolution images, such as satellite imagery, to improve the reconstruction of building models also from those sources.

Acknowledgements This project has received funding from the European Research Council (ERC) under the Horizon Europe Research & Innovation Programme (grant agreement no. 101068452 3DBAG: detailed 3D Building models Automatically Generated for very large areas).

References

- Alidoost F, Arefi H, Tombari F (2019) 2D image-to-3D model: Knowledge-based 3D building reconstruction (3DBR) using single aerial images and convolutional neural networks (CNNs). *Remote Sens* 11(19):2219
- Aspert N, Santa-Cruz D, Ebrahimi T (2002) Mesh: measuring errors between surfaces using the hausdorff distance. In: *Proceedings. IEEE international conference on multimedia and expo*, vol 1. IEEE
- Bauchet JP, Lafarge F (2018) Kippi: kinetic polygonal partitioning of images. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*
- Besuevsky Gonzalo, Beckers Benoit, Patow Gustavo (2018) Skyline-based geometric simplification for urban solar analysis. *Graph Models* 95:42–50
- Biljecki F, Heuvelink GB, Ledoux H, Stoter J (2015) Propagation of positional error in 3D GIS: estimation of the solar irradiation of building roofs. *Int J Geograph Inf Sci* 29(12):2269–2294
- Catita C, Redweik P, Pereira J, Brito MC (2014) Extending solar potential analysis in buildings to vertical facades. *Comput Geosci* 66:1–12
- Chen R (2011) The development of 3D city model and its applications in urban planning. In: *2011 19th International conference on geoinformatics*. IEEE
- Chen J, Qian Y, Furukawa Y (2022) HEAT: holistic edge attention transformer for structured reconstruction. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*
- Czyńska Klara, Rubinowicz Paweł (2014) Application of 3D virtual city models in urban analyses of tall buildings: today practice and future challenges. *Architecturae et Artibus* 6(1):9–13
- Dutch Kadaster (2023) Luchtfoto Actueel Ortho 8cm RGB. <https://opendata.beeldmateriaal.nl/>. Accessed 5 Feb 2023
- Eerenstein W, Jansen MJ, de Groot KM, Carr AJ, Okel LAG, Goris MJJA, van Roosmalen JAM, Bende EE, Jonkman R, van der Sanden R, Bakker J (2015) Tessera: maximizing PV yield performance with size flexibility for bipv. In: *30th European PV solar energy conference and exhibition*
- Kolbe TH, Kutzner T, Smyth CS, Nagel C, Roensdorf C, Heazel C (eds) (2021) OGC city geography markup language (CityGML) part 1: conceptual model standard, Ver 3.0.0

- Lafarge F, Mallet C (2012) Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation. *Int J Comput Vis* 99:69–85
- Lussange J, Yu M, Tarabalka Y, Lafarge F (2023) 3D detection of roof sections from a single satellite image and application to LOD2-building reconstruction. arXiv preprint [arXiv:2307.05409](https://arxiv.org/abs/2307.05409)
- Nauata N, Furukawa Y (2020) Vectorizing world buildings: planar graph reconstruction by primitive detection and relationship inference. In: *Computer vision-ECCV 2020: 16th European conference, Glasgow, UK, 23–28 Aug 2020. Proceedings, Part VIII* 16. Springer International Publishing
- Peters R, Dukai B, Vitalis S, van Liempt J, Stoter J (2022) Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the Netherlands. *Photogram Eng Remote Sens* 88(3):165–170
- Saran S, Wate P, Srivastav SK, Krishna Murthy YVN (2015) CityGML at semantic level for urban energy conservation strategies. *Ann GIS* 21(1):27–41
- Schnabel R, Wahl R, Klein R (2007) Efficient RANSAC for point-cloud shape detection. In: *Computer graphics forum*, vol 26, no 2. Blackwell Publishing Ltd, Oxford, UK
- Wang F, Zhou G, Hu H, Wang Y, Fu B, Li S, Xie J (2023) Reconstruction of LoD-2 building models guided by Façade structures from oblique photogrammetric point cloud. *Remote Sens* 15(2):400
- Zhang F, Nauata N, Furukawa Y (2020) Conv-mpn: convolutional message passing neural network for structured outdoor architecture reconstruction. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*
- Zhang F, Xu X, Nauata N, Furukawa Y (2021) Structured outdoor architecture reconstruction by exploration and classification. In: *Proceedings of the IEEE/CVF international conference on computer vision*
- Zhao Wufan, Persello Claudio, Stein Alfred (2022) Extracting planar roof structures from very high resolution images using graph neural networks. *ISPRS J Photogram Remote Sens* 187:34–45

Enhancing Realism in Urban Simulations: A Mapping Framework for the German National Standard XPlanung and CityGML



Hamza Zahid, Ihab Hijazi, Andreas Donaubaauer, and Thomas H. Kolbe

Abstract 3D spatial data are widely used to simulate various urbanistic phenomena, thanks to their valuable semantic, geometric, and topologic information. CityGML is a highly adopted data standard for semantic 3D city models, providing a standardized description of the cityscape that enables interoperability across different stakeholders. When future scenarios for urban development are simulated, the simulation results can be visualized and further analyzed in synthetically generated 3D city models. However, land use and constructability regulations are often overlooked when generating synthetic 3D city models for simulation purposes, despite some regulatory urban constraints having a direct impact on simulation results. For instance, the roof shape is highly correlated with building solar energy potential, while the zoning maximum allowed number of apartments directly influences the buildings' urban density estimation. Therefore, integrating such constructability knowledge within 3D city models is crucial. This paper proposes a framework for mapping urban planning rules defined in the German XPlanung standard onto 3D city models structured in compliance with CityGML to ensure legislative validity and real-life applicability. We review related work, discuss the structure of CityGML and the main elements concerned by urbanistic laws, explain the main concepts of XPlanung, and investigate the mapping of regulatory information with CityGML entities.

This article was selected based on the results of a double-blind review of the full paper.

H. Zahid · I. Hijazi (✉) · A. Donaubaauer · T. H. Kolbe
Geoinformatics, Technical University of Munich, Arcisstr. 21, Munich, Germany
e-mail: ihab.hijazi@tum.de; eehab@najah.edu

H. Zahid
e-mail: h.zahid@tum.de

A. Donaubaauer
e-mail: andreas.donaubaauer@tum.de

T. H. Kolbe
e-mail: thomas.kolbe@tum.de

I. Hijazi
Urban Planning Engineering Department, An-Najah National University, Nablus, Palestine

Keywords CityGML · XPlanung · Urban simulation · 3D city model · Urban regulations

1 Introduction

Nowadays, 3D city models are increasingly deployed in a number of domains and applications beyond visualization (Kutzner et al. 2020). Since they provide a well-structured description of the urban space, semantic 3D virtual city models are considered as a central asset to simulate various urban phenomena.

The standardization of 3D city models allows interoperability across different simulation domains. Adopting an open and common definition of the cityscape makes the interchange of data between urban planners at different levels possible (Hijazi et al. 2018). CityGML models real-life urban objects by defining four aspects, i.e. geometry, semantics, appearance, and topological interrelationships (Kolbe 2009). Using the CityGML standard, 3D city models have been largely used for different urban simulations that have a spatial context. For instance, simulating the urban spatio-temporal growth (Brasebin et al. 2014; Hijazi et al. 2022), measuring the impact of urban configurations on energy demand (Kaden and Kolbe 2014), evaluating the energy and environmental impacts of large residential building stocks throughout their life cycle (Harter et al. 2020), noise pollution (Stoter et al. 2008), and traffic simulation (Beil et al. 2022). The potential of urban simulations is increasing especially with the availability of tools that can generate synthetic virtual 3D city models for simulating possible future states of the city (Biljecki et al. 2016). Procedural modelling is a commonly deployed approach to generate 3D city models (Biljecki et al. 2015; Müller et al. 2006). While procedural modelling focuses on translating input data and rules into 3D city models, land constructability regulations and limitations are usually disregarded (Brasebin et al. 2016). Consequently, the generated synthetic model may not respect the current land regulations. Thus, the integration of such land constructability rules within the 3D city data model is crucial. Furthermore, the availability of standards that model land use and urban regulations makes the integration process more prominent.

XPlanung is a German national data standard and exchange format that models the urbanistic regulations within a specific scope area. This standardization aims for efficient, loss-free, and secure interchange of data between different actors, whether from the public or the private sector (XLeitstelle 2023). XPlanung harmonically integrates various land regulations. Some of these regulations have a nationwide impact, others are adopted locally. Namely, the development plan, the land use plan (derived from the German laws and regulations Baugesetzbuch (BauGB) and Baunutzungsverordnung (BauNVO) (Benner 2016), the regional plan and state development plan, the landscape plan, as well as other spatial plans that contain additional relevant regulations.

As stated above, many urban simulations have been implemented based on CityGML. Therefore, it makes sense to use CityGML as a data model for generated synthetic 3D city models, that represent a possible state of a city or city district in the future. Thus, a question arises of how to link XPlanung and CityGML data models to make sure that each generated synthetic 3D city model is legislatively valid. This integrated modelling has the potential to make urban simulations more realistic and applicable in real-life use cases. It should be noted that we understand “realistic” in the context of this work not in the sense of photorealistic city models, but of functionally realistic models that could be transferred into reality from the perspective of urban planning rules.

This paper aims to develop ways for mapping urban planning rules defined in XPlanung into the 3D city models structured in compliance with the CityGML standard. The paper is structured as follows: after an introductory first section, Sect. 2 presents related work focusing on integrating regulatory knowledge within 3D city models, while Sect. 3 investigates the reasons behind a schema-level integration of urban regulations and 3D city models. The structure and the main CityGML concerned elements, as well as the relevant concepts of the German XPlanung standard, are presented in Sect. 4. Section 5 investigates the mapping of regulatory information with CityGML entities. The paper ends with a discussion and conclusions.

2 Related Work

The question of integrating land regulation knowledge for 3D urban simulation have been the focus of various works. (Brasebin et al. 2014) present a system that simulates urban expansion based on existing urban regulations. Constructability constraints were retrieved from two local land planning documents: PLU and POS (Plan Local d’Urbanisme and Plan d’Occupation des Sols in French). A multitemporal database has been developed to derive urban evolution rules based on the comparison between the urban configuration in different epochs. Subsequently, the urban density and fabric can be forecasted. A simulator has been developed to reach the target density. The simulator starts by selecting a suitable parcel based on attractiveness and constructability requirements. Afterwards, the parcel is populated with buildings that respect the urban regulations. If the building does not verify the regulations, morphological modifications have to be made to cope with urbanistic rules. The process is repeated until reaching the target density. Figure 1 describes the workflow in more detail.

Brasebin et al. 2016) define a model that represents knowledge contained in a land use regulation. The model focuses on regulations concerning buildings only. Hence, the regulation articles in regard to other objects as the road network and vegetation are omitted. The model is based on existing standards: themes relevant to buildings, relief and roads are adopted from CityGML2.0, whereas cadastral information (i.e., parcels, separation limits) are based on INSPIRE specifications for cadastral parcels

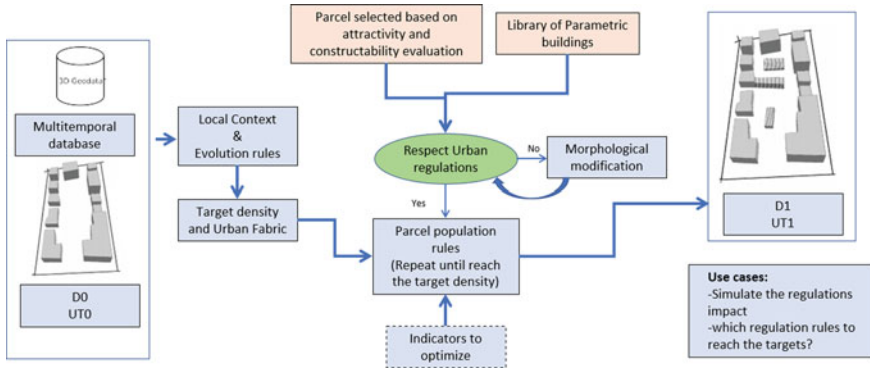


Fig. 1 Simulating the urban expansion based on 3D city model (Brasebin et al. 2014)

(INSPIRE 2012). Zoning elements are modelled based on COVADIS standards (Covadis 2012). Furthermore, these standards are extended with additional classes, attributes, and relationships to cope with land regulation requirements, which is, in this case, the French regulation PLU. The urban rule is expressed, initially, through a textual attribute, and subsequently, through Object Constraint Language formalization. However, the proposed workflow is not intended to satisfy urban simulation requirements but rather perform posterior building permits compliance checks.

Various works have focused on the integration of CityGML with other data models for code compliance checking purposes. (Hbeich et al. 2019) have developed a multi-scale framework for aligning spatial information with the local urban regulation plan PLU in order to generate a compliance report. Building Information Modeling (BIM) is deployed to check building construction rules while CityGML is adopted to verify subzones district restrictions. (Benner et al. 2010) have created a concept for a GML-based checking protocol for building license checking. Semantic and geometric information from BIM and CAD have been transformed to CityGML. These data were harmonized with urban planning data (XPlanGML) and a 3D city model (CityGML) to perform a regulation consistency evaluation. However, the proposed framework is more application-oriented (building permit check) rather than proposing a multi-disciplinary general data model which can support various urban simulation use cases.

The above-mentioned work focuses on using geospatial data combined with regulatory urban information mainly for post compliance checking and building permit validation.

In contrast to the related work, this paper presents a conceptual framework that integrates urban regulation rules within 3D city models at the schema level with a focus on creating synthetic 3D city models for representing future states of the city for urban simulation purposes that comply with urban regulation rules.

3 Schema Level Integration of 3D City Models and Urban Regulations as a Requirement for Realistic Urban Simulations

Urban simulations impose different requirements compared to code compliance use cases. A schema-level harmonization of 3D city models and urban regulations is required based on the following reasons:

- Urban simulations are interdisciplinary and can touch various domains ranging from energy efficiency analysis to urban density simulations.
- 3D city models cover the geo context in regulation and have applicability to issues related to urban planning, the geo context is important for the automatic construction of city objects.

As an example, spatiotemporal urban growth analysis is a common GIS use case that aims to simulate the change of population density over space and time. Urban density change is due to a variety of reasons, including the job market situation and the presence of business structures. These urban internal forces can be well modeled using urban system dynamics for accurate population density predictions. (Hijazi et al. 2022) developed an approach that interconnects urban system dynamics modeling with 3D city models. A schema level integration is crucial regarding this use case due to the following reasons:

- To accurately simulate the housing demand growth by making use of input information related to the existing buildings with respect to their usage. Specifically, the number of buildings can enhance the estimation of the number of inhabitants, whereas the number of commercial and industrial structures can improve the estimation of available job opportunities. As a consequence, information regarding land use and building usage is required.
- To satisfy the urban housing demand growth resulting from the urban simulation, it is fundamental to incorporate information specifying possible vertical and/or horizontal building extension. Hence, knowledge regarding the maximum number of regularly allowed floors and apartments is needed.
- For horizontal densification, land use information about the urban zoning specification is required. Since any created building must respect the applicable constructability regulations. Information regulating the physical building plot location (construction window, building boundary), the allowed floor space ratios, the building use type, and the shape of the building (roof type and ridge direction) must be respected by a synthetic 3D city model, that results from an urban growth simulation.

4 Background on Relevant Standards

4.1 *CityGML Definition and Applications*

In order to guarantee interoperability across users, organizations, and their respective domains, an open standard to represent 3D city models is a must-have. The City Geography Markup Language (CityGML) is a standardized and open data model and XML-based format for the representation, storage, and interchange of 3D city models. CityGML has been developed to create a general definition of urban space. This is achieved by defining four aspects, i.e. geometry, semantics, appearance, and topological interrelationships. CityGML is adopted as a standard by the Open Geospatial Consortium (OGC) and the latest version of CityGML is 3.0, which was released in September 2021 to meet the requirements of new use cases. The latest version of CityGML has been enriched with new modules like Dynamizer, Versioning, Point Cloud, and Construction, whereas other modules have been revised as CityGML Core, Generics, Building, and Transportation (Kutzner et al. 2020). CityGML eases the integration of urban spatial data for different applications.

Beyond visualization, 3D City models are being increasingly used in a number of domains and a wide range of tasks. (Biljecki et al. 2016) demonstrate that 3D city models can be deployed in at least 29 non-visualization use cases that are part of more than 100 applications. Ross (2010) categorized the 3D city models use cases into applications that make use of the geometry only; simulations based on geometry and semantic information, and analysis based on domain-specific extensions. An important issue that arises when generating 3D city models is the conformity of the generated model with the local constructability regulations. Consequently, the integration of regulation knowledge within such frameworks is a necessity. The Building, LandUse, and construction modules are the most prominent thematic CityGML parts to the case of generating synthetic 3D city models, that represent possible future states of the city in a realistic way. These modules will be discussed in more detail in the following sections. The new CityGML 3.0 concepts Space and Space Boundary can, e.g., be deployed to specify the regulatory occupied building space and the obligatory unoccupied spacing that has to be respected between buildings. Since these concepts are abstract core features, only the relevant thematic classes will be further discussed in the following sections.

4.2 *CityGML Structure*

The CityGML LandUse module provides means of defining objects that describe portions of the earth's surface that are either dedicated to a particular land use or possess a specific land cover, with or without vegetation. The LandUse objects in CityGML can account for concepts related to land use (human activities) and land cover (physical/ biological land cover) and can be utilized to depict various features

such as parcels, spatial planning objects, recreational spaces, and physical characteristics of an area in 3D. The top-level feature type `LandUse` is the sole class of the `LandUse` module and represents these `LandUse` objects in the CityGML data model.

Buildings in CityGML are considered as self-supporting, free-standing, normally roofed, and location-fixed constructions. A building can be decomposed logically into building units (`BuildingUnit` and `Storey` features) and functionally into building parts (`BuildingPart` feature). In CityGML, rooms represent the interior of the building, which includes installations and furniture. Installations are usually conceived as permanent parts of the building, like balconies and stairs, whereas furniture is freely movable objects. Structurally speaking, various types of surfaces can bound a building. This includes outer façade elements such as the roof surfaces, ground surfaces, and outer ceiling surfaces. Additionally, indoor room surfaces can be semantically distinguished into interior wall surfaces, floor surfaces, and ceiling surfaces.

Any object that can be manufactured from construction materials, intended to be permanent, and connected to earth is considered a construction object. The CityGML construction module defines concepts to represent construction objects. The purpose of this module is to integrate elements that consist of various categories of construction, namely buildings, tunnels, and bridges. In the Construction module, additional terms such as `Furniture`, `Installations`, and `Constructive Elements` are elaborated on. `Installations` refer to fixed components of a structure that have a significant impact on its exterior or interior appearance and cannot be relocated, such as balconies, chimneys, or stairs. Conversely, furniture pertains to portable objects within a structure, such as tables and chairs. The outer and inner elements of the constructions are differentiated in a similar way to the building module.

4.3 XPlanung as an Urban Plan Content Standard

XPlanung is a German data standard and data exchange format that provides models to describe information related to land use and urban planning. XPlanung is mainly adopted to formalize the content that is related to urban planning, aiming for a loss-free transfer of urban plans between the different actors (XLeitstelle 2023). Starting from 2017, it is mandatory to introduce the XPlanung standard for Information Technology (IT) projects used in the planning and approval procedures (XLeitstelle 2023) for all German municipalities. The main goal of XPlanung is to provide a standardized digital data exchange format to enable simple, fast, and secure interexchange between the different administrative levels and the land planning actors.

UML is used for XPlanung data modelling, whereas the encoding rules have been adopted from ISO 19118 and ISO 19136. XPlanGML is the exchange format derived from the XPlanung data model using the encoding rules mentioned above. XPlanGML is based on GML 3.2.1 (Benner 2016). The latest version 6.0.2 was released in November 2022. XPlanung and XPlanGML are derived from ALKIS/

NAS German standard, which is mainly used in the field of surveying and land registration. Accordingly, XPlanung basic schema uses only one geometry type, which corresponds to the ALKIS model “common geometry” (Benner 2016). XPlanung is based on the following land use and regulation documents:

- Urban development plan (**BPlan**) according to BauGB and BauNVO;
- Land use plan (**FPlan**) according to BauGB and BauNVO;
- Regional plan and state-wide spatial development plan (**RPlan**) according to Raumordnungsgesetz (ROG);
- Landscape plan (**LPlan** core model) in accordance with the Federal Nature Conservation Act.

Besides the nationwide regulations mentioned above, the data model includes a feature called SoPlan for other spatial plans that may contain additional relevant legal foundations. These regulations differ by scope and geographical extent. Thus, XPlanung came to integrate these documents within one standardized data model. Since our focus is on building constructability regulations, we will concentrate on regulations according to BauGB (German national building code) and BauNVO (zoning regulation code), which are modeled by the feature Bplan. These documents define the building constructability rules and the urban planning guidelines at the municipality level.

XPlanung models a spatial planning document as an instance of the plan class. Each specific planning document mentioned above has its own plan class (BP_Plan, FP_Plan, RP_Plan, LP_Plan, SO_Plan), all of which are derived from a common superclass XP_Plan. These plan classes refer to one or more areas that are specific to the respective planning document (BP_Area, FP_Area, RP_Area, LP_Area, SO_Area). These areas may have their own scope and can structure the plan geographically and/or thematically. All classes used for modeling these areas are derived from an abstract superclass XP_Area. Visualizing the plan involves graphically representing the areas using their own maps.

5 Mapping XPlanung and Its Regulations on Relevant CityGML Classes

5.1 Land Use Plan (*Flächennutzungsplan*) and Urban Development Plan (*BebauungsPlan*)

The German Building Code (BauGB) stipulates that municipalities are entitled to publish a land use plan (*Flächennutzungsplan*) and development plans (*Bebauungsplan*). The land use plan outlines the land use type zoning resulting from the projected urban development projects as well as the vision and needs of the municipality. Development plans extend the land use plan with more structural stipulations, the type and extent of the land use areas, the dimensional limitations of the constructable areas,

as well as regulations regarding public and private green spaces. It is important to note that no substantial contradiction in terms of the content of the development plan and the land use plan is allowed. Consequently, the zoning stipulations of the land use plan are present in the development plan with more details in order to make it legally binding for citizens, for building permits allowance, for example:

Land use knowledge is modeled in detail within the XPlanung land use plan FPlan (Flächennutzungsplan) module. This sub-plan section models the following areas:

- areas intended for urban development FP_DevelopmentArea (FP_BebauungsFläche),
- areas that are reserved for waste disposal, excavations, and mineral resources FP_Landfill_Excavation_Mineral resources (FP_Aufschuettung_Abgrabung_Bodenschuetze),
- areas that are conceived for community needs and sports facilities FP_Community_needs_and_SportFacilities (FP_Gemeinbedarf_Spiel_und_Sportanlagen),
- zoning dedicated for agriculture activities and green spaces FP_Agriculture_Forest_GreenSpace (FP_Landwirtschaft_Wald_und_Grün),
- subareas subject to the nature conservation acts FP_NatureConservation (FP_Naturschutz).

Nevertheless, the crucial land use and zoning types are also modeled within the development plan BPlan (BebauungsPlan). As the right part of Fig. 2 shows, the feature class BP_PartialBuildingAreaSurface (BP_BaugebietTeilFläche) includes attributes that specify land use information. The latter XPlanung feature is used within the planning area of a development plan to specify partial areas with a consistent building use, such as purely residential or commercial areas. This XPlanung feature incorporates additionally BauNVO semantic land use information. Namely, the attribute GeneralLandUseType (allgArtDerBaulNutzung) which models the general type of land usage; possible values are Residential building area, Mixed building area, Commercial building area, Special building area, and other building area type. SpecialLandUseType (besondereArtDerBaulNutzung) specifies special area designations including industrial area (code 1800) and rural residential areas (code 1450). Table 1 presents the possible attribute values. In case detailed zoning information is required the attribute DetailedLandUseType (detaillierteArtDerBaulNutzung) can be deployed by defining a corresponding code list. The SpecialUse (sonderNutzung) attribute is used regarding zones that have special characteristics. For instance, if the value 2000 is assigned, the zoning is preserved for military purposes, the SpecialUse codelist is presented in Table 2.

Before a simulation creates a synthetic building model at a specific location in the city using the CityGML standard, verifying if the object's land parcel is legally buildable and under which conditions is a necessity. The CityGML LandUse objects can be used to model various features such as parcels and spatial planning objects. Linking the LandUse object with the municipality-imposed zoning information can be achieved by creating a relationship between the LandUse feature and the XPlanung BP_PartialBuildingAreaSurface (BP_BaugebietTeilFläche) feature as Fig. 2

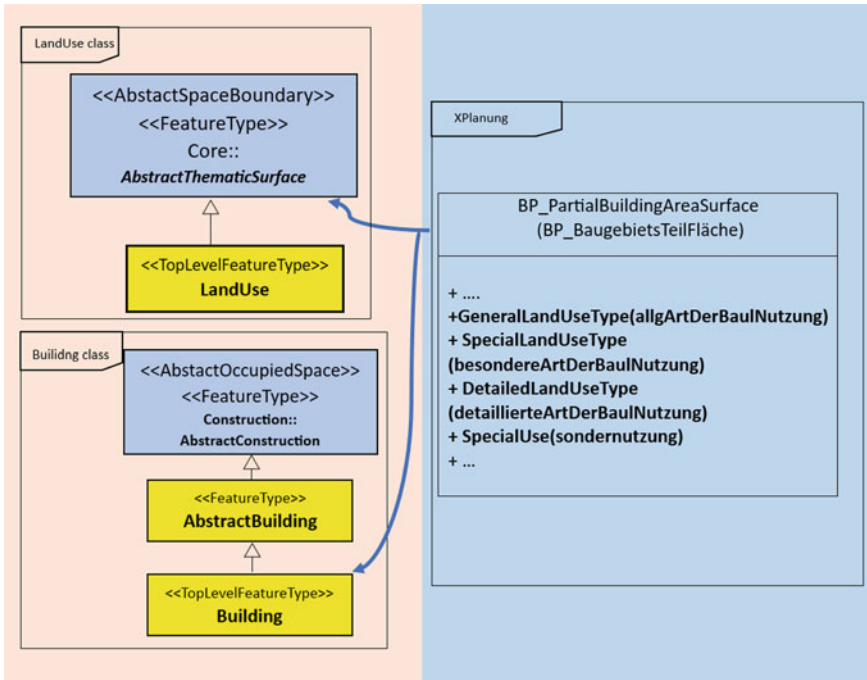


Fig. 2 Relation between land use regulations defined in XPlanung and CityGML classes. The blue arrows show the relations between CityGML classes on the left and an XPlanung class on the right

illustrates. As a consequence, buildings can be enriched with zoning information. Thus, this integration can serve for defining consistency rules in order to make sure that a building is constructed on a buildable area in compliance with the land use plan implications.

5.2 XPlanung Restrictions on the Building Plot

Within the BPlan submodel of the XPlanung standard, there are various classes that have spatial reference and can define areas where buildings can be constructed. This includes the features BP_BesondererNutzungszweckFlaeche (BP_SpecialPurpose-UseArea), BP_SpielSportanlagenFlaeche (BP_SportsFacilitiesArea), BP_GemeinbedarfsFlaeche (BP_CommunityRequirementsArea), BP_GruenFlaeche (BP_GreenArea), BP_VerEntsorgung (BP_DisposalArea), BP_StrassenVerkehrsFlaeche (BP_RoadTrafficArea), BP_VerkehrsflaecheBesondererZweckbestimmung (BP_SpecialPurposeTrafficArea), BP_WohngebaeudeFlaeche (BP_Residential-BuildingArea), and BP_UeberbaubareGrundstuecksFlaeche (BP_Buildable Area).

Table 1

SpecialLandUseType codelist (upper left). SpecialUse codelist (upper right). GeneralLandUseType codelist (bottom right)

XP_Enumerationen::
SpecialLandUseType (BesondereArtDerBaulNutzung)

Small residential area = 1000
 Pure residential area = 1100
 General residential area = 1200
 Special residential area = 1300
 Village area = 1400
 Rural residential area = 1450
 Mixed area = 1500
 Urban area = 1550
 Core area = 1600
 Commercial area = 1700
 Industrial area = 1800
 Special recreation area = 2000
 Other special area = 2100
 Weekend cottage area = 3000
 Special area = 4000
 Other area = 9999

XP_Enumerationen::
SpecialUse
(Sondernutzung)

Weekend cottage area = 1000
 Vacation home area = 1100
 Campground area = 1200
 Spa area = 1300
 Other special recreation area = 1400
 Retail area = 1500
 Large-scale retail area = 1600
 Shop area = 16,000
 Shopping center = 16,001
 Other large-scale retail area = 16,002
 Wholesale special area = 1650
 Traffic training area = 1700
 Port area = 1800
 Renewable energy special area = 1900
 Military special area = 2000
 Agricultural special area = 2100
 Sports special area = 2200
 Health and social special area = 2300
 Clinic area = 23,000
 Golf area = 2400
 Cultural special area = 2500
 Tourism special area = 2600
 Office and administration special area = 2700
 Judicial special area = 2720
 University research special area = 2800
 Fair special area = 2900
 Other special use area = 9999

(continued)

Table 1 (continued)

| |
|---|
| XP_Enumerationen: GeneralLandUseType (AllgArtDerBaulNutzung) |
| Residential building area = 1000 Mixed building area = 2000 Commercial building area = 3000 Special building area = 4000 Other building area = 9999 |

The feature classes `BP_FestsetzungenBaugebiet` (`BP_BuildingAreaDeterminations`) and `BP_ZusätzlicheFestsetzungen` (`BP_AdditionalDeterminations`) are crucial and model the building's dimensional constructability regulations.

Table 2 presents the attributes of the XPlanung class `BP_FestsetzungenBaugebiet` (`BP_BuildingAreaDeterminations`) in more detail. `BP_BaugebietTeilFläche` (`BP_PartialBuildingAreaSurface`) models development plan areas with consistent building types as commercial buildings, and residential building areas. `BP_ÜberbaubareGrundstücksFläche` (`BP_BuildableArea`) represents the area of the land parcel that can be built on. These entities inherit the properties of the classes `BP_GestaltungBaugebiet` (`BP_DesignBuildingArea`), `BP_Flächenschlussobjekt` (`BP_ClosureAreaObject`), `BP_ZusätzlicheFestsetzungen` (`BP_Additional-Determinations`), and `BP_FestsetzungenBaugebiet` (`BP_BuildingAreaDeterminations`).

The feature `BP_UeberbaubareGrundstueckeFlaeche` (`BP_Buildable Area`) is the essential overlay XPlanung class that presents the building structural area and dimensional regulations. It contains specifications regarding the building alignment (`BP_BauLinie`) and the building boundary (`BP_BauGrenze`). These entities are pivotal elements of the development plan. In addition, the building alignment and the building boundary restricts the structure of the building according to Paragraph 1 No. 2 BauGB, §22–23 BauNVO. As illustrated in Fig. 3, the building site must not exceed the building boundary area (`BauGrenze`) and the outer building surfaces must be aligned with the imposed building alignment (`BauLinie`). Consequently, the CityGML features `GroundSurface` as well as `OuterFloorSurface` must cope with the Building alignment and the building boundary. In addition, the CityGML `GroundSurface` is restricted by the maximum and minimum width of the building plot `Bmax` and `Bmin`.

5.3 XPlanung Restrictions on the Building Underground Components

The CityGML feature `Building` is subject to constraints regarding the maximum and minimum number of underground floors (`ZUmax` and `ZUmin`), and if applicable the

Table 2 Relation between attributes of the XPlanung class BP_FestsetzungenBaugebiet (BP_-BuildingAreaDeterminations) and CityGML classes, the attributes define constraints for instances of the CityGML classes

| Attribute | Description | Concerned CityGML feature |
|------------------|---|---------------------------|
| BM | Maximum allowed construction mass | Building |
| BM_Ausn | Exceptionally maximum allowed construction mass | Building |
| Bmax | Maximum width of building plots | GroundSurface |
| Bmin | Minimum width of building plots | GroundSurface |
| BMZ | Maximum permissible structural mass | BuildingPart |
| BMZ_Ausn | Exceptionally maximum permissible structural mass | Building |
| Fmax | Maximum size (area) of a building plot | GroundSurface |
| Fmin | Minimum size (area) of a building plot | GroundSurface |
| GF | Maximum allowed floor area | FloorSurface |
| GF_Ausn | Exceptionally maximum allowed floor area | FloorSurface |
| GFmax | Maximum allowable floor area for an area release | FloorSurface |
| GFmin | Minimum allowable floor area for an area release | FloorSurface |
| GFZ | Floors/plot surface area ratio | FloorSurface |
| GFZ_Ausn | Maximum Floors/plot surface area ratio | FloorSurface |
| GFZmax | Maximum allowable Floors/plot surface area ratio | FloorSurface |
| GFZmin | Minimum allowable Floors/plot surface area ratio | FloorSurface |
| GR | Building footprint area | FloorSurface |
| GR_Ausn | Maximum building footprint area as an exception | FloorSurface |
| GRmax | Maximum building footprint area | FloorSurface |
| GRmin | Minimum building footprint area | FloorSurface |
| GRZ | Percentage of the building footprint over the size of the building plot | FloorSurface |
| GRZ_Ausn | Exceptionally maximum permissible GRZ | FloorSurface |
| GRZmax | Maximum percentage of the building footprint over the size of the building plot | FloorSurface |
| GRZmin | Minimum percentage of the building footprint over the size of the building plot | FloorSurface |
| MaxZahlWohnungen | Maximum number of apartments in residential buildings | BuildingUnit |

(continued)

Table 2 (continued)

| Attribute | Description | Concerned CityGML feature |
|--------------------|--|---------------------------|
| MinZahlWohneinheit | Minimum number of apartments in residential buildings | BuildingUnit |
| Tmax | Maximum depth of building plots | WallSurface |
| Tmin | Minimum depth of building plots | WallSurface |
| Z | Maximum number of full floors above ground | BuildingUnit |
| Z_Ausn | Maximum number of full floors above ground as an exception | BuildingUnit |
| Z_Dach | Maximum number of additional permitted floors that are also full stories | BuildingUnit |
| Z_Staffel | Maximum number of full floors set back above ground as stacked floors | BuildingUnit |
| Zmax | Maximum allowable number of storeys above ground | BuildingUnit |
| Zmin | Minimum permissible number of full stories above ground | BuildingUnit |
| ZU | Maximum allowed number of underground storeys | BuildingUnit |
| ZU_Aus | Exceptionally maximum permissible number of floors below ground level | BuildingUnit |
| ZUmax | Maximum allowable number of underground storeys | BuildingUnit |
| ZUmin | Minimum permissible number of floors below ground level | BuildingUnit |
| ZUzwingend | Mandatory number of underground floors | BuildingUnit |
| Zzwingend | Mandatory number of full stories above ground | BuildingUnit |

mandatory number of underground ZUzwingend. Furthermore, the maximum and minimum building depth must also be considered referring to the attributes Tmax and Tmin.

5.4 XPlanung Restrictions on the Building Overground Components

A CityGML building can be subdivided functionally into building units and storeys, as well as structurally into Building constructive elements. These elements are subject to dimensional regulatory conditions. Thus, a link to the XPlanung classes BP_

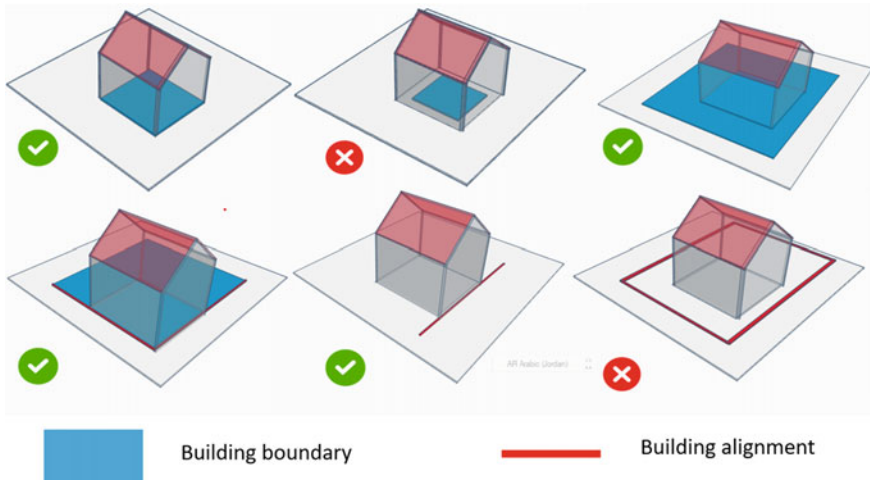


Fig. 3 Building alignment (BauLinie) and Building boundary (BauGrenze) as building site constraints

FestsetzungenBaugebiet (BP_BuildingAreaDeterminations) and BP_Zusätzliche-Festsetzungen (BP_AdditionalDeterminations) must be conceived. This is illustrated in Table 2 as well as Fig. 4. The building GroundSurface must respect conditions related to the maximum and minimum area size of the building plot (Fmax and Fmin). The CityGML Building attribute StoreysAboveGround as well as the feature class Storey are subject to limitations regarding the maximum and minimum number of apartments (MaxZahlWohnungen and MinZahlWohneinheit respectively) and maximum and minimum permissible number of floors above ground (Zmax and Zmin), as well as the mandatory aboveground floors (Zzwingend). The CityGML constructive features FloorSurface and GroundSurface objects are subject to regulatory limitations regarding the maximum and minimum allowed floor area (Gfmin and Gfmax), the maximum and minimum floor area ratio (GFZmax and GFZmin), the maximum and minimum portion of the building plot that can contain buildings (GRmin and GRmax), and the maximum and minimum percentage of the building areas over the size of the building plot (GRZmax and GRZmin). These different area limitations are illustrated in more detail in Fig. 4. Additionally, the building alignment (BauLinie) and building boundary (BauGrenze) attribute elevation (Höhenangabe) specifies the maximum height of the building. Thus, this will restrict the CityGML attributes height and storeyHeightsAboveGround of the Building class and also its geometry.

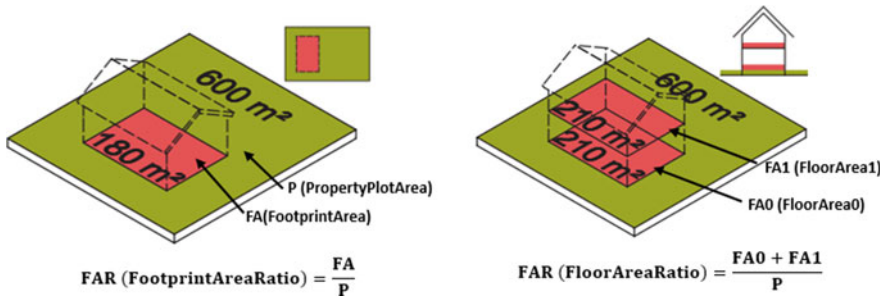


Fig. 4 BauNVO building area constraints [(XLeitstelle: Was ist XPlanung 2023), edited]

5.5 XPlanung Restrictions on the Building Roof

The feature BP_GestaltungBaugebiet (BP_DesignBuildingArea) models some of the building design characteristics as the ridge direction and the roof design parameters (BP_Dachgestaltung) (BP_RoofDesign). These attributes constrict the CityGML constructive object RoofSurface as well as the building class attribute roofType. These limitations as well as the above mentioned regulations are illustrated in Figs. 5 and 6.

6 Discussion and Conclusion

This paper aims to provide a schema-level integration of land planning regulatory information modeled in the German data model standard XPlanung and the 3D city model data standard CityGML as a base for constraining the generation of synthetic 3D city models in urban simulations. This integration is conducted through an investigation process to answer the pivotal research question: “What XPlanung objects constrain semantically and dimensionally a CityGML Building entity?”.

Our integration process starts by investigating land use areas where buildings are allowed to be built on. The XPlanung feature BP_PartialBuildingAreaSurface (BaugebietTeilFläche) incorporates attributes that specify the landscape zoning allocations following the land use plan of a municipality (Flächennutzungsplan). This can be conceptually linked to the CityGML LandUse class, which should be linked to the Building class. This harmonization makes sure that a CityGML building is legislatively allowed in the parcel area. In addition, the building plot is constrained by regulations related to the building alignment (BauLinie) and the building boundary (BauGrenze); these features include the maximum allowed height information as well as limitations on the plot area. The building’s underground and above-ground entities are subject to regulations relevant to the maximum and minimum over-ground and under-ground FloorSurfaces and buildingUnits. The various floor area

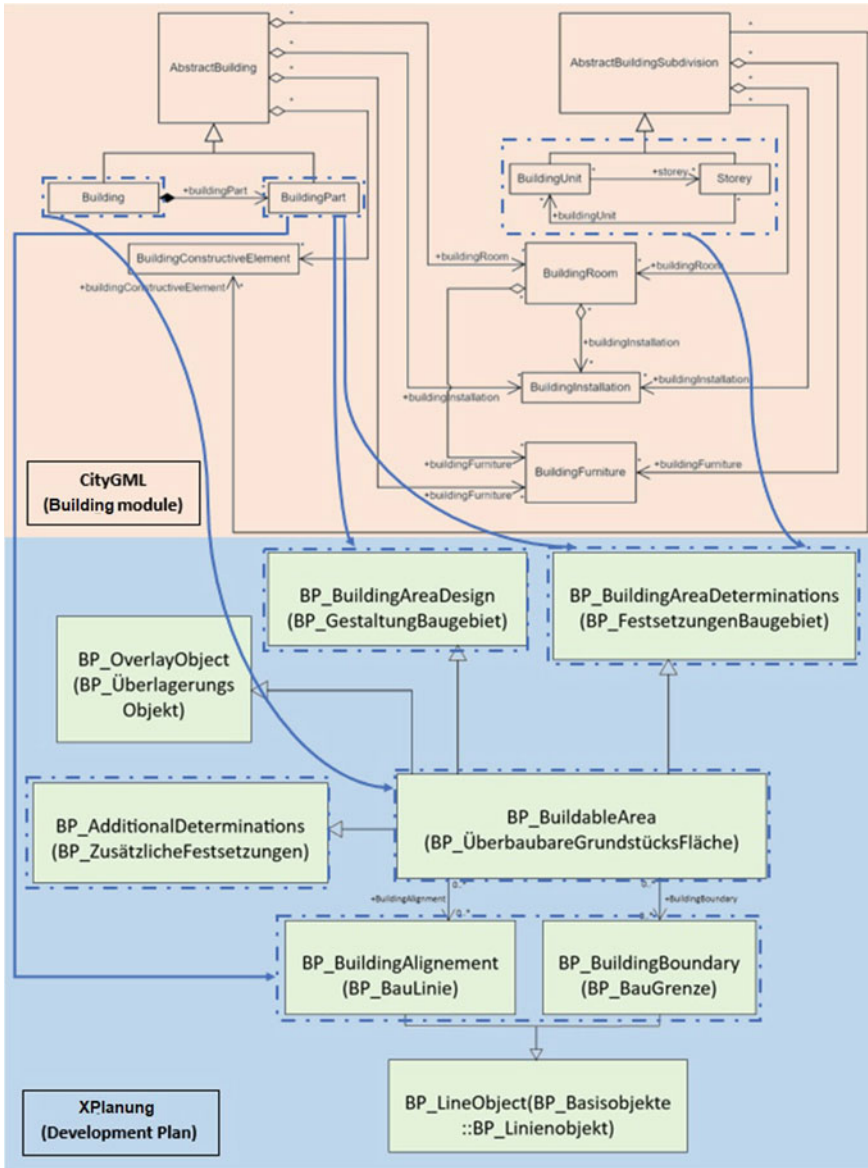


Fig. 5 XPlanung restrictions of classes from the CityGML building module. The blue arrows demonstrate the association between the XPlanung regulations and the corresponding classes in CityGML

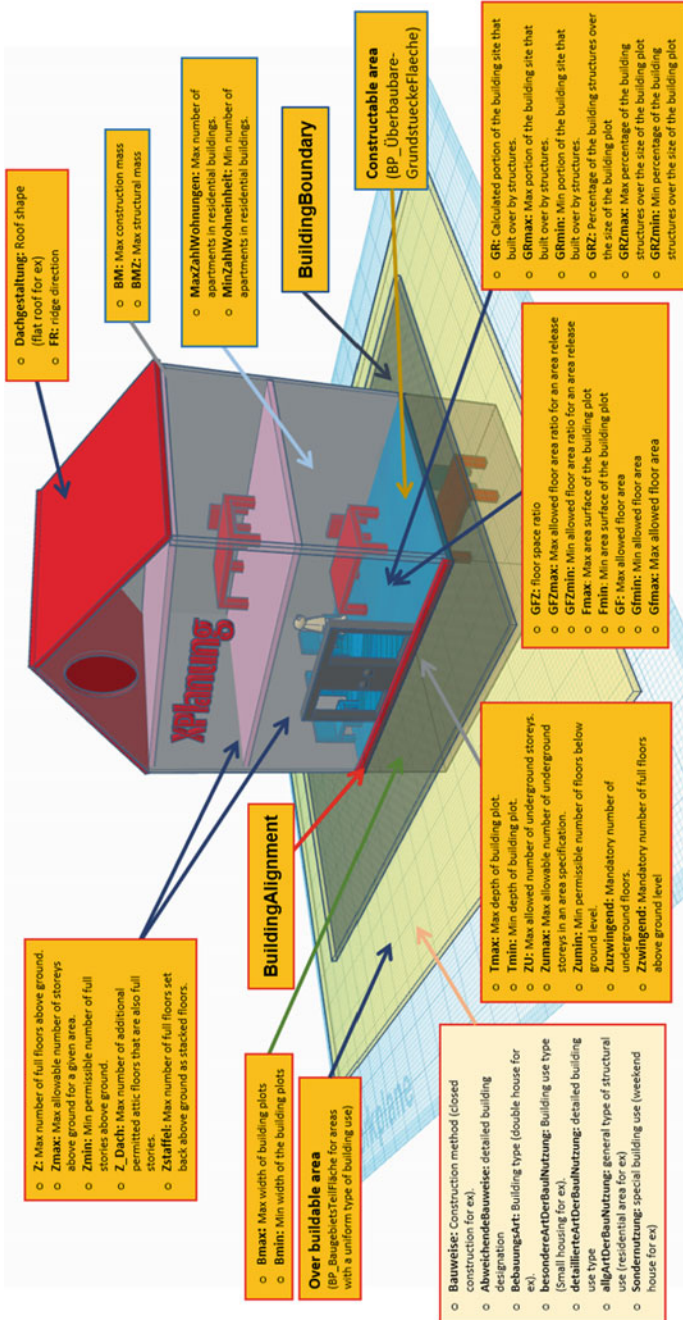


Fig. 6 Dimensional and constructability regulations regarding the building element

constraints that are outlined in the BauNVO legal document are attributed to the XPlanung class BP_BuildingAreaDeterminations (BP_FestsetzungenBaugebiet). Hence, these features will constrain the CityGML constructive FloorSpace elements. The CityGML RoofSurface class object is not neglected within XPlanung and subject to constraints related to the roof shape and the ridge direction. Such integration of landscape regulatory knowledge with 3D city data model ensures the real-life applicability of any urban simulation workflow. Making sure that the synthetic building model generated as part of an urban simulation respects the binding land use regulation enhances the importance of using semantic 3D city model for urban planning use cases.

In addition to the building example, both standards can provide support for other aspects, such as the city's streets and green space area. This integration framework enhances the semantic and topological richness of CityGML 3D models with constructability regulation knowledge provided in the German standard XPlanung. Accordingly, synthetic 3D city models generated by urban simulations will incorporate urban regulations limitations which gives them more realism. Urban simulation results will be in accordance with the urban applicable land use regulations. This includes energetic simulations and solar irradiation potential estimation, which are highly impacted by the roof shape, the height of the building, as well as the building orientation. The latter building features are legally binding and cannot be freely designed. Urban growth simulations must cope with the land use limitations, the maximum and minimum floor space and number of storeys etc. Synthetic 3D city models generated for proposing smart and optimized cityscape configurations must follow the land use and the development plan guidelines. If not, transforming them into real-life projects remains doubtful. Various use cases can benefit from CityGML-XPlanung harmonization, the proposed schema-level integration can be used to study the impact of the development and land use plans on different factors that have a spatial context. This covers for instance, the impact of building ridge direction and the roof shape on the energy consumption as well as the influence of the adopted building alignment (BauLinie) and building boundary (BauGrenze) on the traffic areas and the indoor perceived noise.

Future work will build on this mapping and focus on extending it with Object Constraint Language (OCL) rules to ensure that any CityGML entity generated by an urban simulation respects its relevant urban constraints. This will lead to the development of a 3D city logical model that includes urban regulation knowledge, enabling the generation of synthetic 3D city models for proposing smart and optimized cityscape configurations that follow land use and development plan guidelines. Based on an implementation for the use case urban expansion, we intend to evaluate the conceptual mapping and demonstrate the benefits of our concept.

Acknowledgements We thank the City of Munich for the cooperation in the Connected Urban Twins (CUT) project funded by the Federal Ministry for Housing, Urban Development and Building of Germany (BMWSB).

References

- Beil C, Kendir M, Ruhdorfer R, Kolbe TH (2022) Dynamic and web-based 4D visualization of streetspace activities derived from traffic simulations and semantic 3D city models. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci X-4/W2-2022*:29–36
- Benner J (2016) XPlanung Struktur und Konzepte. <https://xleitstelle.de/downloads/xplanung/releases/XPlanung%20Version%205.4/Struktur%20und%20Konzepte.pdf>. Accessed 2 June 2023
- Benner J, Geiger A, Häfele KH (2010) Concept for building licensing based on standardized 3d geo information. *Int Arch Photogramm Remote Sens Spat Inf Sci XXXVIII-4/W15*:9–12
- Biljecki F, Stoter J, Ledoux H, Zlatanova S, Çöltekin A (2015) Applications of 3D city models: state of the art review. *ISPRS Int J Geo Inf* 4(4):2842–2889
- Biljecki F, Ledoux H, Stoter J (2016) Generation of multi-LOD 3D city models in CityGML with the procedural modelling engine Random3Dcity. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci IV-4/W1*:51–59
- Brasebin M, Perret J, Mustière S, Weber C (2016) A generic model to exploit urban regulation knowledge. *ISPRS Int J Geo Inf* 5(2):14
- Brasebin M, Mustière S, Perret S, Weber C (2014) Simuler les évolutions urbaines à l'aide de données géographiques urbaines. 3D. *Revue internationale de la géomatique* 2:159–180
- Brasebin M, Perret J, Reuillon R (2017) Stochastic buildings generation to assist in the design of right to build plans. In: *Advances in 3D geoinformation*, pp 373–384
- Covadis (2012) Standard de Données COVADIS plan local D'urbanisme—Plan D'occupation des Sols PLU et POS—Version 2; Ministère de l'Écologie, du Développement durable et de l'Énergie, Paris, France
- Grelot JP (1994) Cartography in the GIS age. *Cartogr J* 31(1):56–60
- Harter H, Willenborg B, Lang W, Kolbe TH (2020) Life cycle assessment of technical building services of large residential building stocks using semantic 3D city models. *ISPRS Ann Photogramm.Remote Sens Spat Inf Sci VI-4/W1-2020*:85–92
- Hbeich E, Roxin AM, Nicolas BUS (2019) Aligning BIM and GIS—CityGML and PLU to achieve compliance checking model. In: *Proceedings of the 18th international conference on informatics in economy*, pp 165–170
- He S, Perret J, Brasebin M, Brédif M (2015) A stochastic method for the generation of optimized building layouts respecting urban regulations. In: Harvey F, Leung Y (eds) *Advances in spatial data handling and analysis. Advances in geographic information science*. Springer, Cham, pp 256–288
- Hijazi I, Ebert V, Donaubaue A, Kolbe TH (2018) Combining urban metabolism methods and semantic 3D city models. *Int Arch Photogram Remote Sens Spat Inf Sci XLII-4/W10*:63–70
- Hijazi I, Donaubaue A, Hamm A, Falkenstein A, Kolbe TH (2022) Urban growth simulation using urban dynamics and CityGML: a use case from the City of Munich. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci X-4/W2-2022*:97–104
- INSPIRE (2012) D2.8.III.4 Data specification on land use—draft guidelines; INSPIRE Thematic Working Group Land Use, European Union
- Kaden R, Kolbe TH (2014) Simulation-based total energy demand estimation of buildings using semantic 3D city models. *Int J 3D Inf Model* 3(2):35–53
- Kolbe TH (2009) Representing and exchanging 3D city models with CityGML. In: Lee J, Zlatanova S (eds) *3D geo-information sciences. Lecture notes in geoinformation and cartography*. Springer, Berlin, Heidelberg
- Kutzner T, Chaturvedi K, Kolbe TH (2020) CityGML 3.0: new functions open up new applications. *PFG J Photogramm Remote Sens Geoinform Sci* 88(1):43–61
- Müller P, Wonka P, Haegler S, Ulmer A, Van Gool L (2006) Procedural modeling of buildings. In: *ACM SIGGRAPH 2006 papers*, pp 614–623
- Ross L (2010) Virtual 3D city models in urban land management: technologies and applications. Doctoral dissertation, Fakultät VI—Planen Bauen Umwelt, Techn. Univ. Berlin

- Stoter J, De Kluijver H, Kurakula V (2008) 3D noise mapping in urban areas. *Int J Geogr Inf Sci* 22(8):907–924
- Vandenbroucke D, Biliouris D (2010) Spatial data infrastructures in France: state of play. <https://geoforum.pl/upload/files/pliki/raport.pdf>. Accessed 2 June 2023
- Vandenbroucke D, Biliouris D (2011) Spatial data infrastructures in France: state of play. <http://inspire.ec.europa.eu/reports/stateofplay2011/rcr11FRv102.pdf>. Accessed 2 June 2023
- Xleitstelle (2023) Was ist XPlanung?. https://xleitstelle.de/XPlanung/ueber_XPlanung. Accessed 2 June 2023
- Zeitner R (2019) Baunutzungsverordnung (BauNVO). In: Zeitner R, Marchionini M, Neumann G, Irmscher H (eds) *Flächenmanagement in der Immobilienwirtschaft*. Springer Vieweg, Berlin, Heidelberg

An Alternative Raw Data Acquisition Approach for Reconstruction of LOD3 Models



Florian Frank, Ludwig Hoegner, Peter Buckel, and Kris Dalm

Abstract Visual, autonomous, object-based outdoor vehicle localization premise detailed object-based maps. Semantic-rich and qualitative Level of detail 3 (LOD3) building models fulfill these requirements, but only a few real-world city models are available. These models are mainly reconstructed by LiDAR, have a cost benefit, scalability and reconstruction-complexity problem. Challenging the issues, we propose an alternative data acquisition approach for LOD3 model reconstruction primarily based on images. Thus, we created a moveable handcart mounted height-adjustable, high-precision gimbal. The gimbal enables 360° poses of camera, LiDAR rangefinder, inertial measuring unit. Additionally, GNSS RTK is used for absolute positioning. Our system is capable to record data and ground truth data for system validation in one step. The paper is about the system design, data processing and validation of the proposed reconstruction approach. Resulting real-world reconstruction accuracies are in the millimeter to low centimeter range. So, our system is compared in discussion with competing systems.

Keywords SfM · Camera · Raw data acquisition · LOD3 reconstruction · MVS · LiDAR · MLS · GNSS RTK

This article was selected based on the results of a double-blind review of the full paper.

F. Frank (✉)

Institute for Continuing Education, Knowledge and Technology Transfer,
88045 Friedrichshafen, Germany
e-mail: frank@iwt-bodensee.de

L. Hoegner

Department of Geoinformatics, University of Applied Science Munich, 80333 Munich, Germany

P. Buckel

Baden-Wuerttemberg Cooperative State University Ravensburg (DHBW),
88045 Friedrichshafen, Germany

K. Dalm

Ostbayerische Technische Hochschule Amberg-Weiden, University of Applied Sciences,
92637 Weiden, Germany

1 Introduction

Semantic-rich and qualitative 3D building models could widely be used in urban planning, simulation of automated driving, localization, metaverse, building accessibility for disabled. 3D building models are a cross-sectional technology for further applications. The cross-sectional technology contributes to people and the environment by increasing flexibility, efficiency, sustainability and safety. For building up applications, the semantic description of the building outer shell is necessary. This corresponds to Gerhard Groeger et al. to Level of Detail 3 (LOD3) (Groeger et al. 2008). The research of real-world LOD3 reconstructions that have been already performed revealed that the number is small (see Related work). In addition, companies offer LOD3 building reconstructions (Voxelgrid 2023). In the reconstruction process, the ambivalence of data acquisition and processing are the challenging issues in scalability, cost-benefit and automation level. Most approaches have used raw data from existing data acquisition systems. The acquisition systems are stationary, mobile or flying and mainly use LiDAR, photogrammetry, inertial measuring units (IMU), Global Navigation Satellite System (GNSS) and Real-Time Kinematics (RTK) as sensors. A gap has been identified in acquisition systems for use in city areas and detailed mapping. The problem is that most of the large-scale mobile mapping systems are mounted on cars. Thus, it is difficult to take detailed images while driving or accessing restricted areas. Alternatively, drones, trolley- and backpack-based systems are used to capture details. However, flying in city centers is often only allowed with exceptional permits. Backpacking systems with LiDARs and panoramic cameras are mainly designed for indoor use. They have a limited line of sight due to their low LiDAR position above ground. Based on these circumstances, the paper presents an alternative approach for data acquisition. In this approach, a height-adjustable, high-precision gimbal is mounted on a handcart. The system combines stationary and mobile data acquisition and relies primarily on photogrammetric reconstruction assisted by LiDAR, IMU and GNSS RTK data. The gimbal is equipped with two cameras and different lenses, resulting in a pan tilt zoom system (PTZ) for data acquisition.

1.1 Related Work

A. Gueren et al. propose different approaches for LOD3 modelling from multiple data sources (Gueren et al. 2020). The approach considers raw data acquisition by light detection and ranging (LiDAR) and structure from motion/multi-view stereo (SfM/MVS) photogrammetry technology. Furthermore, the paper described the use of raw data mobile ground and flying vehicles. Thus, the paper covers a wide variety of different raw data possibilities.

Wyskocki et al. (2022) provided an open source semantic city model overview of real and artificial CityGML reconstructions in all detail levels. According to this

overview, there are three LOD3 city models from Espoo, London and Ingolstadt available. Based on the three LOD3 city models inversely, the processing and raw data acquisition methods are explained in detail:

Espoo, Finland, was reconstructed in LOD3. The LOD3 data does not provide semantic information like windows, doors etc. but rather marks texturing (City of Espoo 2023). The reconstruction and data acquisition process are not described.

AccuCities Ltd. provides some reconstructed city models of the UK in LOD3 quality (AccuCities Ltd 2023). They describe the data acquisition by aerial imaging and a LOD3 reconstruction accuracy of 15 cm. The data acquisition is done by flying parallel lines above the city. Due to aerial images the provided LOD3 models are only considering details of the rooftop structure.

In the governmental founded projects SAVE and SAVENoW, in Ingolstadt in Germany, the project partners reconstructed one street in LOD3 (Schwab and Wysocki 2021). The LOD3 model reconstruction is based on point clouds recorded by a mobile laser scanner (MLS) system from 3D Mapping Solutions GmbH. The MLS is driven on road through the area and generates a point cloud with an accuracy of 1–3 cm for reconstruction. The LOD3 reconstructions shows that the ground-based view of MLS makes it hard to reconstruct rooftop structures, rain gutters and chimneys.

Voxelgrid GmbH offers LOD3 reconstructions for single buildings (Voxelgrid 2023). The data acquisition is done drone flights with LiDAR, multispectral cameras and structure for motion (SfM) photogrammetry. The flying strategy is to record the building surfaces in parallel equidistant lines. The façade elements are distinguished by multispectral reflectance. Information regarding the accuracy is not provided. The data acquisition requires governmental and local residents permits.

NavVis GmbH provides LOD3 reconstruction for single buildings (NavVis 2023). They developed a backpack with LiDAR and a panorama camera system for data acquisition. During the acquisition process a person has to walk around the building. Their white paper provides a validation of their sensor setup based on an 20 m high façade. The system achieves an average accuracy less than 12.2 mm.

All related data collection methods differ in combination of different sensors, mechanical integration and acquisition strategy. The proposed acquisition system combines different mentioned approaches and tries to compensate disadvantages.

1.2 Objectives

The general challenges of raw data acquisition for LOD3 reconstruction can be summarized as follows:

1. The different architectures of the data acquisition systems in sensors, mechanics, and strategy limits their field of application. Thus, all real-world LOD3 data acquisition approaches are limited in scalability for large, semantical enriched LOD3 reconstructions.

2. The counterpart of scalability is the cost-benefit of the LOD3 reconstruction. These costs are largely caused by data collection and processing expenses. A solution is the efficiency improvement of the methodology.
3. Permissions for public data acquisition are in most countries challenging legislation and privacy regulations. These regulations make it even difficult to collect data in the appropriate quantity and quality.

To overcome these challenges, we propose a novel human-based mapping approach. This approach is mainly based on photogrammetry and is considering zoom for detailed pictures. Furthermore, the capturing system is mounted on 360° and height-adjustable high precision gimbal on a ground-based handcart. Thus, the system combines stationary and mobile acquisition. The handcart design allows accessibility to cities and is regulation compliant because the point of view is like that of a truck driver. The acquisition strategy is considering LOD2 models and camera-behavior model for accuracy. Additionally, the system can be used for testing future object-based, visual localization approaches.

2 Handcart Data Acquisition Approach

The starting point was the research about localization for autonomous ground vehicles. So, the idea came up to do human like localization via visual photogrammetry on object-level. The idea of the acquisition system came from the need that only a few LOD3 models were available. The models were reconstructed by point clouds. In SAVE model in comparison with real-world taken images, had many differences. So, the models were not useful for photogrammetry object-based localization approaches. Thus, the decision was made to create a photogrammetry-based mapping approach.

2.1 LOD3 Reconstruction Process Pipeline Overview

The LOD3 reconstruction process focusses on a photogrammetric SfM approach. Figure 1 shows: Data acquisition, instance segmentation, 3D model reconstruction, standardized output format.

The Data acquisition includes the recording of images with additional ground truth data for reconstruction validation. In the second step, the images are decomposed via instance segmentation to façade elements. The object classes of façade elements are inspired by CityGML (Groeger et al. 2008) standard. From the segmented façade elements the contours are derived. Next, contours from different points of view are matched. Matched contour points are three-dimensionally reconstructed via SfM triangulation to 3D model. A camera behavior model and the LOD2 data are used as

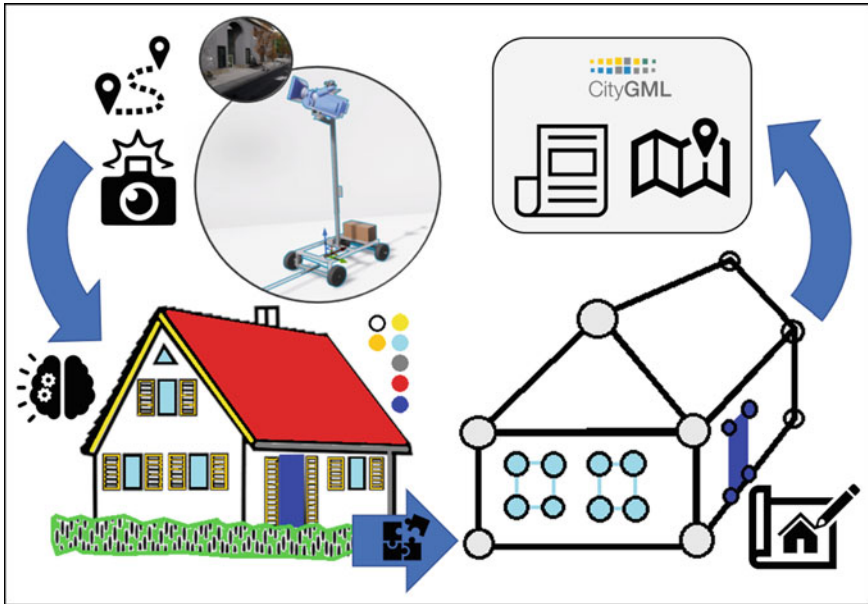


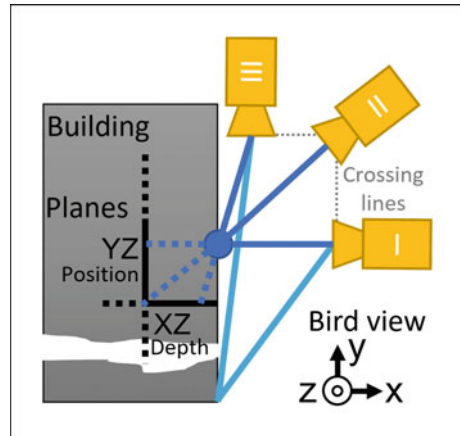
Fig. 1 Photogrammetry SfM LOD3 reconstruction process

support. Finally, the model is transformed to a compliant standard, like CityGML. Due to the complexity of the presented process pipeline, this paper is only about the data acquisition.

2.2 Data Acquisition Strategy

The data acquisition strategy is distinguished in stationary and mobile acquisition. Figure 2 shows the stationary data acquisition in birds view. The stationary acquisition strategy is also splitted into orthogonal XZ and YZ planar capturing. YZ is for façade element positions on surface and XZ for depth estimation. In overlapping the XZ and YZ three-dimensional XYZ position accuracies should increase. So images for YZ plane are taken from the house opposite (I, II). Façade elements depth capturing on XZ plane is done in parallel view direction to the façade plane (II, III). LOD2 models will be used to calculate capturing positions. During capturing, the gimbal is moved to defined positions and 6DoF of gimbal and vehicle position (IMU, motor), date, time, GNSS RTK are logged. The gimbal is additionally equipped with a LiDAR rangefinder for tachymetic measurements. These points are used as ground truth data for photogrammetry reconstruction validation. The rangefinder can be controlled in closed loop by using camera and gimbal to position its laser point. By determining salient points on the façade, a relative 6DoF estimation via LiDAR can be done.

Fig. 2 Static data acquisition strategy with orthogonal planar XZ and YZ approach



The mobile acquisition process is distinguished in low and fast speed. At walk speed the system operates like in stationary acquisition. At faster speed, the gimbal view direction is positioned to front-side of the handcart. The data acquisition is done iteratively by driving along the buildings. During each pass, the yaw, and pitch angles of the gimbal are systematically changed. Ground truth data acquisition like rangefinder measurements are not further considered.

2.3 System Design and Building

This subsection describes only the opto-electromechanical system design and setup. System accuracy and precision are in results Sect. 3. Figure 3 shows the handcart functionality of axes and sensors and Fig. 4 shows the prototype. The handcart explanation of design and building is correlated with numbers to Figs. 3 and 4. The handcart is built by off the shelf components for a cost benefit setup. Its skeleton is build of aluminum 20 and 40 mm profiles and 3D-printed parts (1). Further, the vehicle is on four wheels and is pulled via a handle bar. This design enables its application in rough terrain, on the road and on footpaths. The elevator (2) on top is rest on big real elevators with a counterweight. The elevator can move in range of 1.2 m from 1 m above the ground. The moving part of the elevator is equipped with linear axis for precision. The elevator itself is a 3D printed 360° gimbal mounted (3). The gimbal is limited by wiring to a range of yaw 340° and pitch 100°. This construction enables a five DoF acquisition system (roll is missing). All moveable axis are moved by 1.8° stepper motors. Every axis is reduced by a fix ratio. Each motor is controlled via a separated CAN gateway with TMC2130 stepper driver (TRINAMIC Motion Control GmbH & Co. KG 2023) (4). Movable parts are homed by one site end switches to a reference position. The pitch axis is equipped with two Sony IMX 477 (Song 2018) raspberry pi cameras, Bosch BNO055 (Bosch Sensortec 2021) IMU, a green laser

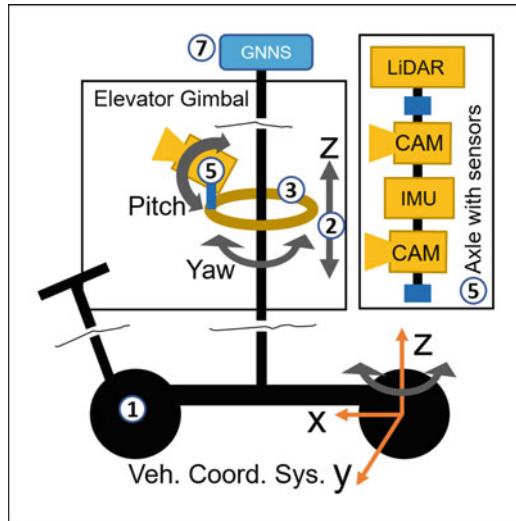


Fig. 3 Functional representation of the handcart including sensors DoF

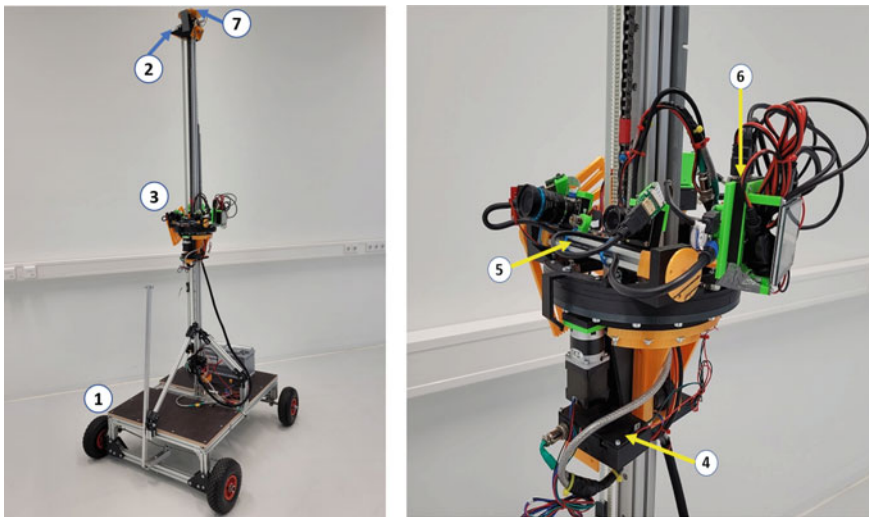


Fig. 4 Handcart prototype

diode and Leica Distro D510 (Leica Geosystems 2023) rangefinder (5). Beside, the system controller NVIDIA Jetson Xavier NX is mounted (6). The camera heads are connected via CSI-2 cable and the rangefinder via Bluetooth to the Jetson. The GNSS RTK ublox ZED-F9P (u bloxx 2023) module is mounted on the top of the elevator pole (7) and connected with USB. All sensors, DIOs, IMU, drivers are connected on CAN operated nodes. The nodes are binded with a CAN gateway to the Jetson. The system is powered by a 120 Ah 12V car battery. Material costs were around 4000 €.

2.4 3D Sensor Pose Reconstruction Methods

The 3D reconstruction process consists of two stages. The first stage is the absolute or relative 6DoF positioning of the vehicle cameras. Followed by the reconstruction of façade elements three-dimensional key point positions. Every stage cloud be proceed by different component combinations. The 6DoF poses are described by a cartesian vector \vec{v} and euler angles.

1. Absolute position determination with GNSS, IMU and gimbal data. Due to lack of IMU yaw accuracy, the yaw was reconstructed by camera with pole objects. The individual camera absolute focal point P_{focal} is calculated by GNSS average position $P_{pos,GNSS,avg}$ and offset vector \vec{v}_{gimbal} . The offset vector depends on motor positions, offsets and is anchored by rotation matrix R to absolute coordinates.

$$P_{pos,GNSS,avg} = \frac{\sum_{i=0}^n P_{i,GNSS}}{n} | P_{i,GNSS} \in \{fixedRTK\}$$

$$P_{focal} = P_{pos,GNSS,avg} + \vec{v}_{gimbal}(\alpha_{m1}, \gamma_{m2}, \alpha_{m3}) \times R(\Delta\alpha_{imu}, \Delta\beta_{imu}, \Delta\gamma_{cam})$$

The pose is calculated by gimbal positions and their IMU Δ angles to absolute coordinates in addition to the basic offsets:

$$\begin{pmatrix} \alpha_{abs} \\ \beta_{abs} \\ \gamma_{abs} \end{pmatrix} = \begin{pmatrix} \alpha_{m3} \\ 0 \\ \gamma_{m2} \end{pmatrix} + \begin{pmatrix} \Delta\alpha_{imu}(\alpha_{m3}, \gamma_{m2}) \\ \Delta\beta_{imu}(\alpha_{m3}, \gamma_{m2}) \\ \Delta\gamma_{cam}(\alpha_{m3}, \gamma_{m2}) \end{pmatrix} + \begin{pmatrix} \alpha_{offset} \\ \beta_{offset} \\ \gamma_{offset} \end{pmatrix}$$

Afterwards vector $\vec{v}_{focal, P_1 P_2}$ is calculated for SFM reconstruction:

$$\vec{v}_{focal, P_1 P_2} = P_{2, focal} - P_{1, focal}$$

2. Relative positioning and selective point measurements with LiDAR rangefinder for ground truth data acquisition. For 6DoF pose estimation, a minimum set of three, spatio-temporally constant points in the environment is necessary. The individual environment point P_{env} is described by gimbal spherical coordinates γ_{m2}, α_{m3} and laser measured distance d LiDAR relative position P_{LiDAR} is triangulated by environmental points $\sum_{i=0}^n P_{env}$. From environmental points the roll angle β_{env} is derived. The camera focal point P_{focal} is calculated by P_{LiDAR} and the pose corrected offset vector $\vec{v}_{cam,offset}$.

$$\begin{pmatrix} \alpha_{abs} \\ \beta_{abs} \\ \gamma_{abs} \end{pmatrix} = \begin{pmatrix} \alpha_{m3} \\ \beta_{env} \\ \gamma_{m2} \end{pmatrix}$$

$$P_{focal} = \vec{v}_{cam,offset}(\gamma_{m2}, \alpha_{m3}, \beta_{env}) + P_{LiDAR}$$

3. Relative 6DoF pose estimation by camera is done in four steps. Frist, a set of corresponding key points in two images are extracted. The the the fundamental matrix is calculated by key points with 8-point algorithm. The fundamental

matrix is enhanced by camera and translational matrix to essential matrix. From this, the pose is recovered by solving the perspective 'n' point problem. Afterwards the transition vector \vec{t} is scaled by length of $\vec{v}_{focal, P_1, P_2}$.

3 Experiments and Results

This section contains the selected components, configuration and its theoretical accuracy. The second subsection contains the calibration methods and determined real-world accuracies of the components. Third to fifth subsection contains the real-world test environment, test description and 3D reconstruction results of different system combinations. The last subsection is about the image annotation process and results.

3.1 Sensor and Motor Driver Setup—Theoretical System Accuracy

The cameras are setup for mono camera data acquisition with zoom. So LOD3 reconstruction is done by different positions and photogrammetry SfM. The two cameras are equipped with a 16 mm telephoto (10MP) and the left 6 mm wide angle (3MP). The image sensor pixel size is $1.55 \mu\text{m}$ (Sony 2018). A pixel of the sensor plane could be transferred via pinhole model to a parallel object plane (Sturm 2016). The 16 mm lens has a resolution of $96.88 \frac{\mu\text{m}}{\text{px}}$ in per meter distance from focal point to a parallel object plane. This ideal assumption ends at one cm in pixel diagonally on object plane at a distance of 72.99 m.

The motors are driven by 256 micro steps. By multiplying the reductions, lengths per step and step width the theoretical accuracies were calculated. The elevator achieves an accuracy of $7.752 \mu\text{m}$. The yaw axis achieves $0.253''$ and pitch achieves $2.195''$.

The Bosch BNO055 IMU (Bosch Sensortec 2021) is an integrated triaxial 16 Bit gyroscope, 14 Bit accelerometer and geomagnetic sensor. Thus, 14 to 16 Bit, the IMU theoretical archives an accuracy of $79.102''$ to $19.775''$. The magometer tolerance is $\max \pm 2.5^\circ$, accelerometer sensitivity tolerance is $\max \pm 4.0 \%$ and gyroscope sensitivity tolerance $\pm 3.0\%$. It is operated in data nine degree fusion mode for absolute euler angle orientation.

The GNSS RTK System consists of a ublox ZED-F9P (u bloxx 2023) rover and a NetwakeVision RoyalBase (2023). Both systems use Radio Technical Commission for Maritime Services (RTCM) 3.3 protocol for RTK correction. The fix point RTK achieves an accuracy of 14 mm.

3.2 Calibration Tests and Real Accuracy Determination

The handcart combines electromechanical and optical components for data acquisition. Thus, there are four separated calibration processes which result in overall system accuracy.

The height adjustable gimbal was calibrated by repeatability tests. These repeatability tests were executed stationary and individually for each axis. The test procedure was to move 100 times from every approaching direction to the same position. For measuring an image of the green laser dot on a 2 m away, orthogonal millimeter paper was taken. Figure 5 shows an example of position reaching. Afterward, based on the images, the standard deviations were calculated. Table 1 contains the gimbal accuracy results. The results are distinguished axis-individually in negative, positive approaching and the backlash between approaching directions. Negative and positive approaching accuracies are close together and much lower than the backlash. Finally, the strategy to pose the gimbal in high precision is a one way position approaching direction.

The component XYZ offsets are measured by meter rule in millimeter accuracy.

The BNO055 IMU is used during the whole capturing process for absolute euler angles determination. Due its tolerance, a average filter is applied to achieve higher accuracy. For determining the IMU roll and pitch accuracy the handcart was rolled or pitched by underlay. The real plumped vector was measured with a 1.2 m plump

Fig. 5 Picture of millimeter paper for accuracy determination

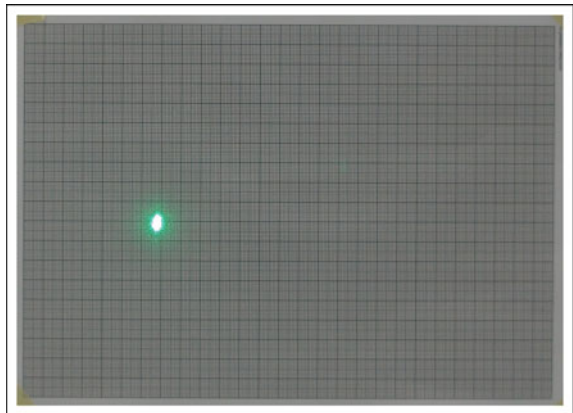


Table 1 Gimbal accuracy results of repeatability test

| Motor | 6DoF pose | Approach direction | | Backlash | Unit |
|-------|-----------|--------------------|--------|----------|--------|
| | | Neg | Pos | | |
| m1 | z | 0.313 | 0.476 | 2.106 | mm |
| m2 | Yaw | 18.004 | 13.516 | 396.441 | acrsec |
| m3 | Pitch | 19.565 | 20.866 | 238.515 | acrsec |

Table 2 Comparison of theoretical and practical component accuracies

| Componet | Type | Additional information | Accuracy | | Unit |
|----------|---------------------|------------------------|-------------|-----------|------------------|
| | | | Theoretical | Practical | |
| LiDAR | Disto D510 | | ±1.0 | ±2.0 | mm |
| GNSS | ZED-F9P | (RTK mode) | ±14; - | ± 14; 456 | mm (fix ; float) |
| Gimbal | TMC2130, 3D printed | Z | ± 8 | ± 476 | µm |
| | | Yaw | ± 0.253 | ±18 | arcsec |
| | | Pitch | ± 2.195 | ± 21 | arcsec |
| Camera | IMX477 | Center pixels | ± 19.982 | ± 19.702 | arcsec |
| IMU | BNO055 | 14 Bit acc. | ± 79.102 * | ± 3.6 ** | *arcsec; **deg |
| | | 14 Bit mag. | ± 79.102 * | ± 2.5 ** | *arcsec; **deg |
| | | 16 Bit gyro | ± 19.775 * | ± 3.6 ** | *arcsec ; **deg |

and meter rule. During testing the gimbal was tilted in yaw and pitch with negative approaching to 100 positions. For each test, afterward the IMU plumped vector was calculated with absolute gimbal positions. The calculated and measured plumped vectors differ less than 1 mm. According to this, the accuracy is $\leq 2.865'$. The yaw was tested by pulling the car around in 360° circles to the same position or tilting gimbals yaw. Pulling around ended in 33.125° difference. This may be caused by vibrations of no damping construction of the cart. But stationary, relative positioning with the gimbal yaw tilting achieved 0.688° accuracy.

The camera with the 10MP 16 mm lens is adjusted to a depth of 15 m and the one with 6 mm is adjusted to a depth of 7 m. Afterward the intrinsic calibration matrix *A* is determined via a chessboard by using openCV 2011. The calculated focal point offset was 16.227 mm so the resolution per pixel increased.

Leica certified the Disto D510 by ISO 16331-1 accuracy to ± 1.0 mm at favorable and ± 2.0 mm at unfavorable conditions.

Table 2 contains theoretical and practical component accuracy results. In comparison , excepting the camera, the accuracies differ between theoretical and practical.

3.3 Outdoor Test Environment

Outdoor testing should challenge camera and LiDAR. Dark, matt surfaces absorb light and uni colors make instance segmentation and edge detection more difficult. Furthermore, LiDAR is challenged by light energy absorption. Glass stress LiDAR and camera by light scattering effects. Building depth offers, like overhangs, are generating shades and disturb line of sight. This makes edge detection and segmentation

tasks difficult. Irregular positions of shutter systems make façade element interpretation complicated. Pitched roofs require greater distances for ground-based data acquisition. The RITZ building at campus Fallenbrunnen in Friedrichshafen, Germany fulfills most of the challenging tasks. Unless the the pitched roofs and dormers. The surrounded buildings have pitched roofs and dormers. Furthermore, ground truth data of is available in CAD, LOD2 and total station measurements.

3.4 Outdoor Test Description

The system was tested in stationary mode at the east side of the RITZ building. The handcart was positioned according the stationary acquisition strategy at three points. The test was about accuracy testing of the system as well as the possibility of façade element segmentation of captured images. Sunny to partly cloudy weather condition was chosen for uniform illumination conditions. Additionally, the building was tagged with markers for reproducibility of accuracy testing.

The acquisition process consists of data acquisition at the three positions. For validation the handcarts 6DoF positions were referenced via total station. During data acquisition the pitch and yaw were stepped equidistant in 5° . Thus, data of 352 poses at a stationary position was taken.

3.5 3D Sensor Pose Reconstruction Results

Results of the three 6DoF pose maximum deviations are shown in Table 3. The maximum deviations were referenced to total station measurements. Due to different positions of Camera and LiDAR only GNSS and Camera euler angels could be directly compared. The table is showing that the LiDAR pose estimation method had the best accuracy. Camera pose estimation method showed worst results in rotation angle reconstruction.

3.6 3D Façade Reconstruction Results

Façade point 3D reconstruction is similar to the sensor pose estimation. The façade points could be measured by LiDAR method or reconstructed by camera ray triangulation. Ray triangulation is based on previous 6DOF sensor pose estimation. So the sensor pose euler angles affect the 3D reconstruction. After testing, best results were archived by GNSS sensor pose estimation method. Point matching in the images was done manually to ensure exact matching. Table 4 shows the object reconstruction results. First and second row contain baseline orthogonal point reconstruction of XZ and YZ plane of the building. Third and fourth row show results of

Table 3 Maximum deviations results of pose estimation methods

| 6DoF pose | Absolute | Relative | | Unit |
|-----------|----------|----------|--------|--------|
| | GNSS | LiDAR | Camera | |
| x | 1 | 1 | 3 | mm |
| y | 1 | 1 | 5 | mm |
| z | 1 | 1 | 5 | mm |
| RMS xyz | 2 | 2 | 5 | mm |
| Alpha | 20.8 | – | 72.69 | arcsec |
| Beta | 355 | – | 355.71 | arcsec |
| Gamma | 19.7 | – | 74.13 | arcsec |

Table 4 Maximum deviations of three-dimensional object point reconstruction based on photogrammetry

| fp = 16 mm | Position pairs | $d_{stereo,GNSS}$ (m) | d_{min} (m) | d_{max} (m) | max. dev. (x ; y ; z) (mm) |
|-----------------|----------------|-----------------------|---------------|---------------|----------------------------|
| Points on YZ | I, II | 8.520 | 8.218 | 11.739 | 3; 2; 2 |
| Points on XZ | II, III | 6.172 | 8.032 | 11.791 | 2; 3; 2 |
| Building length | I, II, III | 6.505 | 48.850 | 55.024 | –; 64; – |
| Building height | I, II | 8.520 | 8.482 | 14.170 | –; –; 3 |

the building dimensions. The first column shows the reconstruction aim. Followed by position pairs and base distance $d_{stereo,GNSS}$. Additional minimal $d_{P,min}$ and maximum point distanced $d_{P,max}$ from positions are listed. Point reconstruction was done like proposed in the acquisition strategy. From position I and II, the parallel points on YZ-plane had an accuracy of 2 mm and a depth of 3 mm. Results of orthogonal XZ plane (II, III) for depth estimation match with YZ plane. For building length estimation, the combinations using all positions were made. The RITZ reference east side length is 54.9867 m, the reconstructed length was 55.051 m. The error of 64 mm is caused mainly due the calculation of the opposite south-east edge point by position I and III. The RITZ building visible rooftop edge is 10.802 m, the reconstructed was 10.799 m.

3.7 Results of Manual Image Annotation

Image annotation, as precursor of machine learning, is the process of labeling and segmenting objects. In segmentation, the contour is determined by human vision and perception. To follow the scientific approach, a set of possible contour and edge

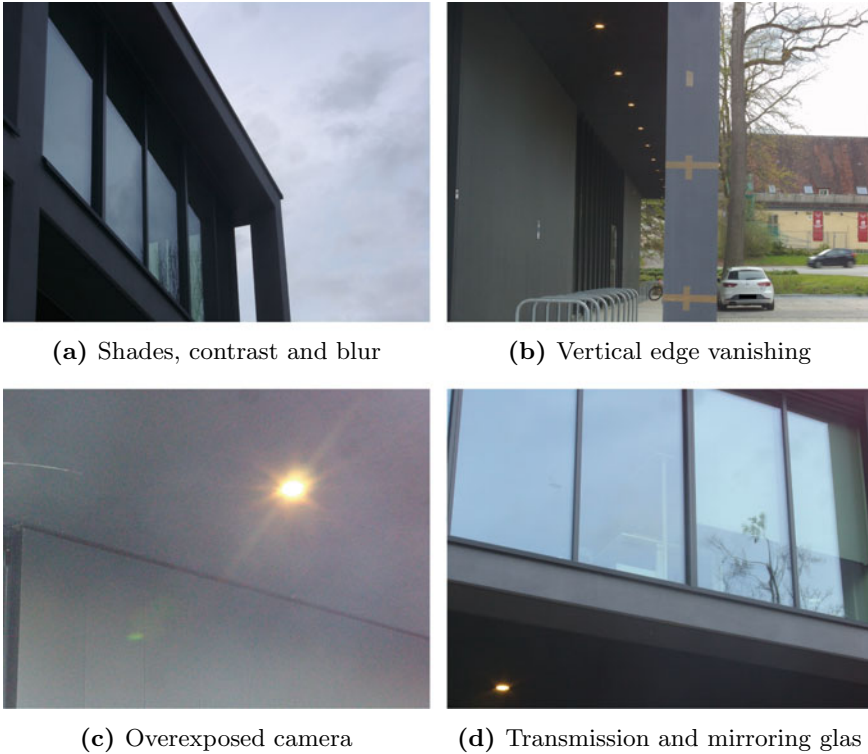


Fig. 6 Examples of annotation challenges

detection problems were proposed: Image quality, blur, illumination, contrast, line of sight, light scattering and contextual reference. An example subset of affected images is shown in Fig. 6. All images were taken during sunset. In Fig. 6a illumination and blur make detection of round sealing of the upper corner hard. Increasing distance shows in Fig. 6b the effect of edge vanishing on vertical façade elements. Figure 6c is an example of boarder-crossing the dynamic range of cameras, resulting in overexposure. Figure 6d shows light scattering effects in the transition from mirroring to looking through glas. This leads to object contextual reference challenges. In Summary, all these challenges can be examined by different filters, edge detection or pattern recognition. But the solution for most of these issues are images from other positions and poses. Additionally, environmental conditions should be considered in acquisition strategy.

4 Discussion

The achieved system results of 3D building façade reconstruction from previous section are discussed in relation to competitive system architectures. Discussion is divided into 3D Reconstruction and image annotation.

The handcart has a redundant system architecture for 3D reconstruction. The LiDAR subsystem can be used for relative positioning and measurements in closed loop mode with camera and gimbal. The Camera pose can be precisely controlled with the gimbal for detailed picture taking. Absolute 6DoF estimation via GNSS RTK and IMU needs to be supported by visual odometry. Good absolute position results were only achieved in fixed RTK mode. This makes the system depended of GNSS signal and RTK communication. Far depth reconstruction with stereo vision over 40m with a small distance baseline is not useful. But these can be solved by other handcart positions or LiDAR measurements. The handcart architecture makes reconstructions in millimeter to low centimeter range possible.

Based on the practical approved reconstruction, the system is compared to competing systems. For the comparison, Table 5 was created. As representatives for competing systems were selected from popular manufacturers. Due to comparable sensor setup of camera and LiDAR, hand-held and trolley-based systems can be represented by a backpack system (Pantoja-Rosero et al. 2022; Nüchter et al. 2015; Blaser et al. 2020). The compared backpack is the evolution of a former trolley-based system (NavVis 2021).

Data acquisition is distinguished in stationary or mobile. The backpack and the vehicle data acquisition are based on movement. Total station and the handcart have a 360° FoV for stationary acquisition. Further, the handcart is designed for both modes.

All systems use different localization methods but are in the same range. Unless the total stations with autarky precise point positioning (PPP) GNSS. The mobile mapping vehicle (MMV) and the handcart need an active connection to a reference station. Simultaneous localization and mapping (SLAM) of the backpack is a relative method. So, the system must be initial referenced to absolute position points.

All systems use cameras and LiDAR. Total station uses a deflection unit and handcart a gimbal 360° field of view (FoV). So, they have the same approach. The Backpack and MMV have fixed mounted vision systems with a defined. Their FoV is varied by moving. Thus, the accuracy of mobile systems and stationary systems differ. The handcart and total station are nearly in the same accuracy range of LiDAR. Backpack and MMV accuracy is lower because of rotational multi-line LiDAR.

The handcart is constructed for selective point reconstruction of façade elements by camera. The 16 mm lens is capable to keep up with the handcart LiDAR accuracy due to its camera angular resolution. The handcart has a camera closed loop LiDAR measurement. A camera comparison to the other system is not useful because necessary parameters are not available.

Times for data acquisition are only available from total station and handcart. Here, the time of the handcart is significantly reduced to the total station. This is caused

Table 5 Comparison of competing data acquisition systems

| | Total station (Zoller + Fröhlich GmbH 2023) | Backpack (NavVis 2021) | Mobile street mapping vehicle (Beil et al. 2021) | Handcart |
|---------------------------------------|---|------------------------------|--|---------------------------|
| Data acquisition | Static | Mobile | Mobile | m/s |
| Self-localization | GNSS,PPP | SLAM | MoSES | GNSS,RTK |
| Localization acc. | <1 m | 6 mm @ 500 m ² | <2 cm | <1 cm (static) |
| Self-calibration | x | – | N/A | x |
| LiDAR, angular acc. | 14.4'' | 0.1° (VLP16) | 0.1° (VLP64) | 20.8'' |
| LiDAR, distance acc. | 1 mm | 1–3 cm | 1–3 cm | 2 mm |
| Cam, res. @ lens | N/A | 4 × 20 MP @ 3.3 mm | N/A | 2 × 12 MP @ 16 mm |
| Cam, res. angular | N/A | N/A | N/A | 19.702'' |
| Closed loop measure | – | – | – | x |
| Time per scan | 48:57 min ** | N/A | N/A | 3:30–15 min* |
| Illumination dep. | ~ | ~ | N/A | x |
| Weather dep. | ~ | ~ | ~ | x |
| Data acquisition area restrictions | – | – | Only streets | Steep slopes, barriers |
| Point accuracy *camera, **LiDAR | 1.2 mm @ 10 m distance** | <14.3 mm @ h = 7.8–14 m** | 1–3 cm @dataset** | 3 mm @ 14 m* |
| Segmentation approach | N/A | N/A | Point cloud | Images |

Legend *camera; **LiDAR; x fully, ~ partly; – not at all, N/A not available

by the 12 MP camera to a $1.1 \frac{M.pt/s}{s}$ sampling rates. But the handcart needs data from various positions for reconstruction.

The handcart camera-based approach needs good weather and illumination conditions. The 12 Bit high dynamic rate (HDR) camera has problems with direct views in sunlight and dark shades. This is a disadvantage compared to the other systems. They use camera images for point cloud coloring. Also the LiDAR laser beams can be disturbed by rain and snowflakes. So, they are partially illumination and weather dependent. Systems' data acquisition area restrictions show that MMV can only be driven on roads. Total station and backpack have no restrictions, but their use in road traffic is difficult. Stairs and strong slopes are restricting areas for the handcart. With this, the handcart can operate in many urban areas, sidewalks and in road traffic as bicycle trailer.

All point cloud accuracies listed in comparison refer to different environmental data acquisitions of buildings. So, the data are in the same context but can not be directly compared. Handcart and total station point cloud deviation are close together.

But the handcart generates the points by photogrammetry and all other systems by LiDAR. MVH and backpack have a significant higher deviation in accuracy.

The handcart and MVH have segmentation approaches. The handcart segmentation approach is based on images and the MMV on LiDAR. So based on this, the handcart is not designed for high density point cloud generation. It is designed to reconstruct the necessary key points of facade elements shape for LOD3 modelling. Surface areas can be photogrammetrically three-dimensionally reconstructed. But questionable is this three-dimensional information necessary? Usually facade elements are extruded, simple, regular shapes, like rectangular, circles. So high-level detailed objects like stuck bring the handcart to its reconstruction borders. Because many images with good illumination must be taken to reconstruct a high density point cloud. Here the LiDAR-based approaches are in advantage.

In summary, the systems are a trade-off between time and accuracy. The handcart is a trade-off design of both for further LOD3 reconstruction based on camera photogrammetry.

5 Conclusion

In this paper, we propose another an alternative data acquisition approach for LOD3 reconstruction based on a camera, high precision gimbal, handcart design. This design was derived from a superior camera SfM based LOD3 reconstruction pipeline. A prototype was built and a data acquisition strategy was developed. It was tested, evaluated and practical approved in a test environment on component, subsystem, and system level. The test results were discussed in relation to competing systems techniques. The primary contributions can be summarized as follows:

1. The system is a multi functional data acquisition system for facade reconstruction. The gimbal enables a 360° surround view of the camera and LiDAR rangefinder. Additionally, the rangefinder is controlled in close loop measurement by camera and gimbal. This enables to data recording for photogrammetry reconstruction and LiDAR-based ground truth in one step.
2. The photogrammetry reconstruction was evaluated in a test environment. Resulting accuracies were in the millimeter and low centimeter range.
3. The system is designed for extruded, simple, regular shapes reconstruction. Highly detailed reconstructions like stuck are conditional possible.
4. The handcart application area is limited by steep slopes and barriers. Its dual use in stationary and mobile allows data acquisition in road traffic, sidewalks and rough terrain. This is an enabler of scalability for LOD3 reconstructions.
5. Summarized, the system is an alternative photogrammetry LOD3 reconstruction approach competing to point cloud-based. Due to its accuracy, it is capable to keep up with competing systems.

In the future, we will focus on: (1) image instance segmentation, (2) automatization building reconstruction supported by LOD2 data, (3) mobile data acquisition, (4) object-based autarky self-localization.

References

- AccuCities Ltd. Central London level 3 tiles. [Online]. Available: <https://www.accucities.com/3d-models-london/level-3-3d-models-london/>
- Beil C, Kutzner T, Schwab B, Willenborg B, Gawronski A, Kolbe TH (2021) Integration of 3d point clouds with semantic 3d city models—providing semantic information beyond classification. ISPRS Ann Photogram Remote Sens Spat Inf Sci VIII-4/W2-2021:105–112
- Blaser S, Meyer J, Nebiker, Fricker L, Weber D (2020) Centimetre-accuracy in forests and urban canyons—combining a high-performance image-based mobile mapping backpack with new georeferencing methods. ISPRS Ann Photogram Remote Sens Spat Inf Sci V-1-2020:333–341
- Bosch Sensortec (2021) Bno055: intelligent 9-axis absolute orientation sensor. [Online]. Available: <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bno055-ds000.pdf>
- Camera calibration and 3d reconstruction (2011) [Online]. Available: https://docs.opencv.org/4.7.0/d9/d0c/group__calib3d.html
- City of Espoo (2023) City model data: Espoo’s comprehensive 3d city information model adheres to the ogc citygml 2.0 city model standard. [Online]. Available: https://kartat.espool.fi/3d/citymodel_en.html
- Groeger G, Kolbe TH, Czerwinski A, Nagel C (2008) Opendgis city geography markup language (citygml) encoding standard. [Online]. Available: https://portal.ogc.org/files/?artifact_id=28802
- Gruen A, Schrotter G, Schubiger S, Qin R, Xiong B, Xiao C, Li J, Ling X, Yao S (2020) An operable system for lod3 model generation using multi-source data and user-friendly interactive editing: final report
- Leica Geosystems, Leica blk360: reality capture. now. [Online]. Available: https://shop.leica-geosystems.com/sites/default/files/2022-06/Leica_BLK360-SpecSheet.pdf
- NavVis GmbH (2021) Evaluierung der Genauigkeit von mobilen Mappingssystemen in Innen- und Außenbereichen: Eine Bewertung der Datenqualität des NavVis VLX im Vergleich zu einem terrestrischen Laserscanner, inkl. Fallbeispielen für Anwendungen in Innen- und Außenbereichen. [Online]. Available: <https://www.navvis.com/de/resources/white-papers/indoor-outdoor>
- NavVis GmbH, Product datasheet navvis vlx: 2nd generation. [Online]. Available: navvis.com/accuracy
- NetwakeVision GmbH, Royalbase. [Online]. Available: <https://www.royal-gps.com/produktubersicht/#RoyalBase>
- Nüchter A, Borrmann D, Koch P, Kühn M, May S (2015) A man-portable, imu-free mobile mapping system. ISPRS Ann Photogram Remote Sens Spat Inf Sci II-3/W5:17–23
- Pantoja-Rosero BG, Achanta R, Kozinski M, Fua P, Perez-Cruz F, Beyer K (2022) Generating lod3 building models from structure-from-motion and semantic segmentation. *Autom Const* 141:104430
- Schwab B, Wysocki O (2021) Lod3 road space models. [Online]. Available: <https://github.com/savenow/lod3-road-space-models>
- Sony (2018) Imx477-aack product information. [Online]. Available: https://www.sony-semicon.com/files/62/pdf/p-13_IMX477-AACK_Flyer.pdf
- Sturm P (2016) Pinhole camera model. In: *Computer vision: a reference guide*, [enhanced credo edition] (ed.). Springer and Credo Reference, New York [New York] and Boston, Massachusetts
- TRINAMIC Motion Control GmbH & Co. KG, Tmc2130 datasheet. [Online]. Available: https://www.trinamic.com/fileadmin/assets/Products/ICs_Documents/TMC2130_datasheet.pdf

- u bloxx (2023) Zed-f9p-04b: high precision gnss module. [Online]. Available: https://content.u-blox.com/sites/default/files/ZED-F9P-04B_DataSheet_UBX-21044850.pdf
- Voxelgrid GmbH (2023) Vgs data processing. [Online]. Available: https://voxelgrid.com/voxelgrid-system-de/#VGS_Data_Processing
- Wysocki O, Schwab B, Willenborg B (2022) Awesome citygml. [Online]. Available: <https://github.com/OloOcki/awesome-citygml>
- Zoller + Fröhlich GmbH, Z+f imager 5016 datasheet. [Online]. Available: info@zf-laser.com

Identification and Interpretation of Change Patterns in Semantic 3D City Models



Son H. Nguyen and Thomas H. Kolbe

Abstract Urban Digital Twins have received significant attention in recent years due to their economic and research importance. Although many definitions exist, the general consensus agrees on a continuous two-way data flow between a physical entity and its virtual counterpart in a digital twin. In the context of smart cities and semantic 3D city models, however, no major breakthrough in realizing such complex change detection and analysis systems has yet been achieved. While several methods for change detection in semantic 3D city models have been proposed, the analysis of found changes, especially the identification of patterns among a large number of changes, has not been given as much attention. Without a proper handling of patterns, it is difficult to provide useful interpretation of changes with respect to stakeholders. Therefore, this research proposes a framework to define, detect and decipher complex semantic change patterns in semantic 3D city models. The approach provides a central rule network to describe aggregation relations between changes as well as methods to identify and capture detected change patterns directly in the graph representation of a city model.

Keywords Semantic networks · Change patterns · Urban digital twins

1 Introduction

Digital Twins have in recent years become a major driving force behind many technological and economic progresses worldwide. In the context of smart cities and urban development, a digital twin of a city - an Urban Digital Twin—is a comprehensive framework for organizing and harnessing the many diverse aspects of a city, ranging

This article was selected based on the results of a double-blind review of the full paper.

S. H. Nguyen (✉) · T. H. Kolbe
Technical University of Munich, Munich, Germany
e-mail: son.nguyen@tum.de

T. H. Kolbe
e-mail: thomas.kolbe@tum.de

from physical components and logical structure to partaking actors and processes. Urban Digital Twins are created for specific purposes. The goal is to gain essential insights into the state of the city and its development by observing and analyzing the information available in its corresponding digital twin, thereby supporting both regular operations and critical urban planning and decision-making. Despite the many definitions of digital twins, the general consensus agrees that a digital twin must involve a *physical entity*, a corresponding *digital representation* and a *continuous feedback loop* between the physical and digital entity. This means that changes in the real world must be reflected on the digital side, and vice versa, as illustrated in Fig. 1. Such systematic two-way synchronization is however difficult to implement and scale efficiently (Grieves and Vickers 2017; Sharma et al. 2022). According to a recent survey among international experts (Lei et al. 2023), updating (including change detection, version management and efficiency) was identified as one of the most commonly cited technical challenges of Urban Digital Twins, for which no known complete solution has yet been achieved. As a result, many current smart city deployments, especially in 3D city modelling, often replace old datasets with newer versions, which not only wastes time and computational resources, but also ignores any meaningful development that may have materialized in the datasets during the recorded time period.

Therefore, acquiring efficient methods not only to detect changes but also to assess and understand the results is of significant advantage. Moreover, the gained knowledge provides a valuable insight into the semantic interrelations among changes. Thus, the general goal is to increase the semantic usefulness and readiness of the model. The modified digital entity can therefore be further enriched with semantic contents in a multi-levelled process called the *Semantic Enrichment*, as illustrated on the right-hand side of Fig. 1. This process consists of three consecutive levels listed in ascending semantic order as follows:

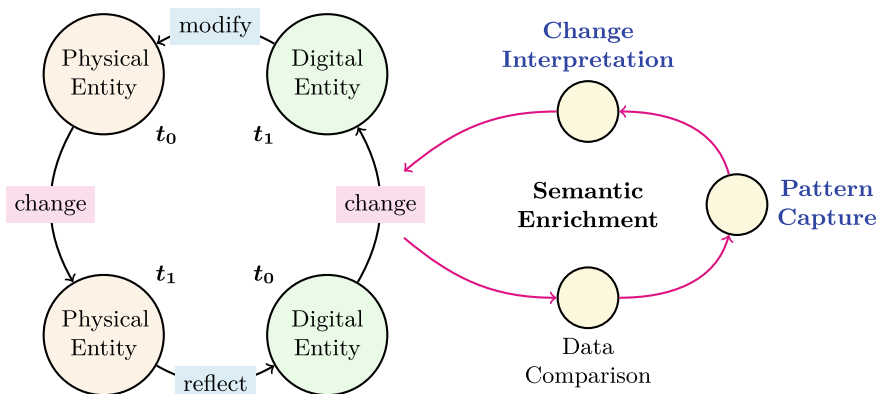


Fig. 1 An overview of an urban digital twin (left) and its semantic enrichment process (right) between two temporal versions of the digital entity. The focus of this research is on the pattern capture and change interpretation (blue)

1. **Data Comparison:** Snapshots recorded at different timestamps are matched and compared. This step is directly linked to the data storage and has the lowest level of semantic detail.
2. **Pattern Capture:** Based on the changes detected in the previous level, patterns of changes are captured to provide additional semantic context on the data. The semantic detail of this level is thus increased.
3. **Change Interpretation:** Combining the results from previous levels, comprehensive semantic interpretation of changes can be produced. This level has the highest concentration of semantic contents.

Change detection and version control in general is not new and has been discussed by several studies in recent years, such as in the field of semantic 3D city modelling (Redweik and Becker 2015; Nguyen et al. 2017). The majority of these studies however focused solely on the detection of literal modifications in the data (Semantic Enrichment, level 1), without further considering their semantic context. Most changes however also have a meaning and purpose relevant to specific groups of stakeholders (Nguyen and Kolbe 2021). A graph-based framework was introduced (Nguyen and Kolbe 2022) to better model and understand the interrelations between changes and stakeholders, as well as the correlations and reasoning of changes in semantic 3D city models (Semantic Enrichment, level 3). An open question, however, remains as to how patterns of changes can be efficiently captured (Semantic Enrichment, level 2). The current approach in many smart city deployments is to create database queries for each pattern and execute them in any necessary order on an ad hoc basis. This not only requires expert knowledge on the structure of the underlying databases, but may further lead to unwanted scheduling and efficiency problems, especially if rules are dependent on each other (forming a “pattern” of patterns).

This research explores the changes that occurred in the digital representation of a city within a digital twin. The goal is to uncover patterns and underlying meanings or reasoning that these changes have on the real city. To achieve this, this paper proposes a framework to define, detect and decipher complex change patterns in semantic 3D city models, namely (1) a single rule network to define all aggregation semantic rules, and (2) methods to detect patterns based on given rules. The proposed methods were developed for Urban Digital Twins represented by semantic 3D city models in CityGML as one of their core components but can also be applied to other fields with similar use cases and semantic object representations, such as in the BIM field.

2 Foundations and Related Work

Since semantic 3D city models are structured as graphs (Falkowski and Ebert 2009; Agoub et al. 2016; Nguyen and Kolbe 2022), most matching methods also naturally store changes in compatible data structures. This leads to the pattern matching prob-

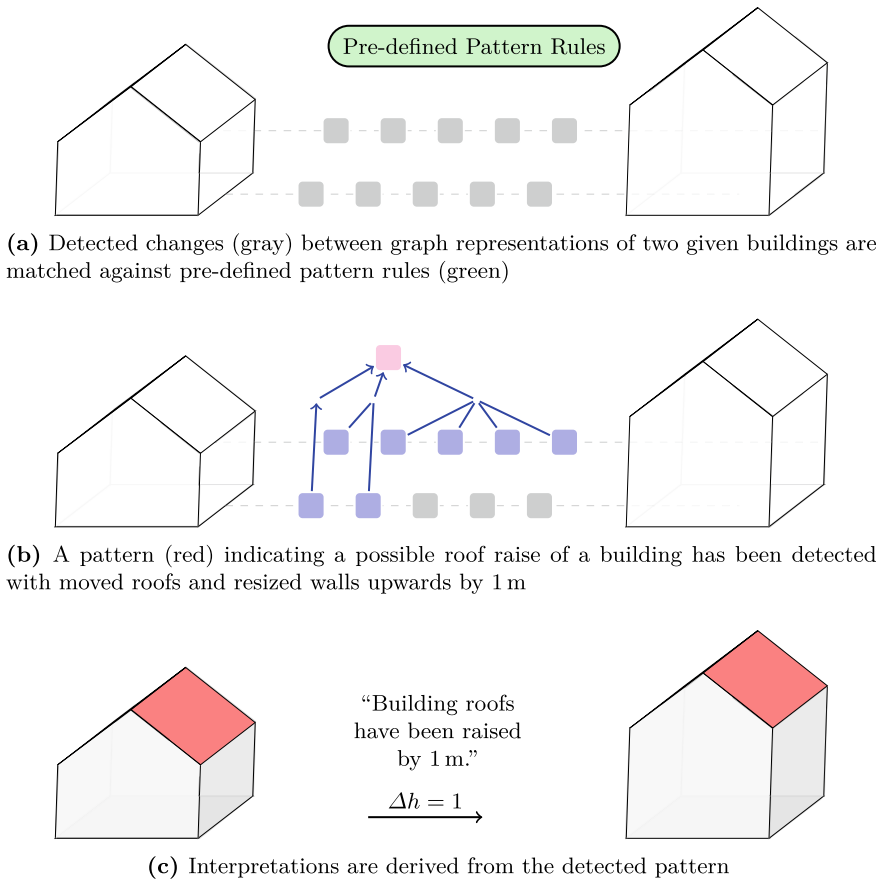


Fig. 2 An illustration of the pattern matching process. Input changes (gray) belonging to a pattern (blue) are aggregated into new interpretation nodes (red)

lem of graphs. Figure 2 gives an overview of the methods introduced in this research, where (1) changes detected between graph representations of two semantic 3D city models are first matched against given pattern rules (Fig. 2a), (2) additional interpretation nodes are then created and attached to the source nodes in the graph database (Fig. 2b), which (3) can ultimately be utilized to derive meaningful interpretations about the changes in the data (Fig. 2c). To achieve this, four key technical requirements of matching change patterns in semantic 3D city models must be fulfilled, namely Dynamic Aggregation, On-the-fly Typing, Origin Handling and Memory Efficiency, as described in Table 1. Thus, this research proposes several concepts, which, to a degree, have their roots in Rete networks, Petri nets and graph transformation systems. This section shall therefore provide a brief introduction to some the most relevant components of each concept.

Table 1 Key requirements for change patterns in semantic 3D city models

| Requirement | Description |
|---------------------|---|
| Dynamic aggregation | Most aggregation rules specify a static number of input objects. In many important use cases, however, input quantity is not known before execution. Therefore, dynamic aggregation must be enabled |
| On-the-fly typing | As input for pattern matching, changes of city objects may be given in any order. To avoid repeated iteration, the system must be able to identify changes based on their types and attributes in real-time |
| Origin handling | In addition to types, changes must be distinguished by their semantic context. For example, when interpreting changes to a building, only changes relevant to that specific building are considered |
| Memory efficiency | Graph representations of cities may become very large, leading to a potentially overwhelming number of produced changes. This requires efficient algorithms with regard to memory consumption |

2.1 Rete Networks

The Rete match algorithm is an efficient method for matching a large number of patterns against a large number of objects (Forgy 1982). The algorithm was originally developed for production system interpreters. A typical production system consists of (1) an unordered collection of conditional statements, (2) a global **working memory** (or database) holding temporary data, and (3) a rule interpreter that can assess rule conditions. To avoid iteration over input, the Rete algorithm stores matched objects in each corresponding rule. When a new element is inserted or an existing element is removed from the shared working element, affected rules are notified and their list of stored objects is updated accordingly. To avoid repeated iteration over rules, a directed acyclic graph representation of rules is used for pattern matching. This graph is called the Rete network (Forgy 1982).

One major advantage of the Rete match algorithm is its processing speed due to the employed working memory containing temporary data of each pattern during execution. The working memory “memorizes” previously read input objects as well as intermediate results in “buckets”, which, if full, shall trigger corresponding actions. Thus, the working memory allows both on-the-fly type checking and on-demand reactivation of pending rules. Moreover, the Rete algorithm excels in use cases, where input is a stream consisting of randomly ordered and differently typed objects, on which a large number of insertion and deletion operations are performed. The performance of the Rete algorithm largely depends on the implementation of its working memory. In general however, in worst-case scenarios, the original Rete algorithm may store all temporary data in main memory, degenerating memory efficiency. The methods proposed in this research employ an extended version of the afore-mentioned working memory while avoiding excessive memory consumption.

2.2 *Petri Nets*

Petri nets were first proposed to model parallel and distributed systems (Petri 1962). A Petri net is a bipartite graph consisting of places and transitions. The partitions are connected via directed arcs. Places may contain **tokens**, which can travel between neighbouring places. In rule-based systems, places semantically represent the current state or conditions of rules, while transitions represent actions. If a sufficient number of tokens exist at a place, its outgoing transition(s) shall be triggered, consuming input tokens and producing new ones. The number of tokens consumed and produced is dictated by the transitions' weights.

Petri nets have strong scalability and modelling potential in rule-based applications. The use of tokens is well-suited for describing many aggregation rules of changes in semantic 3D city models. Petri nets can be represented mathematically as both graphs and matrices. However, tokens in classical Petri nets have neither an attribute, a type, nor an origin, and are thus indistinguishable. In contrast, changes in semantic 3D city models are typed and attributed. Therefore, this research employs a specialized Petri net capable of distinguishing tokens based on their types, attributes and semantic context.

2.3 *Graph Transformation Systems*

Due to the graph-based nature of semantic 3D city models, matching their change patterns can also be considered a use case of graph transformation. Graph transformation was first proposed as a graph grammar for rule-based rewriting of non-linear data structures (Pfaltz and Rosenfeld 1969; Pratt 1971; Heckel 2006). In graph transformation, rules are defined using type graphs and instance graphs. A **type graph** defines the conceptual model of object classes, while an **instance graph** is a snapshot containing concrete values and structure prescribed by the corresponding type graph.

Graph transformation systems are a powerful tool for handling complex semantic structures. Type-enabled graph transformation can utilize hidden context information of objects, enabling more complex analyses. However, graph transformation employs graph isomorphism, whose complexity is neither polynomial nor known to be NP-complete (Steven 2008). In addition, the structure of the type and instance graphs given by the transformation rules must be known, but this information is often unknown until execution. The methods proposed in this research employ a simplified approach to graph transformation. Instead of relying on graph isomorphism, node types and semantic positions in graphs are used to improve the runtime efficiency of the pattern matching process.

3 Defining Pattern Rules

Based on the strengths and limitations of the concepts discussed in Sect. 2, this research proposes a compact aggregative rule network to define graph-based rules for change patterns in semantic 3D city models.

3.1 Definitions

A **content network** is a directed and attributed graph representation of a semantic 3D city model, where all information about the city model is stored. Nodes in a content network are called *content nodes*. An example of a content network is shown in Fig. 3. A content network contains graph representation of both the old and new dataset. Detected changes are attached to both graphs, but for visual clarity, this study only shows one of these graph representations with connected changes, as shown later in Fig. 5.

A **rule network** is a directed acyclic and attributed graph that allows all rules for pattern matching to be defined in one place. It is a type graph capable of describing the characteristic behaviours of different types of changes, allowing the dependencies between rules to be explicitly captured. Nodes in a rule network are called *rule nodes* and can represent both literal detected changes in the data and interpreted changes later on. Figure 4 shows an example of a rule network.

Each rule node is assigned a type corresponding to the changes it represents. Rule nodes are connected using directed rule edges. A rule edge has three components: the next content type, a list of conditions and a weight. The content type acts as a “checkpoint” for the rule interpreter to navigate within the content network. For example, the content nodes associated with the rule edge between *PolygonResized* and *WallResized* are wall surfaces in the content network. Conditions are logical expressions evaluated against properties in the corresponding change nodes. Values can be named and shared as variables across converging rule edges. New properties in previous changes are forwarded to the next one as part of the knowledge gained through the interpretation process. If no condition is needed, the value 1 is used. All conditions must be fulfilled to trigger an aggregation. For example, the rule edge between *RoofMoved* and *RoofRaised* dictates that the translation vector v_T must have no x and y component, while the z value is named Δh to represent any real value. This variable is reused in other rule edges. Moreover, as shown in the rule edge between *HeightChanged* and *RoofRaised*, additional constraints can be introduced to limit property values. Parametrized conditions are evaluated during runtime at the next rule node by matching property values of collected changes, as explained in Sect. 4.

The weight of a rule edge dictates the number of occurrences of changes corresponding to the previous rule node required for the creation of the next interpretation node. If a rule node has multiple incoming edges, it can only be activated when all

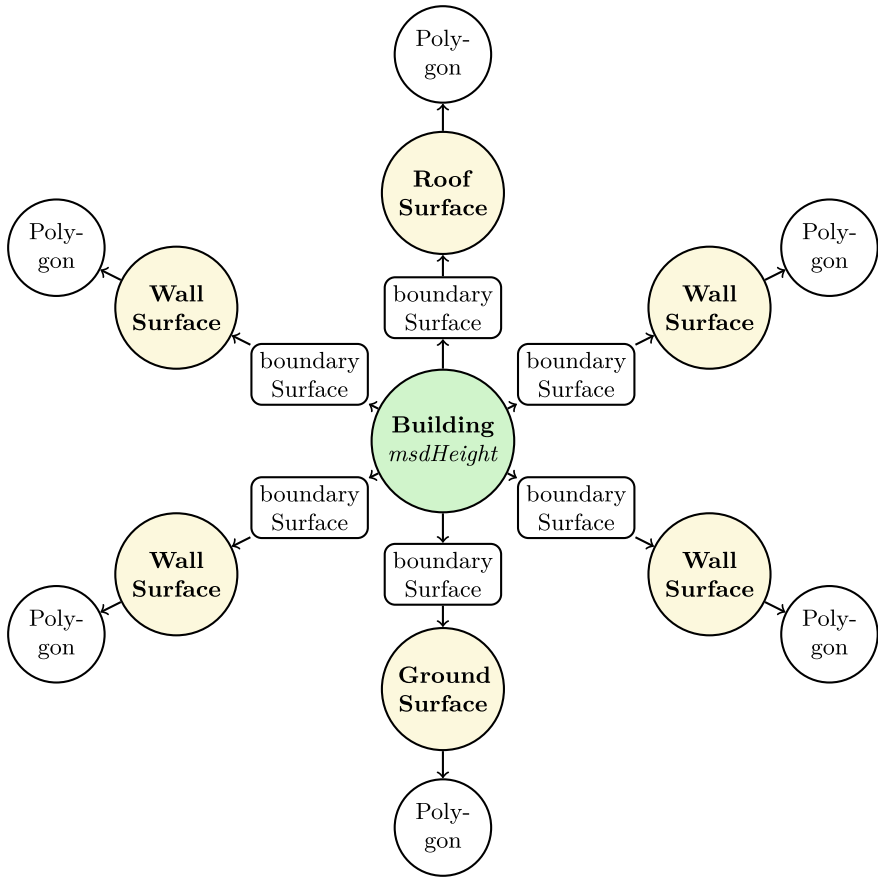


Fig. 3 A content network for a simplified building model, representing multiple 1-to-*n* relations between a building and its boundary surfaces. This is directly derived from a CityGML dataset as described in previous publications (Nguyen et al. 2017)

previous rule nodes have collected a sufficient number of changes. For example, the rule node *RoofRaised* can only be activated if all required occurrences of *RoofMoved*, *WallResized* and *HeightChanged* exist. The weight can be assigned a specific value or a placeholder * for an unknown value. For instance, since each building can have a different number of wall surfaces, the weight of the rule edge between *WallResized* and *RoofRaised* is first initialized with *. This placeholder is updated with a concrete value by the rule interpreter at runtime. The interpreter searches “upwards” in the content network for the next content node that matches the content type given in the rule edge, then traverses all paths “downwards” until a content node specified by the previous rule node is found. The placeholder is then replaced with the number of reached paths. For instance, while processing the rule edge between *WallResized* and *RoofRaised*, the interpreter searches for a *Building* node, as its type is specified as the

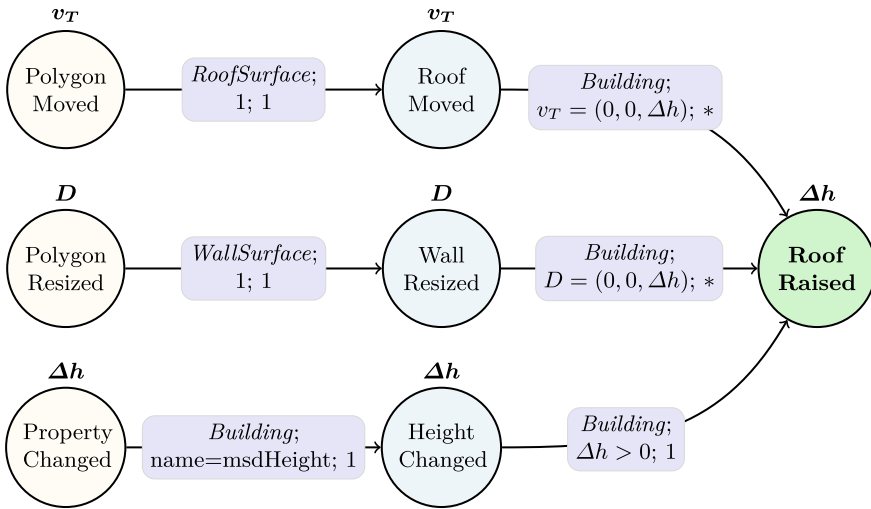


Fig. 4 An example of a rule network for the content network shown in Fig. 3. Each edge is notated with *(next content type; conditions; weight)*

Table 2 A comparison between the proposed rule network and related concepts with respect to the key requirements described in Table 1

| | Dynamic aggregation | On-the-fly typing | Origin handling | Memory efficiency |
|-----------------------------------|---------------------|-------------------|-----------------|-------------------|
| Rete networks ^a | × | ● | × | × |
| Petri nets ^a | × | ⦿ | × | ● |
| Graph transformation ^a | × | ⦿ | ● | × |
| Proposed rule network | ● | ● | ● | ● |

× Not applicable ⦿ Applicable if typing is enabled ● Applicable
^a Original publication is considered. Some variants may differ

next content type of the rule edge. From this *Building* node, all paths to *Wall* nodes are counted. A comparison between this rule network and the concepts previously mentioned in Sect. 2 is summarized in Table 2.

4 Detecting Change Patterns

Given a rule network, change patterns in a content network can be matched. The detected patterns are represented as additional interpretation nodes attached to their corresponding content nodes. For instance, a *PolygonMoved* node is attached to a source *Polygon* node, while a *RoofRaised* is attached to a *Building* object. The results of the pattern matching process are illustrated in Fig. 5.

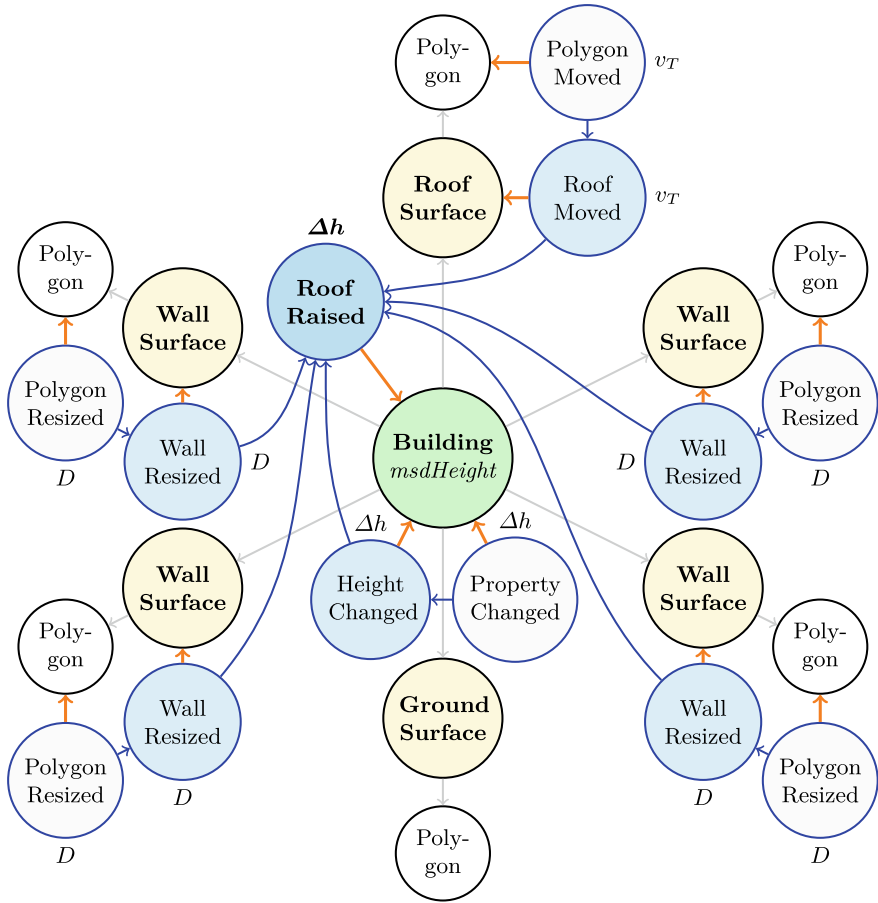


Fig. 5 An example of the results (blue) of the pattern matching process based on the content network shown in Fig. 3 and the rule network shown in Fig. 4. For visual clarity, *boundarySurface* nodes are omitted. Interpretation connections are shown in orange. In this example, the algorithm starts with the changes *PolygonMoved*, *PolygonResized* and *PropertyChanged* at the lowest level and gradually propagates “upwards” in the content network until a content node with a wanted type is encountered, such as the path *PolygonResized* → *Polygon* → *WallSurface*, where *WallSurface* is required by *WallResized*. Once all criteria have been fulfilled, a new interpretation node *WallResized* is created. The propagation proceeds until a *CityModel* node (not shown) is reached, which is associated with global or systematic change patterns. Thus, the pattern matching algorithm is an aggregation process, where changes of lower semantic levels are aggregated to produce new changes of higher semantic levels

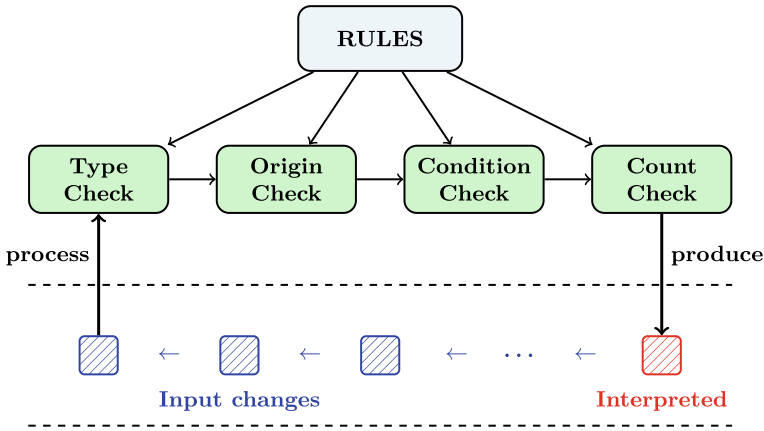


Fig. 6 The pattern matching algorithm uses a queue to process all literal and interpreted changes (blue). Changes that pass all checks are aggregated into a new interpreted change (red) and added to the queue for further processing

The method used to detect change patterns during the pattern matching process is summarized in Fig. 6 and described in Algorithm 1. The algorithm employs a FIFO (First In, First Out) queue that functions like a conveyor belt in an assembly line. The queue contains all literal and interpreted changes, removes the first element for processing, and stores the aggregated changes at the end of the queue until all elements have been processed. The algorithm aims to aggregate input changes into new, higher-level semantic changes. Changes are aggregated if they satisfy all four criteria: *type*, *origin*, *condition* and *count check* required by given rules. The type and condition check can be determined based on the type and attributes available in each change. To conduct the remaining count and origin check, the method employs two of its key concepts, namely an **aggregative memory** and a graph-based **semantic context**.

When a rule is applied, its associated next content node is initialized with a memory to store crucial aggregation information about the current and maximum number of changes collected per type. Each time the content node encounters a change required by its rule, it increases its count of the object type by one until this value reaches a maximum. The count check is considered complete, if the number of all change occurrences per type reaches a maximum. This memory can be implemented as a collection of key-value pairs. Listing 1 gives an example of the memory of the content node *Building* shown in Fig. 5.

The use of memory is similar to that of Rete networks (Forgy 1982) and can eliminate repeated iteration by processing changes on the fly. However, in contrast to classical Rete networks, the proposed method does not store entire objects in its memory. Instead, the algorithm first identifies objects based on their types and attributes, then updates the number of their occurrences accordingly. Moreover, at the start, the memory is empty and only expanded as new rules and changes are

Algorithm 1: Pattern matching algorithm

Data : A content network N_C and a rule network N_R
 A queue Q initialized with detected changes connected with N_C
Result: Interpreted changes connected with N_C

```

1 while  $Q$  is not empty do
2    $q \leftarrow$  dequeue  $Q$ 
3    $r \leftarrow$  find rule in  $N_R$  that accepts  $q.type$ , or else continue
4    $n \leftarrow$  find node in  $N_C$  of  $r.type$  starting from  $q$ , or else continue
5   if  $n.memory$  does not exist then
6      $n.memory \leftarrow$  initialize memory
7   end
8   if  $r.conditions$  are all fulfilled then
9     if  $n$  accepts  $q.origin$  then
10      increase  $n.memory.count(q.type)$  by 1
11      store reference to  $q$  in  $n.memory.refs$ 
12    end
13    if  $n.memory.count(t) = \max, \forall t \in n.memory.types$  then
14       $m \leftarrow$  create interpretation node representing the change pattern
15      initialize  $m.type, m.origin$  and store  $q.properties$  in  $m$ 
16      connect  $n$  and all stored references in  $n.memory.refs$  with  $m$ 
17      enqueue  $m$  into  $Q$ 
18    end
19  end
20 end

```

```

{
  " variables" : [ " deltaH" ],
  " rules" : [
    {
      " rule_type" : " RoofMoved",
      " count_value" : 1,
      " max_value" : 1,
      " properties" : { " vT" : "( 0, 0, 0.969)" }
    },
    {
      " rule_type" : " WallResized",
      " count_value" : 2,
      " max_value" : 4,
      " properties" : { " D" : "( 0, 0, 0.969)" }
    },
    {
      " rule_type" : " HeightChanged",
      " count_value" : 1,
      " max_value" : 1,
      " properties" : { " deltaH" : 0.969 }
    }
  ]
}

```

Listing 1 An excerpt from the memory of a *Building* node shown in Fig. 5 that has collected 1 instance of *RoofMoved*, 2 *WallResized* and 1 *HeightChanged*.

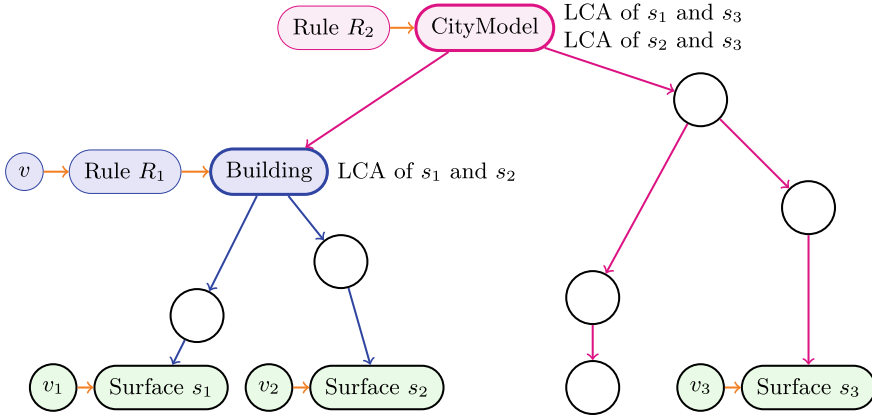


Fig. 7 An example of changes v_1 , v_2 and v_3 associated with surfaces s_1 , s_2 and s_3 (green). The lowest common ancestor (LCA) of s_1 and s_2 is located at the *Building* node (blue), while the LCA of s_1 and s_3 , as well as s_2 and s_3 , is located at *CityModel* (red). Thus, rule R_1 accepts v_1 and v_2 , while rule R_2 accepts both R_1 and v_3 . The interpretation of higher levels only considers the interpretation of objects on the next lower level, or literal changes if no interpretation is available

encountered. This avoids the worst-case memory consumption of Rete networks, where the working memory could hold all input objects at runtime.

Another key concept of the pattern matching algorithm is the ability to distinguish objects based on their semantic context or origin in the network. An origin of a node is a set containing itself and all its ancestors in a directed acyclic network. In an input sequence of changes, such as $(r, w, w, r, r, h, \dots)$, where the change types are *RoofMoved*, *WallResized* and *HeightChanged*, it is unknown whether these changes refer to the boundary surfaces of the same building or several different ones. Therefore, the origin check is employed as an additional guard to ensure changes are aggregated correctly. This is related to the problem of finding the lowest common ancestor (Aho et al. 1976), as illustrated in Fig. 7.

5 Application Examples

In the following experiments, change patterns between two CityGML documents in Level of Detail (LoD) 2 are matched. The datasets used are excerpts, each containing 44 buildings from a selected area of Hamburg, Germany. These datasets were recorded in 2016 and 2022 and are provided publicly by the city.¹ All data are managed within a single graph database, which includes the content network of both the city models and their changes, as well as rule networks for detecting change patterns. The graph database Neo4j is used. The rule interpreter shown in Algorithm

¹ <https://metaver.de>.

Table 3 A summary of thematic changes between 2016 and 2022

| | Property name | Description | B | | P | | R | |
|--------|--|------------------------|----|---|----|---|----|---|
| Insert | <i>Dachhoehe</i> ¹ | Roof height | 9 | × | 25 | × | 0 | × |
| | <i>Firsthoehe</i> ¹ | Ridge height | 21 | × | 63 | ● | 0 | × |
| | <i>Geo.typ2DRef.</i> ¹ | Source geometry type | 44 | ● | 0 | × | 0 | × |
| | <i>Grundrissaktualitaet</i> ¹ | Ground plan update | 44 | ● | 63 | ● | 0 | × |
| | <i>Hauskoordinate</i> ¹ | Building coordinates | 5 | × | 0 | × | 0 | × |
| | <i>Qu.Dacherkennung</i> ¹ | Roof detection quality | 21 | × | 63 | ● | 0 | × |
| | <i>Traufhoehe</i> ¹ | Eaves height | 21 | × | 63 | ● | 0 | × |
| | <i>tridicon_Dachform</i> ¹ | Tridicon roof shape | 21 | × | 63 | ● | 0 | × |
| Update | <i>gmlid</i> | Identifier | 44 | ● | 63 | ● | 81 | ● |
| | <i>creationDate</i> | Modification date | 44 | ● | 63 | ● | 0 | × |
| | <i>measuredHeight</i> | Measured height | 20 | × | 63 | ● | 0 | × |
| | <i>function</i> | Function | 14 | × | 0 | × | 0 | × |
| | <i>roofType</i> | Roof type | 1 | × | 7 | × | 0 | × |
| | <i>Datenqu.Dachhoehe</i> ¹ | Source roof height | 2 | × | 10 | × | 0 | × |
| | <i>Flaechengroesse</i> ¹ | Surface area | 0 | × | 0 | × | 33 | × |
| | <i>Flaechenmeinung</i> ¹ | Surface inclination | 0 | × | 0 | × | 32 | × |
| | <i>Flaechenrichtung</i> ¹ | Surface orientation | 0 | × | 0 | × | 2 | × |
| | <i>Gemeindeschluessel</i> ¹ | Municipality key | 44 | ● | 0 | × | 0 | × |

¹ Generic string attribute **B** Building **P** Building part **R** Roof surface
 × Local or clustered change pattern ● Global change pattern

I can be implemented using Neo4j’s Cypher query language or its Java API. A Java implementation of the rule interpreter can be exported as a user-defined procedure, which can then be invoked directly from Cypher.

Global patterns have been observed based on a total number of 1049 changes on thematic attributes distributed over roof surfaces, building parts and buildings (see Table 3). A visualization of these patterns can be found in Fig. 8.

Out of 638 roof, wall, and ground surfaces, 552 have been observed to either be moved or changed in size. Translation is detected by calculating the offset vector between geometries, while size changes are measured by deviations in the surfaces’ 3D bounding boxes. Notably, all 134 translation and 394 (94 %) of all size changes occurred vertically, with translation offsets ranging from -0.957 m (downwards) to 1.895 m (upwards), and resize margins between -1.836 m (height decrease) and 2.288 m (height increase). These changes are significantly reduced to a few interpretations in the following three steps.

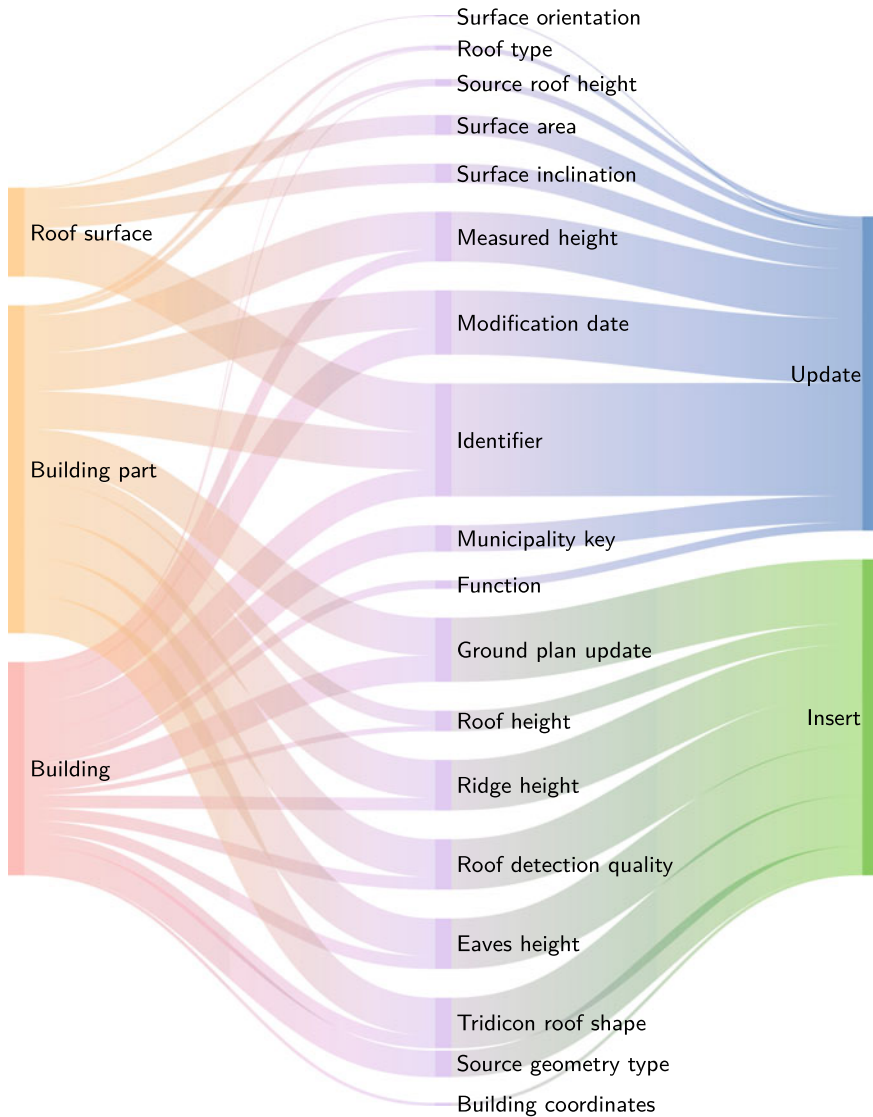


Fig. 8 An overview of change patterns detected in the thematic data of the Hamburg datasets between 2016 and 2022. Changes are categorized by their functions shown on the right column (inserted and updated properties). The left column represents the number of change occurrences grouped by feature types, where most modifications occurred. The property names are given in the middle

Firstly, by extending the rule network given in Fig.4, translation and resize changes of the same margin for all roof, wall or ground surfaces of a building are aggregated into interpretation nodes attached to buildings (see Table 4).

Table 4 A summary of surface-based changes between 2016 and 2022

| Surface changes | Aggregated | Found patterns of same margin ^{a,b} |
|------------------------|------------|--|
| 2 Ground size changes | 2 (100%) | 0 Building with all grounds resized |
| 6 Wall translations | 4 (67%) | 1 Building with all walls moved |
| 31 Roof size changes | 25 (81%) | 5 Buildings with all roofs resized |
| 49 Roof translations | 49 (100%) | 11 Buildings with all roofs moved |
| 79 Ground translations | 79 (100%) | 38 Buildings with all grounds moved |
| 385 Wall size changes | 204 (53%) | 12 Buildings with all walls resized |

^a Only buildings in which *all* surfaces of a given type have been moved or resized by the same amount are considered.

^b A building may be counted multiple times, up to a maximum of three occurrences, with one count for each boundary surface type

Secondly, the aggregated nodes are further combined into tuples. Since each building is bounded by three types of surfaces and each interpretation node indicates a surface translation, resize, or none of the above, there exist 27 combinations to form an interpretation tuple for each building. A tuple RWG denotes a consistent translation (T), resize (S) or none (X) for all roofs (R, first position), walls (W, second) and grounds (G, third) of a building.

Thirdly, the translation and resize margins stored in the interpretation tuples are studied to reveal correlations between geometric changes. For example:

1. All roof, wall and ground surfaces of one building marked with TTT have been vertically shifted by the same offset -0.049 m, meaning the entire building has been moved downwards by that amount.
2. A common pattern has been observed in all ten buildings marked with TST, where each building's roof and ground surfaces were moved by Δz_r and Δz_g , and all wall surfaces resized by Δz_w , such that $\Delta z_w + \Delta z_g = \Delta z_r$ (see Table 5). For example, the roofs of building B1 have been raised by approximately 1 m, supported by an equivalent increase in wall height and a small downward translation of ground surfaces. This information is useful for stakeholders such as urban planners, energy consultants, and city mayors, as it may indicate that a building has been expanded by an additional storey, potentially increasing the amount of available living space. In contrast, buildings with small deviations, such as B8, may be of interest to data brokers and quality managers.
3. Of the six buildings marked with XXX, two have remained geometrically unchanged, as none of their boundary surfaces has been translated or resized.

Therefore, the interpretation nodes produced by the pattern matching process are crucial in providing a deeper understanding of the interrelationships between detected changes in the datasets. Further information on the implementation and additional examples can be found online (work in progress).²

² <https://github.com/tum-gis/citymodel-compare>.

Table 5 An overview of the translation offsets Δz_r and Δz_g of all roof and ground surfaces, resize margins Δz_w of all wall surfaces, and difference in each building’s measured heights Δh (in cm). The correlations $\Delta z_w + \Delta z_g = \Delta z_r$ and $\Delta h = \Delta z_w$ apply in all ten buildings (B1–10) marked with TST

| | B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 | B9 | B10 |
|--------------|-------|------|------|------|-------|-------|------|------|-------|-------|
| Δz_r | 96.9 | 16.1 | 12.0 | 9.4 | -2.8 | -4.5 | -5.7 | -5.8 | -6.0 | -7.4 |
| Δz_w | 102.5 | 25.1 | 16.5 | 12.0 | 10.8 | 11.4 | -3.2 | 0.8 | 5.0 | 18.8 |
| Δz_g | -5.6 | -9.0 | -4.5 | -2.6 | -13.6 | -15.9 | -2.5 | -6.6 | -11.0 | -26.2 |
| Δh | 102.5 | 25.1 | 16.5 | 12.0 | 10.8 | 11.4 | -3.2 | 0.8 | 5.0 | 18.8 |

6 Conclusion and Future Work

Based on the strengths and limitations of well-known concepts for rule-based systems, such as those of Rete networks, Petri nets and graph transformation systems, this research proposed a framework to define rules for matching change patterns in semantic 3D city models. The framework employs graph representations of semantic 3D city models, called the content networks, as a basis for all pattern detection processes. Rules are defined in a rule network, which is a type graph that can describe the characteristic behaviours and interrelations of different classes of changes in aggregative semantic patterns. By applying a rule network to a content network, change patterns can be detected and captured during the pattern matching process. The detected change patterns are represented as interpretation nodes connected to the content network, thus enabling faster retrieval and handling of the semantic patterns of changes.

The method employs aggregative rules to effectively condense a large number of changes into a few interpretation nodes that are more comprehensible to various stakeholders. Based on these gained interpretations, more efficient and complex analyses on the city’s evolution can be performed. Moreover, the employed rule networks are compact yet expressive, and can be used in highly automated processes. It should be noted, however, that the framework is designed to perform aggregation and logical operations exclusively, thus requiring the provision of detected changes, as presented in our earlier publications (Nguyen et al. 2017), including those of complex geometric objects.

Combined with other previous related work (Nguyen and Kolbe 2021, 2022), this research serves as one of the last missing pieces required for a comprehensive understanding of changes in semantic 3D city models. Further experimentation and optimization of the proposed methods on large-scale real-world datasets are planned. An investigation will be conducted to determine the potential reasons for changes, their impact on city models, and how they can be represented and detected using the proposed framework.

References

- Agoub A, Kunde F, Kada M (2016) Potential of Graph Databases in Representing and Enriching Standardized Geodata. Dreiländertagung der DGPF, der OVG und der SGPF, vol 36
- Aho AV, Hopcroft JE, Ullman JD (1976) On finding lowest common ancestors in trees. *SIAM J Comput* 5(1):115–132
- Falkowski K, Ebert J (2009) Graph-based urban object model processing. In: City models, roads and traffic (CMRT'09): object extraction for 3D city models, road databases and traffic monitoring-concepts, algorithms and evaluation, Paris, France, vol 9, pp 115–120
- Forgy Charles L (1982) Rete: a fast algorithm for the many pattern/many object pattern match problem. *Artif Intell* 19(1):17–37
- Grieves M, Vickers J (2017) Digital twin: mitigating unpredictable, undesirable emergent behavior in complex systems. Springer International Publishing, Cham, pp 85–113
- Heckel R (2006) Graph transformation in a nutshell. In: Electronic notes in theoretical computer science, vol 148, pp 187–198
- Lei Binyu, Janssen Patrick, Stoter Jantien, Biljecki Filip (2023) Challenges of urban digital twins: a systematic review and a delphi expert survey. *Autom Const* 147:104716
- Nguyen SH, Kolbe TH (2021) Modelling changes, stakeholders and their relations in semantic 3D city models. In: ISPRS (ed.) Proceedings of the 16th international 3D GeoInfo conference 2021, volume VIII-4/W2-2021 of ISPRS annals of the photogrammetry, remote sensing and spatial information sciences. New York University, ISPRS, pp 137–144
- Nguyen SH, Kolbe TH (2022) Path-tracing semantic networks to interpret changes in semantic 3D city models. In: Proceedings of the 17th international 3D geoinfo conference 2022, volume X-4/W2-2022 of ISPRS annals of the photogrammetry, remote sensing and spatial information sciences. UNSW Sydney, ISPRS
- Nguyen SH, Yao Z, Kolbe TH (2017) Spatio-semantic comparison of large 3D city models in CityGML using a graph database. In: ISPRS (ed.) Proceedings of the 12th international 3D geoinfo conference 2017, volume IV-4/W5 of ISPRS annals of the photogrammetry, remote sensing and spatial information sciences. University of Melbourne, ISPRS, pp 99–106
- Petri C (1962) Kommunikation mit Automaten. Ph.D. thesis, TU Darmstadt
- Pfaltz JL, Rosenfeld A (1969) Web grammars. In: Proceedings of the 1st international joint conference on artificial intelligence, IJCAI'69, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc, pp 609–619
- Pratt Terrence W (1971) Pair grammars, graph languages and string-to-graph translations. *J Comput Syst Sci* 5:560–595
- Redweik R, Becker T (2015) Change detection in CityGML documents. Springer International Publishing, Cham, pp 107–121
- Sharma A, Kosasih E, Zhang J, Brintrup A, Calinescu A (2022) Digital twins: state of the art theory and practice, challenges, and open research questions. *J Ind Inf Integr* 30:100383
- Steven S (2008) Skiena. In: The algorithm design manual, 2nd edn. Springer Publishing Company, Incorporated

3D Data Modelling and Topology

A Level of As-Is Detail Concept for Digital Twins of Roads—A Case Study



David Crampen, Marcel Hein, and Jörg Blankenbach

Abstract The recent rapid rise in the demand for digital methods for planning and management of road infrastructure has led to the development of new concepts for structuring the targeted applications where digital methods shall be applied in the future to improve overall efficiency. This transition towards digital planning, maintenance and even coupling digital representations with the asset in the real world requires a clear structure so that existing potential can be optimally exploited. In this paper, we conduct a case study to test and improve our proposed LOADt concept for the digital representation of roads for the use in digital twins. Since digital planning is just emerging in road construction, the road infrastructure sector currently faces the issue of existing roads not having a digital representation. Therefore, this contribution aims at highlighting the path from reality capturing towards the establishment of the digital representation as-is of existing roads.

Keywords Road infrastructure · Digital twin · Level of detail

1 Introduction

Digitization of the road construction sector is recently gaining much attention in Germany as well as in other European countries. In Germany the obligation to implement digital planning in road construction as defined in the ‘Masterplan BIM Bundesfernstraßen’ (Bundesministerium für Verkehr und digitale Infrastruktur 2021), which is published by the Federal Ministry of Transport and Digital Infrastructure, will come into force in 2025, while the transformation process started in 2021. Over 2 years have passed since the first regulations have been imposed, the comprehensive digital transformation in road construction towards digital planning for public projects is

This article was selected based on the results of a double-blind review of the full paper.

D. Crampen (✉) · M. Hein · J. Blankenbach
Geodetic Institute and Chair for Computing in Civil Engineering & GIS, RWTH Aachen University, Mies-Van-Der-Rohe-Straße 1, 52074 Aachen, Germany
e-mail: crampen@gia.rwth-aachen.de

ongoing and the industry is slowly adopting digital methods. In addition to digital planning as a tool for decision making in the construction process, the further deployment of digital technologies and real construction must be formalized at the same time. Especially these days, when road construction is expecting a radical change, it is important to set a strong foundation in order to be able to fully exploit the potential of this transformation (Guo et al. 2022).

Although for planning of new roads the use of digital planning methods is going to become mandatory, existing roads without digital representation will remain unaffected by this obligation. However, the climate crisis and the associated measures will assumedly slow down the construction of new road infrastructure in the next decade, whilst on the other hand the maintenance of existing roads will draw significantly more attention. In order to also benefit from the digital transformation for existing roads, we are focusing our work in this paper on the process of establishing a digital as-is representation of existing roads using reality capturing technologies. The purpose of the digital representation obtained in this way is to be applied in a digital twin of the road rather than for solely use in site/asset management.

Hence, the method of building information modelling (BIM) is just being adopted for infrastructure planning, the need for smart roads that can interact with vehicles for applications such as autonomous driving or safety monitoring and with their environment in general is rapidly increasing. This is one reason why the transformation of infrastructure has to proceed quickly. New technologies that address wide-ranging challenges such as climate neutral mobility are currently held back by the lack of digitization on infrastructure. To facilitate these new technologies, however, using BIM for planning a road site will be only part of the solution. Much more, the digitalization of the road has to take a further step, a step, where the road environment is functionalized to be adaptable to changing conditions, that will make the establishment of digital twins necessary.

In general, a digital twin enables the bidirectional interaction of real world assets and their digital representation. Sensors in the real world stream measurement data into the digital representation, where various analyses are then conducted. The analyses carried out lead to direct implications and actions in the real world. On a schematic level a digital twin is a system of a physical asset and its digital representation, tightly coupled and interacting through links in the form of sensors and smart control units, that can change specific configurations in real world processes. A digital twin can therefore target a large variety of different use cases (Tao et al. 2019). Our previous work introduced the LOADt concept, which is particularly relevant at this point (Crampen and Blankenbach 2023). Following the idea that a digital twin of a road at a certain location on earth is likely to be applied, not only for a single use case, but for several use cases at the same time, our approach defines a framework that can map the different geometric-semantic representations for all these use cases. A digital twin can thereby combine implications from different subdomains to establish a more complete and more valuable system of information.

Being able to specify necessary alphanumeric information as well as geometry at specific levels of detail, forms a relationship between different use cases and simplifies the process of collecting necessary data at an early stage. By specifying

an optimal representation to fit a use case, the complex system of a digital twin can be better structured. Additionally, stakeholders of specific use cases can evaluate and select the best representation for their application and simplify the definition of missing data. Another aspect that favors the use of a framework like proposed in this contribution is the possibility to store format-independent data which is directly converted into the required format when a specific model is requested. That makes it necessary to implement the process that generates representations from a common set of information, rather than converting between different format standards, where information loss is inevitable. Most existing approaches for automatically modelling roads, focus on directly generating models inside of a standard such as Justo et al. (2021). Since we focus on the application of multiple digital representations for multiple use cases of a digital twin, collecting the set of information and using it for model generation seems to be the better approach since it reduces loss of information and the effort of transforming between standards. While we have defined the general concept in previous work, we further specify the process of establishing the information base necessary for enabling the use of our concept for existing road infrastructure, where no digital representation is available in advance, in this contribution. We therefore conduct a case study as prove of concept in this work to optimize and validate considerations and design decisions met, focusing on the modelling perspective.

This paper is structured as follows: in Sect. 2, we discuss the general concept on the Level of as-is Detail for the geometric-semantic representation of digital twins of roads, we previously proposed on a conceptual level. In Sect. 3 we provide insight into the future process of capturing the real world environment and extracting necessary information in order to generate a geometric-semantic representation of a road. In Sect. 4 we show the results of our case study, we conducted as prove of concept from modelling perspective, where we chose a road segment, to evaluate the set of information necessary for deriving the representations defined in the concept. We then use this information to derive the different representations according to the concept, we proposed. The findings from the case study and also considerations for future work on our concept are the main takeaways of this contribution. In Sect. 5 we discuss our findings, propose meaningful changes to our concept that were identified during the case study and give an outlook on future work regarding this topic.

2 Level of As-Is Detail Concept for Roads

The concept for the Level of as-is Detail (LOADt) (Crampen and Blankenbach 2023) was initially derived from the existing concepts of LOD referring to both, the “Level of Development” from the BIM domain (Alshorafa and Ergen 2021) for construction and planning and “Level of Detail” from the GIM (geospatial information modeling) (Biljecki et al. 2014) as well as computer graphics domain (Heok and Daman 2004). Since a digital twin is a system of real and virtual components, the LOD definition from the BIM domain is not directly applicable, hence it is defined for procedural

models changing in the course of planning of a construction project. Nevertheless, the semantic aspect of BIM is crucial for digital twins, as they encompass not only the geometric component, but also the object types of specific geometric entities and the information associated with them. In BIM the “Level of Information Need” (LOIN) can be used for the definition of information requirements to a model for a specific use case or planning stage. This can be analogously applied in the context of a digital twin representation with the exchange scenario being the starting point of a specific use case, where a certain subset of information is collected and baked into a model that fits the requirements of a use case. The geometric side of the representation is drawn from the well-known GIM concept of “Level of Detail”, where the geometric complexity is defined in different levels, to reduce the size of the model if certain details are out of scope for a specific purpose (Abualdenien and Borrmann 2022).

Accounting for the semantic side we defined the “Level of Semantic Granularity” (LoSG) as a four layered term that holds a hierarchical object class structure which is defined for different functional units separately in order to allow a certain degree of freedom on the side of semantic granularity. We defined these hierarchies according to entities in the road environment that are available in different standards like CityGML (<https://www.ogc.org/standard/citygml/>), IFCRoad (<https://www.buildingsmart.org/standards/calls-for-participation/ifcroad/>) or OKSTRA (Bundesanstalt für Straßenwesen (BASt) 2020), which is the German object catalogue for road assets. We differentiated the functional units, road space, road furniture, traffic, vegetation, and civil structures, which account for engineering structures such as bridges and tunnels.

For the geometry side, we defined a “Level of Geometric Representation” (LoGR) for the geometric granularity of the different entities. This term is structured into three different types of representation, the first being “No Representation” denoted as LoGR 0. For the second and third type, we adapted the idea of different styles of representations from the CityGML 3.0 LoD concept (Beil et al. 2020). For the Transportation Complex in CityGML 3.0 there are two ways of representation, first a linear representation and secondly a planar representation. For the second type of our LoGR we defined a functional type, which is a set of linear and planar representations in four levels 0.1 to 0.4. The third type we call tactile, defines the 3D representations in four levels from 1 to 4. A separation between the geometry and semantic granularity was introduced in the LoD concept of CityGML 3.0, resulting in the possibility of representing models of varying semantic complexities on a single level of geometric representation. Comparable to the concept in CityGML the LoGR and LoSG are decoupled, to allow highly detailed geometrical representation while maintaining a low levels of semantic granularity at the same time. We also incorporated measures for the geometric uncertainty as well as the semantic uncertainty, which are important especially in the context of the digital twin, since representations derived from measurements in the real world environment, will always inherit a certain degree of fuzziness. When deriving a representation from data obtained through reality capturing technologies such as laser scanning or photogrammetry, there are several sources for geometric deviations and errors that have to be quantified in order of a representation to be valid for specific use cases. We call the geometric accuracy measure “Level of Geometric Uncertainty” (LoGU) and the

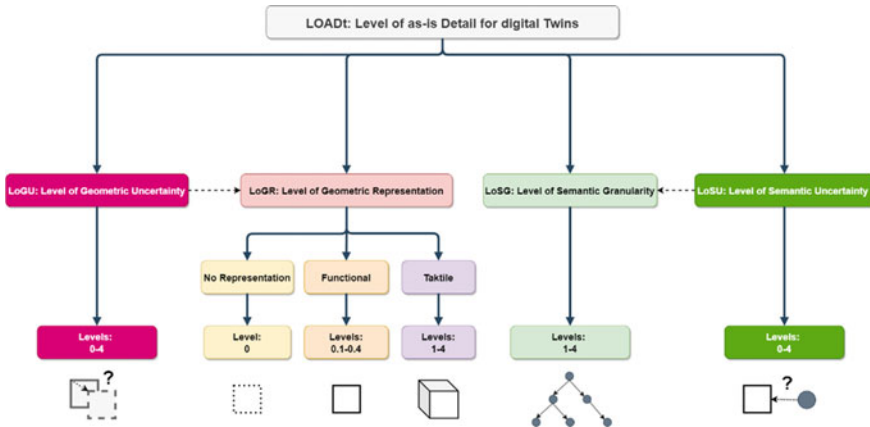


Fig. 1 LOADt-Concept for representations of digital twins of roads

measure for the semantic accuracy “Level of Semantic Uncertainty” (LoSU). The semantic uncertainty is introduced when data is processed and enriched with alphanumeric information such as object classes using methods such as semantic segmentation for point clouds. Depending on the use case a generated representation shall serve, the requirements of a certain semantic granularity as well as the knowledge of the inherent uncertainty of this semantic information can be specified. On the other hand, the geometric complexity of a model can be changed and for use cases with different requirements towards geometric accuracy the LoGU can be used as measure either as requirement for geometric accuracy necessary for a use case or to define the geometric data quality that is provided initially. Figure 1 depicts the terms of our concept.

The general idea of the concept is to serve as a framework for simplifying the specification of best fitting representations for various applications for digital twins of roads. We initially defined each level that is presented in each of the four terms, a schematic example of each level’s definition can be viewed in Table 1. Where the LoSG is an example for a single functional unit over different levels, since each functional unit forms a hierarchical tree of objects and is therefore hard to visualize in a single table.

Since the concept itself shall be applicable to any use case concerning the digital twins of roads, it shall remain open to modifications, such as additional Levels of the different terms, if there are new approaches developed, that allow higher accuracies, or require more differentiation in semantic granularity.

The scales for the LoGU and LoSU have not been validated yet and are therefore first estimates of feasible boundaries. In this contribution we will review the boundaries for the LoGU through our case study to validate and update them to boundaries that can be reached in the practical use of the concept. Additionally, since our concept to this point holds semantic and geometric components as well as quality measures for both, but lacks a clear definition of the necessary set of information required to

Table 1 Example for concept definitions

| Term/ level | LoGU | LoSU (%) | Level of semantic granularity | Level of geometric representation | |
|----------------|-----------|-------------|--|---|-----------------------------------|
| | | | Example: road space | Functional 0.X | Tactile X |
| Level 1 | 5–10 cm | 75 | Road space Only class road space | Centerlines and points | Minimum boundary primitives |
| Level 2 | 3–5 cm | 85 | Carriageways Separation of carriageways and lateral areas/junctions | Centerlines and minimal boundary polygons | Convex hull |
| Level 3 | 1–3 cm | 90 | Lane All visible classes in road space, lanes separate classes | Areal for all classes | Predesigned geometry |
| Level 4 | 1 mm–1 cm | 95 | Carriageway structure Additional hidden classes | 2.5D for visualization | Point cloud derived meshes |
| ... | ... | ... | ... | ... | ... |

automatically generate all possible representations, we will define this set of information as result of our case study as well. We will also briefly discuss the definition of topological relations between different entities in the road environment that are often required for generating models in specific formats such as OpenDRIVE or RoadXML.

3 Model Generation Process

In the following section, we discuss the process we are going to implement for deriving geometric-semantic representations for roads from the real world environment. The general approach roughly follows the automatic Scan-to-BIM idea, that has been a much regarded research topic in construction since a couple of years. Still, there is no robust comprehensive approach that can be applied to different constructions and the solutions that exist (Justo et al. 2021), mostly target small subdomains, while the high complexity in construction makes it hard to develop a generalized process for the full set of relevant assets (Badenko et al. 2019). One advantage of roads over buildings in that regard is the general logic that a road segment follows, in addition to the rather simple geometry of a roadway. Also, the reachability of outdoor structures is much higher than in an indoor environment, since they can be scanned in a large scale using airborne laser scanning (ALS) on different platforms or mobile mapping systems (MMS). A general limitation of laser scanning is that it cannot penetrate opaque surfaces and is therefore only able to capture the surface of a road, leading to the necessity of making assumptions or collecting data from other

sources to being able to model the substructure of a road. Despite these limitations, reality capturing is a necessary step to create digital as-is representations of real world structures.

The process starts with capturing the road environment with a UAV mounted mobile laser scanner. Depending on the solution employed for a survey and the distance to the target object, the resulting point cloud can reach an accuracy up to 1 cm (e.g. datasheet Riegl miniVUX-HA). Choosing UAVs in contrast to mobile mapping systems mounted onto ground vehicles has the advantage, that the result will be less occluded in most cases, since the UAVs point of view is much higher, thereby also capturing objects behind obstacles such as noise barriers or bulky vehicles on the road. Though generally less accurate and less dense than MMS the result obtainable with UAV-ALS is likely to be more complete (Pirotti et al. 2022) The process of capturing the environment introduces geometric uncertainties (e.g. caused by limited spatial resolution, measuring deviations or gaps in data) that have to be assigned to the LoGU measure. This limits the maximum reachable threshold for the highest “Level of Geometric Uncertainty” in our process to accuracies of 1 cm and larger values, indicating that even if there was no additional geometric uncertainty introduced in the further process, a representation can only ever reach the maximum boundary of Level 3 for the initially defined LoGU.

The next step of the workflow is the semantic segmentation of point clouds to separate different objects. To achieve this goal, there are many different approaches, reaching from purely rule based methods over algorithmic approaches such as region growing (Xie et al. 2020) to the most recent approaches where artificial neural networks (ANN) are applied on point clouds to perform semantic segmentation (Zhang et al. 2019). At this stage we will employ a hybrid approach, where we use prior knowledge from the field of GIM, to support a pre-segmentation step to narrow the field of interest, then use different rule based and algorithmic approaches to extract diverse features in advance to increase the information density of the point cloud. This augmented point cloud is then used as dataset for an ANN to perform the fine segmentation of the final object classes. These object classes will be drawn from the LoSG functional units. Since we set the LoSG up as hierarchy, there will likely be different models employed to extract objects from different levels. An approach, where a lower LoSG is segmented first and the resulting subset of the point cloud is used to segment the next higher level in the hierarchy, seems intuitive due to downward compatibility of object classes, but is likely to yield poor results, since model errors will propagate over each consecutive model layer. In this regard, the approach will likely consist of a first, coarse segmentation model, that will separate the classes within the functional units in the LoSG, which is then followed by separated models for each functional unit that extract the highest LoSG object classes for each functional unit. The highest LoSG classes can then be combined back into lower LoSG classes. This step introduces semantic uncertainties that are assigned to the LoSU. However, it is far from trivial to measure the total uncertainty in object class, since not only is the model prone to segmentation errors, it will additionally tend to over-confidence in its estimates. In that way, using a model that was validated on a known validation set, introduces additional uncertainties when dealing with unknown data,

which also has to be considered by the LoSU. In order of minimizing the influence of such epistemic uncertainties there are different approaches of model calibration, such as bayesian methods or ensemble approaches (Gawlikowski et al. 2021), that will be used to get a more accurate measure of the uncertainty that is likely to occur on unseen data.

We structure this process into two steps, with pre segmentation being step two after the capturing of point cloud data and step three being the semantic segmentation where the semantic uncertainty shall be measured. In the fourth step, the relevant information has to be extracted from the point cloud segments. This includes object positions, shapes, and references in form of the centerline of a road, carriageways or single lanes and their boundaries. Additional information for example concerning material or surface roughness are likely to be relatively hard to determine solely from point cloud data. Focusing on the essential information required to generate the representation, the set of information extracted and collectable in this step is crucial for the applicability of the LOADt concept as it is intended. Ideally, there will be a single set of information to serve as basis for deriving many potential representations that are relevant for the use cases of a digital twin. A starting point for this set of information will be set in Sect. 4, where we evaluate the necessary set for the different representations generated. Depending on the information necessary further geometrical uncertainty is added to the uncertainty introduced in step one, for example when fitting a spline to previously extracted centerlines or due to errors in the segmentation in the previous step.

The fifth step is the generation of the digital representation, which involves generating a model in a specific standard, or as combination of logically linked data to provide to a user with a specific demand. By generating geometry from a subset of information collected in step four, additional geometric fuzziness is introduced. On the other hand, the geometry can be compared to the initial point cloud to reduce the total geometric deviation, by a matching and transformation process.

Summarizing the geometric uncertainties that will occur during the whole process, there will be noise in the point cloud data, an additional uncertainty in the extraction of geometric information from the point cloud segments that depend on the semantic segmentation quality and generalizations that are introduced when generating the representation itself. On the semantic uncertainty side, the one step, where it will occur is step 3, where the information is extracted from the point cloud. The steps of our target process for generation of geometric-semantic representations is schematically depicted in Fig. 2.

4 Case Study

The following section will first introduce the data chosen to conduct the case study and will also clarify the simplifications met, since the focus of the case study is to validate the applicability of the concept introduced in Sect. 2 from modelling perspective. After this, we show results and specific examples drawn from the process

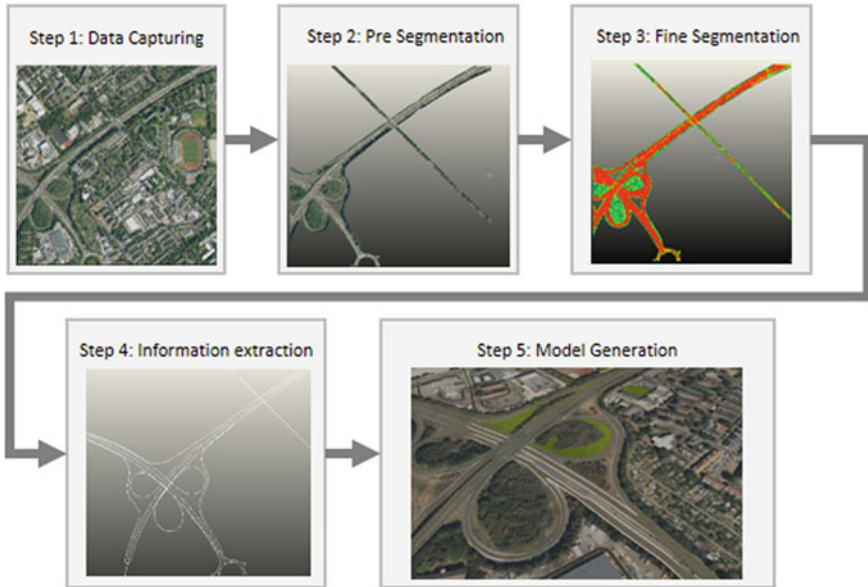


Fig. 2 Model generation process

steps applied in the case study. This section closes with an evaluation of the created geometric-semantic representations.

For the case study, we chose a highway, where point cloud data was already available through open data provided by the German government. The data was captured using ALS, resulting in a point density between 4 and 10 points/m². Since the open data is very sparse, for assets such as road furniture, the detail in the raw data is too low to allow the generation of suitable representations for our purpose. We therefore augment the data foundation with small MLS point clouds of assets such as shield gantries in the same target environment. Since both the ALS as well as the MLS data have been projected into the same spatial reference system, they can be combined to compensate their respective disadvantages. While the MLS data has information condensed very locally, the ALS data lacks detail, but covers a large spatial extend.

In the next step, we perform a pre-segmentation step, by coarsely cutting out the road environment, this can be done automatically using available geospatial data to automate that step. The data used is provided by ATKIS, which is Germany’s official topographic cartographic information system as open data in form of ESRI-Shapefiles, that are projected into the same spatial reference system as the point cloud data. This step reduces the computational cost of performing semantic segmentation significantly. Figure 3 shows the resulting coarsely segmented ALS data in comparison to the initial raw data, which lead to a reduction in data of over 94% from 58 million to approximately 3 million points.



Fig. 3 Step 2—coarse segmentation

Since our main goal is to test the applicability of our concept, we use the pre-segmented point cloud to manually extract centerlines and road furniture object positions, which simplifies the process by removing the semantic segmentation step. Given that step three involves the challenging task of accurately estimating uncertainties of the semantic class, it is considered the most complex step in the entire process. Therefore, in future work, we will concentrate on exploring the best possible approach for this step, while now prioritizing the modeling aspect. This leads to us only being able to specify the LoGU in this case study, whilst LoSU remains unevaluated for now.

By performing step 4 manually, a set of information was gathered consisting of information such as centerlines of the road space, number of driving lanes, or lane width. The full set of information is shown in Table 2. We defined two general types of objects that can be distinguished by their geometrical definition, these are point objects for objects like shield gantries or road signs, that can be positioned via a single point and line objects, that are defined by either a set of points, splines or some other curve definition. The point objects can be positioned with their orthogonal distance to an underlying line object, which is defined as the centerline of the overall road space.

During the extraction of the information set, several simplifications were met. This involves the prerequisite of a fixed number of driving lanes over the whole road segment, so no lane transitions for highway exits or turnouts were recognized. Furthermore, it was assumed that the centerline was extracted accurately in order to being able to quantify the deviations introduced by the definition of point object positions via orthogonal distance to the centerline. Given the centerline is accurate, the lowest positional accuracy of point and line objects can be calculated utilizing the minimal prescribed curve radius for highways in Germany, which is defined as 280 m for city highways, with respect to the centerline point density for a polyline definition. In that case the worst case scenario for positioning an object along the

Table 2 Set of information extracted for model generation

| Information | Description |
|--|---|
| 3D centerline | Line consisting of multiple point coordinates in 3D space, point density is the essential factor for the accuracy of the generated model |
| Number of lanes in each direction | Scalar values for the number of lanes in each direction |
| Lane width | Width of a single lane |
| Median width | Width of the road median if it exists |
| Start/end point of line objects + side of the road | Start and end points of guardrails, barriers and other line objects, that are positioned along the road |
| Position of point objects | The position of road signs, sign gantries and other point objects that are placed along the road |
| Object class | The class of a line or point object determining its function and shape |
| Object-wise point cloud segment for convex hull mesh | Single point cloud segments for generating a mesh directly from points (requires point cloud segments without large errors/outliers and high point density) |
| Layers of road substructure + layer thickness | The layers the substructure of the road consists of and their thickness (only approximated in case study) |

centerline is if its projection point on the centerline lies in the middle of two points defining the centerline. The resulting formulation for determining the maximum deviation is depicted in Fig. 4.

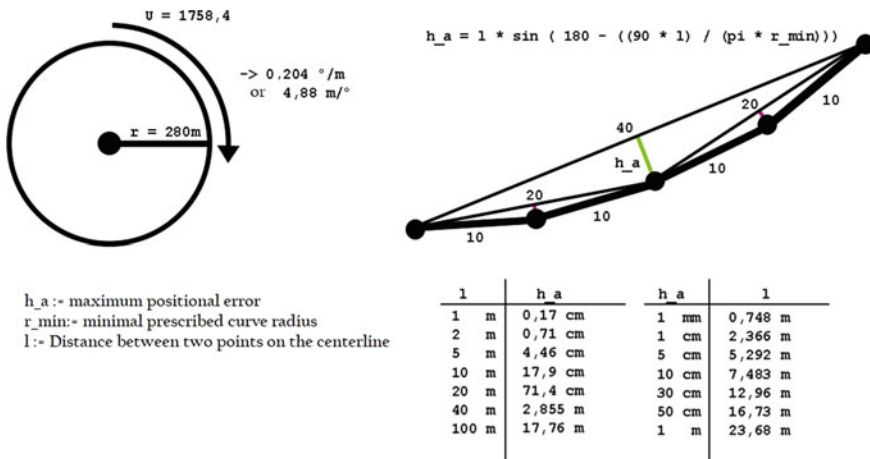


Fig. 4 Positional errors introduced depending on centerline density

According to this formulation, the positional accuracy obtainable for point objects depends on the density of the centerline, when using our modelling approach. This method of course has advantages as well as disadvantages. If the global position of an object is known, it can be positioned directly without the necessity of relating it to the centerline. Nevertheless, if each object is related to one underlying alignment, a topology can be encoded in the model directly by clearly referencing an implication on the road, such as driving rules or such. The referencing of objects to the underlying centerline also allowed for automatically obtaining the correct rotation of point objects, that otherwise would have to be extracted from the point cloud.

It can be noted that in the worst-case scenario the deviation of position for a centerline with a point density of one point per 2.5 m is about 1.3 cm, which is acceptable for large scale models spanning over 3 km like in our case study, while a centerline of that density is still obtainable through point cloud data.

Though quantifying the possible accuracy for positioning objects in the model is subject to more than this factor, we can hereby set the required density of the centerline with respect to our requirements in the LoGU as the component of the uncertainty introduced in process step five of model generation. The components of the full LoGU that are left are the accuracy of the extracted centerline and object position itself. This however requires step 3 which again is not subject to this contribution.

Next, we will show several examples of the actual models generated with our concept, where we focused on the road space and road furniture objects as those assets are the main targets of the concept.

As first impression Fig. 5 depicts a model in LoGR 4 and LoSG 4 in comparison to the underlying point cloud consisting of MLS and ALS data.

The model generation was implemented using SideFx Houdini, which is a Software for procedural model generation. The implemented process takes the set of information that was extracted from the point cloud to automatically generate the models defined in the concept. An example for part of the road segment analyzed in the case study over the LoGR functional representation type is shown in Fig. 6.

The functional units of road space and road furniture can have different LoSG as demonstrated in Fig. 7, where the road furniture remains on the same LoSG, while the road LoSG changes. This allows more variability in the resulting representation to better fit different use cases. Additionally, in the context of the target process, differences in the segmentation accuracy might occur between different object classes, which would also affect the potential representation of an object in the model. By detaching the different units, these differences can also be taken into account in model generation phase.

For road furniture multiple LoGR were modelled for each object class. The LoGR for a functional unit can be selected collectively. An example for stop signs, shield gantries and speed limit signs, is depicted in Fig. 8, where the tactile LoGR representation type was used.

The LoGR 4 in our concept is defined as the as-is representation meaning that objects shall be modelled true to deformation at best, which is not possible in this case due to the limited spatial resolution of the used point clouds. So, we initially

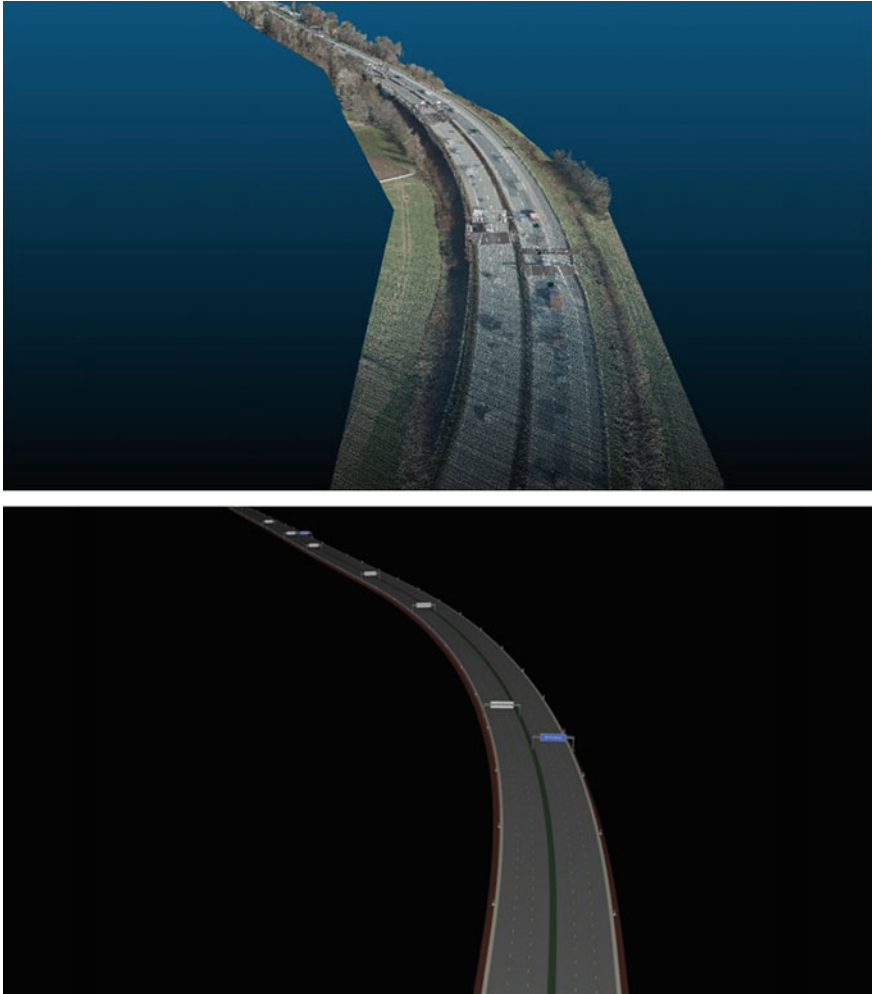


Fig. 5 Comparison of point cloud and model

generated the LoGR 4 without using the point cloud segments, since these had to be complete in order to yield good results.

A relatively simple modification would be using the point cloud segment of the road to deform the surface of the road geometry in a post processing step. In that way, we set the foundation for improvements towards more complex modelling steps in order to better mirror the real world object in the future. For road furniture it could be sensible to either further differentiate sub classes of different entities with varying shapes or implement a matching regiment that uses the point cloud segment to choose a pre-modelled asset depending on the maximum intersection. The simplifications

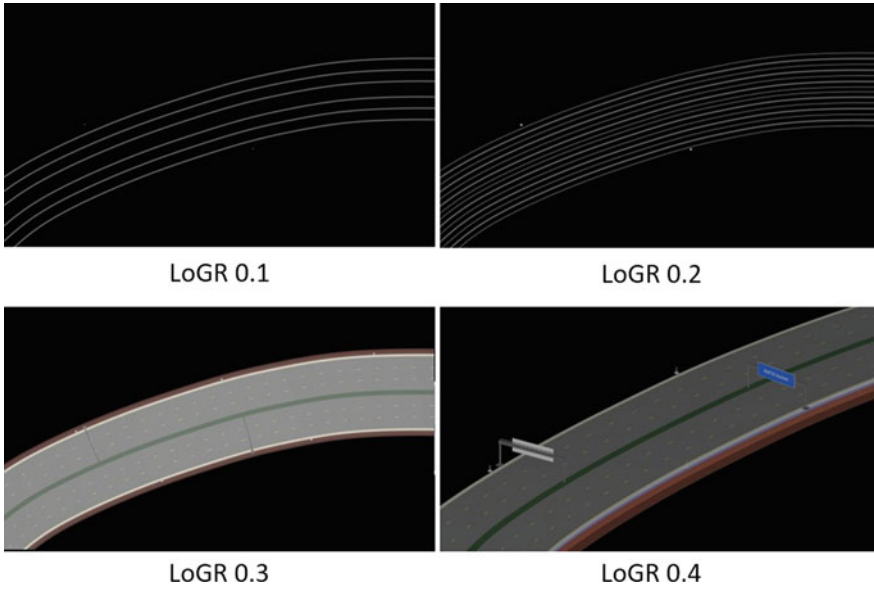


Fig. 6 Example for LoGR functional representation type 0.1–0.4

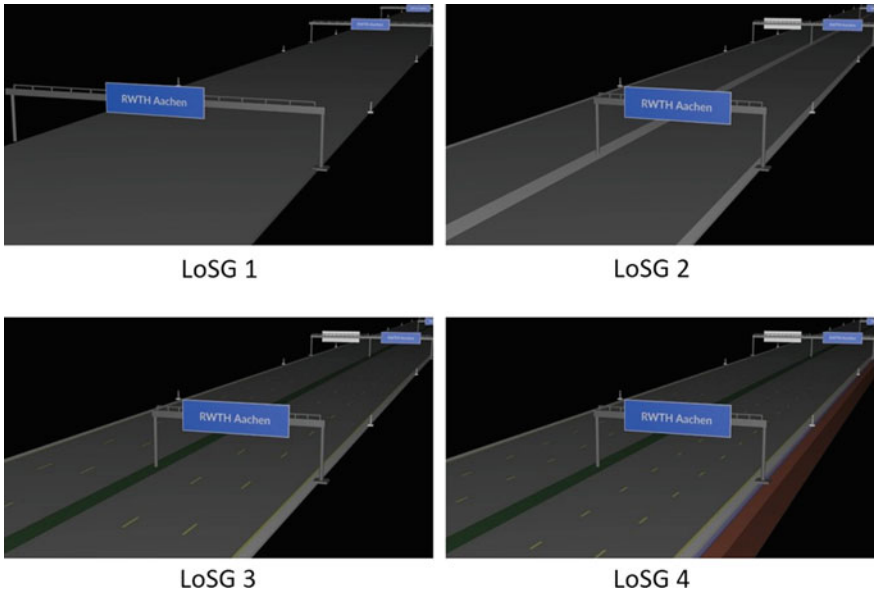


Fig. 7 Example for LoSG 1–4 for road space with constant LoSG 4 for road furniture

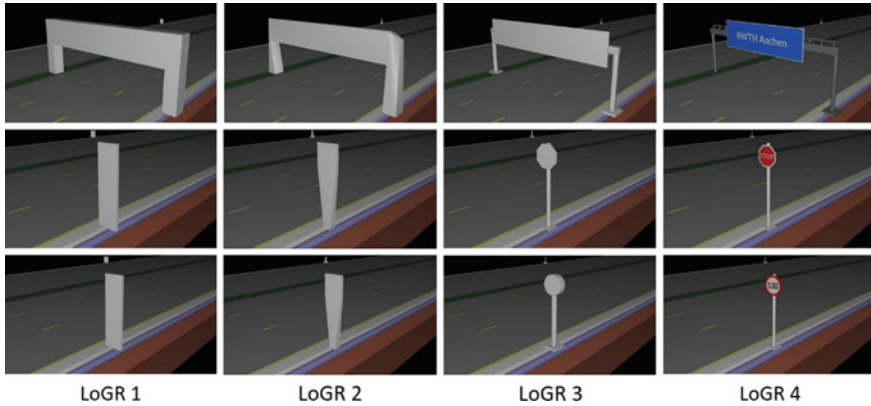


Fig. 8 Tactile LoGR 1–4 for road furniture with fixed LoGR 2 for road space

met in the case study like not recognizing turnouts and highway exits, will be added in future as well.

Although the final geometry generation process will have to incorporate several modifications, as discussed previously, we were able to practically apply the majority of our concept based on a limited set of information that can be extracted solely from point clouds. We were able to accurately position relevant objects in the representations and generate a functional representation of the target environment. In the next section we will summarize our findings and give an outlook on future work concerning the geometric-semantic representation of the digital twin of the road.

5 Discussion

The case study conducted demonstrates that our concept can be applied in modeling practice using a limited set of information, resulting in feasible geometric-semantic representations for several digital twin use cases. However, the study also revealed that for as-is models defined as LoGR 4, higher resolution point clouds are needed than used in the study. The point cloud resolution sets a limitation on models that are true to deformation, requiring a modified modeling approach. Nevertheless, the representations automatically generated by our modelling approach could be utilized for use cases, such as sustainability assessment or traffic simulation. Our findings in the case study have led to the rework of the LoGU thresholds and to distinguishing three different components of uncertainty adding up to the complete LoGU term. This allows for optimizing the single steps towards a better overall geometric accuracy. As discussed in Sect. 3 the components of geometric uncertainty involved in the overall process are: reality capturing, which can be approximated with knowledge about the scanner system used, geometry extraction from point clouds involving semantic segmentation and information extraction from segmentation results, which was not

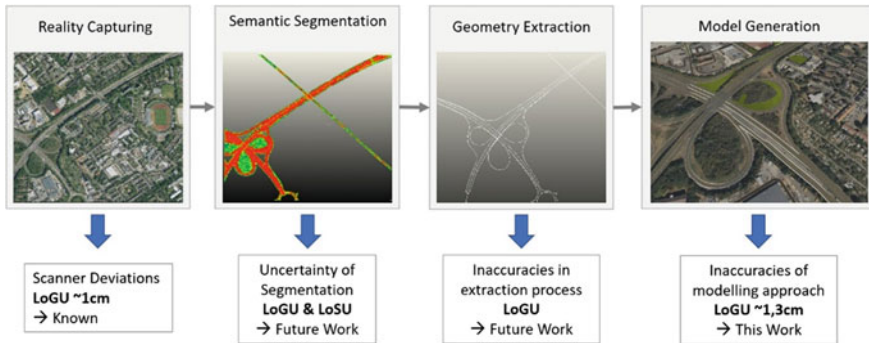


Fig. 9 LoGU and LoSU occurrence in automatic modelling process

addressed in this work and model generation, which we have quantified based on the underlying approach chosen for this work. These different components of LoGU and LoSU relevant in the overall process from reality capturing to model generation are depicted in Fig. 9.

For the overall uncertainty we assume that a geometric accuracy of up to 3 cm on average is achievable, by optimizing the geometry extraction and the model generation step. An essential part of optimization of the derived geometries is the validation with the underlying point cloud, since the deviations between point cloud and model is the sum of the deviations occurring in segmentation and modelling. Accordingly, the LoGU thresholds are redefined as follows: LoGU 1: 12–8 cm, LoGU 2: 8–5 cm, LoGU 3: 5–3 cm, LoGU 4: < 3 cm.

For the modelling part the comparison of point cloud and representation can be used to automatically position objects more accurately. Given the extracted centerline is accurate, the positional error could be corrected by determining curve radius of the road in a specific area and the point distances in the centerline to the projection point of an object to remove the deviation by translating the object position by the resulting offset. Another possible approach would be to directly use the extracted object positions to model them. In that case, however, the representation lacks a close topological relation between road and road furniture. This topological relation is required in many standards for driving simulation such as Vissim or OpenDrive, where road related objects are defined according to the centerlines of the road. Topology has not been addressed in our concept yet, but is crucial for the applicability for various use cases, such as driving simulation or disaster prevention. Also for representing the logical connections of lanes, transitions and turnouts a topological relation of various objects is necessary to establish more than a mere geometric-semantic representation and transition towards smart models that inherit the functional logic that applies in the real world. Incorporating these topological relations into our concept will be subject to future work.

Acknowledgements This research was funded by the German Research Foundation (DFG), as part of the Collaborative Research Center 339 (SFB/TRR 339) (project ID: 453596084). The financial

support from the DFG is gratefully acknowledged. We also acknowledge SideFX Software Inc. for providing us with their software products.

References

- Abualdenien J, Borrmann A (2022) Levels of detail, development, definition, and information need: a critical literature review. *Itcon* 27:363–392. <https://doi.org/10.36680/j.itcon.2022.018>
- Alshorafa R, Ergen E (2021) Determining the level of development for BIM implementation in large-scale projects. *ECAM* 28(1):S. 397–423. <https://doi.org/10.1108/ECAM-08-2018-0352>
- Badenko V, Fedotov A, Zotov D, Lytkin S, Volgin D, Garg RD, Liu M (2019) Scan-to-BIM methodology adapted for different applications. *Int Arch Photogram Remote Sens Spatial Inf Sci XLII-5/W2:S. 1–7*. <https://doi.org/10.5194/isprs-archives-XLII-5-W2-1-2019>
- Beil C, Ruhdorfer R, Coduro T, Kolbe TH (2020) Detailed streetspace modelling for multiple applications: discussions on the proposed CityGML 3.0 transportation model. *IJGI* 9(10):S. 603. <https://doi.org/10.3390/ijgi9100603>
- Biljecki F, Ledoux H, Stoter J, Zhao J (2014) Formalisation of the level of detail in 3D city modelling. *Comput Environ Urban Syst* 48:1–15. <https://doi.org/10.1016/j.compenvurbusys.2014.05.004>
- Bundesanstalt für Straßenwesen (BASt) (2020) OKSTRA 2.020
- Bundesministerium für Verkehr und digitale Infrastruktur (2021) Masterplan BIM Bundesfernstraßen V 1.0
- Crampen D, Blankenbach J (2023) LOADt: towards a concept of level of as-is detail for digital twins of roads. In: Proceedings of the 30th international workshop on intelligent computing in engineering (EG-ICE)
- Gawlikowski J, Tassi C, Rovile N, Ali M, Lee J, Humt M, Feng J, et al (2021) A survey of uncertainty in deep neural networks. Online verfügbar unter <http://arxiv.org/pdf/2107.03342v3>
- Guo X, Tian C, Chen Y, Zhang J (2022) Case study of building information modeling implementation in infrastructure projects. *Transp Res Rec* 2676(2):S. 663–679
- Heok TK, Daman D (2004) A review on level of detail. Proceedings. International conference on computer graphics, imaging and visualization, 2004. CGIV, Penang, Malaysia, 26–29 July 2004 (IEEE, S. 70–75)
- Justo A, Soilán M, Sánchez-Rodríguez A, Riveiro B (2021) Scan-to-BIM for the infrastructure domain: generation of IFC-compliant models of road infrastructure assets and semantics using 3D point cloud data. *Autom Constr* 127:S. 103703. <https://doi.org/10.1016/j.autcon.2021.103703>
- Pirotti F, Piragnolo M, Vettore A, Guarnieri A (2022) Comparing accuracy of ultra-dense laser scanner and photogrammetry point clouds. *Int Arch Photogram Remote Sens Spatial Inf Sci XLIII-B1-2022:S. 353–359*. <https://doi.org/10.5194/isprs-archives-XLIII-B1-2022-353-2022>
- Tao F, Zhang H, Liu A, Nee AYC (2019) Digital twin in industry: state-of-the-art. *IEEE Trans Ind Inf* 15(4):S. 2405–2415. <https://doi.org/10.1109/TII.2018.2873186>
- Xie Y, Tian J, Zhu XX (2020) Linking points with labels in 3D: A review of point cloud semantic segmentation. *IEEE Geosci Remote Sens Mag* 8(4):S. 38–59. <https://doi.org/10.1109/MGRS.2019.2937630>
- Zhang J, Zhao X, Chen Z, Lu Z (2019) A Review of deep learning-based semantic segmentation for point cloud. *IEEE Access* 7:179118–179133. <https://doi.org/10.1109/ACCESS.2019.2958671>

Digital geoTwin: A CityGML-Based Data Model for the Virtual Replica of the City of Vienna



Hubert Lehner, Sara Lena Kordasch, Charlotte Glatz,
and Giorgio Aguiaro 

Abstract This paper presents a CityGML-based data model developed for the semantic 3D city model of Vienna, Austria. The data model consists in a profile of the CityGML 2.0 standard and has been extended by means of an Application Domain Extension (ADE) developed by the Department for Surveying and Mapping of the City of Vienna in order to comply with the current and future needs of the municipality. The definition and adoption of such data model are a fundamental part of Vienna’s “Digital geoTwin” project. The core of the strategy is to process the 3D measurement data of the surveying and mapping department from existing as well as new measurement methods directly into a Digital geoTwin—a virtual, semantic 3D replica of all objects in the city—and to derive other geodata products (city map, elevation models, etc.) from this 3D model. Furthermore, the Digital geoTwin should serve as a geometric and semantic basis for a digital twin of the City of Vienna. In order to define the data model for the Digital geoTwin, 3D modelling of all city objects has been carried out in a test area of the city, followed by a mapping of the objects to the CityGML data model. In an iterative development process, conceptual gaps have been identified, analysed and eventually formalized into a UML-based Application Domain Extension. Additionally, the free and open-source CityGML 3D

This article was selected based on the results of a double-blind review of the full paper.

H. Lehner (✉) · S. L. Kordasch · C. Glatz

Department for Surveying and Mapping, City of Vienna, Muthgasse 62, 1190 Vienna, Austria
e-mail: hubert.lehner@wien.gv.at

S. L. Kordasch

e-mail: sara.kordasch@wien.gv.at

C. Glatz

e-mail: charlotte.glatz@wien.gv.at

G. Aguiaro

Faculty of Architecture and the Built Environment, Department of Urbanism, 3D Geoinformation Group, Delft University of Technology, Julianalaan 134, 2628BL Delft, The Netherlands
e-mail: g.aguiaro@tudelft.nl

City Database (3DCityDB) has been used for storage after being extended accordingly, and FME workbenches have been created to transform and import the original source data into the 3DCityDB and therefore test the suitability of the developed data model.

Keywords Digital geoTwin · Urban digital twin · Data modelling · 3D city model · CityGML ADE

1 Introduction: Geodata from Maps Towards Urban Digital Twins

Urban digital twins are a current trend in the digital transformation of cities. Thus, digital twin projects and initiatives are found in many cities in Europe and all over the world (Digital European Urban Twins (DUET),¹ Connected Urban Twins (CUT),² digital twin of Singapore,³ ...). Originating from Industry 4.0, the Digital Twin is defined a multi-physical and multi-scale representation of a complex system. It uses realistic models of the system, its environment and sensor data in order to mirror physical life of the system in the digital world and vice versa (Durão et al. 2018; Rosen et al. 2015).

Cities are highly complex and growing ecosystems. According to the United Nations it is expected that by 2050 the world's population living in urban areas will increase to 68%.⁴ Constant growth and global challenges like climate change reveal that new strategies such as urban digital twins are needed in order to manage and develop cities for further generations. Many data of many different disciplines and fields such as geodata, management data, sensor data, socioeconomic data, traffic data, etc. are necessary to build such urban digital twins. In our perception, an urban digital twin is not only a technical solution but also a new way of interdisciplinary collaboration. Thus, it is necessary to link many different expert knowledge systems in a virtual environment.

The resulting digital twin of the city can serve as a platform for various use cases and help simulate and understand actions before they are implemented in the real world. An urban digital twin is hardly built up from scratch but rather faces the challenges to develop and grow existing expertise and systems towards an interlinked virtual city.

Due to the complexity of the matter, the term urban digital twin is used in many different ways. While the wrong usage of the term—in our view—originates in some

¹ <https://www.digitalurbantwins.com/>.

² <https://www.connectedurbantwins.de/en/>.

³ <https://www.gim-international.com/content/article/singapore-s-journey-towards-a-nationwide-digital-twin>.

⁴ <https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html>.

cases from a too simple understanding or misinterpretation of the concept, in other cases it is due to marketing reasons. The term has been used as rebranding of existing products, technical solutions or software. Regarding geodata e.g. points clouds or textured 3D meshes have been named digital twin in some cases. These circumstances lead to the fact that the idea and concept of a digital urban twin is hard to define.

In the case of Vienna, there is a long-term vision to create a digital twin of the city. The vision encompasses several aspects related to digitalisation processes, the definition of new workflows to interact with it (both from inside the city administration, but also from outside in terms of services offered to the public or third parties), and the identification of applications and use cases exploiting it. Providing a detailed report on the overarching vision is beyond the scope of this paper.

Within the broader vision, the Department for Surveying and Mapping of the Vienna City Administration started a sub-project to support the Digital Twin of the City of Vienna. In order to work in a virtual environment of a city 3D models are necessary. However, they need to be more than just visualisation models. The objects need semantic information and further attributes in order to support a broader usage. Thus, geometric as well as semantic modelling is necessary.

The Department for Surveying and Mapping has been responsible for delivering and maintaining basic geodatasets for decades. The classic and historic evolution of geodata is similar to many European cities. A more detailed overview of the development in Vienna is given in Lehner and Dorffner (2020). The development usually started with (2D) digital city maps and 2.5D height models to 3D city models. The latter encompassing solely 3D building models, in the beginning, and adding more 3D objects such as 3D bridge models, 3D tree models, etc. later on. Each of these datasets is usually created by specialised software with specialised functions and workflows. Sometimes different input data and sources are used for each dataset, while in other cases—such as in Vienna—it was attempted to use input data multiple times in several product chains. In order to create detailed semantic 3D models of a city, all these different geodatasets are usually combined into one semantic 3D city model (see Fig. 1). Problems usually occur due to temporal incoherence of the datasets. While for visualization purposes, these problems might be neglectable, they have to be tackled in case the resulting 3D city model should serve as basis for an urban digital twin.

This approach does not seem sufficient in order to create data models which can serve as geometric foundation for an urban digital twin. Thus, a new approach for a Digital geoTwin was created in order to overcome the drawbacks of the classical, historically evolved way of geodata production. The centre of the strategy is to use all 3D measurement data of the Department for Surveying and Mapping from existing as well as new measurement methods in order to directly model semantic vector-based 3D geo-objects of the whole city. More generalised geodata products shall be derived from the Digital geoTwin (see Fig. 2).

An advantage of this strategy is that temporal and content-related coherence is achieved for all the geodata products, which are derived from the Digital geoTwin. The prefix *geo* was chosen for the neologism **Digital geo Twin** not only to emphasize our focus on the geodetic, geometric aspect of creating semantic geo-objects, but also

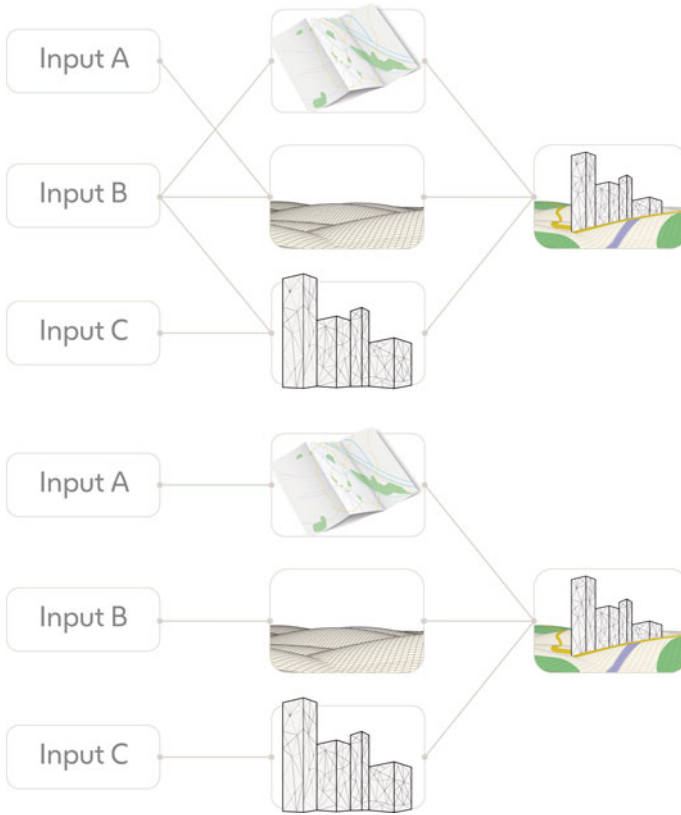


Fig. 1 “Classic” way of producing geodata sets and combining them to a 3D city model. Exemplary delineation of multiple usage of input data in several geodata sets (upper sketch) compared to separate input data for each dataset (lower sketch)



Fig. 2 Representation of the concept for modelling a Digital geoTwin and deriving other geodatasets from it

to distinguish our concept and the resulting model from the wide field of digital urban twins.

2 Digital geoTwin—A Virtual Replica of the City of Vienna

The concept of the Digital geoTwin was already published in 2020 by Lehner and Dorffner. The goal was to rethink the creation of geospatial data of the Department for Surveying and Mapping, detached from existing established systems and structures. Contrary to serial production of several datasets via different workflows and software, all input data shall be used once to create semantic vector-based 3D geo-objects of the whole city. Other datasets of the department shall be derived from this dataset.

With the concept of the Digital geoTwin, the Department for Surveying and Mapping of the City of Vienna is pursuing a unique and innovative approach for which, to the best of current knowledge, there are no comparable examples nationally or internationally. Accordingly, it was not possible to draw upon references. In order to verify the feasibility of the approach a proof of concept (PoC) was established. In the PoC, the focus was on determining whether technical solutions are available for this inverted approach and whether the concept can be implemented productively in the future.

When setting up the project, a deliberate distinction was made between inward-facing and outward-facing project objectives.

The inward-facing project objectives deal with the necessary components from the creation of 3D objects up to the automated derivation of geospatial data products:

- Modelling of semantic, vector-based 3D geo-objects of all objects of the city based on 3D measurement data;
- Usage of point cloud data from different technologies in the modeling process to support filling missing content in the 3D measurement data;
- Development of a 3D semantic, object based data model and a 3D object database for the storage of 3D semantic geo-objects;
- Conceptualization of the automatic derivation of geospatial data from the 3D object inventory.

The outward-facing project objectives encompass interfaces to application scenarios of a Digital Twin of the city of Vienna. They were considered from the beginning, as they may have implications on the creation and maintenance process of the Digital geoTwin. These goals include interlinking the semantic 3D city model with maintenance or management data or sensor data in order to increase the level of information of the individual objects and the whole virtual model to a great extent. Furthermore, it plays a significant role in breaking down data silos within the city administration and enhancing the quality of datasets, contributing significantly to the Data Excellence (DX) strategy of the City of Vienna (Lutz 2019). Another goal deals with the usage of the Digital geoTwin as basis in planning processes. Finally,

all parts together—the Digital geoTwin, interlinked domain-specific data and sensor data as well as planning data—can be used in various urban simulation processes.

Within the Digital geoTwin project a number of different sub-tasks and preliminary studies are being carried out. One of the sub-tasks has the main goal of identifying the requirements and then defining (and adopting) an underlying data model for the semantic 3D city model that will comply with the overall “Digital geoTwin vision”. The developed data model is based on the international standard CityGML 2.0, from which a profile has been extracted, on the one hand, while additional features have been added by means of the Application Domain Extension (ADE) mechanism. The specific parts regarding the generation of the ADE have taken inspiration from existing previous experiences, as described in Agugiaro et al. (2018), Kumar et al. (2019) and Ying et al. (2022). A rather exhaustive overview of CityGML ADEs can be found in Biljecki et al. (2018). However, at least in the authors’ opinion, there are not so many previous experiences regarding a continuous 3D modelling of all objects of a city and an extensive mapping of these objects to nearly all CityGML modules as in the case of Vienna.

3 Digital geoTwin Data Model Requirements

As a result of the first data and user analysis, the following requirements have been identified when it comes to the semantic 3D city model of Vienna, also considering the type of products to be obtained from it. The 3D city model should offer the possibility to:

- Store not only vector-based 3D geometries, but also semantics for each object. This applies also to features generally needed to generate a digital terrain model. For example: distinction between a river bed and the river surface, identification of the terrain surfaces that existed before being covered or built up (e.g. by a building);
- Store objects represented in different Level of Details (LoD);
- Decompose complex objects into smaller components (e.g. buildings and building parts);
- Model and store objects that might be repeated/cloned multiple times by means of templates (e.g. a park bank, a traffic light, or a bus station);
- Store all geometries natively in 3D. In particular, in case of non-planar surfaces, it must be possible to derive them in form of semantic 3D meshes. For example, this applies *also* to terrain models which may contain features that cannot be modelled in 2.5D (such as vertical elements, or overhangs);
- Automatically derive 2D maps or 2.5D height models, which contain a seamless tessellations of space. This means, for example, no holes in a terrain model due to buildings being “taken out”;

- Store some metadata at the level of each object, regarding for example the geometrical accuracy resulting from the surveying campaign, or information about the creation timestamp and the validity/existence interval of time of a given object.

Such shift in the 3D city modelling process does not only call for a revision of the data ETL (Extract, Transform and Load) and integration strategies, but also for the identification of a suitable data model able to store such a 3D city model. Given the above listed set of requirements, the open and international standard CityGML was taken into consideration in order to evaluate its suitability.

3.1 Evaluation of CityGML

Thanks to its availability as open data model, its growing world-wide adoption by several international cities, and its set of characteristics that already align very well with the needs of the Department for Surveying and Mapping, CityGML was considered as the candidate data model for the Digital geoTwin of Vienna.

Despite the availability of CityGML 3.0 in terms of the conceptual model (Kolbe et al. 2022), it was decided to test and evaluate CityGML version 2.0 instead (Gröger et al. 2012) because of the current availability of software tools to work with it, such as the 3D City Database suite (Yao et al. 2018), which includes the free and open-source 3DCityDB and the accompanying Importer/Exporter tool. At the same time, enough documentation and experiences regarding ADE modelling and the associated tools already exist, while the same cannot be said about CityGML version 3.0 to the same extent, yet.

In general, CityGML is a modular and well-structured standard which comes with a set of modules (and their relative classes) that already cover most objects of a city, as represented in Fig. 3. At the same time, it is extendible, i.e. it can be customized by means of the Generics module, or via ADE. At the same time, if certain features (or feature attributes) are not needed, they can be omitted in a subset of CityGML, which is a so-called *profile*.

In the case of Vienna, both strategies were adopted: some features (or some of their attributes) were removed, therefore defining a profile, and at the same time some features were added by means of the ADE mechanism. Although the details of the Digital geoTwin data model will be given in the next chapter, the overarching design decisions will be described here.

3D modelling of all city objects has been carried out in a test area of the city (see Chap. 4.1), followed by a mapping of them, as far as possible, to the closest CityGML equivalent. This means that an analysis was carried out in terms of semantics, but also in terms of available (or missing) attributes, available LoDs and geometry types (solid vs multi-surfaces, implicit vs explicit geometries, etc.). For certain attributes, codelists were defined.

One of the aspects that required major attention was the “mapping” of the digital terrain objects. Given the necessity to store fully 3D objects (e.g. vertical walls), this

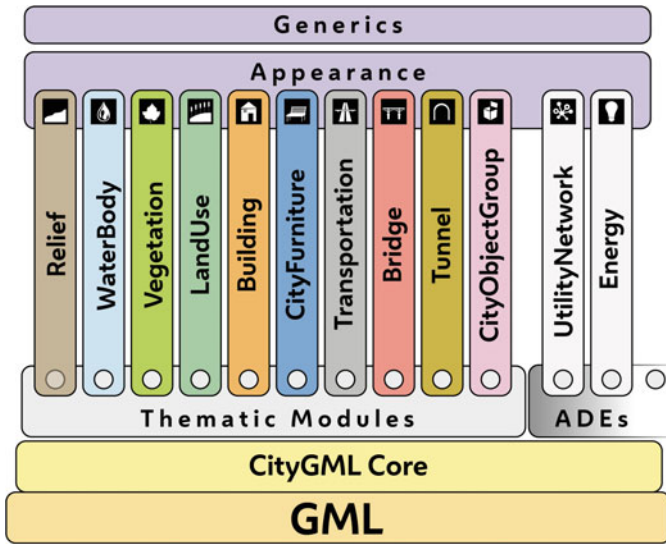


Fig. 3 Overview of the CityGML 2.0 thematic modules. Image adapted from Löwner et al. (2012)

automatically ruled out the use of CityGML’s Relief module, which allows only for 2.5D triangulated meshes in terms of polygon-based geometries.⁵

As a result, a major change in the mapping and modelling rules was introduced for “terrain objects”: the Relief module was discarded and the LandUse module was chosen instead as it is able to store 3D meshes. Additionally, semantically different “terrain” objects such as transportation or waterbody objects have been mapped directly to the corresponding CityGML modules.

When it comes to the intersection between a building (or bridge) and the terrain, the CityGML concept of *TerrainIntersectionCurve* was adopted to store the resulting 3D polyline representing such intersection. However, in order to also store the actual portion of digital terrain model “cut out” by a building (or a bridge) the additional class *LandUseClosureSurface* was added via the ADE mechanism. More details will be given later. The *LandUseClosureSurface* concept extends the CityGML *ClosureSurface*. ClosureSurfaces do not exist in reality and are mainly there to virtually close volumetric objects. They are for example already intended for the Tunnel module. Here the ClosureSurface is used to virtually seal the tunnel portal. The outline of the portal defines the boundaries of a 3D surface which is then in instance of the *TunnelClosureSurface*. A different usage can be found in the WaterBody module. The *WaterClosureSurface* intends to virtually define the end of the modelled area of bigger water bodies in case they exceed the area of interest (e.g. outside the boundaries of the city). The *WaterClosureSurface*, thus, can be defined arbitrarily without any representation in nature.

⁵ Please note that 3D breaklines and 3D masspoints are also supported, but they were considered not suitable for the needs of Vienna.

In order not to store geometries twice, which are deployed in more than one object, CityGML incorporates the xLink concept of the GML-Standard (Gröger et al. 2012). The geometry is stored in one object and can be linked to another object via xLink. Building models can be modelled via thematic surfaces (wall, roof, ground, etc.) and xLink the surfaces to the volume of the building. Another usage is to xLink facade surfaces, which are shared by adjacent buildings. The standard furthermore recommends in the LandUse module to store line segments of adjacent surfaces only once and xLink them from one polygon to the other. While the first example of linking thematic surfaces to volumes is commonly used the other two examples are hardly used in practise. The concept of such complex hierarchical data structures are beneficial for updating city-wide semantic models, as intended with the Digital geoTwin. However, they are hardly suitable for exchanging semantic models. Most software products simply do not support models which are modelled in such complex hierarchical way. Thus, the hierarchical data structure was just used while modelling the data but not when mapping the objects in CityGML.

Regarding thematic modules, CityGML 1.0 was already quite comprehensive. Only the modules bridge and tunnel were missing compared to CityGML 2.0. Still, the focus of 3D city models is the building module. Many software solutions focus solely on 3D building models and not much on other thematic modules. Thus, 3D building models were simply understood as 3D city models. This can also be seen in freely available CityGML datasets. CityGML Wiki,⁶ Github⁷ and the 3D Geoinformation research group of TU Delft⁸ links data sets of cities, regions and in some cases even nations. The datasets which can be found there contain only building models in many cases. Next to buildings some datasets also provide a terrain, tree or transportation models. These reference lists do not claim to be exhaustive and do not show whether there are other datasets, which do use more thematic modules. Still they give a good overview on how and where CityGML is used in practise.

The following chapter will provide further details about the Vienna semantic 3D city data model.

4 The Digital geoTwin Data Model for the Semantic 3D City Model of Vienna

As already mentioned before, one of the main purposes of Vienna's 3D city model is to allow for coherent and integrated representation and storage of all 3D city objects being modelled based on data from different surveying techniques and campaigns. Additionally, it is meant to represent the unique source of all different geodata products that are otherwise provided and used by the City of Vienna. This includes for example the automatic derivation of a city-wide DTM which must geometrically be

⁶ https://www.citygmlwiki.org/index.php?title=open_data_initiatives.

⁷ <https://github.com/OloOcki/awesome-citygml>.

⁸ <https://3d.bk.tudelft.nl/opendata/opencities/>.

closed and topologically correct. In other words, the semantic 3D city model will contain all urban 3D geo-objects, but still, a closed (i.e. “hole-free”) surface can be derived from it to represent just the terrain.

From the development and implementation point of view, the procedure followed to define the data model was based on the following main steps:

1. 3D modelling of all city objects based on existing surveying data and point clouds in the test area;
2. Mapping of the city objects to the CityGML data model, identification and analysis of correspondences and gaps;
3. Extension of the CityGML data model by means of an ADE in order to cope with the specific additional requirements stemming from the previous analysis. Formalisation of the ADE by means of a UML class diagram and generation of the resulting XSD file. The UML class diagram was created in Enterprise Architect, the XSD file was generated by means of ShapeChange, following a similar approach as described by Van den Brink et al. (2013);
4. Generation of DDL database scripts to extend the 3DCityDB and add support for the ADE. Please note that this functionality is already available with version 4.x of the 3DCityDB;
5. Definition of pipelines to convert the source data into CityGML and to import them into the extended 3DCityDB. These ETL procedures were developed using FME 2022;
6. Test of the developed data model and the ETL procedures using the datasets collected for a chosen test area in Vienna;
7. Evaluation of the results.

Please note that, due to the exploratory nature of this work, there have been some iterations in the steps listed above as the process is not completely straightforward.

The following subsections will provide further details on how the CityGML 2.0 data model was adapted in order to comply with the overall requirements described in the previous chapter. To ease readability, first the test area in Vienna will be presented and then the specific mapping decisions regarding the 3D modelling results of the test area will be discussed. For the sake of logical simplicity, the overall structure of CityGML in terms of modules will be followed.

4.1 Test Area

A test area of circa 0.314 km² was defined in the third district of Vienna along the Danube Canal near the Stadionbrücke (see Fig. 4). In this area, a good heterogeneity of urban fabric can be found, ranging from small garden settlement to densely built residential area, forest, waterbodies, irregular terrain, etc. (see Fig. 5).

The available geospatial datasets consist of surveyed 3D points or 3D polylines, associated attributes and codes. An example is given in Fig. 6. All datasets, their characteristics and metadata were first collected, then used to model all city objects,



Fig. 4 Map of Vienna (left) with highlighted position of the test area (right), which is represented in detail (*Source* Stadtplan3D map—Department for Surveying and Mapping Vienna)



Fig. 5 Oblique aerial view of part of the test area (*Source* oblique image 2020—Department for Surveying and Mapping Vienna)

which were finally used as foundation for the development of the data model. The mapping between the 3D objects in the test area and the CityGML classes was carried out manually, while the transformation from the source datasets into CityGML files was carried out using FME.

4.2 LandUse, Transportation and WaterBody Modules

Objects, semantically belonging to these CityGML modules, were generated from 3D points and polylines as 3D meshes. 3D points come from laserscanning or from classic topographic surveying. The 3D points can have a regular as well as irregular

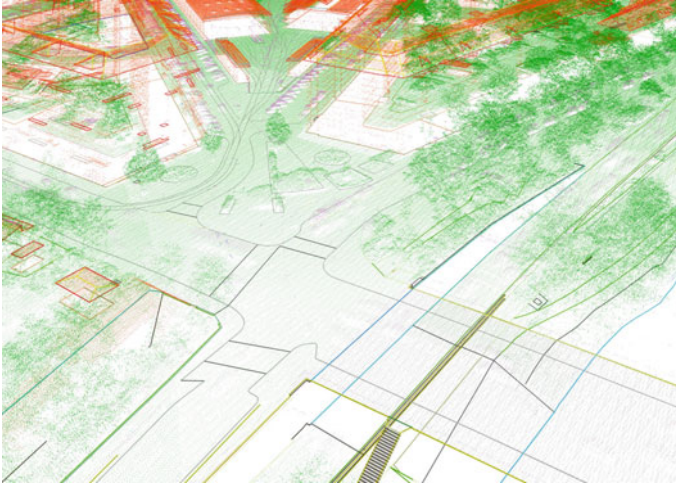


Fig. 6 Example of data available in the test area: 3D surveyed data overlaid with a classified ALS-based point cloud (Source Department for Surveying and Mapping Vienna)

pattern. For example, vertical walls can be represented by means of vertical surfaces without the need to change the underlying data model, as LandUse, Transportation and WaterBody modules already allow to model geometries as MultiSurfaces (instead of TINs), thus overcoming the 2.5D limitation (see Fig. 7).

As in general, there are no clear indications in the CityGML specifications regarding which level of detail to use with the LandUse model. Thus, it was decided to use only LoD2. Unneeded attributes and properties/associations to unnecessary LoDs were therefore removed (see Fig. 8).

When it comes to the Transportation module, all existing classes and subclasses (TrafficArea, AuxiliaryTrafficArea) were used to map and preserve the semantic of the objects available in the test area.

Regarding the WaterBody module, the existing classes and attributes were used to model both waterbodies for which 3D data are available (e.g. the Danube canal) and minor affluent streams (see Fig. 9). An example is given in Fig. 10. In the case of the Danube canal, the ClosureSurface classes were used to seal the river volume contained between the WaterSurface and the GroundSurface.

As mentioned before, in order to create a sealed (e.g. hole-free) “terrain” model once the 3D objects (e.g. buildings) are removed, a new class, called LandUseClosureSurface was defined by means of the ADE mechanism. The class LandUseClosureSurface is associated with the corresponding CityObject it substitutes, and consists geometrically of a 3D mesh. In the case of a building, for example, the LandUseClosureSurface will be a surface whose boundaries coincide with the TerrainIntersectionCurve of that building (see Fig. 7).

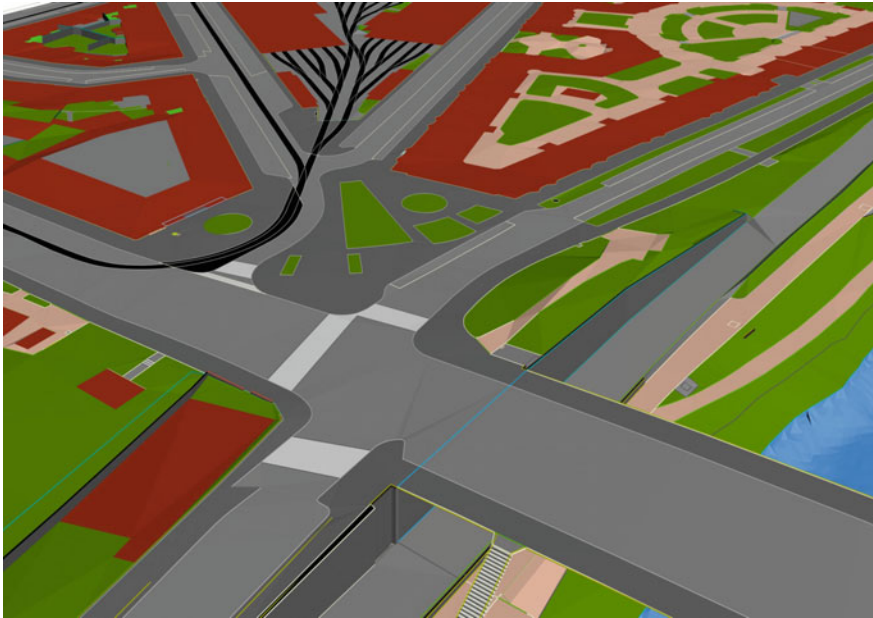


Fig. 7 Example of objects found in the test area modelled using classes from the CityGML LandUse, Transportation and WaterBody modules. Additionally, LandUseClosureSurface objects are represented in red

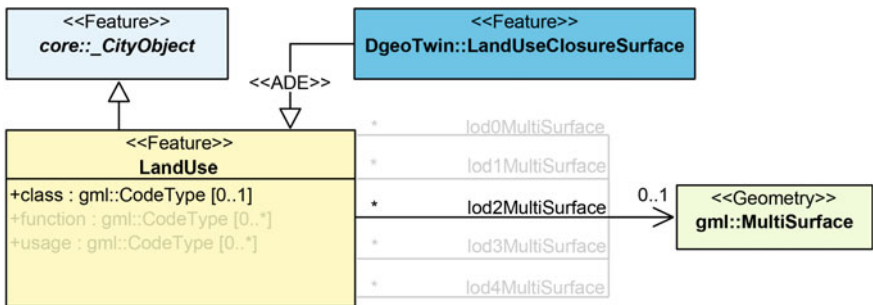


Fig. 8 Class diagram of the Digital geoTwin data model for the LandUse module, consisting in the profile derived from the CityGML corresponding module and the ADE class represented in dark azure. Original CityGML properties not used are represented in light gray. Image adapted from Gröger et al. (2012)

Finally, in order to store information about the status of the surface sealing, an additional attribute (SurfaceSealing) was added via ADE to the core module (see later).

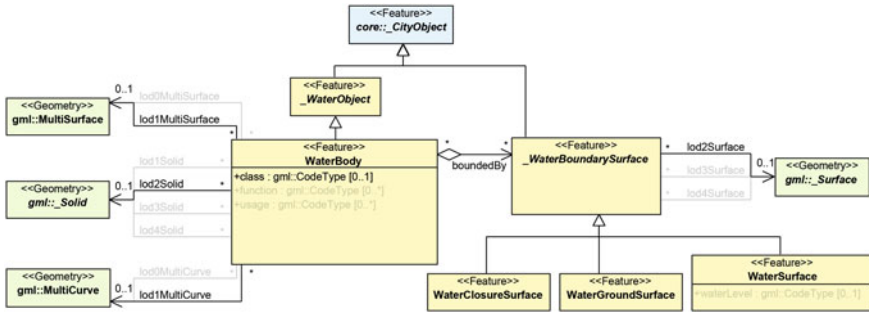


Fig. 9 Class diagram of the Digital geoTwin data model for WaterBody module, represented using the same approach as Fig. 8. Image adapted from Gröger et al. (2012)

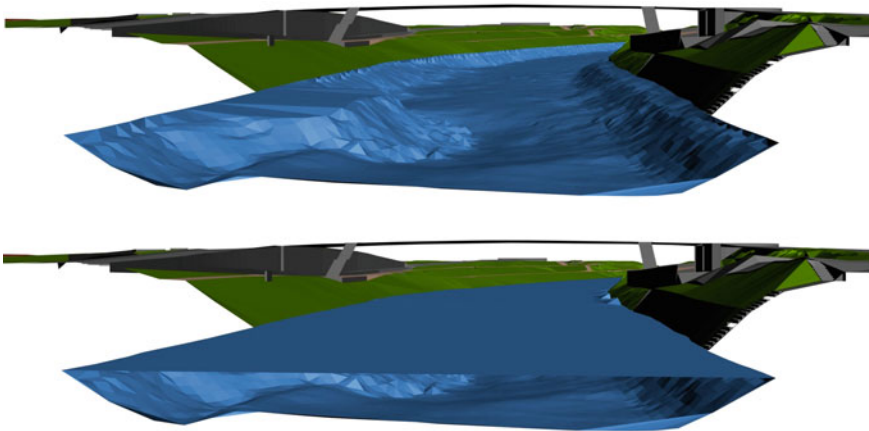


Fig. 10 Example of semantic 3D modelling of the Danube canal in the test area: WaterGroundSurface (top), WaterGroundSurface and WaterSurface (bottom)

4.3 Vegetation and CityFurniture Modules

For both the vegetation and city furniture modules, similar mapping and modelling rules apply. In general, CityObjects are generated automatically from the surveyed source data and both explicit (BRep or solids) or implicit geometries are used, depending on the type of CityObject.

In the case of the Vegetation module, the class SolitaryVegetationObject is used to model single trees, while bushes and cluster of trees are mapped to the PlantCover class (see Fig. 11). For single trees, as the source data only consist of surveyed points and some attributes, the modelling approach consists of using different tree templates and model them geometrically using implicit geometries. Different templates are defined also depending on the LoD (see examples in Fig. 12 covering LoDs from 0 to 3). For the LoD0 implicit representation, this had to be added via ADE as it is not

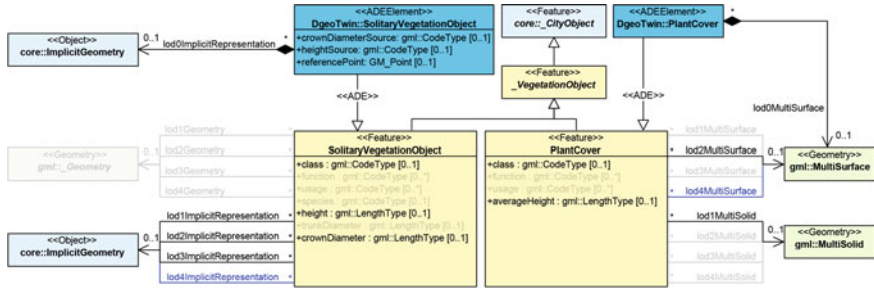


Fig. 11 Class diagram of the Digital geoTwin data model for the Vegetation module, represented using the same approach as Fig. 8. Image adapted from Gröger et al. (2012)

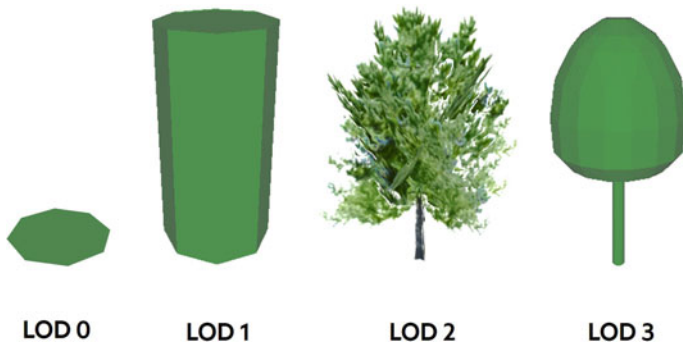


Fig. 12 Example of LoDs for SolitaryVegetationObject trees belonging to class „Laubbaum“ (deciduous tree)

included in the default CityGML class. Attributes are mapped either to the existing or to new classes added via ADE. In particular, a customized codelist was created for attribute “class”.

The PlantCover class is used to model bushes and trees groups, for which it is not possible to distinguish the single plant (see Fig. 13). Source data attributes are mapped to existing ones of class PlantCover (again, a customized codelist was created for attribute “class”), while for geometries, the following rules were defined:

- LoD0 represents the “footprint” of the bush or tree group and is modelled as MultiSurface
- LoD1 represents the closed (solid) geometry of the vegetation group. The height is the average height computed from the corresponding normalised DSM
- LoD2 is used instead only to model (as MultiSurface) the 3D mesh resulting from the vectorization of the corresponding DSM area covered by the PlantCover object.

Similarly to the SolitaryVegetationObject, the LoD0 geometries had to be added via ADE.

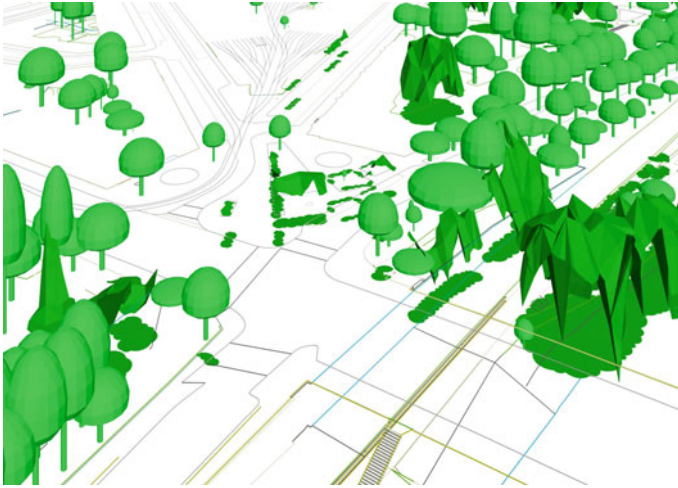


Fig. 13 Example of Vegetation objects in the test area, modelled using both SolitaryVegetationObject and PlantCover classes

When it comes to the CityFurniture module, this was used to represent city furniture objects, although some data transformation rules had to be defined to convert data from the source data accordingly. As the source data consists mainly of surveyed point, line and area objects, enriched with some attributes regarding dimensions and thematic data, these criteria were defined (see Fig. 14):

- CityGML “class” and “function” attributes are used to map the thematic attributes. Customized codelists are created therefore
- Source attributes regarding height, diameter, azimuth were used to generate implicit geometries and stored as metadata via additional ADE attributes

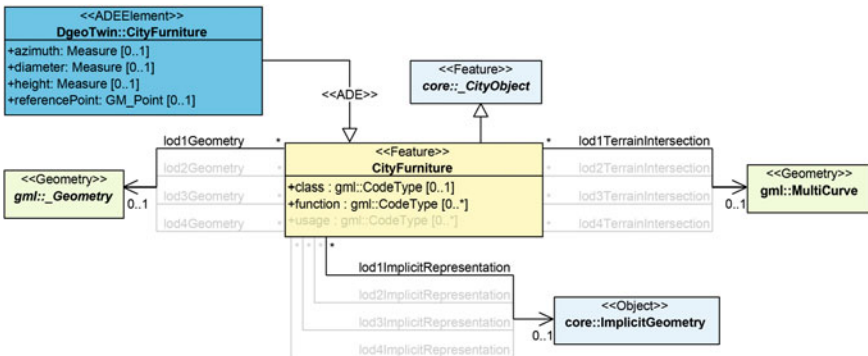


Fig. 14 Class diagram of the Digital geoTwin data model for the CityFurniture module, represented using the same approach as Fig. 8. Image adapted from Gröger et al. (2012)

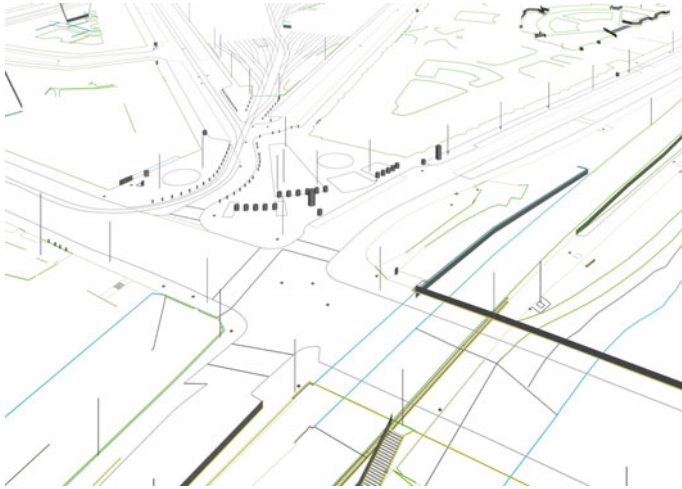


Fig. 15 Example of objects in the test area, modelled using classes from the CityFurniture module

In particular, regarding geometry:

- Only LoD1 was used
- For point-based objects, either implicit geometries for simple shapes (cylinder or box) or simple surfaces were generated, using the dimensional attributes mentioned before for scaling. Volumetric templates are used for objects like masts, bollards, etc. while simple surface templates are used, for example, for manhole lids.
- For line-based objects (e.g. fences or railings), multi-surface geometries are used. Depending on the origin of the information about their height (photogrammetry or terrestrial survey) they are vertically extruded upwards or downwards
- For area-based objects (e.g. the footprints of phone booths), the surveyed 3D multiline is stored as *TerrainIntersectionCurve*. Additionally, the 3D polyline is then flattened to the lowest surveyed height and stored as polygon in order to be extruded using default values depending on the type of object.

An example of the CityFurniture objects is given in Fig. 15.

4.4 Building, Bridge and Tunnel Modules

The last group of thematic modules deals with objects such as buildings, bridges and tunnels. An extensive study regarding building LoDs has already been published (Lehner and Dorffner 2020). The study is based on the works of Biljecki et al. (2016) and extends the therein-included refined specifications for the LoD concepts of buildings. Of particular relevance for Vienna is the systematic classification of buildings depending on their geometric characteristics, i.e. in case of architectural

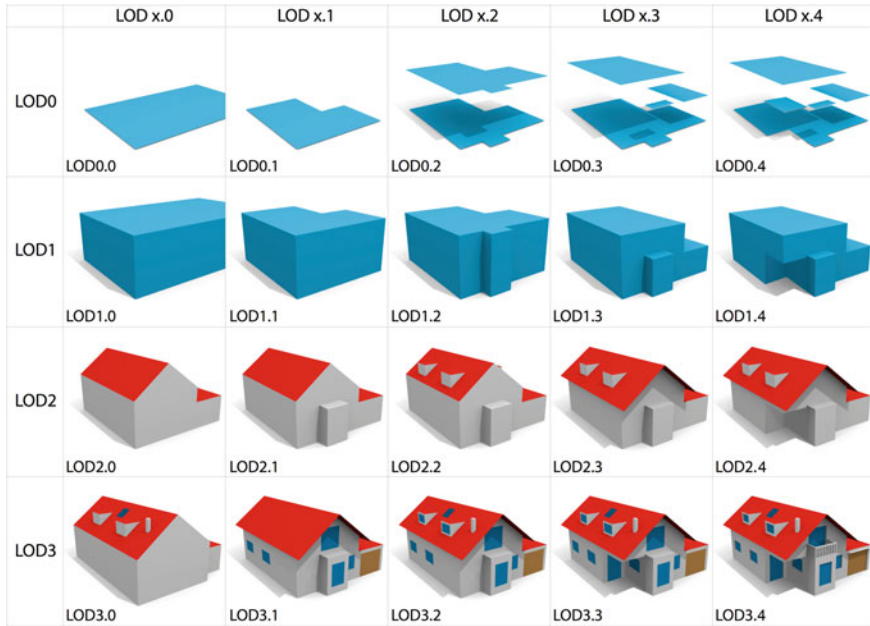


Fig. 16 Refined LoD concept for Building according to Lehner and Dorffner (2020), further developed from Biljecki et al. (2016)

elements such as arcades, passages, bay windows, etc. which eventually best fit with the level 2.4 of the new proposed LoD schema (see Fig. 16).

The underlying idea is to have the building geometry stored at the highest available level of detail, and to procedurally derive lower-LoD models by means of 3D generalization algorithms. At the same time, the LoD2.4 is considered the most future-proof in order to add further details like openings (e.g. windows and doors) whenever they will be available (and thus augmenting the LoD to 3.x). It is noteworthy to remind here that the *TerrainIntersectionCurve* of a building must correspond with the outer boundary of its *LandUseClosureSurface*.

In the case of the Bridge module, the bridge contained in the test area was mapped without major issues to the corresponding CityGML class. For the geometry, the chosen level of detail was LoD2. Again, analogously to the Building module, the *TerrainIntersectionCurve* of the bridge must correspond to the outer boundary of the associated *LandUseClosureSurface*.

An example for buildings and a bridge in the test area is given in Fig. 17.

Although in the test dataset no tunnels were contained, a theoretical mapping was carried out based on the CityGML Tunnel module. However, as already mentioned, a *ClosureSurface* class is indeed already available for the Tunnel module. Thus, in order to support software which uses the tunnel module, the *LandUseClosureSurface* can be *xLinked* to the *TunnelClosureSurface*.

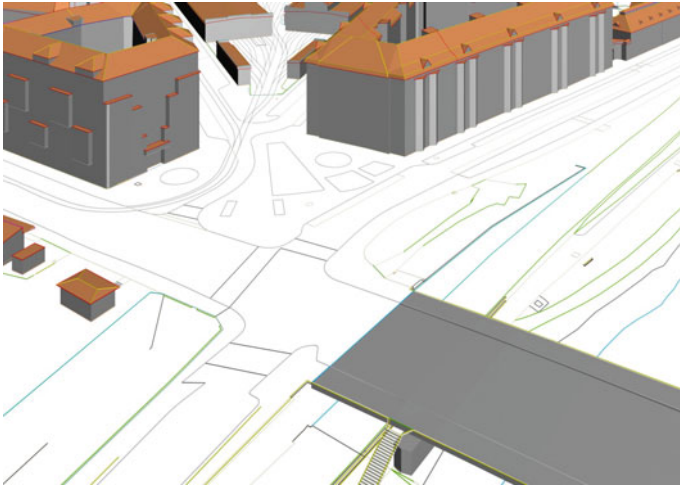


Fig. 17 Example of objects in the test area, modelled using classes from Building and Bridge modules

4.5 Core Module

The CityGML core module contains base classes that are then used (or inherited) by the depending thematic modules (see Fig. 18). It is therefore also the most logical module where to add ADE elements and properties that will be then likewise inherited by all subclasses (see Fig. 19). This applies in particular to the abstract class CityObject. In the case of Vienna, the core module was used to extend the CityObject class by a number of properties, mostly due to the need to store metadata of each object in terms of temporal and geometrical accuracy information. These metadata accompanying the original surveyed data (e.g. points and lines) which are then aggregated separately and associated to the resulting cityobject. In terms of temporal quality information, this is expressed as timestamps when a certain horizontal and vertical value was surveyed, and by means of codelists that define vertical and horizontal accuracy values. Such codelists contain values used in the data processing pipeline, but can be linked to external look-up tables.

Finally, metadata are also stored with regard to the last temporal interval check carried out on the constituent geometries of the cityobject, and finally the timestamp associated to the creation of the 3D geometry of the cityobject.

Finally, the SurfaceSealing property is used to characterize the sealing type of the surface, and it is associated with a codelist, too. This attribute is meant to be used by all modules containing “terrain objects”, i.e. LandUse, Transportation and WaterBody.

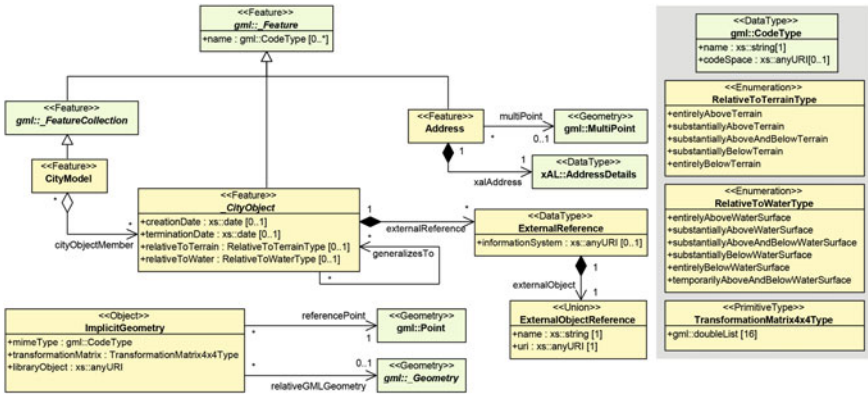


Fig. 18 Class diagram of the CityGML 2.0 core module. Image source: Gröger et al. (2012)

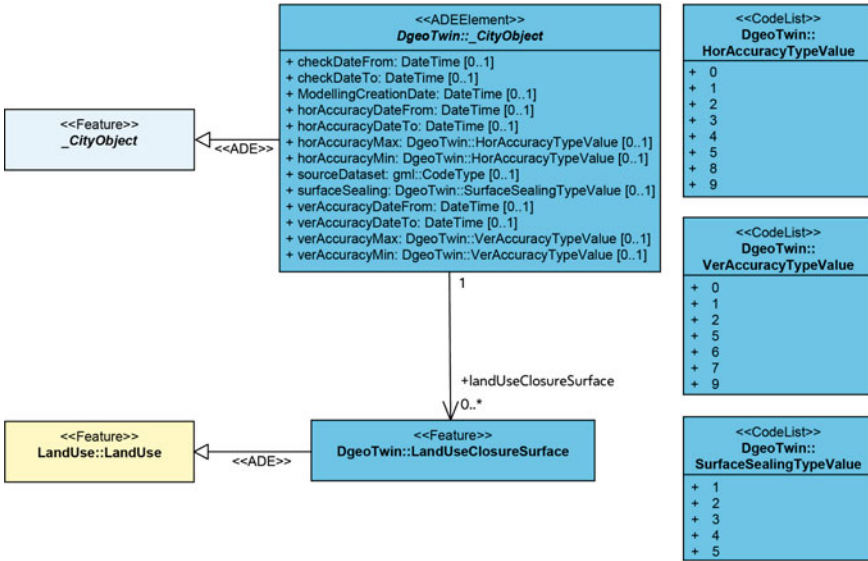


Fig. 19 Additional classes of the core module for the Digital geoTwin data model

5 Evaluation of the Results

Once the 3DCityDB has been extended with the ADE, all test data have been transformed and imported into the database. The results so far allow us to say that:

- In general, the developed data model allows to satisfy the requirements defined at the beginning of the study; the profile based on CityGML is a very valuable starting point upon which to develop and “add” a relatively small ADE;

- All data from the case study area can be successfully converted without data losses using the developed FME workbenches. All objects have been imported “as is”, or have been successfully transformed and enriched as planned;
- We have encountered some limitations in terms of 3DCityDB: there is currently no support for Implicit Geometries in an ADE, therefore we have temporarily used the native LoD4 in lieu of the ADE LoD0 for Vegetation and CityFurniture objects (these limitations have been also acknowledged by the 3DCityDB development team).

Furthermore, when it comes to the major “changes” regarding CityGML:

- There have been no issues associated with the decision to “drop” the Relief module. As originally envisioned, all terrain objects could be mapped to the target classes in the respective modules (WaterBody, Transportation, etc.) with the LandUse module taking up (for the most part) its role. This decision allows, among the rest, to support fully 3D meshes also for terrain objects, instead of 2.5D TINs;
- The LandUseClosureSurface class has been used and it works as planned to “seal” the terrain in places where a 3D object (like a building or a bridge) is removed but preserving a water-tight surface;
- The required metadata for temporal and geometrical accuracy information regarding the geometry objects can be stored successfully.

Figures 20 and 21 show an example of 3D urban objects mapped to and integrated into the semantic 3D city model of Vienna based on the data model presented in this paper.



Fig. 20 Detail view of the heterogeneous objects in the test area, modelled using the Digital geoTwin data model developed for Vienna



Fig. 21 Overview of the heterogeneous objects in the test area, modelled using the Digital geoTwin data model developed for Vienna

6 Conclusions

The creation, management and usage of a (geospatially enabled) digital twin is a complex process and challenge that many cities in the world are starting to learn from and to deal with. On the one hand, the process is strongly dependent on the context and the intended goals, the underlying IT infrastructure, and the intended applications, on the other hand, geodata play in any case a definitely relevant role.

Specifically geodata (modelling, storage, management) was the core of the project Digital geoTwin in Vienna. One of its goals has been the definition of a suitable data model to deal with (vector-based) spatial data of all most relevant urban objects generally available and surveyed in a city, in order to create a semantic (3D) virtual model of the city.

The purpose of this data model is to provide a coherent and integrated source of semantically enriched geospatial data covering all spatial entities that will be part of the Digital geoTwin of Vienna.

Part of the project was to test whether, how and to which extent the international standard CityGML version 2.0 could be adopted (and if needed, modified) for the needs of the Municipality of Vienna. 3D modelling of all city objects was carried out in a test area of the City of Vienna. These results were used to develop and test the data model.

At the end of the project, results and the experiences collected allow us to say that CityGML can be used as a reference data model for the Digital geoTwin of the City of Vienna, from which a profile has been extracted and to which a relatively small ADE has been added. At least to the best of the author's knowledge, there are no other examples in literature where such an extensive work of data and workflow analyses,

3D modelling, mapping and transformation has been carried out in the context of CityGML-based city models at national or international level.

In conclusion, we can say that CityGML is a very rich, but rather complex data model which can be configured and extended in order to comply with specific needs. However, it requires a good level of knowledge, not only in terms of theory, but also about current software availability, functionalities and limitations. From the point of view of geodata producers CityGML represents a good starting point for the development of a semantic 3D city model. From the point of view of the data users, CityGML still suffers from the insufficient support from the side of commercial software. For example, some software products only support the Building module, as this reflects probably the current usage of CityGML in “real life”, as already pointed out in Chap. 3.1. This is also reflected in the definition of the LoD concept, which is quite well defined and established for buildings, but still rather unclear, or poorly documented for other modules. Nevertheless, if support from commercial software might be sometimes a bit behind, it must be noted that support from open-source initiatives has been boosting in the recent years, however with all pros and cons tied to it (especially from the point of view of a department of a municipality). Particularly welcome is a recently released new open-source plugin for QGIS that allows to interact with the CityGML-based model stored in the 3DCityDB directly from QGIS (Agugiario et al. 2023).

6.1 Outlook

Although the results obtained are satisfactory, there are still some topics that need particular attention. Some will be listed here as points of future investigation.

In general, the Digital geoTwin project has been carried out using CityGML version 2.0 and trying to avoid potential incompatibilities with the (back then) upcoming version 3.0. In fact, CityGML 3.0 comes with some interesting changes and additions that are expected to contribute to the reduction of the customization level required by the current CityGML profile plus DigitalgeoTwin ADE. To start with, LoD0 geometries are now supported by all CityObjects, therefore part of the ADE would not be needed anymore.

The new “space concept” brings new possibilities for modelling legal and logical spaces such as clearance gauges or planning schemas. This does not affect the Digital geoTwin but is relevant for a digital urban twin and, thus, will be assessed in the future.

Finally, using the Versioning module of CityGML 3.0 it should be possible to model the time-dependent changes of a city. Since changes play a relevant role both in past and future it is crucial to understand and predict (or plan) the urban evolution trends.

The linking of heterogeneous existing, external datasets plays another major role. CityGML allows to link to external resources via ExternalReference objects. However, linking heterogeneous systems or management processes can be more complex, considering their temporal aspect and their possible validity interval.

Preliminary studies are already on-going, but no final results have been obtained so far.

From the user perspective, the possibility to exploit a digital twin to perform urban planning analyses and simulations opens the door to many new applications. Still, different tools, different set of geometries and attributes are generally used for specific analyses, and another level of complexity is given by the generation of additional alternative scenarios that accompany the “status-quo”. Dealing with such richness and complexity of data (formats, size, scales, resolutions...) is surely within the envisioned “playground” of an urban Digital geoTwin, but still subject of current research worldwide. To start with, at least from the point of view of the (Vienna) municipality, there must be an integrated approach underlying distinct processes for data generation and management between “real and surveyed” data, and simulated, synthetic ones.

References

- Agugiaro G, Benner J, Cipriano P, Nouvel R (2018) The energy application domain extension for CityGML: enhancing interoperability for urban energy simulations. *Open Geospatial Data, Software and Standards*, 3:2. SpringerOpen, United Kingdom (2018)
- Agugiaro G, Pantelios K, León-Sánchez C, Yao Z, Nagel C (2023) Introducing the 3DCityDB-tools plug-in for QGIS (this same conference proceedings)
- Biljecki F, Ledoux H, Stoter J (2016) An improved LOD specification for 3D building models. *Comput Environ Urban Syst* 59:25–37. <https://doi.org/10.1016/j.compenvurbsys.2016.04.005>
- Biljecki F, Kumar K, Nagel C (2018) CityGML application domain extension (ADE): overview of developments. *Open Geospatial Data Softw Stand* 3(1):1–17
- Durão LFCS, Haag S, Anderl R, Schützer K, Zancul E (2018) Digital twin requirements in the context of industry 4.0, product lifecycle management to support industry 4.0. Springer International Publishing, pp 204–214
- Gröger G, Kolbe TH, Nagel K, Häfele K-H (2012) OGC city geography markup language (CityGML) encoding standard
- Kolbe TH, Kutzner T, Smyth CS, Nagel C, Roensdorf C, Heazel C (2022) OGC city geography markup language (CityGML) part 1: conceptual model standard
- Kumar K, Labetski A, Ohori KA, Ledoux H, Stoter J (2019) Harmonising the OGC standards for the built environment: a CityGML extension for Landinfra. *ISPRS Int J Geo Inf* 8(6):246. <https://doi.org/10.3390/ijgi8060246>
- Lehner H, Dorffner L (2020) Digital geoTwin Vienna: towards a digital twin city as Geodata Hub. *Deutsche Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation (DGPF)* 88:63–75. <https://doi.org/10.1007/s41064-020-00101-4>
- Löwner M-O, Benner J, Gröger G, Gruber U, Häfele K-H, Schlüter S (2012) CityGML 2.0—Ein internationaler standard für 3D-Stadtmodelle, Teil 1: Datenmodell. *ZfV—Zeitschrift für Geodäsie, Geoinformation und Landmanagement*, pp 340–349
- Lutz B (2019) Data excellence—IKT-Teilstrategie, Magistratsdirektion der Stadt Wien, Geschäftsbereich Organisation und Sicherheit. Prozessmanagement und IKT-Strategie. <https://digitales.wien.gv.at/wp-content/uploads/sites/47/2019/03/Data-Excellence.pdf>
- Rosen R, Von Wichert G, Lo G, Bettenhausen KD (2015) About the importance of autonomy and digital twins for the future of manufacturing. In: 15th IFAC symposium on information control problems in manufacturing, vol 48, pp 567–572

- Van den Brink L, Stoter J, Zlatanova S (2013) UML-based approach to developing a CityGML application domain extension: UML-based approach to developing a CityGML application domain extension. *Trans GIS* 17(6):920–942
- Yao Z, Nagel C, Kunde F, Hudra G, Willkomm P, Donaubauer A, Adolphi T, Kolbe TH (2018) 3DCityDB—a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. *Open Geospatial Data Softw Stand* 3:5
- Ying Y, Koeva M, Kuffer M, Zevenbergen J (2022) Toward 3D property valuation—a review of urban 3D modelling methods for digital twin creation. *ISPRS Int J Geo Inf* 12(1):2

A Hierarchy of Levels of Detail for 3D Utility Network Models



Zihan Chen, Jacynthe Pouliot, and Frédéric Hubert

Abstract The paper presents a new hierarchy of multiple levels of detail (LoD) designed for the modeling of underground utility networks. The hierarchy of LoDs is applied to water management possible scenarios in the context of determining the aging of distribution networks. The hierarchy of LoDs proposes four levels, two sub-levels, and three quality levels. The various LoDs are defined based on the dimension of the space and the geometric primitives, the topology, the semantic aspects, the contextual information, and the data quality (regarding 3D data collecting). It is believed that this number of levels is sufficient and appropriate to cover the modeling of underground utility networks suitable for various applications and consequently will help increase the use of 3D UUN models and their sharing.

Keywords Level of detail · Underground utility network · 3D modeling · Hierarchy LoD

1 Introduction

Today, 3D models, whether of cities and infrastructures or of natural space, provide great added value to understanding, analyzing, and predicting phenomena (Biljecki et al. 2015; Stoter 2020). The major trend in the digital twin market reveals that 3D models are even more important as they constitute the main skeletons enabling simulation and the internet of objects. Looking back, what are the highlights of this progression in the penetration of 3D models in the market? For instance, if we go back

This article was selected based on the results of a double-blind review of an extended abstract.

Z. Chen (✉) · J. Pouliot · F. Hubert
Department of Geomatic Sciences, Université Laval, Québec City, QC, Canada
e-mail: zihan.chen.1@ulaval.ca

J. Pouliot
e-mail: jacynthe.pouliot@scg.ulaval.ca

F. Hubert
e-mail: frederic.hubert@scg.ulaval.ca

to the beginning of 2000, was it possible to share 3D city models? At that time, 3D city models were sparsely produced and distributed, and the content varied considerably between them (Köninger and Bartel 1998; Gruber 1999; Haala and Brenner 1997; Ranzinger and Gleixner 1997). The use of standards, such as CityGML for 3D city models, is undoubtedly one of the highlights (Kolbe 2009; Gröger et al. 2012; Kolbe and Gröger 2003). Standardization has had an obvious impact on the ability to share data and, therefore, improve the understanding of its content.

If we look more precisely at the modelling of underground networks, even though standardization efforts exist, it is evident that the content of underground utility network (UUN) models is still heterogeneous, and the sharing of these models is yet limited (Kutzner and Kolbe 2017; Yan et al. 2019a, b; American Society of Civil Engineers 2002; Van Son et al. 2018; Wang et al. 2019). Moreover, should or could 3D UUN models be part of digital twin development and deployment? Asking the question is answering it since the underground space is an integral and crucial part of urban systems. It seems appropriate to examine the literature and the practice to better understand why the use and sharing of 3D models in underground network environments are still so limited (Chen et al. 2021). Despite part of the situation is because underground data is often confidential, and the acquisition is still challenging. One of the reasons that appeared possible and would lead to the establishment of standards is to talk about the content of the models, in terms of scale of representation and application domains.

In Chen et al. (2021, 2022), an approach and the establishment of multiple levels of detail (LoD) and task decomposition methods were proposed to make progress in that direction. This paper presents the final proposal about the hierarchy of LoD applied to UUN modelling and based on a simulation task, some applications in the visual aspect are discussed. By investigating the requirements and characteristics of UUN LoD's and the composition of UUN components, considering the key variables defined for UUN, a hierarchy contains four main LoDs, two sub-levels, and three quality levels is designed aiming at solving different classes of tasks. The chosen simulation task is an analysis in managing the underground water supply pipe network. Models in different LoDs are built for demonstrations and for comparing the comprehensibility caused by visual differences between them, which in turn could guide the selection of suitable LoD.

The following sections start with a review of current data model for UUN along with its LoD concept, then analyze the characteristics and key variables for UUN LoD. After the hierarchy of LoD is proposed. Finally, the applications of UUN LoD are introduced and a simulation task has been carried out as demonstration to determine the aging of distribution networks, leading to the final discussion and conclusion.

2 Underground Utility Network Modeling and Level of Detail

Underground Utility Network (UUN) constitutes an important part of the urban system and play a decisive role in ensuring the normal operation of various functions of urban facilities. It contains difference categories such as water supply, sewer, natural gas, electricity, telecommunications lines, etc. (Lieberman and Ryan 2017). With its huge quantity, lacking information and ambiguous representations, it is facing with challenges in detecting, mapping, managing. Considering the fact that obtaining adequate and accurate UUN information and finding a way to manage, visualize them intuitively have become an issue not only indispensable but urgent, there are some major UUN data model proposed by researchers.

2.1 Current UUN Data Model

Here is a list of UUN data models currently proposed by standardization organizations.

- Infrastructure for Spatial Information in Europe (INSPIRE) Utility Networks uses 2D representation models of utility networks in a city or country range (INSPIRE 2010).
- Industry Foundation Classes (IFC) Utility model describes 2D and 3D geometry of utilities within the building and the logical or physical connection between building service components but lacks spatial information (Becker et al. 2013a).
- ESRI Utility model defined edges and junctions as 2D line and 2D point features, but it could be visualized with 3D representation (Grise et al. 2001; Becker et al. 2013b; Willett and Patterson 2022).
- The Model for Underground Data Definition and Integration (MUDDI) is open standards-based way to share information about the below ground, and based on IMKL, Singapore Land Authority model (Yan et al. 2019c; Lieberman and Roensdorf 2020).
- CityGML utility network ADE focuses mainly on the representation of topographical, graph structural and functional information across the multi-utility networks in 3D space.

Whether in 2D or 3D, the model for UUN would consider the geometric representations, network links, complex semantic meanings, integrating with other city objects, visualization, the support of analysis application and data sharing. Modeling them in 3D is the major trend, and a reliable visualization and 3D model is useful to guide the utility location, urban planning and land administration process (Yan et al. 2019a; Ghodsvali 2018).

2.2 *The Level of Detail for Underground*

An accurate and efficient 3D spatio-semantic UUN models looks to be indispensable for the whole cycle of its planning, construction, maintenance, and all kinds of decision-making process. But what should be the content of 3D UUN model?

In 3D GIS, Level of Detail (LoD) is mostly applied to describe the complexity of the geometry and semantics (Kolbe 2009). The concept of LoD is helpful to decrease the size of 3D models, satisfy the demand in different phases such as data acquisition, modeling, visualization and analysis, and combine semantic meanings with cartographic and geometric elements. The construction planning of UUN relies heavily on the use of different scales for representing geometric information on a suitable level of detail. Modeling in 3D UUN should be able to provide multiple representations considering data accessibility and model comprehensibility (American Society of Civil Engineers 2002; Son et al. 2018; Mainroads 2018).

There are a few studies about defining LoD of underground facilities with varieties and numbering: BIM Forum and the American Institute of Architects have defined the geometric elements and contained information on some piping and for each develop level (BIM Forum 2015; Institute and of Architects 2008). Breunig et al. (2011) designed diverse LoDs result from different detailing demands in the individual planning stages. It mostly considered corresponding geometrical representation for subways. Borrmann et al. (2013) shows IFC-based tunnel modeling for different construction planning stages from modeling outer shell to modeling interior. Gröger et al. (2012) illustrates LoDs created to represent either different tubes of a tunnel or different sections. In this LoD, the tunnel volume is represented by different geometry, and can also be differentiated semantically by boundary surfaces.

These attempts of creating LoD are either targeted at one specific facility (subway, tunnels, etc.) or pre-defined application (for planning and construction process). There is no real converging solution that contribute to interoperability and thus limits the usage and integration of LoD in practice. There exist a great diversity of acquisition of the UUN data, with current LoD concept two models with different complexities may be recognized as a same LoD (Biljecki et al. 2014). Researches involved with creating multiple representations model for UUN and an elaborate LoD hierarchy to fulfill more comprehensive tasks are not adequate (Nie 2019; Duijn et al. 2018). Current CityGML ADE LoD is not ready for dealing with higher geometry representation, and mostly about the geometric representations. Data accessibility and data quality are mostly neglected in the design of LoD. It could limit the understandability and communication when sharing and describing UUN models in different phases such as data acquisition, modeling, visualization and analysis, for different applications in real use cases.

3 Proposal of Hierarchy of UUN LoDs

3.1 UUN LoD Characteristics and UUN Components

In order to identify LoD applicable to underground utility networks, several considerations were taken into account in defining the hierarchy:

- LoD represents a series of milestones that correspond to the complexity of the geometry and semantic differentiation of specific features vary on a continuum (Kolbe 2009; Meng and Forberg 2007).
- The definition of LoD should not be subjective; rather, it should be based on measurable and independent variables.
- The list of LoDs must demonstrate a clear hierarchy among those variables.
- The hierarchy of LoDs should enable the automatic conversion from one LoD to another.
- LoD must be suitable for effective visualization and analysis tasks. striking a balance between comprehensiveness and accessibility to users.

The hierarchy of LoDs has been specially developed for features related to UUN, namely Network components. The terminology from CityGML Utility Network ADE was employed to model the network components (Becker et al. 2013c). Table 1 represents the list of network components, as based on (Kutzner and Kolbe 2017).

To describe 3D features with multiple levels of detail, a variety of information (called key variables) are taken into account. We do estimate that these variables are indispensable or atomical to be taken into consideration when modeling 3D city objects. Table 2 shows the list of key variables considered.

As discussed in Chen et al. (2021), after reviewing the different definitions and standards of LoD expressions, the multiplicities in geometry and semantics are generally regarded as main aspects. Meanwhile, for UUN, the 3D topology becomes more significant in practical use, and a hierarchy of topological relationships should also be identified to support modeling complex geometry (Ellul and Haklay 2006). Besides, the accessibility and data quality of UUN data (especially depth information) influence the representation of UUN model (den Duijn 2018). With different data accuracy

Table 1 Components of UUN

| Network components | Examples |
|----------------------|--|
| Distribution objects | Pipes, canals, cables |
| Protection elements | Cable protection package, ductwork |
| Functional objects | Pipe junctions (fittings), valves, connections with buildings, hydrants, meters, regulators, pumps, pressure sensors, temperature sensors, acoustic emission sensors, etc. |

Table 2 List of key variables to describe 3D model

| Variables | Explanation |
|------------------------|---|
| Geometry | Dimension of the space (number of axes) |
| | Dimension of the geometric primitive |
| Topology | Inner topology |
| | Spatial relationship |
| Semantics | Category of network components |
| | Attributes, namely depth, and diameter |
| Contextual information | Presence or absence of additional information, such as ground information, vegetation, roads, etc. |
| Data quality | classification of subsurface utility data quality for the horizontal and vertical position (derived from ASCE 38-02) (American Society of Civil Engineers 2002) |

level, correspondent UUN LoD should be generated. More explanations will be given in following sections.

3.2 Hierarchy of LoDs

Considering the components, the characteristics of UUN LoD, and the key variables to describe 3D UUN model, aiming at task-targeted solution, a hierarchy of LoDs has been proposed. Table 3 shows the proposed hierarchy of LoDs. To understand the hierarchy, it is necessary to specify a number of premises.

- LoD numbering:** We first propose that the primary feature to be modeled for UUN is Distribution objects, which ought to consistently exist across all LoDs, while protection elements and functional objects are optional. The LoD numbering begins at 2, aligning with the space dimension; LoD2 is for 2D space and LoD3,4,5,00.5 represents 3D space. The change of levels depends on the dimension of space or the geometry of distribution or functional components. The hierarchy comprises four main LoDs, two sub-levels, and three quality levels.
 - Main levels:** The main levels refer to a combination of spatial dimension, dimension of the geometric primitive (GM) and finally to the interior.
 - Sub-levels:** “0.5” is added to the LoD number when functional objects are represented, and this sublevel contains the semantic information of Functional Component. The hundredth digit ex. “300” is added when the topology is required.
 - Quality levels:** Based on ASCE 38-02 (American Society of Civil Engineers 2002), a letter (A, B or C) is assigned depending on how the position, depth, and diameter are collected.

Table 3 Hierarchy of LoDs (empty marked by \emptyset)

| LoD number | Dim. of the space | Geometric primitives | | | Topological constraints | Quality LoD number | Geometric attributes | | | |
|------------|-------------------|----------------------|-----------------------|--|-----------------------------|--------------------|----------------------|-----------------------------|---|-------------------------|
| | | Classes of entities | | | | | Acquisition quality | Z coordinates of the entity | | |
| | | Distribution objects | Functional Components | Protection elements and Inner structures | | | | | | |
| LoD 2 | 2D | GM_Curve | \emptyset | \emptyset | \emptyset | \emptyset | \emptyset | \emptyset | | |
| LoD 2.5 | | | GM_Point | | \emptyset | | \emptyset | | | |
| LoD 200 | | | \emptyset | | Connectivity Directionality | | \emptyset | \emptyset | | |
| LoD 200.5 | | | GM_Point | | Connectivity Directionality | | \emptyset | \emptyset | | |
| LoD 3 | 3D | GM_Curve | \emptyset | \emptyset | \emptyset | LoD 3A | A | Location of the entity | | |
| LoD 3.5 | | | GM_Point | | | | | LoD 3B | B | Location of the entity |
| | | | | | | | | LoD 3C | C | Elevation of the ground |
| LoD 300 | 3D | \emptyset | \emptyset | \emptyset | Connectivity Directionality | LoD 300A | A | Location of the entity | | |
| LoD 300 | | | GM_Point | | | | | LoD 3.5A | A | Location of the entity |
| | | | | | | | | LoD 3.5B | B | Location of the entity |
| | | | | | | LoD 3.5C | C | Elevation of the ground | | |

(continued)

Table 3 (continued)

| LoD number | Dim. of the space | Geometric primitives | | | Topological constraints | Quality LoD number | Geometric attributes | |
|------------|-------------------|----------------------|-----------------------|--|-----------------------------|--------------------|----------------------|-----------------------------|
| | | Classes of entities | | | | | Acquisition quality | Z coordinates of the entity |
| | | Distribution objects | Functional Components | Protection elements and Inner structures | | | | |
| LoD 300.5 | 3D | | | | | LoD 300B | B | Location of the entity |
| | | | | | | LoD 300C | C | Elevation of the ground |
| | | | GM_Point | | Connectivity Directionality | LoD 300.5A | A | Location of the entity |
| LoD 4 | 3D | | | | | LoD 300.5B | B | Location of the entity |
| | | | | | | LoD 300.5C | C | Elevation of the ground |
| | | GM_Solid | ∅ | ∅ | ∅ | LoD 4A | A | Location of the entity |
| LoD 4.5 | 3D | | | | | LoD 4B | B | Location of the entity |
| | | | | | | LoD 4C | C | Elevation of the ground |
| | | | GM_Point/ GM_Solid | | ∅ | LoD 4.5A | A | Location of the entity |

(continued)

Table 3 (continued)

| LoD number | Dim. of the space | Geometric primitives | | | Topological constraints | Quality LoD number | Geometric attributes | |
|------------|-------------------|----------------------|-----------------------|--|-------------------------|--------------------|-------------------------|-----------------------------|
| | | Classes of entities | | | | | Acquisition quality | Z coordinates of the entity |
| | | Distribution objects | Functional Components | Protection elements and Inner structures | | | | |
| | | | | Inner topology (same entities) | LoD 4.5B | B | Location of the entity | |
| | | | | | LoD 4.5C | C | Elevation of the ground | |
| LoD 400 | 3D | | ∅ | Connectivity Directionality | LoD 400A | A | Location of the entity | |
| | | | | Connectivity Directionality | LoD 400B | B | Location of the entity | |
| | | | | Connectivity Directionality | LoD 400C | C | Elevation of the ground | |
| LoD 400.5 | 3D | | GM_Point/ GM_Solid | Connectivity Directionality | LoD 400.5A | A | Location of the entity | |
| | | | | Connectivity Directionality | LoD 400.5B | B | Location of the entity | |
| | | | | Connectivity Directionality | LoD 400.5C | C | Elevation of the ground | |

(continued)

Table 3 (continued)

| LoD number | Dim. of the space | Geometric primitives | | | Topological constraints | Quality LoD number | Geometric attributes | |
|------------|-------------------|----------------------|-----------------------|--|-----------------------------|--------------------|----------------------|-----------------------------|
| | | Classes of entities | | | | | Acquisition quality | Z coordinates of the entity |
| | | Distribution objects | Functional Components | Protection elements and Inner structures | | | | |
| LoD 500.5 | 3D | GM_Solid | GM_Solid | Protections elements, inner space, and inner structure | Connectivity Directionality | LoD 500.5A | A | Location of the entity |
| | | | | | Connectivity Directionality | LoD 500.5B | B | Location of the entity |

- **Dimension of space:** Refer to the dimension of the space for modeling, possible value: 2D, 3D.
- **Geometric primitives:** Geometric primitives are non-decomposed objects that present information about geometric configuration. It should be assigned separately for different classes of entities. Possible values: GM_Point, GM_Curve, GM_Surface, GM_Solid. At the same time GM_Complex and GM_CompositeSolid could also be indicated by GM_Solid as needed.
- **Topological constraints:** Based on the literature review, it becomes evident that topological constraints are vital when working with UUN models. Both inner topology (connect, close, directionality, etc.) and topological relationships (touch, connect, network to network connection, network to building features connection, overlap, inside, 3D overlay, etc.) are considered.
- **Data quality of geometric information:** The relevance of integrating geometric data quality into the LoD hierarchy has been the subject of lengthy debate in our group and requires more detailed explanation. If we look at the LoDs for CityGML, for example, this notion of quality is not directly existing in the LoD hierarchy. However, when we look at CityGML specifications, we can observe that the concept of geometric data quality exists, and that there is a link to be made with LoDs and the potential uses of 3D models. Data quality information is usually managed as metadata.

So, why would this information of geometric quality be required in the LoD hierarchy for UUN? Firstly, it must be admitted that all UUNs are underground, so the ability to directly measure their position and size is limited. Some might say that, before burying them, we need to accurately measure their XYZ position and characteristics. We agree with this, but we have to recognize that most networks already exist, and are buried, and that they have not been subject to this kind of care in terms of geometric data collection. As a result, there is several underground networks that have limited and uncertain geometric information, like no precise Z position information. While other underground networks, usually more recently installed, have precise Z position and diameter information. This difference of data accuracy makes a significant distinction on the system requirements and may affect the UUN model's performance. For instance, when excavation or construction simulation are performed, or in doing network analysis to locating and isolate damaged pipes, valves during an emergency, knowing "what the accuracy of the XYZ position means" is crucial. While for urban planning task, the requirement of Z position accuracy would be only at a theoretical or standards level. Consequently, when working with UUNs, we have to take this into account how is collected the geometric data, "it is a must".

Now, how to handle this obligation when designing the hierarchy of LoDs? First, we could simply ignore it and keep this trivial information stored as metadata. We decided that Z coordinates and the diameter of the pipe are too important to be kept as metadata (often ignored by users). We decided to integrate them into the hierarchy of LoDs in order to clearly state "how Z position and diameter information are collected".

There are different proposals specifically designed to state about the quality of Z position and the dimension of the network (American Society of Civil Engineers 2002; Standards and Association 2020; Standards and Institution 2014). They have similar levels considering the methods of acquiring the data or the exact maximum uncertainty of location in centimeters (France 2012). We decided to use ASCE Standard 38-02 proposed by the American Society of Civil Engineers (2002) as main frame, since it is one of the standards well recognized and applied in many real use cases (Meis 2017). ASCE 38-02 is a guideline for the collection and depiction of existing subsurface utility data aiming at improving information reliability. ASCE 38-02 is used in subsurface utility engineering for project owners, engineers, constructors, and utility owners to develop strategies and reduce risk. It presents a system to classify the quality of existing subsurface utility data with four levels. Each of the four levels is proposed by different methods of data collection and interpretation: actual exposure, surface geophysical methods, surveying and plotting above-ground features, and derived from existing records. Meanwhile, we take the idea from geographical uncertainties representation (Olde Scholtenhuis et al. 2018), because it is designed more specifically for 3D representation and extended from CityGML Utility Network ADE. It indicates four parameters used to identify location quality: surveyed, standard, estimated, and unknown, from a low to high quality level. Thus, in our LoD hierarchy, we took the classification for quality of UUN geometric information as follows:

- Level A: Geometric data are obtained by actual excavation or acquired and recorded while being installed. It has a precise location both horizontally and vertically along with other concert geometric properties such as size, shape.
- Level B: Geometric data are obtained through surface geophysical methods (GPR, EML) after the installation. It could approximate horizontal position and rough estimates of depth.
- Level C: Geometric data are requested from owner's oral recollections; investigated/derived from the visible feature; or inferred from standards and installation guidelines. These data are with poorly unknown or incalculable accuracy.

Consequently, we added a sub-level in the hierarchy of LoDs proposed in Table 3, we indicated it as Geometric attributes—Acquisition quality (possible value A, B, or C), depending on how XYZ coordinates and diameter of the network are collected. This sub-level could be optional depending on the interest of the users or the applications. By default, it would be the category C which allow to identify the accuracy of depicted networks as the lowest level of quality (for safety purpose). Furthermore, since in many situations, Z coordinates can be relatively deduced from ground information like digital elevation model information, we need to consider in our hierarchy of LoDs the presence or not of ground information (see Table 3 as Z coordinates).

4 Visualization and Application of the Water Pipe Aging Simulation

With this hierarchy proposal, UUN models in different LoDs can be used to provide a comprehensive solution for different type of networks in municipalities. Though in this paper, after having investigated the literature, and the scientific problems, we focus on the water supply and wastewater network as illustrations, this hierarchy is designed to be generic. It could be applicable to more domains like water supply, sewer, gas, electricity.

To build examples of each LoD and discover the possible values of having this hierarchical UUN LoD, a “fictive” scenario aiming to provide a comprehensive solution for managing the underground water supply pipe network in municipalities is created. The solution will enable the municipalities to monitor the performance and condition of the pipe network, identify potential risks and failures, and optimize the allocation of resources and interventions.

In this scenario, a simulation task that calculates and displays the aging index of pipe is taken as an example to be applied with different LoDs. This implementation is to illustrate the possibility of visualizing 3D UUN model, and trying to distinguish how different LoDs would affect the UUN models’ applicability. In this simulation model, aging index (superannuation) of the pipe is calculated by pipe installation year, material, length, and diameter (Jun et al. 2004).

$$S_i = \frac{YP - YI_i}{D_i} \times \frac{L_i}{TL} \times P_i \quad (1)$$

where S_i = superannuation of pipe, YP = present year, YI_i = the installation year, D_i = the durability of pipe (depending on pipe material), L_i = the length of pipe, TL = the aggregate length of all pipes in this block, P_i = the weight value assigned to pipe i (depending on the pipe size). The superannuation (S_i) is an index to distinguish the tendency based on the material and pipe size. Higher value means the aging problem of this pipe is with higher severity and higher importance. By setting certain thresholds, the municipalities can better understand and arrange the maintenance plan. The UUN model can be used to demonstrate its distribution more intuitively.

The overall view of the sample data block is shown in Fig. 1 in which the simulation results are presented by different colors of the water main pipe and the red rectangle will serve to illustrate the results in Figs. 2 and 3.

Figure 1 shows that if a general overall information is to be investigated, without any hierarchical or categorical organization, a UUN model of all pipeline information appears extremely disordered. Users may find it challenging to comprehend the model due to the presence of redundant information load; the absence of varying LoDs impedes the resolution of practical tasks. Figure 2 and 3 compare different LoDs in a local pipe network block view (red rectangle in Fig. 1), this illustrates a preliminary evaluation and judgment in the visual aspect.

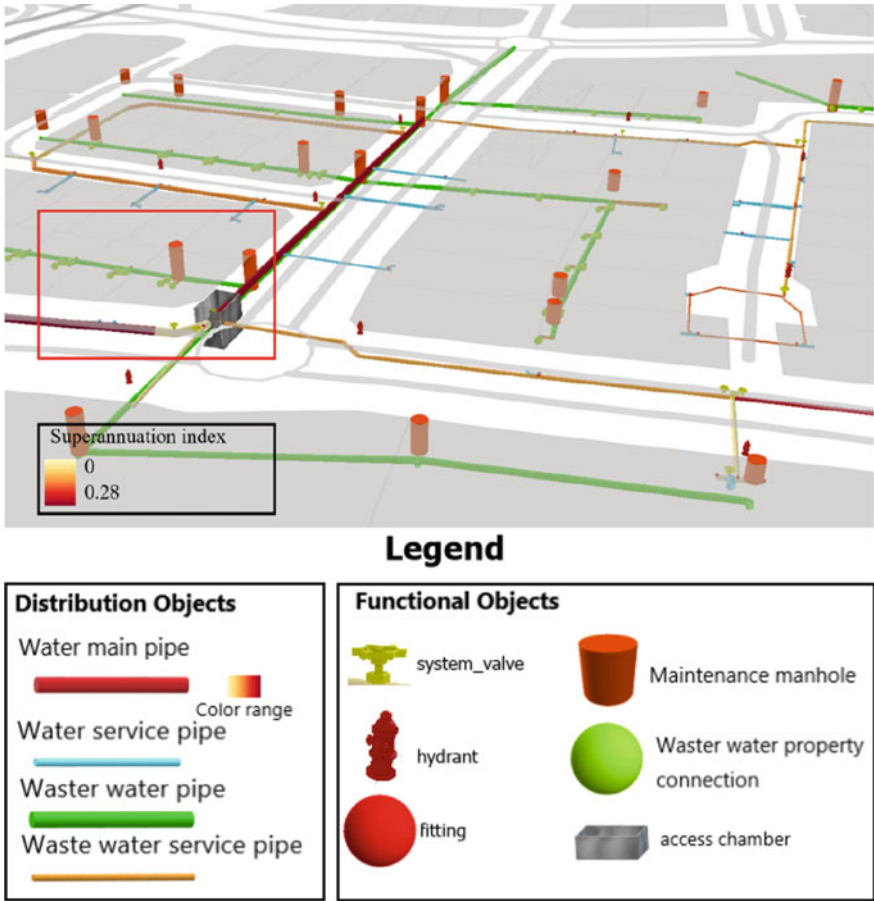
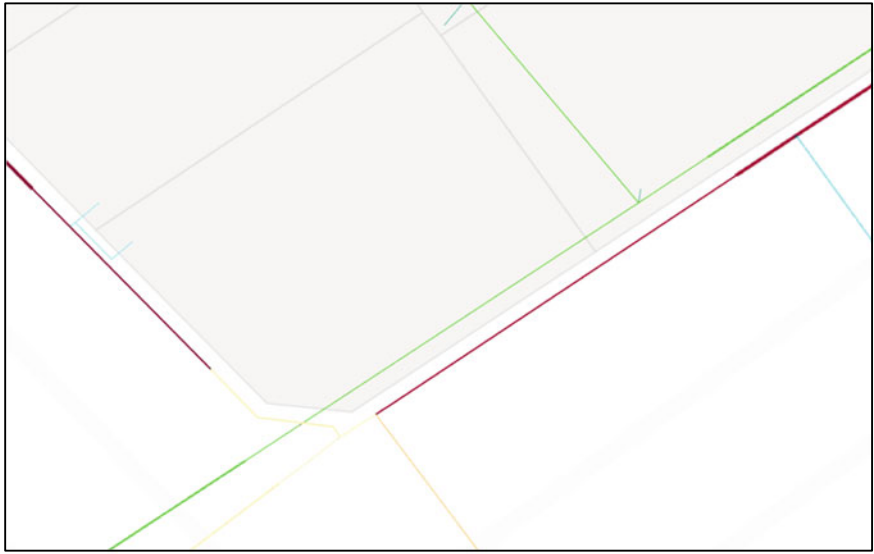
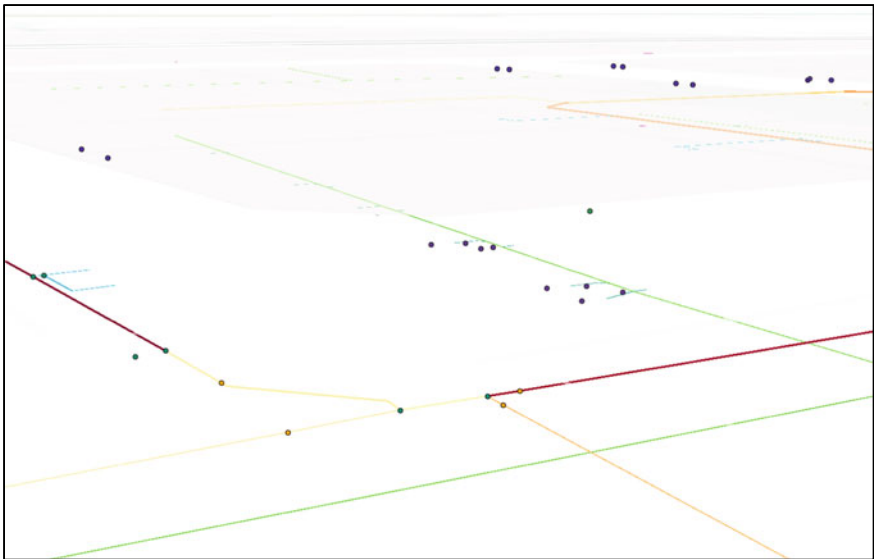


Fig. 1 Overall result view in UUN LoD 4.5 of the study area

To get the aging situation of the main pipe, without the diameter intuitively displayed, the different pipe sizes that affect the replacement priority cannot be well understood (LoD 2, LoD 3.5). Meanwhile, without showing the pipe in 3D space, it is difficult to read the depth relationship (LoD 2). In this specific task, functional components and topology information are not needed for the simulation, and somehow representing them in the model would add more challenges to reading and understanding (LoD 400.5). Thus, in this case, LoD 4 could be the better model to solve its application.

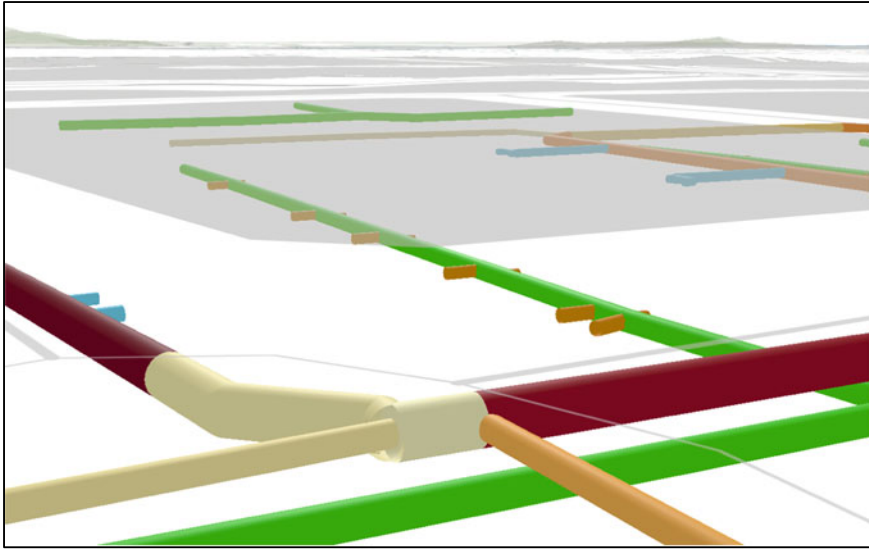


LoD 2

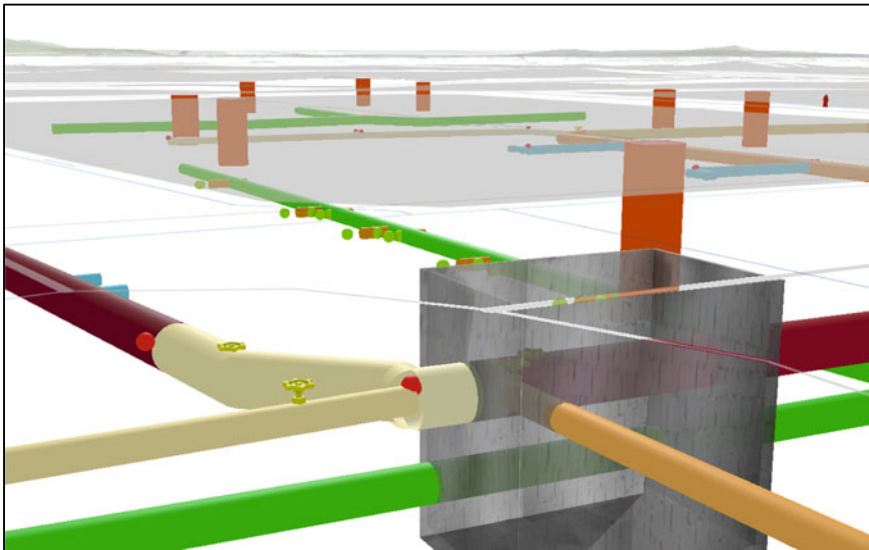


LoD 3.5

Fig. 2 Simulation results showing UUN LoD 2 and LoD 3.5



LoD 4



LoD 400.5

Fig. 3 Simulation results showing UUN LoD 4 and LoD 400.5

5 Discussion and Conclusion

The main contribution of this work has been to propose a new way of modelling UUN for which the concept of LoDs is the core foundation. The hierarchy of LoDs recommends four levels, two sub-levels, and three quality levels. In this LoD hierarchy proposal, the design of LoD is linked to geometric representation, semantic information, data availability and data quality. In this way, the various levels are not arbitrary defined. We also tried to categorize the application of each LoD and use it to guide the selection of best fit LoD.

To illustrate the feasibility and the relevance of applying the hierarchy of LoDs, a simulation task to calculate and display the aging index of pipe is taken as an example to be applied with different LoDs. For now, it is believed that the proposed LoDs is applicable, could be beneficial to various practitioners and may contribute to improving the dissemination of 3D UUN models in practice, building and sharing UUN models in a more efficient way, and guiding the selection of suitable models.

We do estimate the proposed LoD numbering may appear complex in practical implementation. Particularly since most of current UUN applications still work with 2D data. The transition towards 3D is gaining momentum. Potential simulations and advanced analyses put forward the brand-new demands. The enhanced visualization should enable the users to view and understand multiple components in desired level and choose visual representation emphasis based on the tasks. Meanwhile, though a large portion of these application tasks doesn't necessarily require more than one sublevel. Sometimes the users doesn't care about whether functional components exist in their model, or they assume all UUN model should have a network topology representation. However, the topological information, all the connections (interior or exterior), and node positions are not always available, without the design of this topology sublevel, some dataset cannot relate to it. The description of UUN model would not be precise enough to state the differences. If we look back at how CityGML treats Buildings' LoDs, we can find that some content of 3D models for buildings is determined by the specific LoD. But in others' UUN LoD definition, it is missing, our proposal is trying to fill the gap of need.

Thus, we felt it is essential to establish this more comprehensive hierarchy, not only for reducing the ambiguity in model description, but offering practitioners various granularities for to choose from, involving different phases of data collection, visualization, and analysis.

The hierarchy of LoDs could be easily integrated with other standards like CityGML Utility Network ADE, etc. Besides, in the context of digital twin development, being able to replicate the reality, with multiple representations, with multiple levels of detail of the same reality appears to us as required and offers great versatility.

Acknowledgements This work is funded by NSERC RGPIN-2015-05514 (JPouliot) and the author is supported by China Scholarship Council (CSC) Grant #201904190003. The authors would also like to thank all reviewers for their comments and suggestions.

References

- American Society of Civil Engineers (2002) Standard guidelines for the collection and depiction of existing subsurface utility data. American Society of Civil Engineers, Reston, VA
- American Institute of Architects (2008) AIA document E202–2008 Building information modeling protocol exhibit
- Becker T, Nagel C, Kolbe TH (2013a) Semantic 3D modeling of multi-utility networks in cities for analysis and 3D visualization. In: Progress and new trends in 3D geoinformation sciences. Springer, Berlin, Heidelberg, pp 41–62
- Becker T, Nagel C, Kolbe TH (2013b) Semantic 3D modeling of multi-utility networks in cities for analysis and 3D visualization. In: Lecture notes geoinformation and cartography, pp 41–62
- Becker T, Nagel C, Kolbe TH (2013c) Semantic 3D modeling of multi-utility networks in cities for analysis and 3D visualization. In: Pouliot J, Daniel S, Hubert F, Zamyadi A (eds) Progress and new trends in 3D geoinformation sciences. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 41–62
- BIM Forum (2015) Level of development specification. Part I for building information models. www.bimforum.org/lod
- Biljecki F, Ledoux H, Stoter J, Zhao J (2014) Formalisation of the level of detail in 3D city modelling. *Comput Environ Urban Syst* 48:1–15
- Biljecki F, Stoter J, Ledoux H, Zlatanova S, Çöltekin A (2015) Applications of 3D city models: State of the art review. *ISPRS Int J Geo Inf* 4:2842–2889
- Borrmann A, Kolbe TH, Donaubaue A, Steuer H, Jubierre JR (2013) Transferring multi-scale approaches from 3D city modeling to IFC-based tunnel modeling. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci* 2:75–85
- Breunig M, Hinz S, Menninghaus M, Vögtle T, Rank E, Schilcher M, Borrmann A, Mundani RP, Ji Y, Donaubaue A, Steuer H (2011) Towards computer-aided collaborative subway track planning in multi-scale 3D city and building models. In: Proceedings of the 6th 3D geoinfo conference
- British Standards Institution (2014) PAS 128: specification for underground utility detection, verification, and location
- Canadian Standards Association (2020) Mapping of underground utility infrastructure. Canada
- Chen Z, Pouliot J, Hubert F (2021) A first attempt to define level of details based on decision-making tasks: application to underground utility network. *Int Arch Photogramm Remote Sens Spat Inf Sci XLVI-4/W4*:137–144
- Chen Z, Pouliot J, Hubert F (2022) Task decomposition and level of complexity to select the content of underground utility network model. *Int Arch Photogramm Remote Sens Spat Inf Sci XLVIII*:19–21
- den Duijn X (2018) A 3D data modeling approach for integrated management of below and above ground utility network feature
- den Duijn X, Agugiaro G, Zlatanova S (2018) Modelling below- and above-ground utility network features with the CITYGML utility network ADE: experiences from Rotterdam. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci IV-4/W7*:43–50
- Ellul C, Haklay M (2006) Requirements for topology in 3D GIS. *Trans GIS* 10:157–175
- France (2012) Code de l'environnement relatif à l'exécution de travaux à proximité de certains ouvrages souterrains, aériens ou subaquatiques de transport ou de distribution. France
- Ghodsvali M (2018) 3D modelling of underground space for urban planning and management-providing basic planning insight
- Grise S, Idolyantes E, Brinton E, Booth B, Michale Z (2001) Water utilities. *ArcGIS Data Models*
- Gröger G, Kolbe T, Nagel C, Häfele K.-H (2012) OGC city geography markup language (CityGML) encoding standard. OGC, pp 1–344
- Gruber M (1999) Managing large 3D urban databases. *Photogramm Week* 99:341–349
- Haala N, Brenner C (1997) Generation of 3D city models from airborne laser scanning data. In: EARSEL workshop on LIDAR remote sensing on land and sea. Tallinn/Estonia, pp 105–112
- INSPIRE Data Specifications Drafting Team (2010) D2.5: INSPIRE generic conceptual model

- Jun C, Koo J, Koh J (2004) Developing a water pipe management system in Seoul using the GIS. In: Istanbul T (ed) XXth ISPRS congress technical commission II. Istanbul, Turkey, pp 290–293
- Kolbe TH (2009) Representing and exchanging 3D city models with CityGML. In: 3D geoinformation sciences. Springer, Berlin, Heidelberg, pp 15–31
- Kolbe TH, Gröger G (2003) Towards unified 3D city models. In: Proceedings of joint ISPRS commission IV work. Challenges geospatial analysis integration visual II Stuttgart, vol 4
- Königer A, Bartel S (1998) 3D-GIS for urban purposes. *GeoInformatica* 2:79–103
- Kutzner T, Kolbe TH (2017) CityGML utility network ADE—scope, concepts, and applications, New York City
- Lieberman J, Roensdorf C (2020) Modular approach to 3D representation of underground infrastructure in the model for underground data definition and integration (MUDDI). *Int Arch Photogramm Remote Sens Spat Inf Sci XLIV-4/W1-2020*:75–81
- Lieberman J, Ryan A (2017) OGC underground infrastructure concept study engineering report Mainroads, West Australia (2018): Underground Utilities Survey Standard 67-08-121
- Meis PJ (2017) ASCE standards: recording and exchanging utility infrastructure data (“As-Builts”) collection and depiction of existing subsurface utility data
- Meng L, Forberg A (2007) 3D building generalisation. In: Generalisation of geographic information. Elsevier, pp 211–231
- Nie S (2019) 3D BIM-GIS for underground network management 3D BIM-GIS for underground network management
- Olde Scholtenhuis LL, den Duijn X, Zlatanova S (2018) Representing geographical uncertainties of utility location data in 3D. *Autom Constr* 96:483–493
- Ranzinger M, Gleixner G (1997) GIS datasets for 3D urban planning. *Comput Environ Urban Syst* 21:159–173
- Stoter JE, Arroyo O, Hori GAK, Dukai B, Labetski A, Kavisha K, Vitalis S, Ledoux H (2020) State of the Art in 3D City Modelling: Six Challenges Facing 3D Data as a Platform. *GIM International: the worldwide magazine for geomatics*, 34
- Van Son R, Jaw SW, Yan J, Khoo HSV, Loo WKR, Teo SNS, Schrotter G (2018) A framework for reliable three-dimensional underground utility mapping for urban planning. In: International archives of the photogrammetry, remote sensing and spatial information sciences—ISPRS archives, vol 42, pp 209–214
- Wang M, Deng Y, Won J, Cheng JCP (2019) An integrated underground utility management and decision support based on BIM and GIS. *Autom Constr* 107:102931
- Willett J, Patterson WS (2022) Subsurface utility data modeled in 3D. <https://www.esri.com/about/newsroom/arcnews/subsurface-utility-data-modeled-in-3d>. Accessed 21 July 2022
- Yan J, Jaw S, Soon K, Wieser A, Schrotter G (2019a) Towards an underground utilities 3D data model for land administration. *Remote Sens* 11:1957
- Yan J, Soon KH, Jaw SW (2019b) A LADM-based 3D underground utility data model: a case study of Singapore underground utility in land administration. Kuala Lumpur, Malaysia
- Yan J, Soon KH, Jaw SW (2019c) A LADM-based 3D underground utility data model: a case study of Singapore. In: 8th international fig workshop on the land administration domain model. Kuala Lumpur, Malaysia, pp 83–98

Topological Representation of a 4D Cell Complex and Its Dual-Feasibility Study



Pawel Boguslawski 

Abstract Representations of an object in different granularity using the Level of Detail concept are usually separated without links between corresponding components. Extension of a 3D model to the fourth dimension opens new possibilities for spatial analysis. Integration of scale using additional spatial dimension can help in scale-dependent analysis and preserving consistency, especially in case of model updates. In this paper, an extended version of the dual half-edge structure for topological representation of 4D cell complexes is proposed. This feasibility study shows implementation of the Poincaré duality theorem in practice. Thanks to that, the data structure remains simple, where only two atomic elements are used in a construction process i.e. nodes and edges. This solution lays the groundwork for future research, where topological links in the fourth dimension will be used to connect consecutive object representations of different granularity.

Keywords 4D modelling · Data structures · Topology · Levels of detail

1 Introduction

The Levels of Detail (LoD) concept implemented in 3D modelling allows for representation of individual objects in a different granularity. One of the widely-used examples is the CityGML standard designed for city model storage and exchange. Buildings can be represented as simple shapes—low LoD—or have a detailed representation—high LoD. This is used in applications to quickly visualize a rough model as soon as possible, while a detailed model is shown gradually piece by piece as it is read from the source dataset. Also, spatial analysis of a wide area can be performed

This article was selected based on the results of a double-blind review of an extended abstract.

P. Boguslawski (✉)

Institute of Geodesy and Geoinformatics, Wrocław University of Environmental and Life Sciences, Grunwaldzka 53, 50-357 Wrocław, Poland

e-mail: pawel.boguslawski@upwr.edu.pl

in the low LoD, while detailed analysis utilises higher LoD in order to provide better results, which also requires higher computation resources (Ohori et al. 2015).

This approach works well only if the consistency between consecutive LoDs is properly maintained. In case of static models, where lower LoDs are generated from the highest available one, it does not usually cause problems. They may appear when the model is updated and changes are introduced to the objects of lower granularity. Because of lack of topological connection to the modified elements at consecutive levels, propagation of changes to all available levels and the integrity of the whole model may be disturbed. The solution to the problem is the introduction of topological relationships between objects (Ohori et al. 2017). This requires 4D representation of the model, where the fourth dimension is used to map elements between the consecutive LoDs.

Existing solutions dealing with spatial models of different granularity are often focused on generalisation, where new layers with generalised objects are automatically generated from the ones with higher spatial resolution, or individual layers are created independently and common components are mapped (Claridades et al. 2022; Wabiński and Mościcka 2019; Ziouel et al. 2015). Many of them utilise a graph representation and hierarchical representation based on layers interconnected using special links between consecutive layers and specific nodes from different granularity levels are connected. This solution allows to move from a certain granularity level to another through these links. Thus, rough and quick analysis may be performed in models of lower granularity, while it is relatively easy to switch to more detailed model for more sophisticated analysis. This approach is well suited for linear models as well as static models, where no changes are introduced after the model is completed.

Another approach providing complete topological connection between all components is proposed in tGAP model (Oosterom et al. 2014). 2D vector maps in a small scale are generalised in order to obtain a set of maps in a bigger scale. Consecutive scale levels are interconnected in the third dimension by vertical faces used for mapping the same objects represented in a different scale. Such the 2D + 1D model was used to generate a smooth scale by intersecting the model with an arbitrary surface. The idea presented in this paper is based on tGAP but applied to 3D + 1D models.

There are few topological data structures that are capable to implement the aforementioned 4D model, however there is no a specific candidate, which can effectively cover all implementation aspects (Ohori et al. 2015). A solution for a 4D cell implementation was proposed by Ohori et al. (2012), where one of the potential solution was the dual half-edge (DHE) (Boguslawski and Gold 2016). It is a primal–dual data structure designed for cell complex representation. Nodes and half-edges are basic elements used in a construction process. Thanks to the implemented Poincaré duality theorem, following polytopes are available: vertices, edges, faces and cells. Edges of a cell are topologically connected in the primal space, while adjacent cells of the complex are linked using dual edges. However, the DHE is limited to 3D models, and 4D cell complexes are not possible to create.

GIS is not the only field where models of higher dimensionality can be applied. 4D objects are also investigated in the context of computer graphics and gaming (Cavallo 2021). 4D cell complexes are used for analysis of 4D images, where topological aspects are essential in computation (Pacheco and Real 2009).

In this paper, a preliminary research on a new data structure—4D DHE—is presented. It is designed to represent 4D cell complexes including the geometry and topology using basic atomic elements: nodes and half-edges. Attributes attached to these elements allow for storage of semantic information. Two models were implemented using this data structure: a 4D cell complex of cubes and 3D + 1D model where 3D models of different granularity were topologically connected in the fourth dimension.

2 4D DHE

4D DHE is a new data structure, which extends the 3D version of DHE. The concept of half-edges and nodes used as atomic elements is preserved. The Poincaré duality concept is also applied (see Fig. 1). This put a strong requirement on topology representation in the fourth dimension. Each primal node is enclosed by a dual 4D cell, primal edge corresponds to a dual 3D cell, primal face—to a dual face, primal 3D cell—to a dual edge, and primal 4D cell encloses a dual node. In order to meet this requirement it is necessary to topologically link atomic elements into a primal–dual cell complex.

In order to limit a number of pointers necessary to update in a construction process, a simplified version of the 3D DHE (Boguslawski and Gold 2016) was adapted. It allows building individual elements of a cell complex, where 3D cells are joined by a shared face (cell joining by a shared edge or node is not allowed). This requires three pointers to connect half-edges S , N_V and D_{3D} : S joins two half-edges (two half-edges form an edge), N_V joins half-edges sharing the same node and D_{3D} joins half-edges from the adjacent face shared by two 3D cells. Navigation operators to move from one edge to another within an individual 3D cell and between adjacent cells within individual 4D cell are the same as in case of the 3D DHE (Boguslawski 2011).

It should be noted that dual half-edges in 3D are not present in this simplified data structure. It is still capable of representing 3D cell complexes. i.e. individual 4D cells, but does not allow for joining separate 4D cells into a complex, where 4D cells are joined by adjacent (shared) 3D cells. Therefore, a new D_{4D} pointer and dual half-edges in 4D were introduced. D_{4D} points at an associated half-edge in the dual space (see Fig. 2a). A new $Adjacent_{4D}$ operator, shown in Fig. 2b, is defined as a sequence of pointers: $D_{4D}.S.D_{4D}.S$. It links two 4D cells.

The construction process without dedicated operators is a tedious work. Adding new entities and connecting them involves updates of many pointers. Thus, a basic set of construction operators was introduced. Most of them are adapted from the 3D DHE. They are used to build separate 4D cells. From the technical point of view, they are 3D cell complexes. At the same time, an external cell, enclosing the original one,

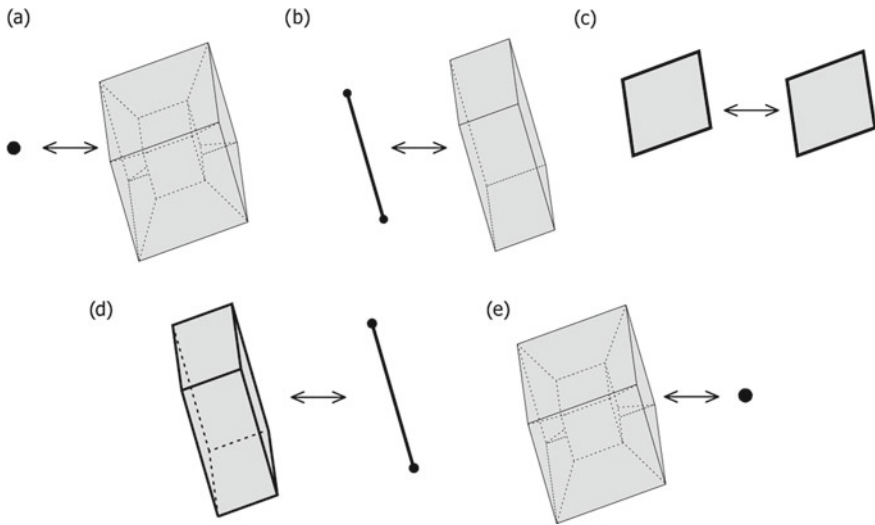


Fig. 1 4D Poincaré duality: **a** primal node corresponds to a dual 4D cell; **b** primal edge corresponds to a dual 3D cell; **c** primal face corresponds to a dual face; **d** primal 3D cell corresponds to a dual edge; **e** primal 4D cell corresponds to the dual node

is created. It is used in next construction steps to join cells into a 4D cell complex, where cells are joined by a shared 3D cell. This required development of a new operator, which works in the fourth dimension. Due to ongoing works related to the construction operators, implementation details are not provided in the paper.

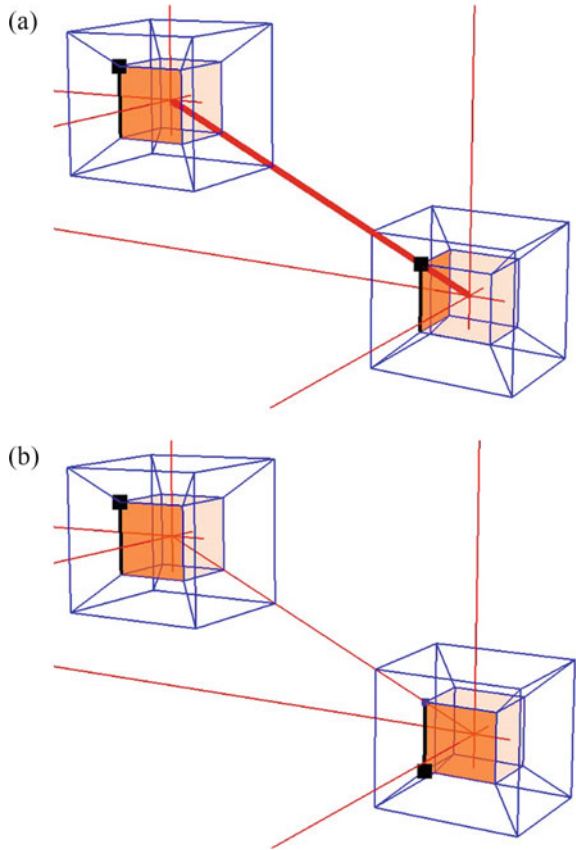
3 4D Cell Complex

A cell complex developed in this research consists of tesseracts (4D cubes) joined by shared 3D cells through dual edges. There are 16 tesseracts sharing one node, which is shown in Fig. 3. A basic visualisation of the model where 4D coordinates are ignored is shown in Fig. 3a, while in Fig. 3b 4D cells are separated and projected into the 3D space in order to show connections between them, while in fact there are no space between them. A primal node shared by all the cells, located in the middle of the complex, is marked as a square dot.

Dual edges are connected with the primal ones using the D_{4D} link, while the $Adjacent_{4D}$ operator is used to link edges in the same space (primal or dual) belonging to the shared faces of the shared 3D cells. The dual edges form a 4D cell (which is also a tesseract) enclosing the shared node (see Fig. 1a).

All dual edges associated with primal edges of an individual 3D cell are bounded by the same end nodes and correspond to a shared cell from the adjacent 4D cell (see Fig. 1d). The dual edges form a bundle, but they belong to different dual cells.

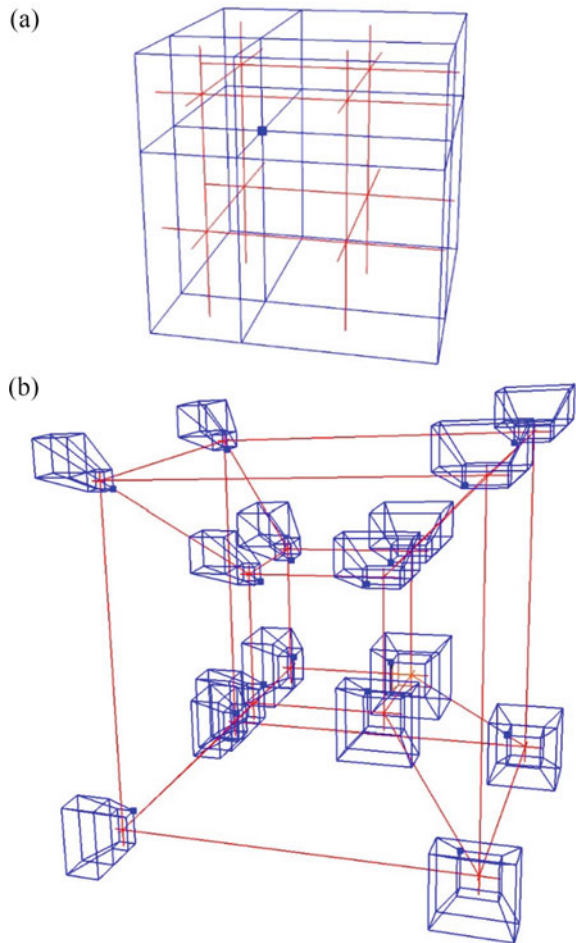
Fig. 2 Two 4D cells in a complex linked through the dual: **a** two primal half-edges (black) from two adjacent 3D cells (orange) are connected to one edge in the dual (thick red); **b** two half-edges (black) linked by $Adjacent_{4D}$



Also, all faces bounded by the same set of nodes are associated with a dual face. There is a primal face shared by four adjacent cells shown in Fig. 4a. Technically this is a bundle of faces belonging to different cells. The face corresponds to the dual face shown in Fig. 4b (also see Fig. 1c).

There are 81 nodes in the primal and 17 in the dual space. Nodes in the dual are associated with 16 4D cells and one external cell enclosing the cell complex. There are 2,688 edges in the cell complex, which gives in total 10,752 half-edges in primal and dual spaces. They represent 216 geometrically unique edges. A geometrically unique edge is represented by a bundle of several edges in the model, and each edge consists of two dual half-edges, which consists of two half-edges one in the primal and one in the dual space.

Fig. 3 16 4D cells sharing a vertex connected via dual into a cell complex. Blue edges belong to the primal structure, red edges belong to the dual. **a** Basic visualisation of the model where 4D coordinates are ignored; **b** Visualisation with projection applied



4 Vario-scale Model

A simple model was developed to test if the proposed solution is applicable for combining 3D separate models into consistent 4D model. A single 3D object is represented in three levels of granularity (see Fig. 5a): two connected polyhedra, a single cube with complex boundary, and a six-face cube. The first one, consisted of two polyhedral, can be considered as a simplification of a model with a complex internal structure, e.g. indoor model or a building divided into sections. The second one, with a complex boundary, is a generalised model (two sections merged into one) but with complex representation of the exterior, e.g. building façade. The third model has the most rough representation, e.g. a building in CityGML LoD1.

3D models at each level of granularity were used to create three 4D models—each volume was extruded in 4D with all spatial relationships preserved. It should be noted

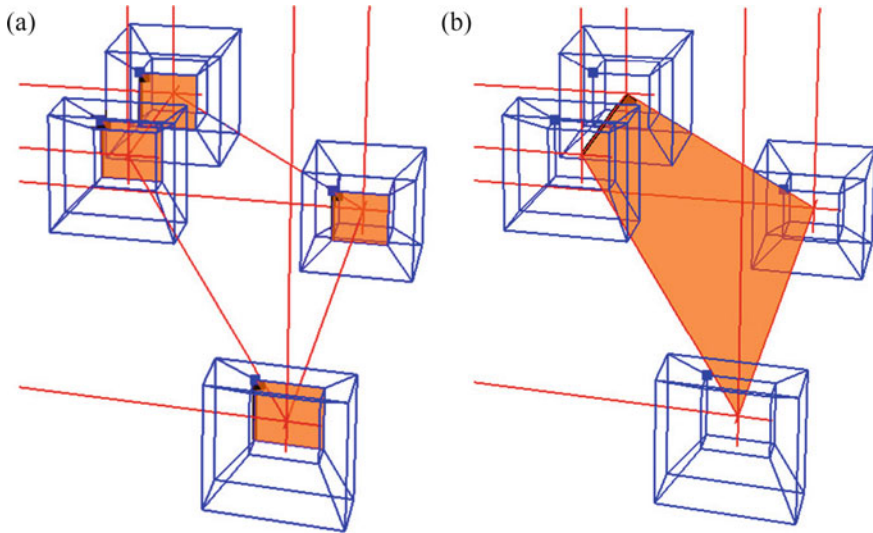
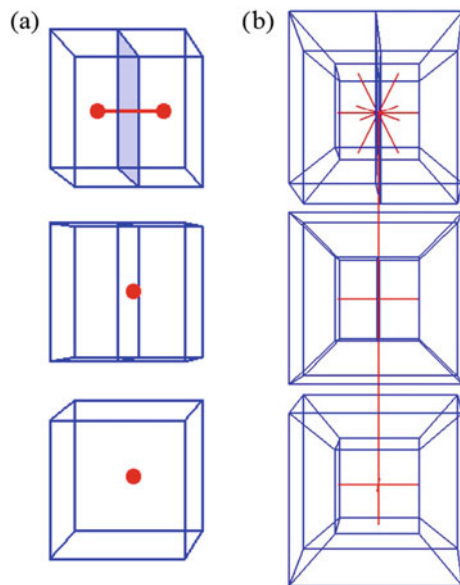


Fig. 4 Shared faces of adjacent 4D cells (a) correspond to a face in the dual (b)

Fig. 5 Vario-scale model:
a original 3D model
 represented in three levels of
 granularity—red dots are
 dual nodes representing
 volumes, red edge—a link
 between volumes; **b** 4D
 model (projection is used for
 visualisation)



that extruded models are 3D cell complexes and no 4D data structure is necessary at this stage. There are three types of cells: cells of the original model, extruded cells and one external cell. This external cell has an infinite volume in 3D but in 4D is a top closing cell. By analogy to extrusion of a polygon into a prism, it is a top face of a prism.

It was important that faces of the external cell have the same geometry as the original model from consecutive level of granularity. This allows for matching and linking component parts of the model. Then 4D models could be linked by a shared cell using dual edges (see Fig. 5b). Moving from one level of granularity to another is possible using extruded cells and dual space.

5 Summary

The 4D DHE data structure is an extension of a 3D version of DHE. It introduces new pointers and construction operators necessary to connect 4D cells into a cell complex using a dual structure. The aim of this preliminary research was to carry out a feasibility study of the Poincaré duality implementation in practice. It proved the correctness of simple models, cell complexes, implemented using the new data structure.

One of disadvantages is the complexity of the model and high storage requirements. In return the model includes rich topology, which can significantly improve computation efficiency. Spatial analysis performed on topological models reduce the need for use of geometrical properties. For instance, finding neighbours of a cell in a complex does not require calculations based on the geometry, as dual edges sharing the dual node associated with the cell are connected with nodes representing adjacent cells. Another example is computation of a path between two locations in the complex, which can be implemented using graph algorithms for the shortest path finding. It is based on topological connections among edges and weight attributes attached to nodes and edges.

Future research includes development of operators for intuitive construction. It will be applied to integrate 3D models represented in a different LoD into one consistent 4D model. It will also require a method for automatic mapping of individual components in consecutive levels of granularity. Id attributes and geometrical properties should allow for matching components for connection in the fourth dimension. The final model can be used for spatial analysis, such as: smooth scale generation and propagating changes introduced in a certain LoD to others.

Another task will be related to introduction of a dual space in each dimension (not only in the fourth dimension). It will increase the complexity of the data structure, operators and storage requirements for models, but at the same time it will allow for multi-dimensional model representation. This will remove limitations of a simplified representation, such as lack of dual nodes for 3D cells and faces, which are needed to store semantical information.

Acknowledgements This research work was funded in whole or in part by the National Science Centre (NCN), Poland as part of the research program OPUS, project no.: 2021/41/B/ST10/03178.

References

- Boguslawski P (2011) Modelling and analysing 3D building interiors with the dual half-edge data structure. Faculty of Advanced Technology, vol. PhD. University of Glamorgan, Pontypridd, Wales, UK, 134 pp
- Boguslawski P, Gold C (2016) The dual half-edge—a topological primal/dual data structure and construction operators for modelling and manipulating cell complexes. *ISPRS Int J Geo-Inf* 5:19
- Cavallo M (2021) Higher dimensional graphics: conceiving worlds in four spatial dimensions and beyond computer graphics forum 40:51–63
- Claridades ARC, Choi H-S, Lee J (2022) An indoor space subsampling framework for implementing a 3D hierarchical network-based topological data model. *ISPRS Int J Geo-Inf* 11:76
- Ohuri KA, Ledoux H, Stoter J (2015) An evaluation and classification of nD topological data structures for the representation of objects in a higher-dimensional GIS. *Int J Geogr Inf Sci* 29:825–849
- Ohuri KA, Ledoux H, Stoter J (2017) Modelling and manipulating spacetime objects in a true 4D model. *J Spat Inf Sci* 14:61–93
- Ohuri KA, Boguslawski P, Ledoux H (2012) Representing the dual of objects in a four-dimensional GIS. In: Rahman AA, Boguslawski P, Gold C, Said MN (eds) *Developments in multidimensional spatial data models*. Springer
- Ohuri KA, Ledoux H, Stoter J (2015) Storing a 3D city model, its levels of detail and the correspondences between objects as a 4D combinatorial map. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci II-2/W2:1–8*
- Pacheco A, Real P (2009) Getting topological information for a 80-adjacency Doxel-based 4D volume through a polytopal cell complex. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Lecture Notes in Computer Science, Vol. 5856* pp 279–286
- van Oosterom P, Meijers M, Stoter J, Šuba R (2014) Data structures for continuous generalisation: tGAP and SSC. In: Burghardt D, Duchêne C, Mackaness W (eds) *Abstracting geographic information in a data rich world: methodologies and applications of map generalisation*. Springer International Publishing, Cham, pp 83–117
- Wabiński J, Mościcka A (2019) Automatic (tactile) map generation—a systematic literature review. *ISPRS Int J Geo-Inf* 8:293
- Ziouel T, Amieur-Derbal K, Boukhalfa K (2015) SOLAP On-the-fly generalization approach based on spatial hierarchical structures. Springer International Publishing, pp 279–290

3D Topology Rules Implementation in Spatial Database



Syahiirah Salleh, Uznir Ujang, and Suhaibah Azri

Abstract Topological information consists of connectivity, adjacency and containment properties which describes relationships between 3D objects. This information provides the basis for more complex 3D spatial analyses. However, only 2D topology rules are available in spatial databases. Consequently, the results based on a 2D topology will also be limited to 2D. This hinders the capabilities of a spatial database in handling 3D data. This study proposed an implementation of 3D topology rules within a spatial database to define valid topological relationships between 3D objects. A 36-Intersection Model that describes intersections between objects interiors, boundaries and exteriors were developed as an additional PL/SQL script for Oracle spatial database. The intersections included ten topological groups with dimension of intersection from 0 to 3D, which are point-to-point, point-to-line, point-to-region, point-to-volume, line-to-line, line-to-region, line-to-volume, region-to-region, region-to-volume and volume-to-volume. Six cases were tested which included 3D objects with disjoint, meet (touches), equals, contains, inside (within) and overlaps topological relationship. As a result, the intersection matrix describing the dimension of intersections between objects could be determined. The 36IM topology rules were able to determine the topological relationships between 3D objects without requiring object decomposition and additional storage. Therefore, 3D topology rules are a lightweight but effective method to maintain topological information between 3D objects.

Keywords 3D topology · Topology rules · Topological relations

This article was selected based on the results of a double-blind review of an extended abstract.

S. Salleh (✉) · U. Ujang · S. Azri
3D GIS Lab, Department of Geoinformation, Faculty of Built Environment, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia
e-mail: syahiirah@graduate.utm.my

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
T. H. Kolbe et al. (eds.), *Recent Advances in 3D Geoinformation Science*, Lecture Notes in Geoinformation and Cartography, https://doi.org/10.1007/978-3-031-43699-4_35

573

1 Introduction

Spatial data which consist of 3D geometrical objects can be stored in various 3D data formats. Spatial databases store 3D data differently based on the capabilities and applications of the respective databases. For example, a desktop GIS such as ArcGIS stores 3D data as a multi-patch feature while database management system (DBMS) with spatial subsystem such as Oracle stores 3D data as a SDO geometry data type. ArcGIS has limited capabilities in storing 3D data due to its multi-patch feature, which converts 3D data into a multi-polygon object. In contrast, the SDO geometry type in the Oracle database stores 3D data as a 3D solid or composite. Parallel to that, topology is an accompaniment to the geometric properties of a spatial object. Topology can be defined as properties that describe how objects in a space are related, which includes containment, adjacencies, and connectivity information (Ellul and Haklay 2006). While topological information may seem simple, it plays a fundamental role in enabling more complex analyses such as 3D data validation and various applications (e.g., indoor navigation, 3D spatial queries, and semantic reconstruction).

A crisp clustering algorithm designed specifically to quantize 3D geospatial vector data was introduced a study that aims to group similar objects together, facilitating efficient storage and analysis of 3D data (Azri et al. 2015). Besides that, a new 3D data structure called the Voronoi classified and clustered data constellation was also developed for geomarketing strategies (Azri et al. 2020). This introduces an innovative approach to organising and analyzing 3D geospatial data for marketing purposes. Even though Voronoi structure is used, there is a need to find the best way to preserve the topological relationship for spatial objects.

As a spatial property, the preservation of the topology within a spatial database is important. A node-edge graph is often used to preserve connectivity between elements stored in a spatial database. For example, building elements can be represented in a connectivity graph where corridors and stairs are nodes connected by edges to room nodes (Song et al. 2016). Similarly, spatial data that have a hierarchical structure also rely on topology to execute spatial queries. A 3D volumetric connectivity network can be derived from topological links of neighboring 3D space subdivisions within a building based on its semantics (Diakité and Zlatanova 2018). It is evident that the 2D graph-based topology is sufficient for navigation and network analysis. However, the connectivity represented by the graphs remains in 2D. Therefore, capabilities of 3D topological information should also be used to accurately represent 3D spatial data connectivity, containment, and adjacencies.

Various approaches can be implemented to preserve topological information within a spatial database. A 3D topological data structure specifies how the topological properties of an object are physically stored within a database. For example, a 3D topological data structure of compact abstract cell complexes (CACC) stores topological primitives as atomic cycles that represent connectivity between primitives (Ujang et al. 2019). In addition to that, a topological model can also be used as a

schema to maintain topological information within a spatial database without physically implementing a data structure (Ohuri et al. 2015). For example, a table storing topological information of connected nodes can be populated from object geometries in the Oracle spatial database (Kothuri et al. 2007). On the contrary, topology rules are a set of conditions that define topological relationships between objects stored in a spatial database. Topology rules do not require a physical data structure or topology model to determine topological relationships between objects. Therefore, topology rules are a lightweight approach to maintain topological information in a spatial database.

This paper attempted to implement a 36-Intersection Model (36IM) as 3D topology rules within a spatial database. The 3D topology rules were developed as an additional PL/SQL script for an Oracle spatial database that stored 3D spatial data. Six simple cases were used as an initial test which are “meets (touches)”, “disjoint”, “equals”, “overlaps”, “contains”, and “inside (within)”.

2 Background

Most spatial databases already have built-in topology mechanisms. However, the mechanisms are based on 2D topology rules which determines intersections between nodes, edges and faces. Desktop GIS software such as QGIS employs the Dimensionally Extended Nine Intersection Model (DE-9IM) as a set of topology rules that determine topological relationships based on the intersections of objects’ interiors and boundaries. Similarly, DBMS such as PostgreSQL with PostGIS extension and Oracle Spatial also implements the DE-9IM via additional building of topology tables or layers (Salleh et al. 2021a). Besides that, desktop GIS software such as ArcGIS has its own set of 32 topology rules that define valid interactions between 0D, 1D, and 2D objects which can be applied directly to the features (Martinez-Llario et al. 2017). Although most of the spatial databases mentioned are capable of supporting 3D objects, the implementation of 2D topology rules also limits the highest dimension of intersection between objects to 2D (Ellul and Haklay 2006). Therefore, 3D objects must be decomposed into lower-dimensional objects before being able to apply 2D topology rules. As a result, the interpretation of 2D topological relationships is limited to 2D and does not describe 3D topological interactions. Consequently, the limitation to the absence of 3D topology also limits the execution of simple 3D spatial queries and query performance (Solihin et al. 2017). This is counterproductive to the current capabilities of spatial databases that support 3D objects, spatial queries, and applications.

In this study, a set of 3D topology rules is implemented based on the 36IM. In essence, the 36IM was extended from the 27IM by adding a third dimension, where intersections between 3D volumes are also taken into account (Salleh et al. 2022; Shen et al. 2017). Ten topological groups can be tested in the 36IM which consists of point-to-point, point-to-line, point-to-region, point-to-volume, line-to-line, line-to-region, line-to-volume, region-to-region, region-to-volume and volume-to-volume.

The 36-IM is represented by a 12×3 intersection matrix that can be simplified into a 3×3 intersection matrix that holds the highest dimension of intersection. The 36IM is denoted in Eq. (1). The topological groups supported by 36IM ensure that each intersection can be represented from 0 to 3D.

$$R_{36IM}(A, B) = \begin{bmatrix} \chi_{0,1,2,3}(A^o \cap B^o) & \chi_{0,1,2,3}(A^o \cap \partial B) & \chi_{0,1,2,3}(A^o \cap B^-) \\ \chi_{0,1,2,3}(\partial A \cap B^o) & \chi_{0,1,2,3}(\partial A \cap \partial B) & \chi_{0,1,2,3}(\partial A \cap B^-) \\ \chi_{0,1,2,3}(A^- \cap B^o) & \chi_{0,1,2,3}(A^- \cap \partial B) & \chi_{0,1,2,3}(A^- \cap B^-) \end{bmatrix} \quad (1)$$

3 Methodology

This study attempted to implement 3D topology rules for determining topological relationships between 3D objects stored in an Oracle spatial database. The capability of Oracle to store 3D solids provides the support needed to determine intersections between each dimension of geometries in the 3D space based on the 36IM. The 3D object is composed of the 3D interior volume, while the boundaries are composed of 2D faces, 1D edges, and 0D nodes. Figure 1 shows the graphical representation of the geometries that make up a 3D object interior and boundary.

3.1 36IM Topology Rules Development

The 3D topology rules were defined based on each intersection between objects' interiors, boundaries and exterior. The intersection of objects must be a subset of both objects. Therefore, the dimension of intersection does not exceed the lowest dimension of the set of objects as defined by Eq. 2. The definitions of each intersection between object A and object B are described in Table 1.

$$\dim(S_1 \cap S_2) \leq \min(\dim(S_1), \dim(S_2)) \quad (2)$$

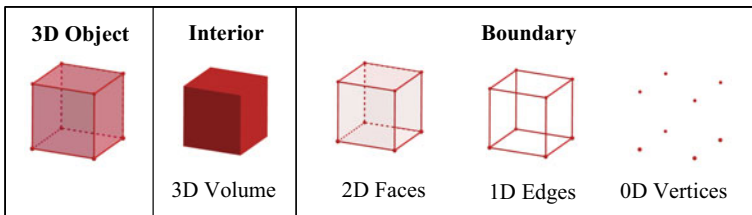


Fig. 1 Interior and boundary geometries of a 3D solid object

Table 1 Definition of intersections between 3D objects

| Intersection | Definition | Intersection matrix value |
|----------------------|---|---|
| Interior-to-interior | $dim(A^\circ \cap B^\circ) = 3 \Leftrightarrow A \supseteq B \vee A \subseteq B$ | $\begin{bmatrix} 3 & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$ |
| Interior-to-boundary | $dim(A^\circ \cap \partial B) = \{\partial B \subset A^\circ dim(\partial B) = (0..2)\}$ | $\begin{bmatrix} * & 0, 1, 2 & * \\ * & * & * \\ * & * & * \end{bmatrix}$ |
| Boundary-to-interior | $dim(\partial A \cap B^\circ) = \{\partial A \subset B^\circ dim(\partial A) = (0..2)\}$ | $\begin{bmatrix} * & * & * \\ 0, 1, 2 & * & * \\ * & * & * \end{bmatrix}$ |
| Boundary-to-boundary | $dim(\partial A \cap \partial B) = \{(\partial A \subset \partial B) \vee (\partial A = \partial B) dim(\partial A) = (0..2)\}$ | $\begin{bmatrix} * & * & * \\ * & 0, 1, 2 & * \\ * & * & * \end{bmatrix}$ |
| Interior-to-exterior | $dim(A^\circ \cap B^-) = 3 \Leftrightarrow A^\circ \not\subset B$ | $\begin{bmatrix} * & * & 3 \\ * & * & * \\ * & * & * \end{bmatrix}$ |
| Exterior-to-interior | $dim(A^- \cap B^\circ) = 3 \Leftrightarrow B^\circ \not\subset A$ | $\begin{bmatrix} * & * & * \\ * & * & * \\ 3 & * & * \end{bmatrix}$ |
| Boundary-to-exterior | $dim(\partial A \cap B^-) = 2 \Leftrightarrow \partial A \not\subset B$ | $\begin{bmatrix} * & * & * \\ * & * & 2 \\ * & * & * \end{bmatrix}$ |
| Exterior-to-boundary | $dim(A^- \cap \partial B) = 2 \Leftrightarrow \partial B \not\subset A$ | $\begin{bmatrix} * & * & * \\ * & * & * \\ * & 2 & * \end{bmatrix}$ |

The construction of 3D topology rules for Oracle spatial database is carried out by writing PL/SQL scripts to determine intersections based on the previous definitions. The intersections between the 3D objects' interior, boundary and exterior were tested based on the 10 topological groups of the 36IM. The highest dimension of intersection is input into the intersection matrix. If no intersection occurs, the value remains – 1. The flowchart for the PL/SQL script is shown in Fig. 2.

The PL/SQL scripts are implemented in Oracle, incorporating logical conditions that determine the topological relationships between 3D objects. A check is executed to determine which condition of the 36IM topology rules matches the intersection

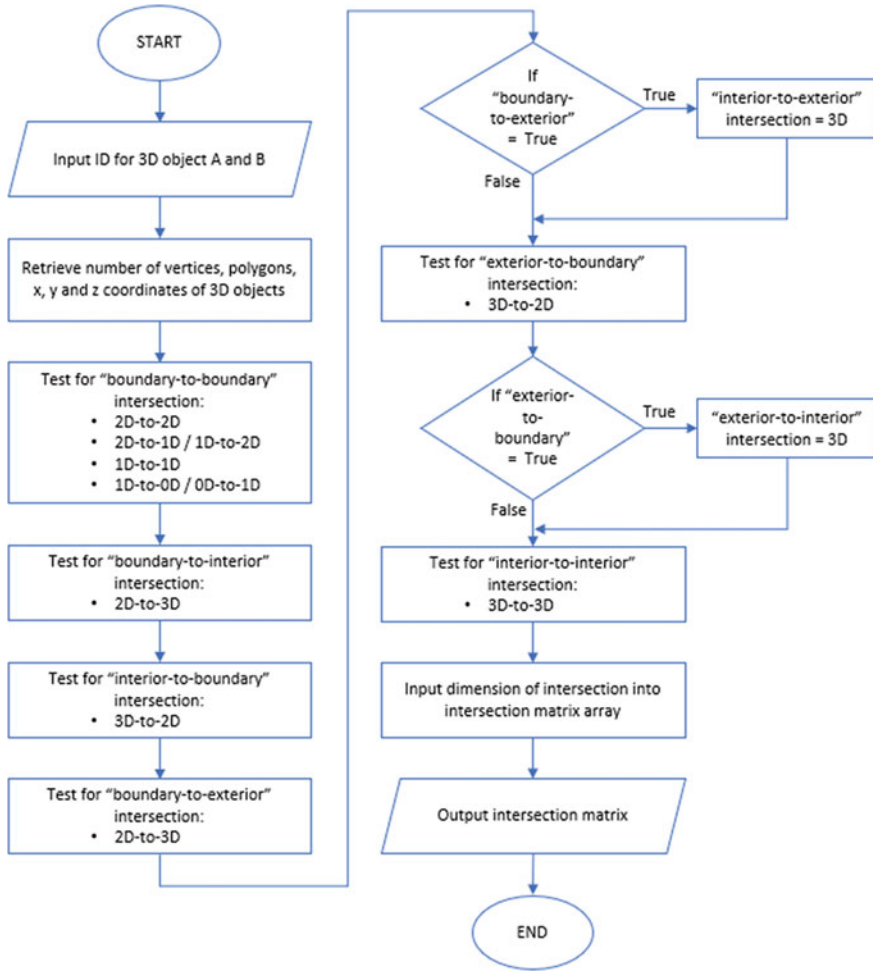


Fig. 2 Flowchart for 3D topology rules PL/SQL script

matrix of the 3D objects tested. As a result, the topological relationship of the objects is displayed. Six cases were used as an initial test for this study, which consisted of a pair of 3D objects for each case. The 3D objects stored in the database were simple water-tight objects with no holes. This may not replicate real-world complex objects but is sufficient as an initial test for implementation.

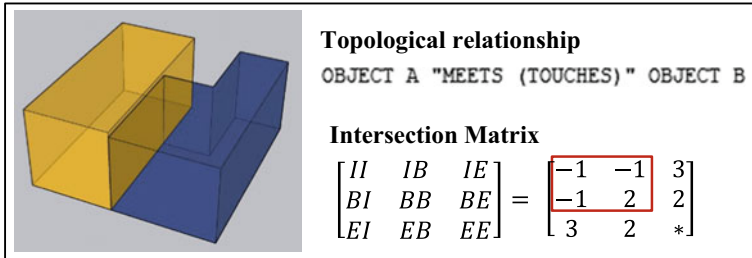


Fig. 3 Meets (touches) topological relationship and intersection matrix results

4 36IM Results and Discussion

Six cases were tested which consisted of 3D objects with a topological relationship of equal, disjoint, meet (touch), contains, inside (within), and overlaps. It was found that the topological relationships for 3D objects were able to be determined based on the 36IM topology rules. The proposed implementation of 3D topology rules within an Oracle spatial data-base allowed topological relationships between 3D objects to be determined without any decomposition into lower dimensions. This is due to the nature of 36IM that handles ten topological groups which includes cross and equal dimension objects. As a result, the geometrical integrity of objects is preserved while maintaining accurate 3D topological information. This implementation also required no additional storage of data structures or models.

4.1 Case 1: Meet (Touches) Topological Relationship

3D objects that “meet (touches)” at a 2D face is shown in Fig. 3. The resulting intersection matrix shows that no intersection occurs between the 3D volume interiors (II) and 3D volume interiors to object exteriors (IE). In addition, the intersection matrix also shows that 2D face intersections occur between object boundaries (BB). This in turn matches the logical condition for meet topological relationship.

4.2 Case 2: Equals Topological Relationship

Two 3D objects that are equal must have the exact same intersecting boundaries and interiors. The intersection matrix and topological relationship is shown in Fig. 4 where boundary-to-boundary (BB) is 2 and interior-to-interior (II) is 3. In addition, no intersection occurs between boundary-to-exterior (BE) and interior-to-exterior (IE).

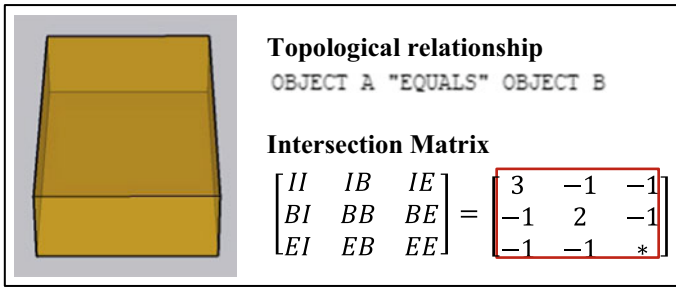


Fig. 4 Equals topological relationship and intersection matrix results

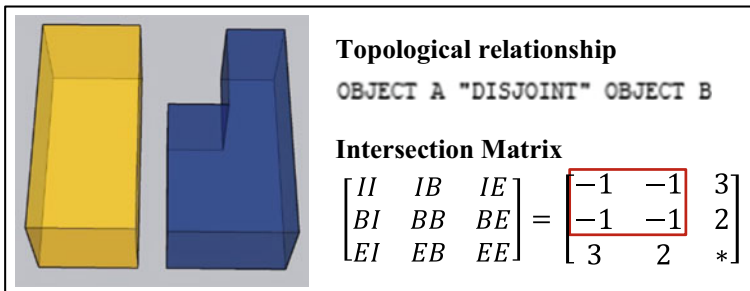


Fig. 5 Disjoint topological relationship and intersection matrix results

4.3 Case 3: Disjoint Topological Relationship

In order to satisfy the conditions for a “disjoint” topological relationship, no intersections can occur between objects’ boundaries and interiors (II, BB, and IB). Each objects’ boundaries and interiors will intersect with the exteriors (BE and IE). The resulting intersection matrix based on the 36IM is shown in Fig. 5.

4.4 Case 4: Contains Topological Relationship

Object A that contains Object B cannot have any boundary-to-interior intersection (BI). Apart from that, the interiors (II) and interior-to-boundary (IB) must always be true. No boundaries of Object B can intersect with exterior of Object A (EB). The intersection matrix of Object A “contains” Object B is shown in Fig. 6.

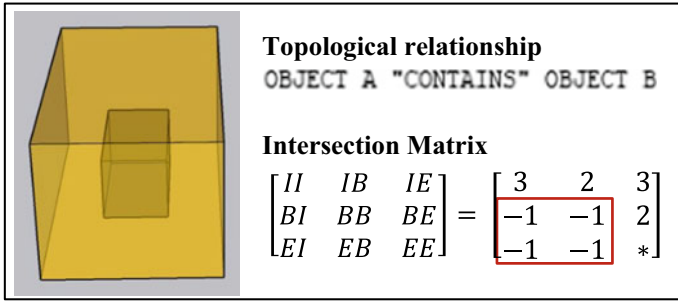


Fig. 6 Contains topological relationship and intersection matrix results

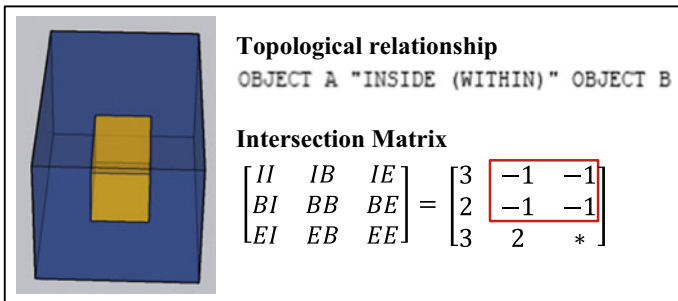


Fig. 7 Inside (within) topological relationship and intersection matrix results

4.5 Case 5: Inside (Within) Topological Relationship

Opposite to the previous case, Object A is inside Object B when intersection occurs between boundary-to-interior (BI) and interior-to-interior (II). No intersection can occur between the boundaries and interiors of Object A with the exterior of Object B (BE and IE). Figure 7 describes the case and resulting intersection matrix based on the 36IM.

4.6 Case 6: Overlaps Topological Relationship

An overlaps topological relationship is a more general condition where the main intersection is between 3D interior-to-interior (II). The resulting intersection matrix based on 36IM is shown in Fig. 8.

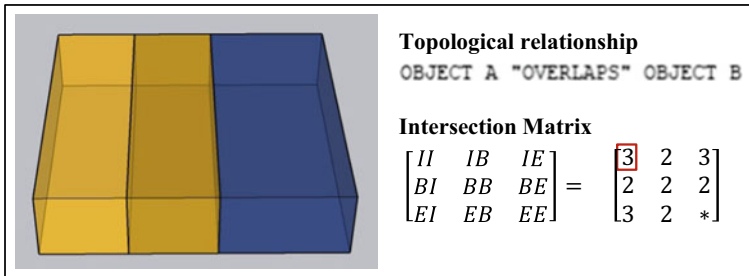


Fig. 8 Overlaps topological relationship and intersection matrix results

5 Conclusion

Spatial databases as the foundation of 3D applications have the capability of storing 3D spatial objects. Parallel to that, the topological information that describes the interactions between 3D objects should also be maintained within a spatial database. As database storage can be costly, a method to maintain topological information while executing on-the-fly spatial queries is required. 3D topology rules that define valid topological interactions between 3D objects based on the 36IM could determine topological relationships without requiring additional storage within the database. In addition, by maintaining 3D topological information, no decomposition of objects into lower dimensions is required. This safeguards the geometric and topological properties of the 3D objects. Future studies would address the use of 36IM topology rules for complex objects and real-world situations such as in BIM or creating a 3D virtual environment (see Basir et al. 2018; Salleh et al. 2021b).

Acknowledgements This research was supported by Ministry of Education (MOE) through Fundamental Research Grant Scheme (FRGS/1/2021/WAB07/UTM/02/2).

References

- Azri S, Anton F, Ujang U, Mioc D, Rahman AA (2015) Crisp clustering algorithm for 3D geospatial vector data, lecture notes in geoinformation and cartography. Springer, pp 71–85
- Azri S, Ujang U, Abdul Rahman A (2020) Voronoi classified and clustered data constellation: a new 3D data structure for geomarketing strategies. ISPRS J Photogramm Remote Sens 162:1–16
- Basir WA, Majid Z, Ujang U, Chong A (2018) Integration of GIS and BIM techniques in construction project: a review. Int Arch Photogramm Remote Sens Spat Inf ISPRS Arch 4/W9:307–316
- Diakité AA, Zlatanova S (2018) Spatial subdivision of complex indoor environments for 3D indoor navigation. Int J Geogr Inf Sci 32(2):213–235. <https://doi.org/10.1080/13658816.2017.1376066>
- Ellul C, Haklay M (2006) Requirements for topology in 3D GIS. Trans GIS 10(2):157–175
- Kothuri R, Godfrind A, Beinat E (2007) Pro oracle spatial for oracle database 11g. Apress, Berkeley, California

- Martinez-Llario J, Coll E, Núñez-Andrés M, Femenia-Ribera C (2017) Rule-based topology system for spatial databases to validate complex geographic datasets. *Comput Geosci* 103:122–132
- Ohuri KA, Ledoux H, Stoter J (2015) An evaluation and classification of nD topological data structures for the representation of objects in a higher-dimensional GIS. *Int J Geogr Inf Sci* 29(5):825–849
- Salleh S, Ujang U, Azri S (2021a) 3D topological support in spatial databases: an overview. *Int Arch Photogramm Remote Sens Spat Inf Sci* 46(4/W5):473–478
- Salleh S, Ujang U, Azri S (2021b) Virtual 3D campus for Universiti Teknologi Malaysia (UTM). *ISPRS Int J Geo Inf* 10(6):356
- Salleh S, Ujang U, Azri S (2022) Representing 3D topological adjacencies between volumes using a 36-intersection model. *Geomatics Environ Eng* 16(2):127–155
- Shen J, Zhou T, Chen M (2017) A 27-intersection model for representing detailed topological relations between spatial objects in two-dimensional space. *ISPRS Int J Geo Inf* 6(2):37
- Solihin W, Eastman C, Lee Y-C (2017) Multiple representation approach to achieve high-performance spatial queries of 3D BIM data using a relational database. *Autom Constr* 81:369–388. <https://doi.org/10.1016/j.autcon.2017.03.014>
- Song Y-Q, Niu L, He L, Wang R (2016) A grid-based graph data model for pedestrian route analysis in a micro-spatial environment. *Int J Autom Comput* 13(3):296–304. <https://doi.org/10.1007/s11633-016-0979-1>
- Ujang U, Anton Castro F, Azri S (2019) Abstract topological data structure for 3D spatial objects. *ISPRS Int J Geoinf* 8(3):102

Indoor/Outdoor Modelling and Navigation

RGB-D Semantic Segmentation for Indoor Modeling Using Deep Learning: A Review



Ishraq Rached, Rafika Hajji, and Tania Landes

Abstract With the availability and low cost of RGB-D sensors, indoor 3D modeling from RGB-D data has gained more interest in the research community. However, this topic is still challenging because of the complexity of indoor environments and the poor quality of RGB-D data. To deal with this problem, a focus on semantic segmentation as a first and crucial step in 3D modeling process is primordial. The main purpose of this paper is to offer a review of recent researches carried out on RGB-D semantic segmentation. Especially approaches based on deep neural network, their datasets, their metrics, and their challenges and limits are presented. Based on this state of the art, guidelines to improve research in this field are proposed.

Keywords Indoor modeling · RGB-D · Semantic segmentation · Deep learning

1 Introduction

In the past few decades, Terrestrial Laser Scanning (TLS) sensors have become the most commonly used 3D data acquisition systems, which generate dense points clouds with high accuracy. However, most existing laser scanning systems are expensive and are short of RGB information as shown in Tang et al. (2019). Alternatively, Simultaneous Localization and Mapping (SLAM) systems are less expensive than

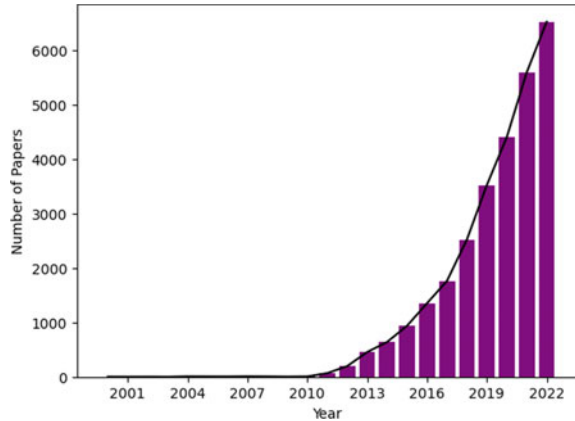
This article was selected based on the results of a double-blind review of the full paper.

I. Rached (✉) · R. Hajji
College of Geomatic Sciences and Surveying Engineering, Institute of Agronomy
and Veterinary Medicine, 6202 Rabat, Morocco
e-mail: Ishraq.rach@gmail.com

R. Hajji
e-mail: r.hajji@iav.ac.ma

T. Landes
ICube Laboratory UMR 7357, Photogrammetry and Geomatics Group, National Institute of
Applied Sciences (INSA Strasbourg), 24, Boulevard de la Victoire, 67084 Strasbourg, France
e-mail: tania.landes@insa-strasbourg.fr

Fig. 1 Trend of publications about RGB-D semantic segmentation (google scholar)



TLS systems and allow a quick scan of the scene with a good accuracy. However, the price of some SLAM systems remains not affordable for many users.

Recently, 3D low cost acquisition systems such as RGB-D cameras are increasingly gaining more interest within the scientific community. They have the advantages of low cost and fast data acquisition. Among them. Recent researches focus on semantic segmentation as a first step in 3D modeling and BIM (Building Information Model) generation from RGB-D data. This increasing interest is illustrated in Fig. 1 which reports the number of published papers about RGB-D semantic segmentation from Google Scholar between 2001 and 2022.

In indoor scenes, the depth information provided by an RGB-D sensor has many advantages: it helps to distinguish objects that have similar colors, and it can also help to better estimate the shape and boundaries of objects. In outdoor scenes, the depth information can be used to separate objects from the background, such as trees, cars, buildings. That's why the automation of the process of 3D modeling from RGB-D data has been the subject of several researches.

One current trend in the literature is to rely on semi-automatic or fully automatic frameworks by using Deep Learning (DL). DL is a type of machine learning that involves training artificial neural networks on large amounts of data in order to learn complex patterns and features as mentioned in several studies (Chen et al. 2014, 2017; Long et al. 2015).

This paper gives an overview of existing RGB-D semantic segmentation approaches based on DL, their datasets and metrics. We then propose guidelines for future research that aims to reconstruct automatically BIM models based on RGB-D sensors exclusively.

The remainder of this paper is structured as follows: Sect. 2 present a review of the RGB-D semantic segmentation approaches based on DL. Section 3 summarizes the existing RGB-D datasets, their challenges and the metrics proposed in the literature to assess DL models. In Sect. 4, we propose future research agenda that will help to advance the development of RGB-D semantic segmentation methods especially for indoor modelling. We conclude the paper and give research insights in Sect. 5.

2 RGB-D Semantic Segmentation Approaches

Computer vision has made significant advances in recent years, thanks to Deep Convolutional Neural Networks (DCNNs) and the increasing performance of the processors. Semantic segmentation, being a field of computer vision, has also benefited from this advancement. The use of deep neural networks and large RGB-D datasets has led to significant improvements in the performance of semantic segmentation algorithms. Semantic segmentation models leverage the depth information provided by depth sensors of a scene in addition to the standard RGB image for improving the segmentation performance as mentioned in Barchid et al. (2021). There are several approaches and architectures used in RGB-D semantic segmentation with DCNN. In this section, we propose a classification of the methods applied in recent years organized in five categories from the point of view of network structure.

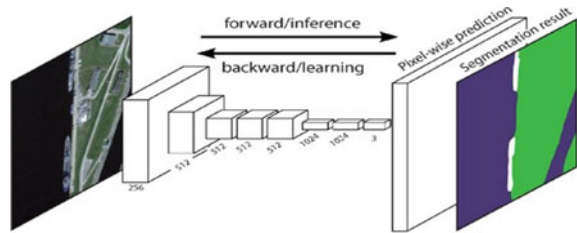
2.1 Encoder Decoder Architecture

In recent years, the encoder-decoder architecture has become one of the most commonly used frameworks for RGB-D semantic segmentation (Fooladgar and Kasaei 2019; de Oliveira Junior et al. 2018). The basic idea of encoder-decoder structure is to use an encoder network to extract features from the input image, and a decoder network to map the features back to the original image resolution for semantic segmentation as shown in Fig. 3. The encoder network is typically a CNN that is used to extract features from the input image. It is typically implemented using transposed convolutions, also known as “up sampling layers”.

For RGB-D semantic segmentation, Badrinarayanan et al. (2015) propose SegNet network, which consists of an encoder network, a corresponding decoder network, and a pixel wise classification layer. It achieves this by using pooling indices from the encoder stage to perform up sampling in the decoder stage, which reduces the number of parameters required for training. However, SegNet uses max pooling indices for up sampling without learning, which may lead to loss of information and artifacts in the output image. In Jiang et al. (2018), the authors propose RedNet: a residual encoder-decoder network, that combines the short skip connections and long skip connections to improve the accuracy of the segmentation results.

Fully convolutional networks (FCNs) are a popular approach in RGB-D semantic segmentation with deep neural networks. The basic idea is to use convolutional layers to preserve spatial information and improve segmentation results. As shown in Fig. 2, this is achieved by replacing fully connected layers in traditional CNNs (Convolutional Neural Networks) with convolutional layers. For RGB-D semantic segmentation, (Song et al. 2015; Gupta et al. 2014) encode depth information into HHA (Horizontal Disparity, Height above ground, and Angle) image as input. Based on FCN network, HHA is used to encode the depth information, because neural networks are weak to treat directly the depth information which is unstruc-

Fig. 2 Fully connected network architecture (Long et al. 2015)



tured geometric information. The suggestion put forward in Wang et al. (2016) was to introduce a feature transformation network to link convolutional and deconvolutional networks, with the underlying assumption that the deconvolution network has the ability to predict semantic information at the pixel level. Su and Wang (2016) propose the use of regularized fully convolutional networks (FCN) which inputs are composed of features and depth. Cheng et al. (2017) proposes the locality sensitive deconvolution network with gated fusion (LSD-GF) based on FCN network. This network has achieved remarkable results, but the high resolution feature maps are ignored, resulting in the loss of the edge information, which causes memory consumption increase. It integrates both dilated and vanilla convolution to recreate FCN architecture. Dilated convolution is used to increase the receptive field of the network without increasing the number of parameters, which helps to capture more context information (Chen et al. 2019).

FCN structure has been shown to be effective in various applications such as building segmentation and semantic image as shown in different articles (Song et al. 2015; Gupta et al. 2014; Wang et al. 2016; Su and Wang 2016; Cheng et al. 2017; Amit Kamran and Shihab Sabbir 2017; Chen et al. 2019). However, FCN can struggle to capture fine grained details due to the pooling and down sampling operations performed in the network. This can lead to loss of spatial resolution and difficulty in accurately localizing object boundaries; moreover, FCN may not be suitable for handling objects of varying scales or sizes, since it applies a fixed-size pooling operation on the feature maps. This can result in over-segmentation or under segmentation of objects of different sizes (Sun et al. 2016).

The encoder-decoder structure is a popular architecture for RGB-D semantic segmentation. However, this structure has some limitations in handling complex scenes with large variations in object scales and shapes. In such cases, the encoder-decoder structure may struggle to capture fine-grained details and object boundaries, leading to sub-optimal segmentation results. To address this issue, recent works have proposed various modifications to the encoder-decoder structure, such as incorporating multi-scale features (Zhou et al. 2020a), attention mechanisms proposed by Bai et al. (2022), Hu et al. (2019), and Zhou et al. (2020b), or shape-aware convolutional layers as shown in Wang and Neumann (2018). Other methods have been proposed to tackle different problems and take advantage of the capabilities of Encoder-Decoder architecture, like AMF proposed by Zhang et al. (2022), MMAFNet by Fooladgar and Kasaei (2019) and PD-CNet by Yang et al. (2023).

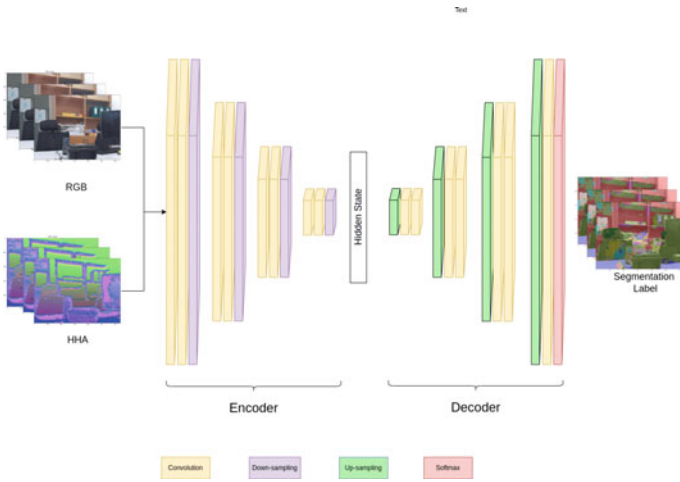


Fig. 3 Encoder decoder architecture (Badrinarayanan et al. 2015)

2.2 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are a type of neural networks that incorporate temporal information by processing sequential data (Fig.4), such as RGB-D video frames. The basic idea is to use RNNs to model the temporal dependencies between RGB-D frames and improve segmentation results.

RNNs have been shown to be effective in semantic segmentation task due to their ability to model long-term dependencies and sequential information (Schmidhuber and Hochreiter 1997). Recent researches have proposed various RNN-based architectures for RGB-D semantic segmentation, which have achieved state-of-the-art performance on benchmark datasets. They use different strategies to combine local feature extraction, mostly using CNN, and RNN (Emre Yurdakul and Yemez 2017; Liang and Hu 2015).

To capture the spatial and temporal dependencies between pixels in the image, Liang and Hu (2015) proposed a Convolutional LSTM (ConvL-STM) network, which achieved better performance than traditional methods that only use RGB or depth information. In a more recent study, Valipour et al. (2017) proposed an end-to-end trainable RNN-based approach for RGB-D semantic segmentation, using a Gated Recurrent Unit (GRU) network. Chen et al. (2014) proposed a model that uses CNNs in combination with fully connected conditional random fields (CRFs). Overall, the use of LSTM, ConvLSTM, GRU, and ConvGRU (Valipour et al. 2017) networks allows for capturing both spatial and temporal dependencies in the RGB-D image data, leading to significant improvement in performance. Kong and Fowlkes (2018) proposed to integrate depth-aware gating into a recurrent convolutional neural network; unlike common multi-scale approaches, this method processes the input image at a single fixed scale but performs pooling at different scales to capture context information.

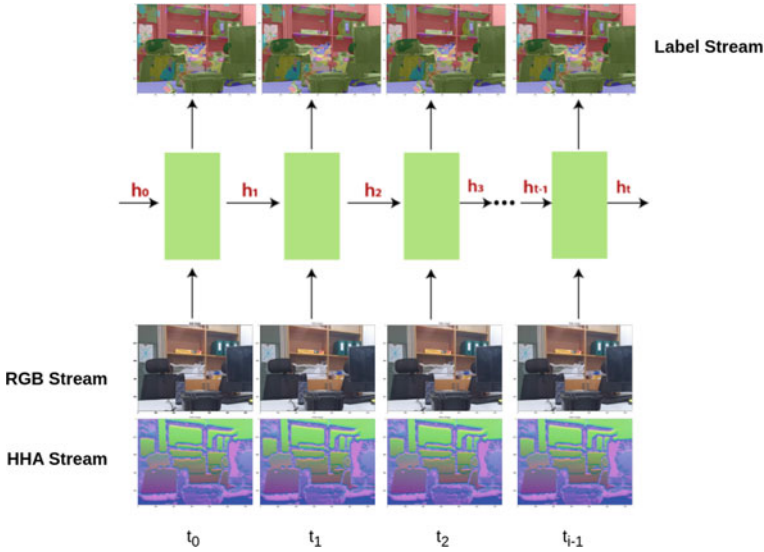


Fig. 4 Recurrent neural network architecture

2.3 Multi-modal Fusion

Multi-modal fusion structure is a technique used to improve the accuracy of semantic segmentation based on RGB-D images. This technique involves combining features extracted from both RGB and depth data to achieve better results (Liu et al. 2022; Deng et al. 2019). The multi-modal fusion structure uses attention mechanisms and residual feature fusion to extract and combine features from both modalities as shown in Deng et al. (2019) and Park et al. (2017). This approach has been shown to be effective in improving the accuracy of semantic segmentation in various applications (Park et al. 2017). In Liu et al. (2022), the authors propose CMX: a cross-modal fusion framework for RGB-X semantic segmentation, which is a model that uses multi-modal fusion to improve semantic segmentation of RGB images. It has been tested on various datasets and has excellent generalization performance in outdoor scenes, with many parameters (He et al. 2023).

The SA-Gate Network proposed by Chen et al. (2020) is a bidirectional cross-modality guided encoder; it recalibrates and fuses complementary information from the two modalities using the SA-Gate unit. Wang et al. (2022) proposed TokenFusion, a multimodal fusion method for transformer-based vision tasks such as RGB-D semantic segmentation. It fuses point cloud and RGB information with a transformer model. The TokenFusion network is designed to effectively fuse multiple modalities of data, which can improve the mIoU of semantic segmentation by 1.8.

Multi modal fusion is based on providing the network with more information and context from different modalities, which can be more robust to variations in

lighting, texture, and object pose (Liu et al. 2022; Deng et al. 2019; Park et al. 2017; Liu et al. 2022; Chen et al. 2020; Wang et al. 2022). For example, depth and lidar can provide geometric cues that are useful for identifying boundaries and shapes, while RGB can provide color and texture information that are useful for recognizing categories and attributes. However, it is important to note that executing multiple tasks simultaneously can increase the processor’s workload, leading to larger memory requirements and slower calculation speed. Therefore, further improvements are necessary to address these issues (Zhang et al. 2022).

2.4 Generative Adversarial Network

Adversarial training is a popular approach to improve the realism of synthetic data and make the model more robust to real-world variations. GANs consist of a generator and a discriminator network (Fig. 5) trained to generate synthetic data similar to real data.

GANs have been used by Liu et al. (2019) and Zhang et al. (2021) for semantic segmentation tasks to improve the performance of segmentation networks. GANs can be used to generate realistic images with pixel-level annotation labels, which can help balance the semantic label distribution and increase segmentation accuracy as shown in Liu et al. (2019). This type of architecture that allows the generation of high quality images is widely used for data augmentation leading to improvement in segmentation accuracy. GANs have also been used for optical coherence tomography (OCT) applications by Kugelman et al. (2022), as well as semi-supervised learning settings where training is performed with a reduced set of annotated images and additional non-annotated images (Peng et al. 2020).

This section has surveyed recent RGB-D semantic segmentation algorithms and categorized them based on their network structure. We have also highlighted the key innovations and challenges of RGB-D methods. Next, we will introduce some common datasets for RGB-D semantic segmentation.

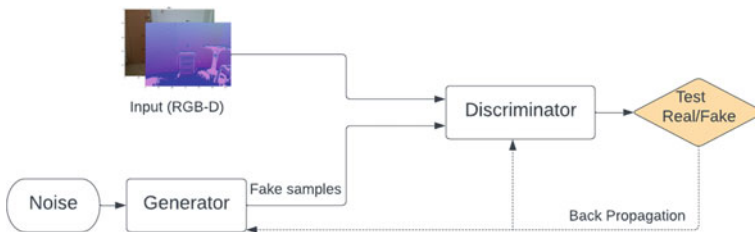


Fig. 5 Generative adversarial network architecture

3 Benchmarks

3.1 Datasets

In recent years, public RGB-D datasets have been released and used in many fields, such as scene reconstruction, semantic classification, and object recognition (Yuan et al. (2021)). RGB-D datasets provide a source of annotated data for training and evaluating machine learning models. They provide a large number of RGB-D images that have been annotated with labels indicating the class of each pixel (e.g., wall, floor, chair, etc.) or object in the scene. In the subsections below, we give an overview of the most popular public RGB-D datasets.

3.1.1 RGB-D Datasets for Indoor Semantic Segmentation

Several RGB-D datasets are available for the user community. The aim of these datasets is to give the possibility to evaluate the performance of indoor semantic segmentation models. In the following, we introduce the most popular semantic segmentation RGB-D datasets and analyze the main challenges related to them:

- SUN RGB-D proposed by Song et al. (2015): captured using different depth cameras, this dataset contains 10,335 RGB-D images of various indoor scenes from 37 classes, such as living rooms, kitchens, and offices. It is an indoor dataset that contains a relatively small number of images, which may limit its ability to train large models. It also contains data from multiple sensors, which may make it difficult to compare performance across different sensor types.
- NYU Depth v2 proposed by Silberman et al. (2012): this dataset is the most popular for RGB-D indoor segmentation due to its large size and diversity of indoor scenes (Wang et al. 2023; Zhou et al. 2022). It contains 1,449 images with depth maps, evaluated on 40 classes. It is captured using Kinect V1 structured light sensor, which may limit their ability to generalize to other sensor types.
- ScanNet proposed by Dai et al. (2017): this dataset contains 2.5M RGB-D images of indoor scenes, such as living rooms, bedrooms, and offices, collected using the Microsoft Kinect v2 sensor. The images include pixel-level annotations for 20 object classes; this dataset also includes 3D reconstruction of the scenes. Scan Net is a big dataset, which may make it computationally expensive to train models on.
- Stanford 2D-3D-S proposed by Armeni et al. (2017): This dataset includes 70,496 RGB images with the associated depth maps. It also provides point clouds, which allows for the evaluation of both 2D and 3D-based algorithms for semantic segmentation. Stanford 2D-3D-S dataset contains images of objects with varying levels of complexity, lighting conditions, and occlusion. This variability makes it difficult for computer vision algorithms to generalize across the entire dataset (Fig. 6).

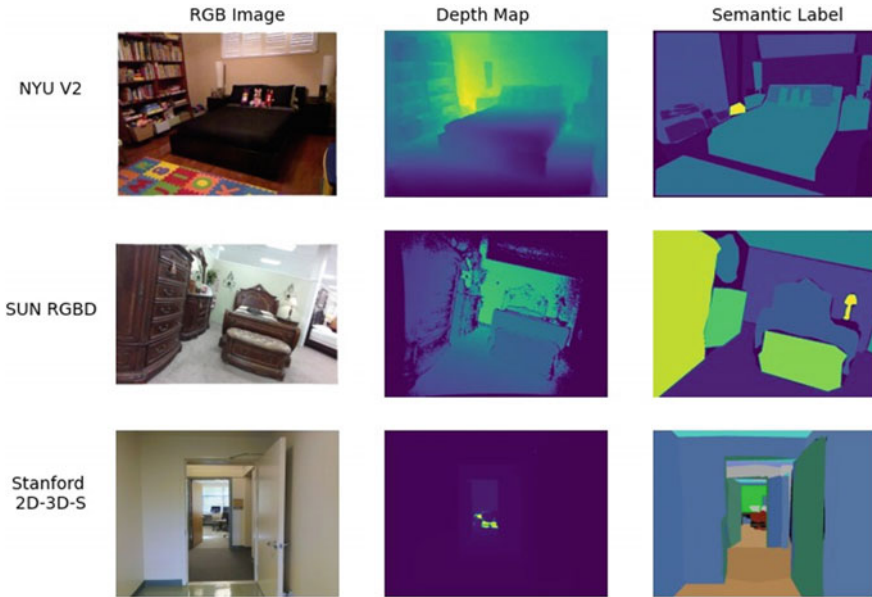


Fig. 6 Samples of indoor RGB-D datasets: NYUDepth-V2 by Silberman et al. (2012), SUN RGB-D by Song et al. (2015), Standford 2D-3D-S by Armeni et al. (2017)

3.1.2 RGB-D Datasets for Outdoor Semantic Segmentation

Outdoor environments present specific challenges for semantic segmentation, such as varying lighting conditions and occlusions; therefore, the used dataset should be designed to reflect these challenges. In the following, we introduce the most popular semantic segmentation RGB-D datasets used for outdoor segmentation and we analyze the main challenges related to them:

- KITTI proposed by Geiger et al. (2012) and Geiger et al. (2015) is a dataset of RGB and depth images collected from a moving car. It includes a total of 7481 training and 7518 test images, along with 20,000 disparity images labeled with semantic information. The main challenges of the dataset are limited variability in scenes and a limited number of classes.
- Cityscapes (Cordts et al. (2016)) is a collection of 30,000 RGB and depth images from a moving car, providing bounding box annotations for object detection models. It is imbalanced, with some classes appearing more frequently than others, which can affect the accuracy of the models trained on the dataset (Atif et al. 2022).

There are many other datasets available for this task, such as Structured3d (Zheng et al. 2020), InteriorNet (Li et al. 2018), and more. Some of these datasets are summarized in Table 1, which shows their main characteristics and challenges (Fig. 7).

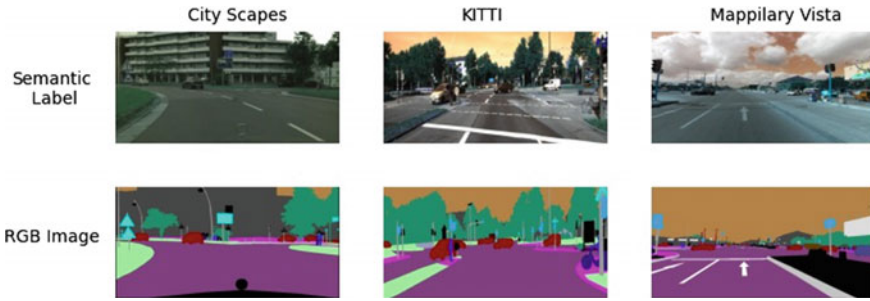


Fig. 7 Samples of outdoor RGB-D datasets: city scapes (Cordts et al. 2016); KITTI (Geiger et al. 2012, 2015); Mappillary Vista (Neuhold et al. 2017)

In this section, we have reviewed some of the most relevant RGB-D semantic segmentation datasets. We have discussed their main characteristics, such as the number of classes, the size of the images, the type of sensors, and their applications such as indoor or outdoor. Table 1 presents some recently used RGB-D datasets for semantic segmentation on indoor and outdoor.

In the next section, we will present some of the most important metrics in use to evaluate the performance of RGB-D semantic segmentation models.

3.2 Metrics

In order to achieve improved results in RGB-D semantic segmentation, it is important to assess the performance of each proposed approach, as such; this section introduces the most widely-used metrics for evaluating RGB-D semantic segmentation.

There are several metrics commonly used in the literature to evaluate semantic segmentation models such as Intersection over Union (IoU), Pixel Accuracy, F1 Score, Confusion matrix, Mean Class Accuracy, Mean Class Intersection over Union (mIoU). Class imbalance in the dataset (Fooladgar and Kasaei 2019):

- **Intersection over Union (IoU)** or Jaccard similarity is a widely used metric in semantic segmentation; it measures the ratio of the intersection of predicted and ground truth pixels (Ulku and Akagündüz 2022) to the union of predicted and ground truth pixels. To calculate the IoU in terms of true positives (TP), false positives (FP), and false negatives (FN), we can use the following formula:

$$IoU = \frac{TP}{TP + FP + FN} \quad (1)$$

- **Pixel accuracy** is another widely used metric; it measures the overall accuracy of the model, but it can be misleading if there is a class imbalance in the dataset. F1 score is the harmonic mean of precision and recall, and it's commonly used to

Table 1 RGB-D datasets with their characteristics

| Dataset | Images/scenes | Classes | Year | Application | Sensor type | Sensor name |
|---|--|---------|------|---------------------|-------------------------|---------------------------------------|
| NYUV1 Silberman and Fergus (2011) | 64 scenes, 108.6K frames, 2347 labeled frames | 40 | 2011 | Indoor | Structured light | Kinect V1 |
| SunCG Song et al. (2017) | 45.6K scenes | 40 | 2017 | Indoor | – | Synthetic |
| Starter Eftekhari et al. (2021) | 14.6M images, multiple scenes | 80 | 2021 | Indoor, in the wild | Structured light | Synthetic, Matterport Pro2, N/A |
| Unreal Mancini et al. (2018) | 21 sequences, 100k images | 20 | 2018 | Driving, outdoor | – | Synthetic |
| TartanAir Wang et al. (2020b) | 1037 scenes, over 1M frames | 20 | 2020 | Indoor, outdoor | – | Synthetic LiDAR |
| Dynamic Scene Shin Yoon et al. (2020) | 9 scenes | – | 2020 | Indoor, outdoor | SCS | Stereo camera |
| InteriorNet Li et al. (2018) | 20M images | – | 2018 | Indoor | – | Synthetic |
| Taskonomy Garcia-Hernando et al. (2018) | 4.5M scenes | 26 | 2018 | Indoor | Structured light | N/A |
| Structured 3D Zheng et al. (2020) | 3.5K scenes, 21.8K rooms, 196.5K frames | 13 | 2020 | Indoor | – | Synthetic |
| Gibson Xia et al. (2018) | 572 scenes, 1400 floor spaces from 572 buildings | – | 2018 | Indoor | LiDAR, structured light | NavVis, Matterport camera, DotProduct |
| TUM RGB-D Sturm et al. (2012) | 39 sequences | – | 2012 | Indoor | Structured light | Kinect |
| S3DIS Armeni et al. (2016) | 6.5K RGB-D images, 271 scenes | 13 | 2016 | Indoor | – | Microsoft Kinect V2 |
| SUN3D Xiao et al. (2013) | 415 Sequences | – | 2013 | Indoor | Structured light | Asus Xtion Pro Live |
| SceneNet RGB-D McCormac et al. (2017) | 15K trajectories (scenes), 5M images | – | 2017 | Indoor | – | Synthetic |
| Mid-Air Fonder and Van Droogenbroeck (2019) | 420k images | – | 2019 | Outdoor | – | Synthetic |

evaluate the performance of binary classification problems. The formula for Pixel Accuracy is:

$$Pixel\ Accuracy = \frac{\#of\ Correctly\ Classified\ Pixels}{Total\ Number\ of\ Pixels} \tag{2}$$

- **Mean Class Accuracy and Mean Class Intersection over Union (mIoU)** are metrics that are used when there is class imbalance in the dataset. These metrics take into account the accuracy and the number of false positives and false negatives

for each class. Specifically, Mean Class Accuracy is calculated as the average of the per-class accuracy values:

$$\text{Mean Class Accuracy} = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FP_i} \quad (3)$$

mIoU is calculated as the average of the per-class IoU values:

$$mIoU = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FP_i + FN_i} \quad (4)$$

where TP_i is the number of true positives for class i , FP_i is the number of false positives for class i , and N is the total number of classes.

The results presented in Tables 2 and 3 demonstrate significant advancements in RGB-D semantic segmentation achieved by recent methods. Notably, many of the approaches discussed in the paper focus on addressing the challenge of feature fusion to enhance segmentation performance. Based on the NYUDv2 data set, CMX b5 proposed in Liu et al. (2022) achieve better results in RGB-D semantic segmentation tasks which reached 56.9% MIOU. It is a model based on multi fusion modal structure, which proposes to improve the performance of semantic segmentation by exploiting informative features from supplementary modalities by fusing these features with RGB images. Based on the SUN RGB-D dataset, the state-of-the-art method is Tokenfusion proposed in Wang et al. (2022) which reached 53.0% MIOU. Incorporating depth information in RGB images has been shown to be effective in semantic segmentation tasks. Cross-modal fusion techniques, such as CMX (Liu et al. 2022), TokenFusion (Wang et al. 2022) have been developed to exploit informative features from supplementary modalities and improve the performance of semantic segmentation; However, fusing an arbitrary number of modalities remains under-explored.

4 Challenges and Research Directions

The purpose of automating the process of RGB-D semantic segmentation by using deep learning is to enable faster, more accurate, and more efficient analysis of visual data. This can be used to generate as-built BIM models from reality capture data, such as RGB-D images or point clouds. As-built BIM can provide accurate and comprehensive information about the geometry, topology, and semantics of the building elements, which can facilitate various applications such as progress monitoring, quality control, facility management, and renovation planning. However, RGB-D semantic segmentation still faces several challenges:

Table 2 Performance of RGB-D semantic segmentation algorithms on NYUv2

| Model | mAcc | Pix accuracy | mIoU | Year | References |
|-------------------------|------|--------------|------|------|--------------------------------------|
| CMX(B5) | – | 80.1 | 56.9 | 2022 | Liu et al. (2022) |
| TokenFusion | 66.9 | 79.0 | 54.2 | 2022 | Wang et al. (2022) |
| FSFNet | – | 77.9 | 52.0 | 2021 | Su et al. (2021) |
| PSD-ResNet50 | 58.6 | 77.0 | 51.0 | 2020 | Zhou et al. (2020a) |
| AMF | 65.3 | 77.5 | 52.5 | 2022 | Zhang et al. (2022) |
| DCANet | – | 78.2 | 53.3 | 2022 | Bai et al. (2022) |
| PDCNet | – | 78.4 | 53.5 | 2023 | Yang et al. (2023) |
| SA-Gate | – | 77.9 | 52.4 | 2020 | Chen et al. (2020) |
| CANet | 63.8 | 76.6 | 51.2 | 2020 | Zhou et al. (2020b) |
| ACNet | 63.1 | – | 48.3 | 2019 | Hu et al. (2019) |
| RedNet | – | – | – | 2018 | Jiang et al. (2018) |
| MMAF-Net-152 | 59.2 | 72.2 | 44.8 | 2019 | Hu et al. (2019) |
| Recurrent scene parsing | – | 72.1 | 44.5 | 2017 | Kong and Fowlkes (2018) |
| Depth-aware CNN | 56.3 | – | 43.9 | 2018 | Wang and Neumann (2018) |
| LSTM-CF | – | – | 49.4 | 2016 | Emre Yurdakul and Yemez (2017) |
| SegNet | – | – | – | – | Badrinarayanan et al. (2015) |
| LSD-GF | 60.7 | 71.9 | 45.9 | 2017 | Cheng et al. (2017) |
| FCN 2s RGB | 47.1 | 62.6 | 32.3 | 2017 | Amit Kamran and Shihab Sabbir (2017) |

- The domain gap between synthetic and real-world RGB-D images may affect the generalization ability of the models: when training an RGB-D semantic segmentation model using synthetic data, it may perform well on synthetic images but may not generalize well to real-world images due to the domain gap.
- The complexity and diversity of indoor scenes may contain occlusions, clutter, illumination variations, and geometric distortions.
- The trade-off between accuracy and efficiency of the models may limit their applicability in real-time scenarios.

Table 3 Performance of RGB-D semantic segmentation algorithms on SUN RGB-D

| Model | mAcc | Pix accuracy | mIoU | Year | References |
|-------------------------|------|--------------|------|------|--------------------------------|
| CMX(B5) | – | 83.8 | 52.4 | 2022 | Liu et al. (2022) |
| TokenFusion | 64.1 | 84.7 | 53.0 | 2022 | Wang et al. (2022) |
| FSFNet | – | 81.8 | 50.6 | 2021 | Su et al. (2021) |
| PSD-ResNet50 | 57.3 | 84.0 | 50.6 | 2020 | Zhou et al. (2020a) |
| AMF | 62.1 | 81.8 | 49.6 | 2022 | Zhang et al. (2022) |
| DCANet | – | 82.6 | 49.6 | 2022 | Bai et al. (2022) |
| PDCNet | – | 83.3 | 49.6 | 2023 | Yang et al. (2023) |
| SA-Gate | – | 82.5 | 49.4 | 2020 | Chen et al. (2020) |
| CANet | 60.5 | 82.5 | 49.3 | 2020 | Zhou et al. (2020b) |
| ACNet | 60.3 | – | 48.1 | 2019 | Hu et al. (2019) |
| RedNet | – | 81.3 | 47.8 | 2018 | Jiang et al. (2018) |
| MMAF-Net-152 | 58.2 | 81.0 | 47.0 | 2019 | Fooladgar and Kasaei (2019) |
| Recurrent scene parsing | – | 80.3 | 45.1 | 2017 | Kong and Fowlkes (2018) |
| Depth-aware CNN | 53.5 | – | 42.0 | 2018 | Wang and Neumann (2018) |
| LSTM-CF | – | – | 49.4 | 2016 | Emre Yurdakul and Yemez (2017) |
| SegNet | 44.8 | 72.6 | 31.8 | | Badrinarayanan et al. (2015) |
| LSD-GF | 60.7 | 71.9 | 45.9 | 2017 | Cheng et al. (2017) |

To address these challenges, some possible research directions are following:

- Developing novel data segmentation techniques or synthetic data generation methods which can increase the quantity and diversity of training data, leading to better results in RGB-D semantic segmentation.
- Exploring domain adaptation or domain generalization methods to reduce the domain gap and improve the robustness of the models.
- Exploiting the effectiveness of multimodal fusion techniques in fusing multi source of data such as depth and Lidar to design better models that adapts well to environment variations and real world scenarios.

- Incorporating prior knowledge or contextual information to improve the semantic consistency and completeness of the segmentation results. For example by incorporating a description of the environment with a neural network architecture.

5 Conclusion

This paper presents approaches and architectures used in RGB-D semantic segmentation with deep neural networks. It outlines the challenges faced and provides an overview of recent advancements in semantic segmentation using deep learning and DCNNs. The accuracy of different methods for RGB-D semantic segmentation has been compared on NYUDepth-V2 and SUN RGB-D datasets, and the advantages and disadvantages of these methods have been analyzed. In the future, further investigations on the challenges associated with RGB-D semantic segmentation will focus on proposing a comprehensive benchmark dataset for RGB-D semantic segmentation in indoor scenes, which can provide a standard evaluation platform for comparing different methods and proposing a novel as-built BIM generation framework, which can automatically generate accurate and complete BIM from RGB-D semantic segmentation results.

References

- Amit Kamran S, Shihab Sabbir A (2017) Efficient yet deep convolutional neural networks for semantic segmentation. arXiv e-prints, arXiv-1707
- Armeni I, Sax S, Zamir AR, Savarese S (2017) Joint 2D3D-semantic data for indoor scene understanding, pp 1–9, arXiv preprint [arXiv:1702.01105](https://arxiv.org/abs/1702.01105)
- Armeni I, Sener O, Zamir AR, Jiang H, Brilakis I, Fischer M, Savarese S (2016) 3d semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1534-1543
- Atif N, Balaji H, Mazhar S, Ahamad SR, Bhuyan MK (2022) Semantic masking: a novel technique to mitigate the class-imbalance problem in real-time semantic segmentation. In: 2022 National conference on communications (NCC). IEEE, pp 407–412
- Badrinarayanan V, Kendall A, Cipolla R (2015) Segnet: a deep convolutional encoder-decoder architecture for image segmentation. CoRR abs/1511.00561
- Bai L, Yang J, Tian C, Sun Y, Mao M, Xu Y, Xu W (2022) DCANet: differential convolution attention network for RGB-D semantic segmentation. arXiv preprint [arXiv:2210.06747](https://arxiv.org/abs/2210.06747)
- Barchid S, Mennesson J, Djéraba C (2021) Review on indoor RGB-D semantic segmentation with deep convolutional neural networks. In: 2021 International conference on content-based multimedia indexing (CBMI). IEEE, pp 1–4
- Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2014) Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint [arXiv:1412.7062](https://arxiv.org/abs/1412.7062)
- Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Patt Anal Mach Intell 40(4):834–848

- Chen G, Li C, Wei W, Jing W, Woźniak M, Blažauskas T, Damaševičius R (2019) Fully convolutional neural network with augmented atrous spatial pyramid pool and fully connected fusion path for high resolution remote sensing image segmentation. *Appl Sci* 9(9):1816
- Cheng Y, Cai R, Li Z, Zhao X, Huang, K (2017) Locality sensitive deconvolution networks with gated fusion for rgb-d indoor semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 3029-3037
- Chen X, Lin KY, Wang J, Wu W, Qian C, Li H, Zeng G (2020) Bi-directional cross-modality feature propagation with separation-and-aggregation gate for RGB-D semantic segmentation. In: *Proceedings of the computer vision-ECCV 2020: 16th European conference, Glasgow, UK, August 23-28, 2020, Part XI*. Cham: Springer International Publishing, pp 561–577
- Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B (2016) The cityscapes dataset for semantic urban scene understanding. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 3213–3223
- Dai, A., Chang, A.X., Sava, M., Halber, M., Funkhouser, T., Nießner, M., (2017). Scannet: Richly-annotated 3D reconstructions of indoor scenes. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 2432-2443
- de Oliveira Junior LA, Medeiros HR, Macêdo D, Zanchettin C, Oliveira AL, Ludermir T (2018) SegNetRes-CRF: a deep convolutional encoder-decoder architecture for semantic image segmentation. In: *2018 International joint conference on neural networks (IJCNN)*. IEEE, pp 1–6
- Deng L, Yang M, Li T, He Y, Wang C (2019) RFBNet: deep multimodal networks with residual fusion blocks for RGB-D semantic segmentation. *arXiv preprint arXiv:1907.00135*
- Eftekhari A, Sax A, Malik J, Zamir A (2021) Omnidata: a scalable pipeline for making multi-task mid-level vision datasets from 3D scans. In: *IEEE International conference on computer vision (ICCV)*, pp 10786–10796
- Emre Yurdakul E, Yemez Y (2017) Semantic segmentation of RGBD videos with recurrent fully convolutional neural networks. In: *Proceedings of the IEEE international conference on computer vision workshops*, pp 367–374
- Fonder M, Van Droogenbroeck M (2019) Mid-air: a multimodal dataset for extremely low altitude drone flights. In: *IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pp 553–562
- Fooladgar F, Kasaei S (2019) Multi-modal attention-based fusion model for semantic segmentation of RGB-depth images. *arXiv preprint arXiv:1912.11691*
- Garcia-Hernando G, Yuan S, Baek S, Kim TK (2018) First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In: *IEEE conference on computer vision and pattern recognition (CVPR)*, pp 409–419
- Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving the KITTI vision benchmark suite. *IEEE conference on computer vision and pattern recognition* 2012:3354–3361
- Geiger A, Lenz P, Stiller C, Urtasun R (2015) The KITTI vision benchmark suite. <http://www.cvlibs.net/datasets/kitti>, 2, 5
- Gupta S, Girshick R, Arbeláez P, Malik, J (2014) Learning rich features from RGB-D images for object detection and segmentation. In: *European conference on computer vision* Springer, Cham, pp 345–360
- He S, Yang H, Zhang X, Li X (2023) MFTransNet: a multi-modal fusion with CNN-transformer network for semantic segmentation of HSR remote sensing images. *Mathematics* 11(3):722
- He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 770–778
- Hu X, Yang K, Fei L, Wang K (2019) Acnet: attention based network to exploit complementary features for RGB-D semantic segmentation. In: *2019 IEEE international conference on image processing (ICIP)*. IEEE, pp 1440–1444
- Jiang J, Zheng L, Luo F, Zhang Z (2018) Rednet: residual encoder-decoder network for indoor rgb-d semantic segmentation. *arXiv preprint arXiv:1806.01054*
- Kong S, Fowlkes CC (2018) Recurrent scene parsing with perspective understanding in the loop. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 956–965

- Kugelman J, Alonso-Caneiro D, Read SA, Collins MJ (2022) A review of generative adversarial network applications in optical coherence tomography image analysis. *J Optometry*
- Liang, M., Hu, X. (2015). Recurrent convolutional neural network for object recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3367-3375)
- Li W, Saeedi S, McCormac J, Clark R, Tzoumanikas D, Ye Q, Huang Y, Tang R, Leutenegger S (2018) Interiornet: mega-scale multi-sensor photo-realistic indoor scenes dataset. In: British machine vision conference (BMVC), pp 1–13
- Liu Y, Yoshie O, Watanabe H (2022) Application of multimodal fusion attention mechanism in semantic segmentation. In: Proceedings of the Asian conference on computer vision, pp 1245–1264
- Liu S, Zhang J, Chen Y, Liu Y, Qin Z, Wan T (2019) Pixel level data augmentation for semantic image segmentation using generative adversarial networks. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 1902–1906
- Liu H, Zhang J, Yang K, Hu X, Stiefelhagen R (2022) CMX: cross-modal fusion for RGB-X semantic segmentation with transformers. arXiv preprint [arXiv:2203.04838](https://arxiv.org/abs/2203.04838)
- Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
- Mancini M, Costante G, Valigi P, Ciarfuglia TA (2018) J-mod 2: joint monocular obstacle detection and depth estimation. *IEEE Robot Autom Lett* 1490–1497
- McCormac J, Handa A, Leutenegger S, Davison AJ (2017) Scenenet RGB-D: can 5M synthetic images beat generic imagenet pretraining on indoor segmentation? In: IEEE international conference on computer vision (ICCV), pp 2697–2706
- Neuhold G, Ollmann T, Rota Bulò S, Kotschieder P (2017) The mapillary vistas dataset for semantic understanding of street scenes. In: Proceedings of the IEEE international conference on computer vision, pp 4990–4999
- Park SJ, Hong KS, Lee S (2017) Rdfnet: RGBD multilevel residual feature fusion for indoor semantic segmentation. In: Proceedings of the IEEE international conference on computer vision, pp 4980–4989
- Peng J, Estrada G, Pedersoli M, Desrosiers C (2020) Deep co-training for semi-supervised image segmentation. *Patt Recogn* 107:107269
- Schmidhuber J, Hochreiter S (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
- Seichter D, Köhler M, Lewandowski B, Wengefeld T, Gross HM (2021) Efficient RGB-D semantic segmentation for indoor scene analysis. In: 2021 IEEE international conference on robotics and automation (ICRA). IEEE, pp 13525–13531
- Shin Yoon J, Kim K, Gallo O, Park HS, Kautz J (2020) Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 5335–5344
- Silberman N, Fergus R (2011) Indoor scene segmentation using a structured light sensor. In: IEEE international conference on computer vision workshops (ICCV Workshops), pp 601–608
- Silberman N, Hoiem D, Kohli P, Fergus R (2012) Indoor conference on 3D vision (3DV). Segmentation and support inference from RGB-D images. In: European conference on computer vision (ECCV), pp 667–676, 746–760
- Song S, Lichtenberg SP, Xiao J (2015) Sun RGB-D: a RGB-D scene understanding benchmark suite. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 567–576
- Song S, Yu F, Zeng A, Chang AX, Savva M, Funkhouser T (2017) Semantic scene completion from a single depth image. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 1746–1754
- Sturm J, Engelhard N, Endres F, Burgard W, Cremers D (2012) A benchmark for the evaluation of RGB-D slam systems. In: IEEE/RSJ international conference on intelligent robots and systems, pp 573–580
- Sun C, Paluri M, Collobert R, Nevatia R, Bourdev L (2016). Pronet: learning to propose object-specific boxes for cascaded neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3485–3493

- Su W, Wang Z (2016) Regularized fully convolutional networks for RGB-D semantic segmentation. In: 2016 Visual communications and image processing (VCIP). IEEE, pp 1–4
- Su Y, Yuan Y, Jiang Z (2021) Deep feature selection and fusion for RGB-D semantic segmentation. In: 2021 IEEE international conference on multimedia and Expo (ICME). IEEE, pp 1–6
- Tang S, Zhang Y, Li Y, Yuan Z, Wang Y, Zhang X, Wang W (2019) Fast and automatic reconstruction of semantically rich 3D indoor maps from low quality RGB-D sequences. *Sensors* 19(3):533
- Ulku I, Akagündüz E (2022) A survey on deep learning-based architectures for semantic segmentation on 2d images. *Appl Art Intell* 36(1):2032924
- Valipour S, Siam M, Jagersand M, Ray N (2017) Recurrent fully convolutional networks for video segmentation. In: 2017 IEEE winter conference on applications of computer vision (WACV). IEEE, pp 29–36
- Wang J, Wang Z, Tao D, See S, Wang G (2016) Learning common and specific features for RGB-D semantic segmentation with deconvolutional networks. *European conference on computer vision*. Springer, Cham, pp 664–679
- Wang Y, Chen X, Cao L, Huang W, Sun F, Wang Y (2022) Multimodal token fusion for vision transformers. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp 12186–12195)
- Wang M, Hu L, Bai Y, Yao X, Hu J, Zhang S (2023). AMNet: a new RGB-D instance segmentation network based on attention and multi-modality. *Vis Comp* 1–15
- Wang W, Neumann U (2018) Depth-aware CNN for RGB-D segmentation. In: *Proceedings of the European conference on computer vision (ECCV)*, pp 135–150
- Wang W, Zhu D, Wang X, Hu Y, Qiu Y, Wang C, Hu Y, Kapoor A, Scherer S (2020b) TartanAir: a dataset to push the limits of visual slam. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp 4909–4916
- Xiao J, Owens A, Torralba A (2013) Sun3D: a database of big spaces reconstructed using SfM and object labels. In: *IEEE international conference on computer vision (ICCV)*, pp 1625–1632
- Xia F, Zamir AR, He Z, Sax A, Malik J, Savarese S (2018) Gibson env: realworld perception for embodied agents. In: *IEEE conference on computer vision and pattern recognition (CVPR)*, pp 9068–9079
- Yang J, Bai L, Sun Y, Tian C, Mao M, Wang G (2023) Pixel difference convolutional network for RGB-D semantic segmentation. *arXiv preprint [arXiv:2302.11951](https://arxiv.org/abs/2302.11951)*
- Yuan Z, Li Y, Tang S, Li M, Guo R, Wang W (2021) A survey on indoor 3D modeling and applications via RGB-D devices. *Front Inf Technol Electr Eng* 22(6):815–826
- Zhang C, Tang Y, Zhao C, Sun Q, Ye Z, Kurths J (2021) Multitask gans for semantic segmentation and depth completion with cycle consistency. *IEEE Trans Neur Netw Learn Syst* 32(12):5404–5415
- Zhang H, Sheng VS, Xi X, Cui Z, Rong H (2022) Overview of RGB-D semantic segmentation based on deep learning. *J Amb Intell Human Comput* 1–19
- Zhang Y, Yang Y, Xiong C, Sun G, Guo Y (2022) Attention-based dual supervised decoder for RGBD semantic segmentation. *arXiv preprint [arXiv:2201.01427](https://arxiv.org/abs/2201.01427)*
- Zheng J, Zhang J, Li J, Tang R, Gao S, Zhou Z (2020) Structured3D: a large photo-realistic dataset for structured 3D modeling. In: *European conference on computer vision (ECCV)*, pp 519–535
- Zhou F, Lai YK, Rosin PL, Zhang F, Hu Y (2022) Scale-aware network with modality-awareness for RGB-D indoor semantic segmentation. *Neurocomputing* 492:464–473
- Zhou L, Cui Z, Xu C, Zhang Z, Wang C, Zhang T, Yang J (2020) Pattern-structure diffusion for multi-task learning. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 4514–4523
- Zhou H, Qi L, Wan Z, Huang H, Yang X (2020) RGB-D co-attention network for semantic segmentation. In: *Proceedings of the Asian conference on computer vision*

A Framework for Generating IndoorGML Data from Omnidirectional Images



Misun Kim , Jeongwon Lee , and Jiyeong Lee 

Abstract Due to its efficiency and effectiveness, image data is widely used in many fields to express indoor space. Image data has the advantage of providing rich visual elements and having low construction cost. However, most spatial applications built on image data are limited to visualizing the indoor space because combining image with topology data is difficult. Topology data that expresses spatial relationships is essential to provide services such as routing and spatial queries in applications. To overcome those limitation, this study proposes the framework of generating topology data from image data. This paper discusses the methods of capturing image data from indoor space, detecting spatial entities and spatial relationships from omnidirectional images, and generating IndoorGML NRG (Node-Relation Graph) data. The methodologies proposed in this study can create topology data using only images without additional data and build topology data at a low cost. Using the suggested framework, we expect to be able to provide a variety of services for more indoor spaces.

Keywords IndoorGML · Indoor GIS · Omnidirectional image · Indoor topology

1 Introduction

Applications that model indoor space have mainly used 2D vector data such as floor plans, 3D BIM data, and point cloud data generated by scanning with lasers. However, these data have the disadvantages that pre-collected data must be required to model the indoor space, and it consumes much labor and cost to build. For these reasons,

This article was selected based on the results of a double-blind review of an extended abstract.

M. Kim (✉) · J. Lee · J. Lee
Department of Geoinformatics, University of Seoul, Seoul, South Korea
e-mail: kmisun@uos.ac.kr

J. Lee
e-mail: leejw576@uos.ac.kr

J. Lee
e-mail: jlee@uos.ac.kr

recently, images have been getting attention as indoor space modeling data. Image data has advantages in that it is easy to acquire, has low construction costs, and is easy to service because the data is light. However, most indoor image-based maps (e.g., Google Maps) do not have topology data, so they do not provide routing and spatial query services. Applications must use topology data together with images to provide indoor space LBS (location-based services) that include various functions based on these images. Nevertheless, since image data provide little information besides visual information about space, it is not easy to link with topology data.

In order to link image data and topology data, numerous research has presented methods of using external reference data such as floor plans and POI (point of interest) data. Jung (2016) established the relationship between images and topology data by converting data from floor plans into topology data. Ahn et al. (2020) expressed topological relationships through image data by linking the objects' image coordinates and the topological data nodes when the user clicks on the object in the image. However, these methods are limited because they are not applicable when external reference data is unavailable.

Studies have also presented methods of generating topological data directly from images without external data. Kim et al. (2016) and Claridades et al. (2018) detected door objects using an edge detection algorithm in image data and created topology data for the detected doors. These studies built the topology data for other spaces except for doors using floor plans in Kim et al. (2016) and shooting points in Claridades et al. (2018). However, these studies did not accurately detect objects in images, and there was a hassle of manually creating nodes after doors were detected. Kim and Lee (2023) presented a process of automatically detecting spatial objects required for constructing topology data using deep learning to overcome these limitations.

Claridades et al. (2023) conducted a study to embed topology information on image data and correspondingly extract network-based topology data from the embedded images to enable image-based indoor space analysis. This study defines a single omnidirectional image as a scene and connected the scenes represented by nodes using the concept of Linkpoint. This paper also presents methods for searching and saving POIs contained in the scene using the SEP (Spatial Extended Point) concept, which proposed by Ahn et al. (2020).

Previous studies have presented various methodologies of generating topology data, but they do not cover the entire process from image acquisition to generation of topological data. Thus, this study proposes an overall framework for generating topological data from image data and suggests methods for each step. Using this framework, we expect indoor LBS to be implemented in more regions because topology data can be easily built from indoor space information services modeled as images.

2 Methodologies

Figure 1 presents the overall framework presented in this study. First is the acquisition of image data. Second is the detection of spatial entities required for topology data generation, and the final step is creating network-based topology data that represents the relationship between spatial objects. This study uses IndoorGML (OGC 2023), which uses Node-Relation Graph (NRG) structure among topology models.

2.1 Image Data Capture

When modeling indoor space with images, omnidirectional images that capture all 360 degrees around from the shooting point are generally used. In this study, we also use omnidirectional images as image data.

Since each shooting point will be represented as a node, the shooting point is selected after sub-spacing the indoor space first. Sub-spacing divides the space (nodes) to express better the indoor space's geometrical characteristics (OGC 2023), which can increase navigation efficiency. Figure 2 shows why sub-spacing is necessary. In Original NRG(a), to go to room 103 or 104 from $n1$, you have to go to $n2$'s location in the center of the corridor, then turn around and move to $n3$ and $n5$. However, dividing nodes (spaces), as shown in (b), provides users with the shortest efficient route (move from $n1$ to $n2-1$ and then directly to $n3$ or $n5$).

As a sub-spacing rule, we use level 3—Room among the NRG levels of detail of the indoor space suggested by Claridades et al. (2022). In this level, the NRG is firstly divided into floors (level 1), semantically grouped zones within the same floor (level 2), and then divided into room units (Claridades et al. 2022). That is, the indoor space can be primarily divided according to the floor, and some spaces with connectivity relationships with various spaces, such as hallways and corridors, are divided into several spaces based on the room's entrance to provide more accurate route. In here, the room includes the general room and the elevator and stair space.

We basically capture one image per space type for each indoor subspace. However, according to the subspacing rule, in the case of a hallway or lobby, we divide the space based on the room's doorway (e.g., door, elevator door, emergency exit door)

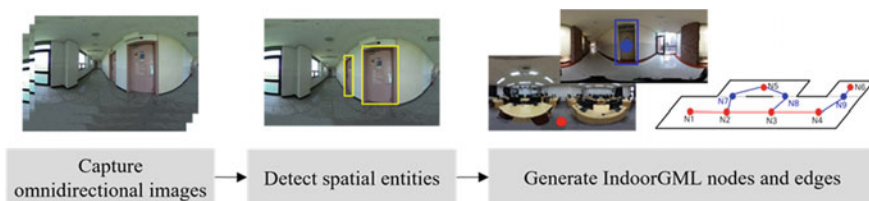


Fig. 1 The framework for generating topology data from image data

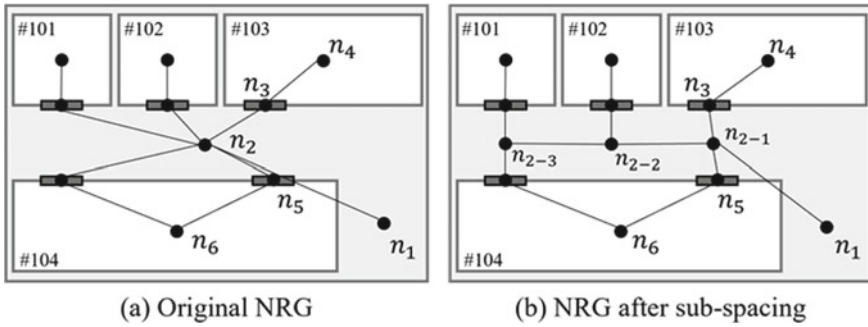


Fig. 2 Example of sub-spacing

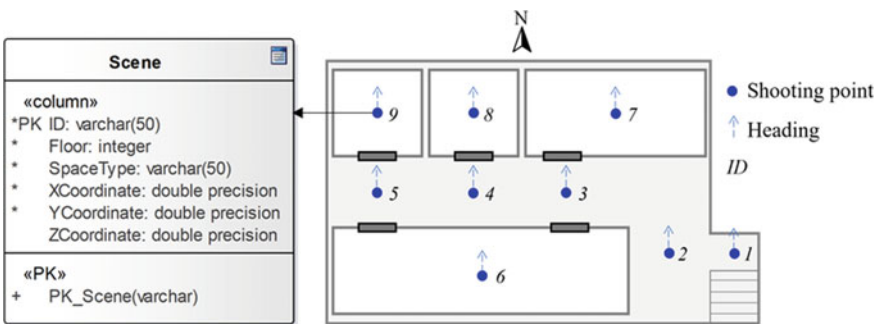


Fig. 3 Data structure of scene and example of ID assignment

and capture images in front of all doorways. In addition, we also capture the stair space so that the image data acquired on each floor can be connected.

In this study, referring to our previous research (Claridades et al. 2023), each omnidirectional image is defined as a *Scene*. All *Scenes* are filmed heading north. Also, when establishing a *Scene* as data, we include attribute information that includes floor number, ID, space type, and coordinates (Fig. 3). When allocating IDs, first group the space by type and assign sequentially according to the order of space. In SpaceType, we specify the space type, such as room, hallway, or stairs. Finally, the shooting point's x, y, and z coordinate values correspond to its geographic location based on a global or local coordinate system.

2.2 Detecting Spatial Entities

After establishing the Scene dataset, the next step is to detect spatial objects required for constructing IndoorGML data in each image. In the Navigation module of

IndoorGML, the TransitionSpace and ConnectionSpace classes (OGC 2023) represent the spaces required for indoor space exploration. TransitionSpace includes stair and corridor space, and ConnectionSpace includes door and vestibule space. In other words, stairs, corridors, doors, and vestibules should be identified in each omnidirectional image. Among these, corridors and vestibules have a strong concept of space than objects, and the shooting point can be used as a node representing the space. On the other hand, stairs and doors are spatial objects, so there is no need to create a corresponding shooting point, and these become the target objects for detection in the image.

This study uses a deep learning-based object detection algorithm to detect stairs and doors in images. Our previous study (Kim and Lee 2023) describes this process in detail. It presents a method for constructing a training dataset that can efficiently detect door and stair objects in an indoor omnidirectional dataset. We obtain a huge number of images, including doors and stairs, by refining an open dataset according to three conditions. First of all, we removed images in which external phenomena obstruct doors and stairs. Second, contain images that shows various shapes and materials of doors and stairs. However, we also remove images of stairs and doors which are not passable for people or used only in special situations (e.g., ladders and scaffolds). In order to use a more objective criterion, we checked the types of doors and stairs classified in Omniclass (The Construction Specifications Institute 2012) and re-select only such types of images as training data. Finally, we also include various door states in the data, such as images of closed, half-open, and wide-open doors. Furthermore, contain image data of various brightness.

In addition to the open dataset images, indoor omnidirectional images are included in the dataset to train the model. We performed this process because the images in the open dataset are not omnidirectional, but most of them are taken with a standard lens, so they do not reflect the characteristics of omnidirectional images with severe distortion. For this reason, we manually obtained omnidirectional images of doors and stairs with various colors and textures using a fisheye lens camera. Then we cropped the images with a redundancy of about 40% to prevent objects from being clipped and selected images that contain doors and stairs.

Our previous experiments used the YOLOv4 (You Only Look Once) model (Bochkovskiy et al. 2004). YOLOv4 has lower accuracy than models such as R-CNN (Region-based Convolutional Neuron Networks) and Faster R-CNN, but has the advantage of very fast detection speed. The image we use to build IndoorGML is an omnidirectional image, and since the omnidirectional image is composed of many pixels, the YOLO model is suitable. YOLOv4 uses CSPDarknet53 as the backbone, SPP (Spatial Pyramid Polling), and PAN (Path Augmented Network) as the neck and uses the previous YOLO model YOLOv3 as the head. In previous studies, we performed object detection using the basic architecture of the YOLOv4 model, almost as it is. For the hyper-parameters, the learning rate was 0.001, and the model used momentum as the optimizer (Fig. 4).

The detection results are saved after the spatial object detection is completed. If two or more objects are detected in one image, compare the objects' bounding box area value and store the widest bounding box's object. Here, if door objects on both

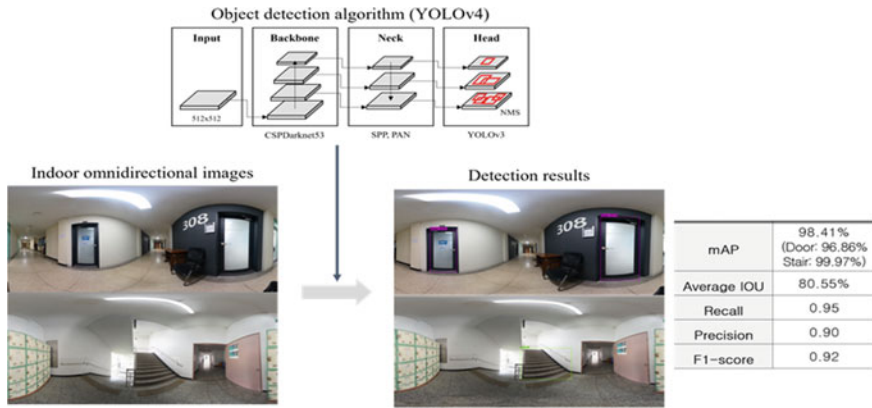
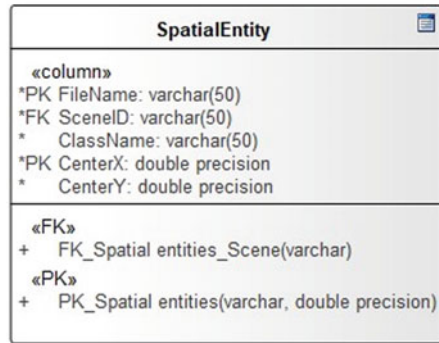


Fig. 4 Detecting stair and door in indoor omnidirectional images (Kim and Lee 2023)

Fig. 5 Data structure of detected spatial entities



walls are simultaneously detected, and the bounding box area is similar, store both objects respectively. Saving the results of object detection as Fig. 5. FileName means the file name of the image in which the spatial object is detected, and SceneID means the ID of the Scene that includes the spatial object. ClassName is the name of the detected spatial object (e.g., door or stair), and CenterX and CenterY indicate the location of the door or stair on the image coordinate system.

2.3 Generating IndoorGML Nodes and Edges

Build NRG by creating IndoorGML nodes and edges using Scene and SpatialEntity data. Figure 6 briefly shows the procedure for building the NRG.

First, create Scene nodes for each floor using the X, Y coordinates and floor number data in the Scene (Fig. 6a). Since IndoorGML does not define a data format,

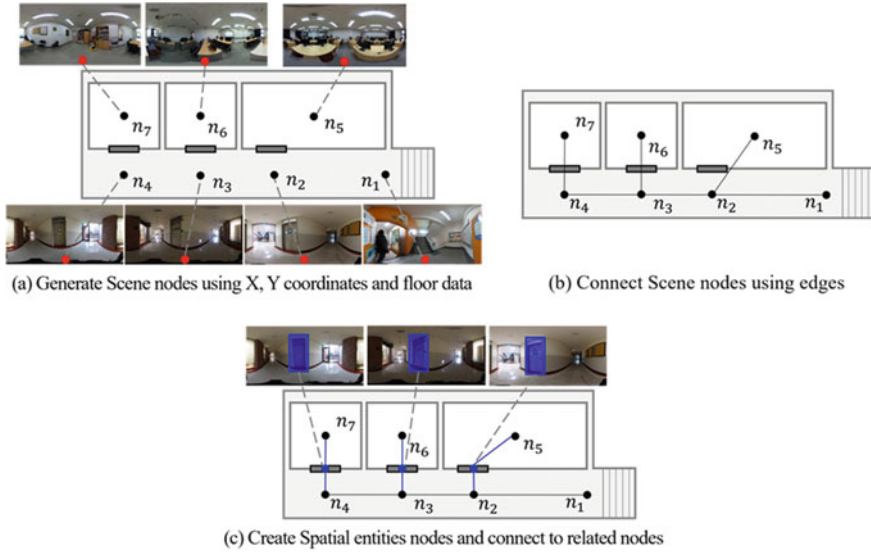


Fig. 6 A simple flow diagram to generate IndoorGML data

this study uses JSON (JavaScript Object Notation) format to store node data. Figure 7 shows the data structure of nodes. ID represents the unique value of each node, and SceneID, SpaceType, X, Y, and Floor are the same as those expressed in the Scene data. Connectivity saves the IDs of other nodes connected to the corresponding node.

Next, connect nodes with a connectivity relationship as edges (Fig. 6b). This study establishes the following rules for connecting nodes and creating edges for this step.

- Rule 1: Connect nodes whose SpaceType is a hallway to each other according to sequential ID.
- Rule 2: For nodes whose SpaceType is room, find and connect the nearest node whose SpaceType is lobby or hallway.
- Rule 3: In uncommon situations (for example, when two hallways are connected through a door or when hallways are located on either side of a room), connect the nodes manually.

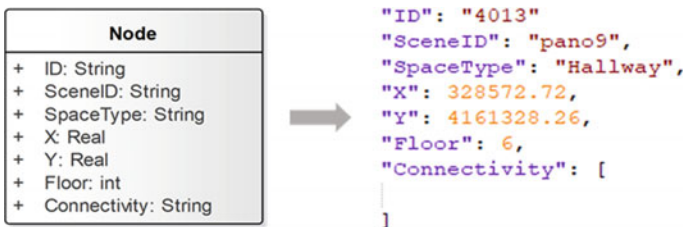


Fig. 7 Data structure of nodes and example of JSON format

When all Scene nodes are linked, create a ConnectionSpace node and TransitionSpace node using SpatialEntity data and connect them with existing Scene nodes (Fig. 6c). This process proceeds using the azimuth between connected Scene nodes and the horizontal angle of the detected object. The steps are as follows.

- Step 1: If the ClassName of *SpatialEntity* is a door, follow steps 2 to 4. If ClassName is a stair, follow step 5.
- Step 2: For each *Scene* node, calculate the nodes' azimuth to all connected nodes and save the nodes' set of azimuths. The formula for calculating the azimuth is:

$$azimuth = \frac{180}{\pi} \times \arctan\left(\frac{N_{i+1}^y - N_i^y}{N_{i+1}^x - N_i^x}\right) \quad (1)$$

N_i^y, N_i^x , coordinates of node i .

- Step 3: Calculate the horizontal angle from the image's center to the detected object's central point. The calculation is performed as follows:

$$horizontalangle = \frac{centerX}{p} - 180, \quad where \ p = \frac{ImageSize}{360} \quad (2)$$

centerX, X coordinates of *SpatialEntity*'s central point;

ImageSize, number of pixels in the horizontal direction.

This process is necessary because the position of the detected object is expressed in the image coordinates (pixel coordinates). By comparing the azimuth and horizontal angle, the proximity of the *Scene* node, which is expressed in global coordinates, and the *SpatialEntity* node, represented in image coordinates, is compared. This comparison is possible because all the omnidirectional images were taken with the north heading.

- Step 4: Compare the horizontal angle to the nodes' set of azimuths and determine the azimuth with the smallest difference between horizontal angles. Generate a new *SpatialEntity* node between *Scene* nodes having the smallest difference in azimuth and update connectivity data for each *Scene* and *SpatialEntity* node. In this study, we set the X and Y coordinates of the *SpatialEntity* node to the middle of two *Scene* nodes.
- Step 5: If the object is a stair, generate a new node between nodes with the smallest difference in X and Y coordinates but different Z values. In this case, the X and Y coordinates of the stair node are also set to the middle of the two *Scene* nodes.

3 Experimental Implementation

We conduct an experimental implementation on part of the 21st Century Building of the University of Seoul, South Korea. For the study area, omnidirectional images were taken according to sub-spacing rules explained in 2.1 (Fig. 8b). To get omnidirectional images, we used a 35mm DSLR camera with a fisheye lens and RICOH’s THETA camera.

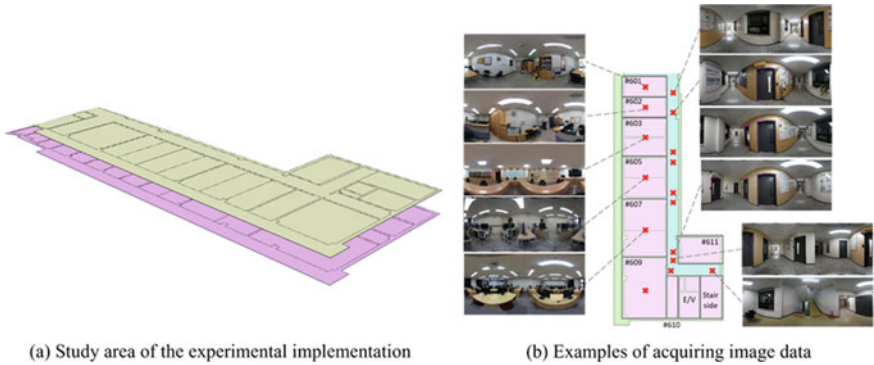


Fig. 8 Study area and example of captured image data

After each image was saved according to the Scene data structure, door and stair objects were detected using a deep learning-based object detection algorithm and stored as SpatialEntity data. We use Open Image Dataset and Kaggle for the open dataset, and we also add omnidirectional images into training data obtained from the Civil Engineering and Humanities buildings of the University of Seoul. The open dataset images used for training are 4,440, and the omnidirectional images are 1344. The trained model showed satisfactory performance with a mAP of 98.41%, average IOU of 80.55%, recall of 0.95, and precision of 0.90.

Using Scene data and SpatialEntity data, we create nodes and edges and save nodes data as JSON files (Fig. 9).

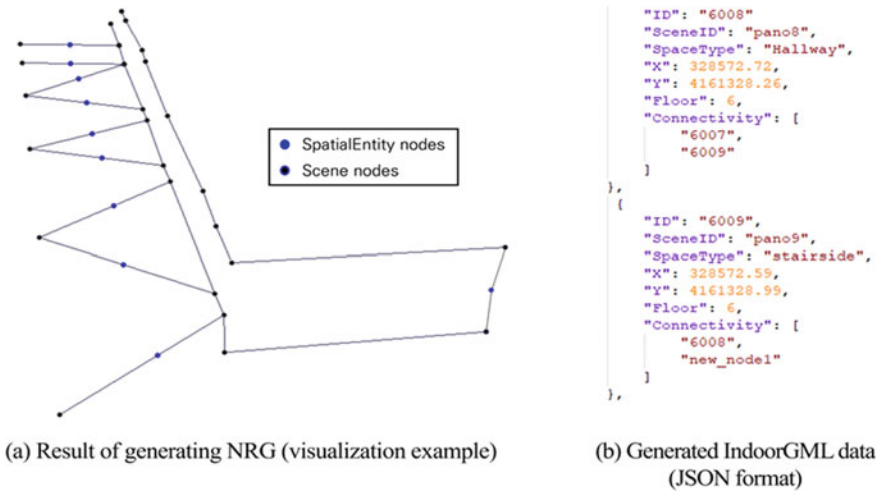


Fig. 9 Final results of experimental implementation

4 Conclusion

This study proposes an overall framework for generating topological data in NRG format from images. We present a method for acquiring omnidirectional images in indoor spaces and detecting door and stair objects necessary for generating IndoorGML data from omnidirectional images. In addition, we propose a method to create NRG by creating IndoorGML nodes from the acquired omnidirectional image data, detecting spatial entities data, and connecting those generated nodes.

This framework can be used for indoor LBS, providing services using images and topology data. Using the method presented in this study, images, and topology data can be built simultaneously at a low construction cost, providing various services for more indoor spaces. However, this study has a limitation in that it cannot deal with all types of indoor space, and the geographic location of the Spatial Entity node is not accurate. Developing a method for geo-referencing omnidirectional images in a global coordinate system may address this limitation. Furthermore, a method for automatically processing all framework procedures should be considered.

References

- Ahn D, Claridades A, Lee J (2020) Integrating image and network-based topological data through spatial data fusion for indoor location-based services. *J Sensors* 2020:1–12
- Bochkovskiy A, Wang C, Liao HM (2020) YOLOv4: optimal speed and accuracy of object detection [arXiv:2004.10934v4](https://arxiv.org/abs/2004.10934v4)
- Claridades ARC, Lee J, Blanco A (2018) Using omnidirectional images for semi-automatically generating IndoorGML data. *J Korean Soc Surv Geod Photogramm Cartogr* 36(5):319–333

- Claridades ARC, Choi H-S, Lee J (2022) An indoor space subsampling framework for implementing a 3D hierarchical network-based topological data model. *ISPRS Int J Geo Inf* 11(2):76
- Claridades ARC, Kim M, Lee J (2023) Developing a model to express spatial relationships on omnidirectional images for indoor space representation to provide location-based services. *ISPRS Int J Geo Inf* 12(3):101
- Jung H (2016) Development of indoor space applications data model based on integrating IndoorGML with 3D image. Master's thesis, University of Seoul, Korea
- Kim M, Lee J (2023) Developing a method to detect indoor spatial entities in omnidirectional images for constructing IndoorGML data. *J Korean Soc Surv Geod Photogramm Cartogr* 41(1):59–74
- Kim M, Kang H, Lee J (2016) A study on the implementation of indoor topology using image data. *J Korean Soc Surv Geod Photogramm Cartogr* 34(3):329–338
- OGC (Open Geospatial Consortium) (2023) IndoorGML v1.1. Available online: <https://docs.ogc.org/is/19-011r4/19-011r4.html>, last accessed 2023/7/17
- The Construction Specifications Institute (2012) OmniClass[®]—a strategy for classifying the built environment; Table 23-Products, National Standard

Deep Adaptive Network for WiFi-Based Indoor Localization



Afnan Ahmad and Gunho Sohn

Abstract A growing trend for indoor localization is relying on the existing WiFi signal strength. However, this method faces challenges due to WiFi's received signal strength (RSS) being susceptible to multipath, signal attenuation, and environmental variations, making it an unreliable measure of signal strength. To address this issue, a study was conducted which combined WiFi signals from various locations to create a localization system with high accuracy within a few meters. The study utilized WiFi propagation characteristics as a type of location fingerprinting, providing a new method for indoor localization using Wi-Fi RSSI fingerprinting. To adapt to new surroundings, the system employed a Variational Autoencoder to distribute WiFi signal properties, an LSTM network to analyze temporal relations of Wi-Fi signals, and a feature backpropagating refinement module to adjust neural network weights during inference. These tools were instrumental in achieving the system's main objective of domain adaptability. The accuracy of localization improved by approximately 18% compared to the baseline neural network.

Keywords WiFi · Indoor localization · VAE · LSTM · fBRS

1 Introduction

Location-based services (LBS) are becoming increasingly popular as mobile devices become standard in most people's daily lives. Consumers gain an advantage from LBS because it heightens their awareness of the nearby environment, but companies gain an advantage since it makes it easier to disseminate location-specific content. There are many different uses for indoor localization, some of which include gaming,

This article was selected based on the results of a double-blind review of the full paper.

A. Ahmad (✉) · G. Sohn
York University, Toronto, M3J 1P3, Canada
e-mail: aafnan@yorku.ca

G. Sohn
e-mail: gsohn@yorku.ca

search and rescue, asset monitoring, and autonomous robot navigation. Conventional positioning systems such as GNSS lose the capacity to precisely identify a device once it travels out of its satellite's direct Line-of-Sight (LOS) broadcast channel. Therefore, indoor localization is an essential component of the LBS system. This implies that a device that uses GNSS for position identification may lose the ability to establish its precise location if it enters a building, a parking garage, or travels under a bridge. These environments can interfere with the signals that GNSS sends and receives. Users need to have continual access to data that has a precision of at least one meter in order for location services to be provided in their entirety. A resolution of at least one meter is required of the data as well.

Wi-Fi signals serve as a prevalent method for indoor location tracking Liu et al. (2007). Given the ubiquity of Wi-Fi technologies today, they are suitably positioned to provide indoor location-based services. In the context of the 802.11 wireless local area network, the location of a user is commonly determined by measuring the frames transmitted from diverse access points (APs) through RSS fingerprinting-based localization techniques. Such Wi-Fi RSS measurement-based localization approaches encompass two stages. Initially, a fingerprint database is constructed to recognize each fingerprint. Each of these fingerprints comprises RSS data collected from access points by assessing the received signal strength at a specific reference location. Subsequently, in the localization stage, the user's position is estimated by comparing the RSS signal measurements from APs with the data in the fingerprint database via mathematical modeling methods Zegeye et al. (2016).

Both device-dependent and device-independent Wi-Fi-based localization systems are currently available. Device-dependent systems leverage the pre-calculated positions of WiFi Access Points (APs) to estimate a Wi-Fi enabled device carried by a user. Conversely, device-independent systems detect the physical presence of a user by passively monitoring Wi-Fi signal alterations caused by the human body. These techniques do not require any specialized equipment or additional Wi-Fi overhead. As a result, device-independent solutions offer more adaptability, mobility, and ease of installation. Users in different locations record Wi-Fi propagation attributes such as the Received Signal Strength Indicator (RSSI) Yang et al. (2012b) and Channel State Information (CSI) Yang et al. (2012b). By linking these propagation characteristics to location labels or coordinates, device-independent Wi-Fi location fingerprints can be formulated. The systems subsequently infer user locations in real-time by juxtaposing current propagation characteristics with those recorded in fingerprints.

Wi-Fi signals are influenced by a variety of environmental factors, such as signal variation, the multipath effect, interference, signal degradation over time, the effects of temperature and humidity on signal strength, and susceptibility to channel inconsistencies. These elements pose difficulties for neural networks in adapting to environmental shifts. To address the issue of consistency and provide reliable location information, we introduce research proposing a deep neural network capable of adjusting to changes without the need to retrain the network with new fingerprints.

We utilize the IPIN2016 datasets Ruiz et al. (2016) to evaluate the proposed method. Our contributions are summarized as follows:

1. We introduce a network that handles Wi-Fi signal (RSSI) data while learning about spatial and temporal relationships.
2. We suggest a backpropagation refinement function that allows networks to adapt to unexpected data or environmental shifts, thereby improving localization accuracy.
3. We conducted experiments on the IPIN2016 dataset to illustrate enhancements in location accuracy in comparison to standard algorithms.

2 Background

Indoor localization primarily employs two categories of technologies: computer vision techniques A comprehensive survey of indoor localization methods based on computer vision (2020) and wireless transmission methods Liu et al. (2020). Wireless transmission approaches leverage technologies such as ultrawideband, wireless local area networks, and radio frequency identification to locate a device. These technologies often necessitate the deployment and installation of physical infrastructures like Wi-Fi routers and Bluetooth beacons within the indoor environment Mautz (2012). Some methods, like the RSS-Based Lateration-based technique, offer excellent positioning accuracy Yang and Chen (2009).

Fingerprinting-based localization techniques typically follow a two-step process: (1) an offline training phase and (2) an online testing phase. The training phase involves the creation of a fingerprint database, where position-related survey data are gathered and preprocessed. Deep learning algorithms may be utilized in this phase to train fingerprints, extracting key features for improved localization.

Received Signal Strength (RSS) is an estimation of the power received by a client RF device from a router or access point. Weaker signals and greater distances result in lower overall data throughput and slower wireless data rates. The receive signal strength indicator (RSSI), which gauges the signal quality, generally indicates how well a particular radio can pick up signals from remote client radios EnGenius (2019).

Owing to its simplicity and minimal hardware prerequisites, many contemporary indoor localization systems use RSS as a fingerprint. For example, the Horus system uses a probabilistic approach for location determination based on RSS information Youssef and Agrawala (2005).

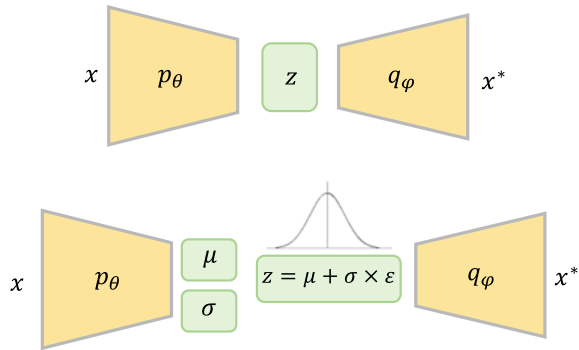
3 Proposed Method

3.1 Background

3.1.1 Variational Autoencoder (VAE)

VAE is a kind of generative probabilistic model renowned for learning embedding strategies capable of inferring the generative factors of most training data. This makes VAEs particularly effective for modeling Wi-Fi RSS signal propagation behavior for

Fig. 1 Network diagram for autoencoder and variational autoencoder (VAE)



localization tasks. Autoencoders were first introduced in the 1980s and later adopted by Hinton and Zemel (1993) in 1994, after which they have seen widespread use. The primary concept behind autoencoders is to encode the input vector into a condensed vector and subsequently reconstruct the original input from this coded representation. This process results in a compact and minimized vector that represents the data, aiding in dimension reduction and information retrieval. Autoencoders are a type of unsupervised neural network and comprise three stages: an Encoder denoted by p , the Bottleneck by z , and a Decoder by q . The encoder maps the input to a compressed vector z , and the decoder attempts to recreate it. The reconstruction loss is then described as:

$$Loss_{\theta\phi} = \frac{1}{n} \sum_{i=1}^{\infty} (x_i - q_{\phi}(p_{\theta}(x_i)))^2 \tag{1}$$

where the weights and biases of the network are denoted by θ and ϕ (Fig. 1).

Since the inception of autoencoders, various adaptations have been investigated. The VAE is a probabilistic version of autoencoders that was initially proposed by Kingma and colleagues in 2013. A distinct feature of VAE is the presence of a continuous vector in the bottleneck. Unlike other autoencoders that map their input to a discrete vector, VAE maps the input to a Gaussian distribution.

Given that the latent vector is not explicitly defined, the reconstruction loss in autoencoder transitions to a reconstruction likelihood. Based on Bayes' theorem, this loss is the expected negative log-likelihood of the data given the latent space distribution, which is an average of all possible values weighted by their respective probabilities:

$$Loss_E = -E[\log q_{\theta}(x|z)] = - \sum_{i=1}^{\infty} (x_i p(x_i)) \tag{2}$$

As previously noted, in VAE, there is an assumption that the latent space follows a Gaussian distribution $\mathcal{N}(0,1)$. This assumption must be taken into account in the loss, measuring how much data is lost due to this approximation. In other words, we need to regularize between the encoder distribution $p(z|x)$ and the latent space

distribution $p(z)$. As explained by Kingma in Eq. 3, the Kullback-Leibler divergence is utilized to measure the difference between these two terms as:

$$Loss_{KE} = KE(p(z), p(z|x)) \quad (3)$$

The final loss will be:

$$Loss = Loss_E + Loss_{KE} \quad (4)$$

3.1.2 Long Short-Term Memory (LSTM)

Recurrent Neural Networks (RNNs) are useful when we have data with sequences and we want to have the effect of the temporal dynamic behavior of the data in our network. In other words, we want to pass some information in form of a hidden layer from one instance to the next instance. As it is shown:

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \quad (5)$$

$$y_t = \sigma_y(W_y h_t) + b_y \quad (6)$$

The hidden layer is saved at each instance h_{t-1} . Then along with the weighted input $W_h x_t$ and current bias b_h , a weighted copy of the previous values of the hidden layer is passed to the activation function, σ_h . This value is then used for calculating the output vector y_t .

RNN networks are constrained in various ways. One of them is the long-term dependency problem, which results in disappearing gradients because the network is unable to keep track of long-term dependencies. Another issue might arise when it gives large weight by accident, a condition known as an explosive gradient. Long Short-Term Memory (LSTM) is introduced to address this problem. Hochreiter first invented it in 1991 for his thesis Eq. 3, and it was afterward improved upon and given in several forms.

To solve the aforementioned limitations addition to the one gate existing in the network, input i_t , output o_t , forget f_t and memory cell c_t gates were added and in this way, it can memorize long term and short term dependencies and pass it to the next instances:

$$f_t = \sigma_f(W_f x_t + U_f h_{t-1} + b_f) \quad (7)$$

$$i_t = \sigma_i(W_i x_t + U_i h_{t-1} + b_i) \quad (8)$$

$$o_t = \sigma_o(W_o x_t + U_o h_{t-1} + b_o) \quad (9)$$

$$\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \quad (10)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \quad (11)$$

$$h_t = o_t \cdot \sigma_h(c_t) \tag{12}$$

where W s and b s are weights and biases respectively.

3.2 Proposed Deep Adaptive Network

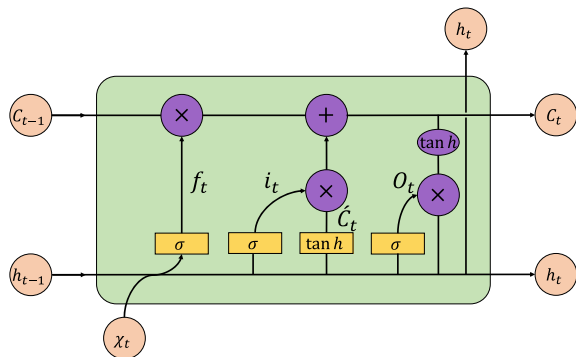
Figure 3 shows an overall architecture of the proposed deep adaptive network for Wi-Fi-based indoor positioning. In this section, we will explain the modules implemented for this novel deep-learning network.

3.2.1 Feature Extraction

This layer extracts the feature from labeled and unlabeled Wi-Fi fingerprint data. It shares an extracted feature with both the reconstruction path and the regression path.

Both the reconstruction and the reconstruction paths are built on top of the feature extraction layers. All labeled and unlabeled data will flow through these layers to produce their feature vectors. As shown in Fig. 3, we'll then examine augmented data XA to see how data travels via several layers. The flow of unlabeled data XU is the same in these layers. Figure 2 shows this data flow from left to right. An improved data point called $xA[t]$, XA is sent to the first linear layer, converting this 90-dimensional vector into a 360-dimensional vector. The second linear layer increases the 480-dimensional vector. The $zA[t]$ feature vector is output by the last linear layer, which increases the dimensions to 600. It normalizes each input in order to produce useful features. Between the first and second linear layers, as well as between the second and last linear layers, we add a sigmoid activation layer, a batch normalization layer, and a dropout layer (with a dropout rate of 0.3). The sigmoid layers try to introduce some nonlinearity, while the batch normalization layers try to speed up the training and the dropout layers strive to avoid overfitting.

Fig. 2 Network diagram for long short-term memory (LSTM)



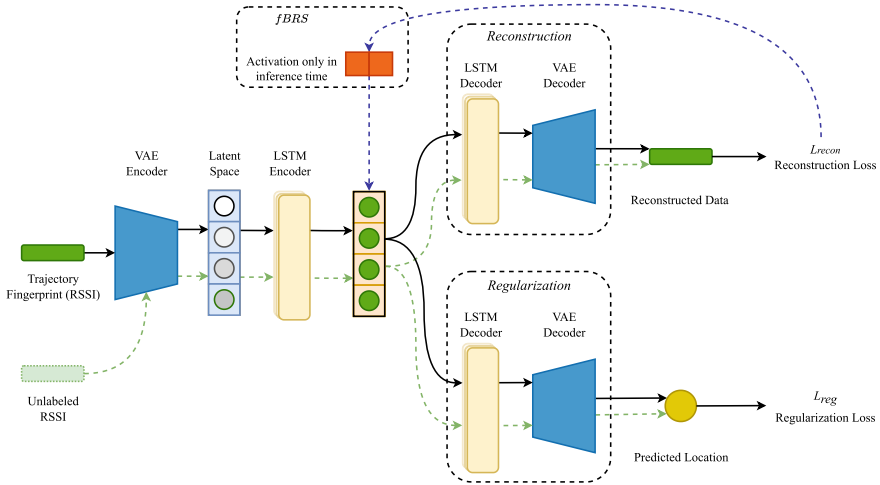


Fig. 3 Neural network architecture for proposed method

3.2.2 Reconstruction Decoder

The reconstruction decoder reconstructs the fingerprint and determines a reconstruction loss using extracted features for both labeled and unlabeled fingerprints.

The reconstruction path flows into the reconstruction layers after emerging from the feature extraction layers. The first and second linear layers, as illustrated in Fig. 3, shrink a 600-dimensional feature vector $zU[t]$ to a 480-dimensional vector and then to a 360-dimensional vector, respectively. The final linear layer attempts to build a 90-dimensional vector $xU[t]$ from this 360-dimensional vector that as closely as possible reproduces $xU[t]$. Equation (9) is used to calculate an aggregated reconstruction loss for all data with and/or without labels (or a batch of unlabelled and/or augmented data points).

3.2.3 Regularization Decoder

The extracted feature vectors are passed to the regularization decoder, which regresses and outputs the predicted location.

The regularization path flows into the regularization layers after the feature vectors are extracted. These layers are only open to feature vectors produced from augmented data, i.e., ZA , during the training phase. A feature vector $zA[t]$ goes through the regularization layers to generate a position prediction, as shown in Fig. 3.

The first and second linear layers, respectively, shrink a 600-dimensional feature vector $zA[t]$ to a 300-dimensional vector and subsequently a 100-dimensional vector. The third linear layer uses this 100-dimensional vector to forecast an N-dimensional one-hot encoded location probability vector $yA[t]$. We use Eq. 10 to calculate the regularization loss $lC[t]$ of one data point given the vector $yA[t]$.

3.2.4 Feature Backpropagating Refinement (fBRS)

This module simply does inference operations. After each iteration of reconstruction, it modifies the extracted feature weight that was learned during the training period to minimize reconstruction loss, which then aids the regularization decoder in reducing the impacts of signal propagation/loss and expanding its coverage.

For optimization purposes, we'd like to save time by only computing backpropagation for a subset of the network. Some parameters in the network's intermediaries can be optimized instead of the input. Maximizing the outputs of some of the last layers is a simple strategy that can be used to compute backpropagation only through the network's head. However, the desired result could not be achieved with such a basic approach. The final layer convolutions have a restricted receptive field compared to the network outputs.

Therefore, an optimization target can be easily achieved by changing just a few components of a feature tensor from loss generate during the reconstruction branch.

Let us reparameterize the function f and introduce auxiliary variables for optimization. Let $\hat{f}(x, z)$ denote the function that depends both on the input x and on the introduced variables z . With auxiliary parameters fixed $z = p$ there parameterized function is equivalent to the original on $\hat{f}(x, p) \equiv f(x)$. Thus, we aim to find a small value of Δp , which would bring the values of $\hat{f}(x, p + \Delta p)$ in the clicked points close to the user-provided labels. We formulate the optimization problem as follows:

$$\lambda \|\Delta p\|_2 + \sum_{i=1}^n \left(\hat{f}(x, p + \Delta p)_{u_i, v_i} - l_i \right)^2 \rightarrow \min_{\Delta p} \quad (13)$$

For the sake of minimization, we refer to this optimization job as f-BRS, which stands for feature backpropagating refinement, and we employ an unconstrained L-BFGS optimizer. We need to pick a reparameterization that a) does not have a localized effect on the outputs and b) does not require a backward pass through the entire network in order to optimize it in order for f-BRS to be effective.

Scaling and biasing the activations of the final layers of the network channel by channel is one of the possibilities for this kind of reparameterization, which is one of the options. When contrasted with optimization with regard to activations, optimization with respect to scale and bias cannot lead to degenerate solutions (that is, changes in reconstruction loss that are just slightly more concentrated in certain areas).

Let us denote the output of some intermediate layer of the network for an image x by $F(x)$, the number of its channels by h , a function that the network head implements by g . Thus, f can be represented by $f(x) \equiv g(F(x))$. Then the reparameterized function \hat{f} looks as follows:

$$\hat{f}(x, s, b) = g(s \cdot F(x) + b) \quad (14)$$

where $b \in R^h$ is a vector of biases, $s \in R^h$ is a vector of scaling coefficients and \cdot denotes a channel-wise multiplication. For $s = 1$ and $b = 0$ we have $\hat{f}(x) \equiv f$, thus we take these values as initial values for optimization.

A natural trade-off between accuracy and speed is achieved by adjusting the region of the network where auxiliary scale and bias are applied. Different optimization strategies are depicted in Fig. 2 alongside the network design we used in this work. We discovered that applying f-BRS to the final few layers results in considerable speed-up while only incurring a minor accuracy hit compared to full-network BRS.

3.2.5 Training

Our network works with a joint structure of reconstruction-regression for forming the aforementioned layers together, it utilizes a shared feature extraction layer that represents both labeled and unlabeled data which assists the regression layer to adapt to the unlabeled domain. This helps the network to readjust its weights for unseen scenarios or users. The feature back-propagating layer works at inference time and adjusts the features to minimize the reconstruction loss thus adapting and mitigating the environmental effect on Wi-Fi RSS signals.

4 Experiment

4.1 Datasets

We used IPIN Completion 2016 dataset. The dataset is collected inside a shopping mall. The collection location contains 168 access points (2.4G, 5G including Virtual Access Points). The training dataset includes 105 reference points.

4.1.1 Characteristics

There are a total of four different data sets inside IPIN2016. For training, we utilized the UAH, CAR, and UJIUB datasets; for evaluation, we used the UJITI dataset. These three data sets can be broken down into two distinct scenarios: those with floor switching (UAH, UJIUB, and UJITI) and those without (CAR). These four subsets of data were gathered in four distinct buildings in four distinct Spanish cities, each of which had a distinctive layout and group of neighbours. The number of Wi-Fi connections is lowest in the CAR data set and greatest in the UAH data set, respectively. The four sub-datasets are very different from one another.

4.1.2 Preprocessing

There are a total of 17 training log files included in the IPIN2016 datasets. These comprise four files for CAR, six files for UAH, five files for UJIUB, and two files for UJITI. Before being utilized for the purpose of training the network, these datasets are subjected to some preliminary processing and then blended together to produce the following files: CAR_train, UAH_train, UJIUB_train, and UJITI_train.

In this section, we provide an overview of the collection times for the four sub-datasets. On March 31, participants in both the UAH train and the UAH test will congregate. Within the UAH sub-data set, there is no gap in time between the training data and the test data at any point. The CAR train will be collected on March 8 and then again on April 14, in that order. The CAR test is gathered on July 21, approximately three months after the CAR train ends its collection period and before the CAR test begins its collection period. The UJIUB exam will be collected until the 23rd of May, while the UJIUB train will be collected until the 5th of April. The UJIUB train and the UJIUB test are separated by approximately one month from one another. While responses to the UJITI test can be submitted until July 19th, responses to the UJITI train can be submitted until April 25th. Between the UJTI train and the UJITI test, there is approximately a three-month gap in time.

4.2 Setup

In order to evaluate the computational difficulty of our method, we first determine the total amount of time required to run our proposed algorithm, which takes into account both the amount of time required to completely train on the UAH data set as well as the amount of time required to make a forecast for a single sample of the UAH data set. All algorithms can be run on a machine that has an Intel i9-10900KF central processing unit, 64 gigabytes of random access memory (RAM), and an RTX 3080 graphics processing unit (GPU). During the phase of feature extraction, our methods make use of encoders such as the Variational Auto Encoder (VAE) and the Long Short-Term Memory (LSTM). The same subnetwork that the decoder uses is used by this component so that it may obtain positioning data for use throughout the regularization and reconstruction stages. The network topology described in this article is comprised of six residual units, an LSTM, and a layer that is fully linked. This approach has the longest forecast time for a single sample, despite the fact that the delay is almost nonexistent at the millisecond level. In addition, the network is trained for real-world scenarios on a dedicated GPU, which shortens the amount of time needed for training and increases the network's inference performance.

4.3 Online Evaluation

For each sample in the test set, the algorithm produces a predicted location. The localization error is measured as the Euclidean distance (in meters) between the observed location and the projected location. We build a CDF of the localization error that occurred across all test samples, as shown in Tables 1, 2 and Fig. 4 in this section for various experimental settings.

4.4 Performance of Our Proposed Algorithm Under Different Parameters

4.4.1 Convergence of the Proposed Algorithm

The positioning algorithm will cease iterating thanks to the EarlyStopping mechanism that has been included in the network. This will occur when the validation data loss no longer lowers after 10 iterations of the algorithm. As a consequence of this, when the iteration process of the algorithm is complete, the result has converged.

Table 1 RMSE results of localization error

| Method | Percentage of estimated position (m) | | | | | | | | |
|---------------|--------------------------------------|------|------|------|------|------|------|------|------|
| | 25% | 50% | 75% | 80% | 85% | 90% | 95% | 99% | 100% |
| VAE | 0.51 | 0.91 | 1.26 | 1.39 | 1.54 | 1.67 | 1.91 | 2.26 | 2.87 |
| LSTM | 0.47 | 0.84 | 1.42 | 1.56 | 1.68 | 1.79 | 2.01 | 2.58 | 3.61 |
| VAE-LSTM | 0.42 | 0.82 | 1.16 | 1.42 | 1.59 | 1.72 | 1.98 | 2.19 | 2.3 |
| VAE-fBRS | 0.43 | 0.82 | 1.19 | 1.48 | 1.67 | 1.74 | 2.06 | 2.25 | 2.58 |
| LSTM-fBRS | 0.46 | 0.81 | 1.28 | 1.39 | 1.55 | 1.68 | 1.97 | 2.11 | 2.43 |
| VAE-LSTM-fBRS | 0.42 | 0.79 | 1.06 | 1.31 | 1.52 | 1.63 | 1.82 | 1.97 | 1.98 |

Table 2 MAE results of localization error

| Method | Mean error (percentile) | | |
|---------------|-------------------------|-------|-------|
| | 50% | 75% | 90% |
| VAE | 0.194 | 1.638 | 2.336 |
| LSTM | 0.192 | 2.154 | 3.159 |
| VAE-fBRS | 0.191 | 1.148 | 2.123 |
| VAE-LSTM | 0.231 | 0.431 | 2.196 |
| VAE-LSTM-fBRS | 0.183 | 0.318 | 1.829 |

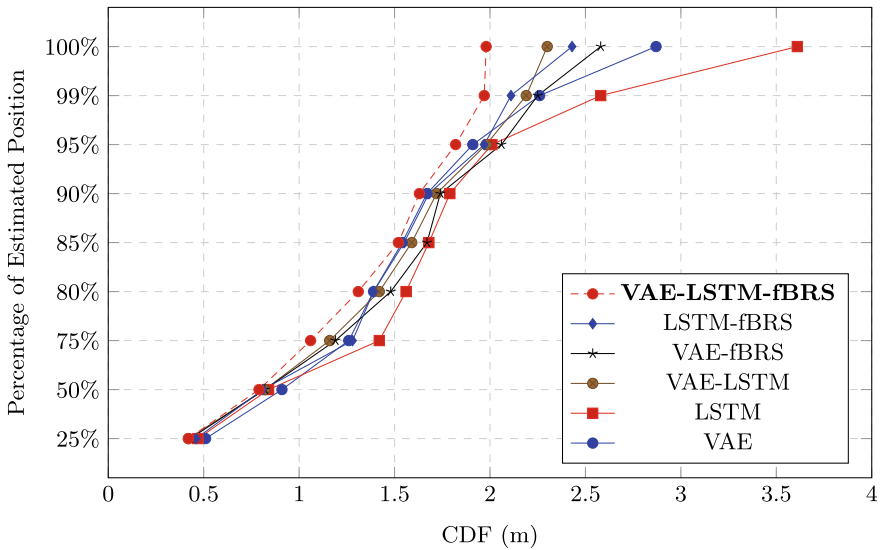


Fig. 4 Comparison of localization error between baselines and our proposed method

4.4.2 Determined-AI Hyper-parameter Optimization

In order to do hyperparameter optimization, a combination of the Early Stopping method implemented on the Determined AI stack and the utilization of the Python library that was provided to optimize the network were utilized.

4.5 Positioning Performance of Proposed Algorithm

Table 1 shows the results produced from experiments. We can see that our proposed method outperforms *LSTM* network in localization accuracy. Generally, we saw an improvement from 16 to 20%. Figure 4 shows plots of localization results, we can see that both *VAE - fBRS* and *VAE - LSTM* are performing better than *LSTM* and predicting closer to group truth location and especially around the corner where there are more doors and window (which affects Wi-Fi signal propagation) they are much smoother and finally *VAE - LSTM - fBRS* further improves the localization accuracy by having feature backpropagating refinement which is able to adapt to environmental changes.

5 Conclusion

In this research, we introduce a Deep Activation Optimization Technique for localization systems. This method delivers near one-meter localization accuracy by combining Variational AutoEncoder (VAE) and Long Short-Term Memory (LSTM) with a backpropagation refinement module. This integration allows the system to manage the inconsistency of WiFi fingerprint data across multiple users as well as varying times and environmental conditions. The use of VAE enables the training of networks with a range of signal values, while LSTM is employed to unearth the temporal correlations in Wi-Fi signal data over multiple time slices. These operations occur over several time intervals. This approach aids the network in striking an optimal balance between extracting domain-independent features from unlabeled data and localizing labeled data.

Acknowledgements The project, entitled “Intelligent Systems for Sustainable Urban Mobility (ISSUM)”, has been supported by the Ontario Research Fund (ORF) Research Excellence Program, the Natural Sciences and Engineering Research Council (NSERC), and Data Analytics and Visualization (DAV) CREATE Program.

References

- Ali AH, Razak MRA, Hidayab M, Azman SA, Jasmin MZM, Zainol MA (2010) Investigation of indoor WIFI radio signal propagation. In 2010 IEEE symposium on industrial electronics and applications (ISIEA)
- Bahl P, Padmanabhan V (2000) Radar: an in-building RF-based user location and tracking system vol 2, pp 775–784
- Bolliger P (2008) Redpin-adaptive, zero-configuration indoor localization through user collaboration. In: Proceedings of the first ACM international workshop on mobile entity localization and tracking in GPS-less environments, MELT '08, Association for Computing Machinery, New York, NY, USA, pp 55–60
- Brajdic A, Harle R (2013) Walk detection and step counting on unconstrained smartphones. In: Proceedings of the 2013 ACM international joint conference on pervasive and ubiquitous computing, UbiComp '13, Association for Computing Machinery, New York, NY, pp 225–234
- Chen X, Li H, Zhou C, Liu X, Wu D, Dudek G (2020) FiDo: ubiquitous fine-grained WiFi-based localization for unlabelled users via domain adaptation. Association for Computing Machinery, New York, NY, USA, pp 23–33
- Cheng J, Yang L, Li Y, Zhang W (2014) Seamless outdoor/indoor navigation with WiFi/GPS aided low cost inertial navigation system. *Phys Commun* 13:31–43
- Chen X, Ma C, Allegue M, Liu X (2017) Taming the inconsistency of Wi-Fi fingerprints for device-free passive indoor localization. In: IEEE INFOCOM 2017—IEEE conference on computer communications, pp 1–9
- Chen C, Zhao P, Lu CX, Wang W, Markham A, Trigoni N (2018) Oxiod: the dataset for deep inertial odometry
- Corral-Soto ER, Tal R, Wang L, Persad R, Chao L, Solomon C, Hou B, Sohn G, Elder JH (2012) 3d town: the automatic urban awareness project. In 2012 Ninth conference on computer and robot vision. IEEE, pp 433–440

- Dou F, Lu J, Xu T, Huang C-H, Bi J (2021) A bisection reinforcement learning approach to 3-d indoor localization. *IEEE Internet Things J* 8(8):6519–6535
- EnGenius (2019) What is RSSI and its acceptable signal strength? <https://helpcenter.ingeniustech.com/hc/en-us/articles/234761008-What-is-RSSI-and-its-acceptable-signal-strength>
- Fuller RXDK (2009) *Mobile entity localization and tracking in GPS-less environments*. Springer, Berlin Heidelberg
- Fuller R, Service SO (2009) *Mobile entity localization and tracking in GPS-less environments*. Springer, Berlin
- Gulo E, Sohn G, Afnan A (2018) Indoor positioning using WLAN fingerprint matching and path assessment with retroactive adjustment on mobile devices. *ISPRS Int Arch Photogr Rem Sens Spat Inf Sci XLII-4*:253–260
- Hinton GE, Zemel RS (1993) Autoencoders, minimum description length and helmholtz free energy. In: *Proceedings of the 6th international conference on neural information processing systems, NIPS'93*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 3–10
- Hoang MT, Yuen B, Dong X, Lu T, Westendorp R, Reddy K (2019) Recurrent neural networks for accurate RSSI indoor localization. *IEEE Internet of Things J* 6(6):10639–10651
- Hor A, Gunho S, Claudio P, Jadidi M, Afnan A (2018) A semantic graph database for BIM-GIS integrated information model for an intelligent urban mobility web application. *ISPRS Ann Photogram Rem Sens Spat Inf Sci* 4(4)
- Jimenez A, Seco F, Prieto C, Guevara J (2009) A comparison of pedestrian dead-reckoning algorithms using a low-cost mems IMU. In: *2009 IEEE international symposium on intelligent signal processing*, pp 37–42
- Klein G, Murray D (2007) Parallel tracking and mapping for small AR workspaces. In: *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE, pp 225–234
- Lin S, Clark R, Birke R, Schönborn S, Trigoni N, Roberts S (2020) Anomaly detection for time series using VAE-LSTM hybrid model. In: *ICASSP 2020—2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp 4322–4326
- Liu F, Liu J, Yin Y, Wang W, Hu D, Chen P, Niu Q (2020) Survey on WiFi-based indoor positioning techniques. *IET Commun* 14(9):1372–1383
- Liu H, Darabi H, Banerjee P, Liu J (2007) Survey of wireless indoor positioning techniques and systems. *IEEE Trans Syst Man Cybern Part C (Appl Rev)*, 37(6):1067–1080
- Mautz R (2012) *Indoor positioning technologies*. In: *Doctoral dissertation, Department for Civil, Environmental and Geomatic Engineering, Institute of Geodesy and Photogrammetry, ETH Zurich, Zurich, Switzerland*
- Mobile entity localization and tracking in GPS-less environments (2009) *mobile_entity_localization_and_tracking*
- Morar A, Moldoveanu A, Mocanu I, Moldoveanu F, Radoi IE, Asavei V, Butean A (2020) A comprehensive survey of indoor localization methods based on computer vision. *Sensors* 20:2641
- Niu J, Wang B, Cheng L, Rodrigues JJPC (2015) Wicloc: an indoor localization system based on wifi fingerprints and crowdsourcing. In: *2015 IEEE international conference on communications (ICC)*
- Ramos F, Trilles S, Torres-Sospedra J, Perales FJ (2018) New trends in using augmented reality apps for smart city contexts. *ISPRS Int J Geo-Inf* 7(12):478
- Ruiz ARJ, Mendoza-Silva GM, Montoliu R, Seco F, Torres-Sospedra J (2016) Datasets and supporting materials for the IPIN 2016 competition track 3 (smartphone-based, off-site)
- Ruohomäki T, Airaksinen E, Huuska P, Kesäniemi O, Martikka M, Suomisto J (2018) Smart city platform enabling digital twin. In: *2018 International conference on intelligent systems (IS)*. IEEE, pp 155–161
- Sofiuk K, Petrov IA, Barinova O, Konushin A (2020) f-BRS: rethinking backpropagating refinement for interactive segmentation. *CoRR*. [arXiv:2001.10331](https://arxiv.org/abs/2001.10331)
- Wang B, Chen Q, Yang LT, Chao H-C (2016) Indoor smartphone localization via fingerprint crowdsourcing: challenges and approaches. *IEEE Wireless Commun* 23(3):82–89

- Wang R, Luo H, Wang Q, Li Z, Zhao F, Huang J (2020) A spatial-temporal positioning algorithm using residual network and LSTM. *IEEE Trans Instrum Measure* 69(11):9251–9261
- Wikipedia (2019) Channel state information. <https://bit.ly/2WieoZA>
- Xiang Z, Song S, Chen J, Wang H, Huang J, Gao X (2004) A wireless LAN-based indoor positioning technology. *IBM J Res Develop* 48(5.6):617–626
- Yang J, Chen Y (2009) Indoor localization using improved RSS-based lateration methods. In: *GLOBECOM 2009 IEEE global telecommunications conference*, pp 1–6, Nov 2009
- Yang S, Claudio P, Ahmad A, Park P, Sohn G (2019) Development of a computer-vision based on real-time pedestrian comfort estimation system (poster). In: *Transportation association of Canada and ITS Canada 2019 joint conference and exhibition*, 2019
- Yang Z, Wu C, Liu Y (2012) Locating in fingerprint space. In: *Proceedings of the 18th annual international conference on Mobile computing and networking—Mobicom 12*
- Yang Z, Wu C, Liu Y (2012) Locating in fingerprint space: wireless indoor localization with little human intervention. In: *Proceedings of the 18th annual international conference on mobile computing and networking, Mobicom 12*, Association for Computing Machinery, New York, NY, USA, pp 269–280
- Yan H, Herath S, Furukawa Y (2019) Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, and new methods
- Yan H, Shan Q, Furukawa Y (2017) RIDI: Robust IMU double integration
- Youssef M, Agrawala A (2005) The horus WLAN location determination system. In: *Proceedings of the 3rd international conference on mobile systems, applications, and services, MobiSys '05*, ACM, New York, NY, USA, pp 205–218
- Zegeye WK, Amsalu SB, Astatke Y, Moazzami F (2016) WiFi RSS fingerprinting indoor localization for mobile devices. In: *2016 IEEE 7th annual ubiquitous computing, electronics and mobile communication conference (UEMCON)*
- Zhuang Y, Syed Z, Li Y, El-Sheimy N (2016) Evaluation of two WiFi positioning systems based on autonomous crowdsourcing of handheld devices for indoor navigation. *IEEE Trans Mob Comput* 15(8):1982–1995 Jan
- Zou H, Zhou Y, Jiang H, Huang B, Xie L, Spanos C (2017) Adaptive localization in dynamic indoor environments by transfer kernel learning. In: *2017 IEEE wireless communications and networking conference (WCNC)*, pp 1–6

MoLi-PoseGAN: Model-based Indoor Relocalization Using GAN and Deep Pose Regression from Synthetic LiDAR Scans



Hang Zhao, Martin Tomko, and Kourosh Khoshelham

Abstract Model-based LiDAR localization systems provide accurate pose estimation but they highly rely on the accuracy of 3D models. The inaccurate parts of 3D models will introduce localization errors. This paper presents a novel LiDAR relocalization method using synthetic LiDAR scans generated from a LiDAR generative adversarial network. Synthetic LiDAR scans are generated in a 3D model using the poses of a set of real LiDAR scans and to train a change detection network together with the corresponding real LiDAR scans to detect differences between the 3D models and the real environments. The synthetic and real data, and the differences are used in a generative adversarial network to correct the difference in synthetic LiDAR scans. A pose regression network is then trained using the corrected synthetic LiDAR scans and tested using new real LiDAR data. Experimental results show the proposed method achieves a higher accuracy than previous model-based pose regression methods.

Keywords Relocalization · Synthetic LiDAR scans · Generative adversarial network · 3D model · Change detection

1 Introduction

LiDAR sensors have been widely used for various applications such as building change detection and robot path planning and unknown place exploration Meyer et al. (2022) and Kada and McKinley (2009), and are extensively studied for LiDAR

This article was selected based on the results of a double-blind review of an extended abstract.

H. Zhao (✉) · M. Tomko · K. Khoshelham
Department of Infrastructure Engineering, The University of Melbourne, Parkville Victoria 3010, Australia
e-mail: zhaohz@student.unimelb.edu.au

M. Tomko
e-mail: tomkom@unimelb.edu.au

K. Khoshelham
e-mail: k.khoshelham@unimelb.edu.au

localization and navigation systems Shan and Englot (2018) and Zhao et al. (2020). In indoor environments, the low availability of the global navigation satellite system (GNSS) poses mitigating localization failure a challenge. Model-based LiDAR localization and navigation systems are developed to provide accurate LiDAR pose with an accurate 3D model Zhao et al. (2022) and Hendriks et al. (2021). These methods, however, highly rely on the accuracy of the 3D model and therefore, the inaccurate parts of the 3D model introduce localization errors or pose estimation failure.

Geometry-based LiDAR odometry approaches estimate the current LiDAR pose with respect to previous poses incrementally but these methods suffer from drifts Zhang and Singh (2014) and Wang et al. (2021a). Several learning-based relocalization methods have been proposed to perform real time LiDAR pose estimation accurately Wang et al. (2021b) and Komorowski et al. (2021), but these methods require a large real LiDAR dataset to train the network. Collecting such LiDAR dataset with ground truth poses is a challenge, requiring huge labour and time.

To overcome the above problems, this paper presents a novel LiDAR relocalization method using LiDAR generative adversarial network (GAN) model. A 3D model and a small number of real LiDAR sample data are used to generate sufficient synthetic LiDAR data with exact ground truth poses using a 3D model, the range values of which are sufficiently close to these real LiDAR data. An off-the-shelf change detection algorithm is used to perform change detection between the real LiDAR scan and the synthetic LiDAR scan Zhao et al. (2023). The real LiDAR scans, synthetic LiDAR scans with corresponding labels of changes, representing the differences between the real environment and the 3D model, are used to train a LiDAR GAN model to minimize the differences between the real LiDAR scan and the synthetic LiDAR scan. The output synthetic LiDAR scans, sufficiently close to real LiDAR scans acquired in real environment, are used to train a pose regression network. The trained network is then used to perform relocalization for real LiDAR data. The contributions of this paper are as follows:

- (1) Detecting changes between the 3D model and the real environment to improve the robustness and accuracy.
- (2) A LiDAR generative adversarial network is proposed to minimize the differences between the synthetic LiDAR data and real LiDAR data.
- (3) An experimental comparison of the proposed method with existing relocalization methods and vision-based methods that use synthetic images from a 3D model, demonstrating that the proposed method achieves higher accuracy than previous relocalization methods.

2 Related Work

Deep learning regression has been used for sensor pose estimation recently and achieves promising results Acharya et al. (2019b), Kendall and Cipolla (2017) and Yu et al. (2022). Shotton et al. (2013) proposed a deep regression method to estimate

the camera pose using RGB and depth values of image pixels. RGB and depth values of pixels were used to train a regression network and then the trained network is used to perform relocalization on images. Kendall and Cipolla (2017) analyzed the performance affected by the definition of loss function and how a beta value is used to balance the location loss and orientation loss. These camera-based methods use real images with known poses to train the deep regression network. However, the ground truth poses are hard to acquire Acharya et al. (2019a) and Acharya et al. (2019c). Therefore, Acharya proposed camera-based relocalization methods using synthetic images. Synthetic images were generated using a 3D model with exact ground truth poses and these synthetic images were used to train a deep regression network. The trained network was then used to perform relocalization on real images. BranchNet was proposed to estimate the location and orientation separately instead of learning location and orientation jointly Wu et al. (2017). CNN, LSTM and the random sample consensus (RANSAC) algorithm were combined in Walch et al. (2017) to improve the robustness to outliers. Regression trees and neural networks were combined to take advantage of deep learning and machine learning Dong et al. (2021) and dynamic objects in images were removed by a outlier rejection algorithm to improve accuracy and robustness.

Inspired by camera-based deep regression methods, LiDAR-based deep regression networks have been also proposed to perform relocalization on the LiDAR sensor. Zhao et al. (2020) proposed a LiDAR relocalization method using deep learning and synthetic LiDAR data. Synthetic LiDAR scans were generated from a 3D model with exact ground truth poses and the generated LiDAR data were then used to perform relocalization on real LiDAR scans. The limitation of this approach is the difference between real LiDAR scans and synthetic LiDAR scans. PointNet and PointNet++ were used as backbone architectures to perform LiDAR relocalization Wang et al. (2021b) and Yu et al. (2022). These methods achieve promising relocalization accuracy but the test environment is limited to a small environment and these methods require a large real LiDAR dataset to train the network.

To overcome the above problem, this paper presents a novel LiDAR relocalization method using GAN to minimize the differences between synthetic LiDAR scans and real LiDAR scans in Zhao et al. (2020). The proposed method requires only a small number of real LiDAR scans and a low-detail 3D model to generate sufficient synthetic LiDAR data with exact ground truth poses for the training purpose. The trained network is then used to perform relocalization on real LiDAR data.

3 Methodology

3.1 Framework of MoLi-PoseGAN

Figure 1 shows the framework of the proposed MoLi-PoseGAN. As Fig. 1 shows, the pose of a set of real LiDAR scans is estimated and used to generate synthetic LiDAR

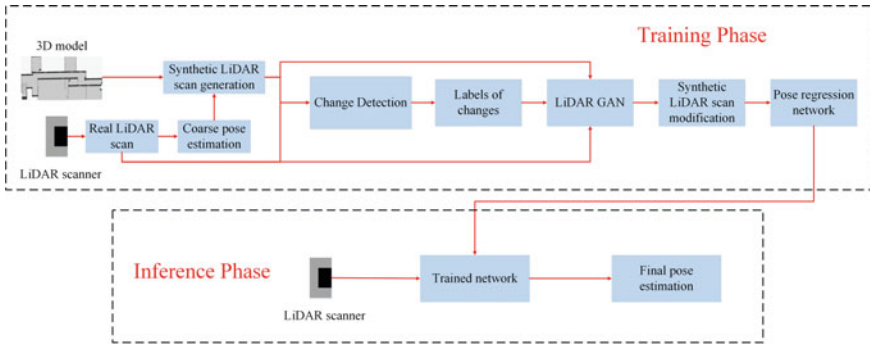


Fig. 1 Framework of MoLi-PoseGAN

scans. The real LiDAR scans and synthetic LiDAR scans are used to train an off-the-shelf change detection network to detect the differences between the synthetic LiDAR scan and the corresponding real LiDAR scan Zhao et al. (2023). The real LiDAR scan, synthetic LiDAR scan and corresponding labels of changes are used to train a LiDAR generative adversarial network to minimize the difference between the synthetic LiDAR scan and real LiDAR scan. The modified synthetic LiDAR scans with exact ground truth poses are used to train a regression network and the trained network is used to perform relocalization on real LiDAR scans.

3.2 Network of LiDARGAN

Figure 2 shows the structure of the LiDAR GAN network. As shown in Fig. 2, the synthetic LiDAR scans and the labels of changes are used to train an encoder and decoder to reconstruct a new synthetic LiDAR scan. The loss is defined as the difference between the real LiDAR scan and synthetic LiDAR scan and is used to optimize the weights in the encoder and decoder. The encoder and decoder are modified from a UNet-based network Ronneberger et al. (2015), to have two inputs for LiDAR scans and labels representing changes of each point respectively. LiDAR scans have the shape of a narrow strip, which requires changing the size of the input layer of the network to take LiDAR scans with a small number of rows and a large number of columns. The number of rows is the same as the number of laser channels of the LiDAR scanner (32 in our experiments). The number of columns depends on the horizontal angular resolution of the LiDAR scanner (2160 in our experiments) and therefore the input shape is set to $32 \times 2160 \times 3$. The input shape for the other input layer is set to $32 \times 2160 \times 1$. The output is the regenerated LiDAR scan which has the same shape as the input LiDAR scans ($32 \times 2160 \times 3$ in our experiments).

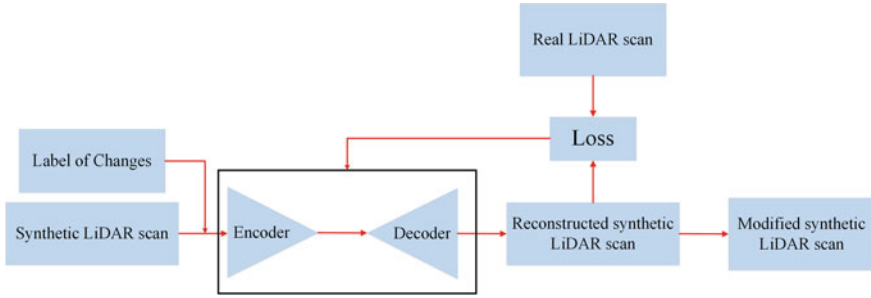


Fig. 2 The structure of the LiDARGAN network

3.3 Pose Regression Network

We applied the PointNet as the backbone architecture to perform pose regression Qi et al. (2017). We applied the main structure of the PointNet and introduced several modification to perform pose regression. To perform the regression task, the global average pooling layer is replaced by two dense layers. One dropout layer is added after each dense layers with 50% dropout rate. The weights of each convolutional layer are initialized by small random values drawn from a truncated normal distribution centered on zero (‘he_normal’ initializer).

3.4 Loss Function

3.4.1 LiDARGAN

The loss function of the LiDARGAN network is defined as the difference between the ground truth LiDAR scan and the predict LiDAR scan as follows:

$$loss_GAN = \frac{1}{n} \sum_{i=1}^n \sqrt{[(x - \hat{x})^2 + (y - \hat{y})^2 + (z - \hat{z})^2]} \quad (1)$$

Where n represents the number of points in each LiDAR scan, and (x,y,z) and (x̂,ŷ,ẑ) represent ground truth coordinates and predict coordinates respectively.

3.4.2 Pose Regression

To perform pose regression, quaternion is used to describe orientation and the location is described as (x, y, z). Therefore, the pose of the LiDAR scanner is described using a

7-element vector. In this paper, the location and orientation are learnt jointly Kendall and Cipolla (2017) and Walch et al. (2017). Accordingly, the loss function is defined as follows:

$$L = \|x_{pre} - x_{tru}\| + \beta \times |q_{pre} - q_{tru}| \quad (2)$$

where

- L = loss value;
- x_{pre} = predicted location;
- x_{tru} = ground truth location;
- q_{pre} = predict quaternion;
- q_{tru} = ground quaternion;
- β = a constant;
- $\|\cdot\|$ = norm;
- $|\cdot|$ = absolute.

In the equation above, the β is used to balance the location loss and orientation loss since they have different ranges of values. For example, in our experiment environment, the range of location can range from 0 to 40 while the range of a unit quaternion is between 0 and 1.

4 Experimental results

4.1 LiDAR Dataset

We selected 40 locations uniformly in a University corridor environment to acquire a total of 40 real LiDAR scans using a 32-channel Velodyne LiDAR and the location of these 40 real LiDAR scans are estimated using the 3D model Zhao et al. (2022). The estimated poses were then used to generate 7200 synthetic LiDAR scans using a 3D model. At each location of a real LiDAR scan, 180 synthetic LiDAR scans were generated as shown in Fig 3. As shown in Fig 3, the left part shows how synthetic LiDAR scans were generated and the right part shows a zoom in part of the entire environment. 180 synthetic scans were generated with respect to the pose of that real LiDAR scan. The blue stars represent the location of real LiDAR scans while the start point of each red arrow represents the location of each synthetic LiDAR scan with the direction of each arrow demonstrating the orientation of each synthetic scan. The distance between the location of synthetic LiDAR scans and real LiDAR scans ranges from 0.05 meters to 0.25 meters with 0.05-meter increments, at a 10-degree interval from 0 to 360 degrees with each distance.

The total of 7200 pairs of synthetic LiDAR scans and real LiDAR scans are the input of a pre-trained change detection model to perform change detection Zhao et al. (2023) and train the LiDARGAN network with predict labels and pose regression model.

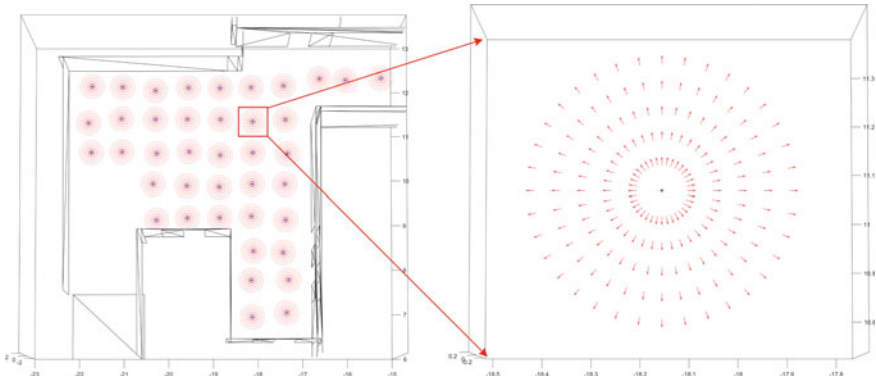


Fig. 3 Generation of synthetic LiDAR scans

4.2 LiDARGAN

The Synthetic LiDAR scans and real LiDAR scans with predict labels are the input of the LiDARGAN model to minimize the differences between synthetic LiDAR scans and real LiDAR scans. Figure 4 shows and example of the output synthetic LiDAR scans from the LiDARGAN model. As shown in Fig. 4, the differences between the synthetic LiDAR scans and real LiDAR scans such as the pedestrian, furniture and a new room caused by an open door were generated in the output synthetic LiDAR scan.

4.3 Accuracy and Comparison

We used the 7200 output synthetic LiDAR scans from the LiDARGAN model to train a PointNet-based regression network Qi et al. (2017) and the trained network then performed relocalization on 200 real LiDAR scans. Table 1 compares the relocalization accuracy between MoLi-PoseGAN and previous LiDAR-based relocalization methods. The LiDAR PoseNet, DH3D, EgoNN are trained with original synthetic LiDAR scans without using LiDAR GAN to modify the differences. Compared with previous LiDAR relocalization networks, the proposed MoLi-PoseGAN achieves the best accuracy, in terms of location and orientation.

We compared MoLi-PoseGAN with vision-based relocation methods, which trained the pose regression networks using synthetic images. Table 2 compares the median error, mean error and mean absolute deviation by implementing MoLi-PoseGAN and vision-based approaches. As shown in Table 2, MoLi-PoseGAN achieves higher relocalization accuracy than existing vision-based approaches.

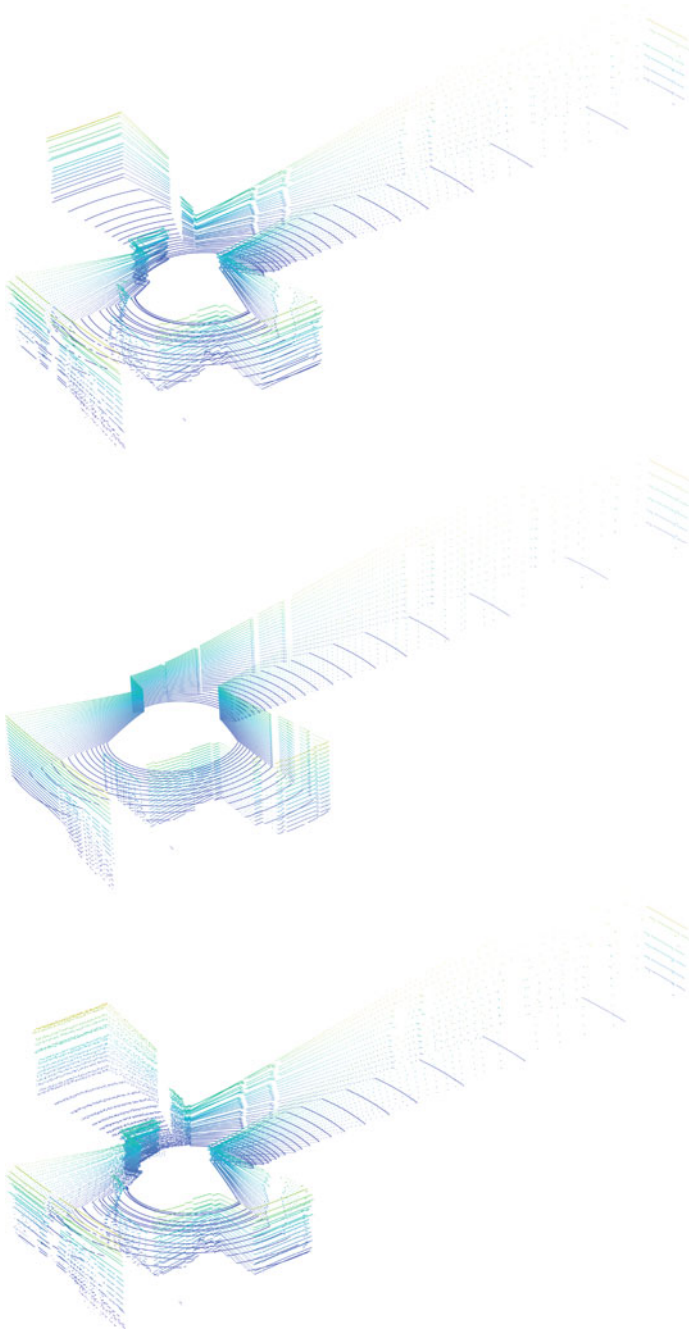


Fig. 4 An example of an output synthetic LiDAR scan (top) input real LiDAR scan with pedestrians, furniture and an open door (middle) synthetic LiDAR scan generated in the 3D model (bottom) output synthetic LiDAR scan with generated objects in the point cloud

Table 1 The median, mean and mean absolute deviation of location error and orientation error (m, deg) of MoLi-PoseGAN and three open-source LiDAR-based networks

| Method | Median error | Mean error | Mean absolute deviation |
|----------------------------------|---------------------|---------------------|-------------------------|
| LiDAR PoseNet Zhao et al. (2020) | (0.67, 4.67) | (0.91, 7.96) | (0.51, 5.98) |
| DH3D Du et al. (2020) | (0.71, 6.81) | (1.01, 8.33) | (0.43, 5.44) |
| EgonNN Komorowski et al. (2021) | (0.64, 6.12) | (0.88, 7.43) | (0.53, 4.14) |
| MoLi-PoseGAN | (0.32, 3.15) | (0.54, 4.38) | (0.46, 3.69) |

Table 2 The median, mean and mean absolute deviation of location error and orientation error (m, deg) of MoLi-PoseGAN and four open-source camera-based relocalization methods

| Method | Median error | Mean error | Mean absolute deviation |
|--|---------------------|---------------------|-------------------------|
| BIM-PoseNet Acharya et al. (2019a) | (1.88, 7.73) | (2.42, 12.51) | (1.53, 8.98) |
| Recurrent BIM-PoseNet Acharya et al. (2019b) | (1.87, 11.15) | (2.18, 12.33) | (1.51, 7.74) |
| Bayesian BIM-PoseNet Acharya et al. (2019c) | (1.98, 7.33) | (3.21, 14.19) | (2.00, 5.37) |
| Segmentation BIM-PoseNet Acharya et al. (2022) | (1.19, 6.25) | (1.88, 6.97) | (1.35, 2.72) |
| MoLi-PoseGAN | (0.32, 3.15) | (0.54, 4.38) | (0.46, 3.69) |

5 Discussion

This paper presents a novel LiDAR relocalization method using a 3D model, which requires a small number of real LiDAR scans. Sufficient synthetic LiDAR scans with exact ground truth poses can be generated to train a regression network. LiDARGAN has been used to minimize the differences between real LiDAR scans and synthetic LiDAR scans. Using the synthetic LiDAR scans generated from LiDARGAN to train the pose regression networks achieves higher accuracy than using original synthetic LiDAR scans.

However, the accuracy of the MoLi-PoseGAN relies on the accuracy of the change detection approach. A lower change detection accuracy will result in more differences between synthetic LiDAR scans and real LiDAR scans. The inaccurate parts will finally introduce relocalization error to MoLi-PoseGAN.

6 Conclusion

This paper presents a novel LiDAR relocalization method using a 3D model and the generative adversarial network. Synthetic LiDAR scans are generated using the 3D

model and poses of a small set of real LiDAR scans. The pairs of synthetic LiDAR scans and real LiDAR scans are the input of a change detection network to perform change detection. The pairs of synthetic LiDAR scans and real LiDAR scans with predict labels are the input of a generative adversarial network to minimize the difference between synthetic LiDAR scans with corresponding real LiDAR scans. The generated synthetic LiDAR scans with ground truth poses are used to train a pose regression networks. The trained regression network is finally used to predict poses of real LiDAR scans. Experimental results show that the proposed MoLi-PoseGAN outperforms previous LiDAR LiDAR regression methods and vision-based relocation methods.

References

- Acharya D, Khoshelham K, Winter S (2019) BIM-PoseNet: Indoor camera localisation using a 3d indoor model and deep learning from synthetic images. *ISPRS J Photogram Rem Sens* 150:245–258
- Acharya D, Ramezani M, Khoshelham K, Winter S (2019) BIM-tracker: a model-based visual tracking approach for indoor localisation using a 3d building model. *ISPRS J Photogram Rem Sens* 150:157–171
- Acharya D, Roy SS, Khoshelham K, Winter S (2019) Modelling uncertainty of single image indoor localisation using a 3d model and deep learning. *ISPRS Ann Photogram Rem Sens Spatial Inf Sci* 4:247–254
- Acharya D, Tennakoon R, Muthu S, Khoshelham K, Hoseinnezhad R, Bab-Hadiashar A (2022) Single-image localisation using 3d models: combining hierarchical edge maps and semantic segmentation for domain adaptation. *Automa Construct* 136:104, 152
- Dong S, Fan Q, Wang H, Shi J, Yi L, Funkhouser T, Chen B, Guibas LJ (2021) Robust neural routing through space partitions for camera relocation in dynamic indoor environments. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 8544–8554
- Du J, Wang R, Cremers D (2020) Dh3d: deep hierarchical 3d descriptors for robust large-scale 6dof relocation. In: *Proceedings of the computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, Part IV* 16, pp. 744–762. Springer (2020)
- Hendrikx R, Pauwels P, Torta E, Bruyninckx HP, van de Molengraft M (2021) Connecting semantic building information models and robotics: an application to 2d lidar-based localization. In: *2021 IEEE international conference on robotics and automation (ICRA)*, pp 11,654–11,660. IEEE
- Kada M, McKinley L (2009) 3d building reconstruction from lidar based on a cell decomposition approach. *Int Arch Photogram Rem Sens Spat Inf Sci* 38(Part 3):W4
- Kendall A, Cipolla R (2017) Geometric loss functions for camera pose regression with deep learning. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 5974–5983
- Komorowski J, Wysoczanska M, Trzcinski T (2021) Egonn: egocentric neural network for point cloud based 6dof relocation at the city scale. *IEEE Robot Autom Lett* 7(2):722–729
- Meyer T, Brunn A, Stilla U (2022) Change detection for indoor construction progress monitoring based on BIM, point clouds and uncertainties. *Autom Construct* 141:104, 442
- Qi CR, Su H, Mo K, Guibas LJ (2017) Pointnet: deep learning on point sets for 3d classification and segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 652–660
- Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: *Proceedings of the 18th international conference on medical image computing*

- and computer-assisted intervention—MICCAI 2015, Munich, Germany, October 5-9, 2015, Part III 18. Springer, pp 234–241
- Shan T, Englot B (2018) Lego-loam: lightweight and ground-optimized lidar odometry and mapping on variable terrain. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 4758–4765
- Shotton J, Glocker B, Zach C, Izadi S, Criminisi A, Fitzgibbon A (2013) Scene coordinate regression forests for camera relocalization in RGB-D images. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2930–2937
- Walch F, Hazirbas C, Leal-Taixe L, Sattler T, Hilsenbeck S, Cremers D (2017) Image-based localization using lstms for structured feature correlation. In: Proceedings of the IEEE international conference on computer vision, pp 627–637
- Wang H, Wang C, Chen CL, Xie L (2021) F-loam: fast lidar odometry and mapping. In: 2021 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 4390–4396
- Wang W, Wang B, Zhao P, Chen C, Clark R, Yang B, Markham A, Trigoni N (2021) Pointloc: deep pose regressor for lidar point cloud localization. *IEEE Sens J* 22(1):959–968
- Wu J, Ma L, Hu X (2017) Delving deeper into convolutional neural networks for camera relocalization. In: 2017 IEEE international conference on robotics and automation (ICRA). IEEE, pp 5644–5651
- Yu S, Wang C, Wen C, Cheng M, Liu M, Zhang Z, Li X (2022) Lidar-based localization using universal encoding and memory-aware regression. *Patt Recogn* 128:108, 685
- Zhang J, Singh S (2014) Loam: Lidar odometry and mapping in real-time. In: *Robotics: science and systems*, vol 2. Berkeley, CA, pp 1–9
- Zhao H, Acharya D, Tomko M, Khoshelham K (2020) Indoor lidar relocalization based on deep learning using a 3d model. *Rem Sens Spat Inf Sci Int Arch Photogram*
- Zhao H, Tomko M, Khoshelham K (2023) Interior structural change detection using a 3d model and lidar segmentation. *J Build Eng* 106628
- Zhao Y, Zhao H, Radanovic M, Khoshelham K (2022) A unified framework for automated registration of point clouds, mesh surfaces and 3d models by using planar surfaces. *Photogram Rec* 37(180):366–384

Digital Twins: Simulating Robot-Human Sidewalk Interactions



Ali Hassan, Muhammad Usman, Melissa Kremer, Seungho Yang,
Michael Luubert, Petros Faloutsos, G. Brent Hall, and Gunho Sohn

Abstract This paper investigates the interactions between sidewalk delivery robots and pedestrians in urban settings to evaluate pedestrian safety and robot efficiency in shared spaces. We developed a 3D digital twin environment model that allows realistic simulation of robot-human and robot-cityscape interactions, adopting the Pedestrian Aware Model (PAM) to ensure effective and safe navigation. Using an agent-based modeling approach, we analyze various scenarios involving pedestrians, wheelchair users, and robots sharing sidewalk spaces. Our findings reveal that robots do not inherently contribute to sidewalk congestion and can be programmed to maintain a buffer zone that allows for both safety and efficiency, suggesting their potential for smooth coexistence with pedestrians. We observed that robots caused most collisions, while pedestrians were primarily responsible for proximity violations of minimum distance thresholds, emphasizing the need for further research and strategies to reduce further the risks associated with these incidents. The results suggest that with careful design-oriented planning and continued research, robots can safely and comfortably share sidewalks with pedestrians, contributing to a more

This article was selected based on the results of a double-blind review of the full paper.

A. Hassan (✉) · G. Sohn

Department of Earth and Space Science Engineering, York University, Toronto, ON M3J 1P3, Canada

e-mail: alihc.me@gmail.com

G. Sohn

e-mail: gunho.sohn@gmail.com

M. Usman

Department of Information and Computer Science, King Fahd University of Petroleum and Minerals, Dhahran 31261, Eastern Province, Saudi Arabia

M. Kremer · P. Faloutsos

Department of Electrical Engineering and Computer Science, York University, Toronto, ON M3J 1P3, Canada

S. Yang

Department of Urban Engineering, Hanbat National University, Daejeon, South Korea

M. Luubert · G. B. Hall

Esri Canada, 900-12 Concorde Pl 4700, Toronto, ON M3C 3R8, Canada

harmonious and efficient urban landscape. Our proposed simulation model can assist urban planners, policymakers, and researchers in evaluating the influence of various design interventions and policies on human-robot coexistence in cities, marking a crucial step toward accommodating both humans and robots in urban spaces.

Keywords Digital twin · Pedestrian behaviour · Sidewalk autonomous delivery robots · Human-robot interaction

1 Introduction

The utilization of multi-functional robots in various human-centered environments has progressed steadily over time. Originally, such robots were limited to fixed locations on assembly lines to facilitate manufacturing production Krüger et al. (2009); Weckenborg et al. (2020). However, recent technological advancements have resulted in robots' increased autonomy and mobility. Mobile robots are currently designed to perform tasks involving public spaces in urban areas, including sidewalks, roadways, and even rail corridors Corominas Murtra et al. (2010); Correa et al. (2012). As they operate in such spaces, they interact with humans. However, the efficiency of deterministic algorithms that govern robot behaviour and operation tends to diminish when there is a greater degree of uncertainty and the movements of humans and robots are mutually influenced Trautman et al. (2015). An instance of such human-robot interaction in public spaces is the employment of last-mile delivery robots, which use sidewalks to traverse between their starting points and delivery destinations. As the deployment of last-mile delivery robots has increased in the last several years, the issue of ensuring the safe and efficient coexistence of robots and humans in urban areas has become of paramount importance. Hence, a new body of literature has started to emerge that seeks to comprehend the interaction between humans and robots, while enhancing the mobility efficiency of both in shared spaces and assisting the navigation of robots through pedestrian zones and intersections.

The rise of e-commerce has significantly increased consumer demand for parcel delivery, leading to higher costs and complexity, especially in last-mile delivery Jennings and Figliozzi (2020). Various solutions have been proposed, including cargo bikes, crowd shipping, self-pickup from parcel lockers, drones, and autonomous delivery robots. One of these options, sidewalk delivery robots show considerable potential due to their ease of operation, cost-effectiveness, and positive impact on freight efficiency and the environment Jennings and Figliozzi (2020); Figliozzi (2020). However, a primary concern for lawmakers and municipal bodies is pedestrian safety in relation to last-mile sidewalk robots Hoffmann and Prause (2018). Limited studies have investigated interactions between robots and humans on sidewalks or robots and vehicles at road intersections Du et al. (2019); Chugo et al. (2021). Evaluating the safety of last-mile delivery robots coexisting with pedestrians and vehicles on sidewalks and roads is crucial Fagnant and Kockelman (2015).

Sidewalk Autonomous Delivery Robots (SADRs) and Road Autonomous Delivery Robots (RADRs) are two types of autonomous robots designed to deliver packages or goods to customers. SADRs are small autonomous robots typically designed to operate on pedestrian sidewalks or footpaths. These robots are designed to operate at low speeds and carry small to medium-sized packages over short distances. SADRs have sensors and cameras that enable them to detect and avoid obstacles, including pedestrians, and they are designed to operate safely in pedestrian areas. On the other hand, RADRs are larger autonomous robots designed to operate on roads, typically in dedicated lanes or areas. The scope of the research examined in this paper is limited to SADRs.

The actual deployment of SADRs in urban areas necessitates a more nuanced approach than RADRs, owing to their increased interaction with pedestrians and various street features. The diversity in the design of city sidewalks, in terms of their dimensions, usage, and traffic characteristics, is dictated mainly by their geographical location. This implies that the interactions of SADRs with other objects within the urban milieu can be quite variable when operational in real-world urban environments. Establishing SADRs as a novel mode of urban transportation requires formulating appropriate regulations. These regulations, to be effective, should be substantiated by empirical data on the influence of SADRs on their immediate surroundings.

To this end, we propose a novel framework to analyze human-robot interactions, their movement patterns, and the benefits they can provide by developing a comprehensive spatial and functionally accurate 3D digital twin of a contained urban area. This controlled digital setting allows users to simulate robot-human and robot-cityscape interactions on the digital twin under near real-world conditions and run a range of simulated interaction scenarios across multiple conditions. Several simulation scenarios are evaluated. The results show that robot-human and robot-cityscape interactions and interventions can enhance the safety and mobility of human sidewalk and roadway users.

2 Literature Review

SADRs, equipped with sensors and cameras, are designed to navigate sidewalks and deliver goods, maneuvering around obstacles Liyanage et al. (2021). While research has addressed legislation, technical capabilities, and potential benefits of SADRs, real-world performance assessments remain limited Jennings and Figliozzi (2020). Autonomous delivery solutions can tackle logistics challenges by reducing costs, traffic, and ecological impact, improving accessibility, and helping supply chain companies enhance their services Ranieri et al. (2018).

However, concerns arise from the use of SADRs, such as potential sidewalk congestion and obstruction for pedestrians, as well as the risk of accidents from collisions with pedestrians or other objects Fagnant and Kockelman (2015); Hoffmann and Prause (2018). To mitigate hazards, like colliding with pedestrians, crashing into

buildings and other fixed objects, or disorientation due to limited sensor-detected features, the use of simulations and simulators that can improve the performance of SADR is crucial Tian and Shi (2022). These tools provide an essential evaluation of autonomous system safety and effectiveness before real-world deployment.

Multi-Agent Simulation (MAS) has emerged as a viable and effective method for representing human interactions with the layout and design of real-world virtual environments. Various pedestrian navigation and collision avoidance methods have been developed over the past few decades, each of which emulates different aspects of human-like movement Kapadia et al. (2015); Pelechano et al. (2016). Physical force-based approaches which employ attraction and repulsion forces to simulate individual pedestrian interactions have been utilized to model pedestrian movement Karamouzas et al. (2009). These approaches act upon the location and trajectories of pedestrians to steer them toward their intended destinations, while also preventing collisions with obstacles and other pedestrians in the environment. They have been specifically used to evaluate and assess the dynamic characteristics of human-human interactions and human-building interactions in virtual environments Usman et al. (2019, 2021). The work presented in Usman et al. (2020) investigated different navigational policies for social distancing using a multi-agent system.

SADRs predominantly function on sidewalks, complying with pedestrian regulations to transport packages autonomously, presenting a novel solution to address last-mile delivery challenges Mintrom et al. (2022). As shown in Table 1, various studies have explored different aspects of autonomous robots and their navigation methods. These studies encompass a diverse range of robot types, including SADRs, quadrupedal robots, humanoid torsos, and robotic wheelchairs. The navigation methods employed in these studies vary from adhering to pedestrian laws Morales et al. (2009), utilizing sensors and cameras Kümmerle et al. (2015), and employing LiDAR-based obstacle detection Wen et al. (2022), to incorporating human-like collision avoidance and pedestrian social behavior emulation Du et al. (2019). Notably, Nguyen et al. (2018) focused on robotic wheelchairs, considering the social relationship and walking behavior of humans during side-by-side navigation. A recent study demonstrated a framework for IndoorSim-to-OutdoorReal transfer for navigation, further supporting the potential of autonomous robots to navigate robustly novel environments and obstacles Truong et al. (2023).

3 Digital Twin: 3D University Campus

Web-based digital twin simulators have become an advantageous tool for integrating various types of data sources, including geospatial data, to provide critical information about a location's physical and environmental characteristics Lei et al. (2022). The use of Geographic Information System (GIS) data integrated with 3D building information model (BIM) renditions of actual buildings as well as other three-dimensional objects such as trees, outside light fixtures, street furniture, and details such as roadside curbs, and variable terrains allows researchers to construct

Table 1 Research on autonomous robots and their navigation methods

| Study | Short description | Type of robot | Navigation method |
|------------------------|---|---------------------|---|
| Morales et al. (2009) | Explores traffic congestion alleviation | SADRs | Navigating pedestrian areas while adhering to pedestrian laws and regulations |
| Kümmerle et al. (2015) | Equipped with sensors and cameras | SADRs | Autonomous navigation using sensors and cameras |
| Wen et al. (2022) | Velodyne-16 LiDAR for situational awareness | SADRs | LiDAR-based obstacle detection |
| Sorokin et al. (2022) | Quadrupedal robot for sidewalk navigation | Quadrupedal robot | Uses onboard sensors, a route plan, and a two-staged learning framework |
| Wen et al. (2022) | Vision-based sidewalk navigation algorithm | SADRs | Navigation through sidewalk segmentation, costmap generation, and motion planning |
| Du et al. (2019) | Pedestrian social behaviour emulation | SADRs | Group surfing, curb-following, human-aware collision avoidance |
| Nguyen et al. (2018) | Robotic wheelchairs in crossing situations | Robotic wheelchairs | Side-by-side navigation considering the social relationship and walking behaviour of humans |
| Shiomi et al. (2014) | Human-like collision avoidance | Humanoid torso | Collision avoidance system using the social force model designed for shopping malls and similar pedestrian environments |

realistic immersive virtual environments that accurately mimic the conditions of a real-world area, making them ideal for studying the interactions between sidewalk delivery robots and pedestrians Ranieri et al. (2018). In this section, we present the data and methods utilized to create a 3D digital twin model of York University’s Keele campus in Toronto, Ontario (Fig. 1).

3.1 Building Models

York University’s campus comprises many diverse academic, administrative, residential, and utility buildings as well as significant treed amenity spaces. Twelve 3D BIM and, numerous additional 2D-computer-assisted design (CAD) building floor plans were obtained from the Campus Facilities Department. The BIMs were



Fig. 1 Snapshot York University campus digital twin

included in the 3D campus model and were supplemented with 29 SketchUp models of the textured exterior of buildings and 32 other textured building exterior models created using the open-source platform Blender (Fig. 2). All 12 BIMs were provided in Autodesk Revit format, five of which included interior details and six exterior details. LiDAR data from 2018 were used to ascertain building heights as well as to geolocate features such as exterior light standards, and trees and to add detail to a digital elevation model. Overall, we processed 11 BIMs, 84 2D-CAD, and 75 3D-CAD models using techniques like georeferencing, extrusion, and annotation extraction. Campus Planning, Design, and Construction (CPDC) supplied the BIM and CAD models. The former was converted from AutoDesk Revit format to building scene layer packages in ArcGIS Pro, whereas the SketchUp and Blender models were exported to .obj format into Esri's City Engine software and from there into a scene layer package. The 2D CAD building floor plans were not used in the current 3D campus model, although there are plans to integrate some of these as interiors for the SketchUp and Blender models. Georeferencing was performed using Esri's ArcGIS Pro software, and room annotations were pre-served as 2D point scene layers.

3.2 Terrain

The digital twin integrated a Digital Elevation Model (DEM) and as noted above, other land use information was obtained both from contour data and point shapefiles as well as from LiDAR data, classified semi-automatically in the summer of 2019. Ground LiDAR points generated the DEM at 0.5 m x 0.5 m resolution, while the GIS 2D layers captured land use data such as roads, parking lots, and sidewalks. Street furniture models provided context for the 3D simulation. Campus Services and Business Operations (CSBO) supplied land-use data, which informed simula-

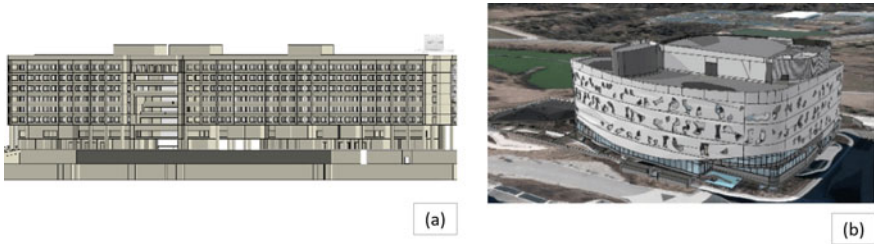


Fig. 2 An example of BIM model **a** is an example of an exterior model for Ross building **b** georeferenced for The Bergeron Centre

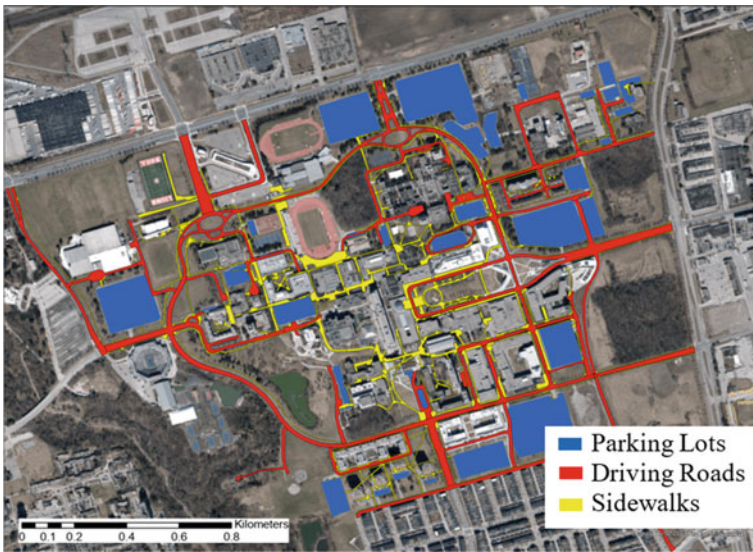


Fig. 3 Landuses dataset for driving road, parking, and sidewalks

tion object definitions. The database had three categories: driving roads, parking lots (type, name, and spaces), and sidewalks. Figure 3 shows parking lot, road, and sidewalk data.

4 Methodology

We employed agent-based modeling with a Multi-Agent System (MAS) to simulate pedestrian and sidewalk robot interactions within the digital twin of York’s campus. As noted in the introduction, the study focuses on safety, comfort, and efficiency, with various experiments for sensitivity analysis. The proposed TwinWalk simulation system makes use of the detailed 3D campus model outlined in the previous section, and a Predictive Avoidance Model (PAM) is used to enable situational awareness and obstacle avoidance for pedestrians and robots.

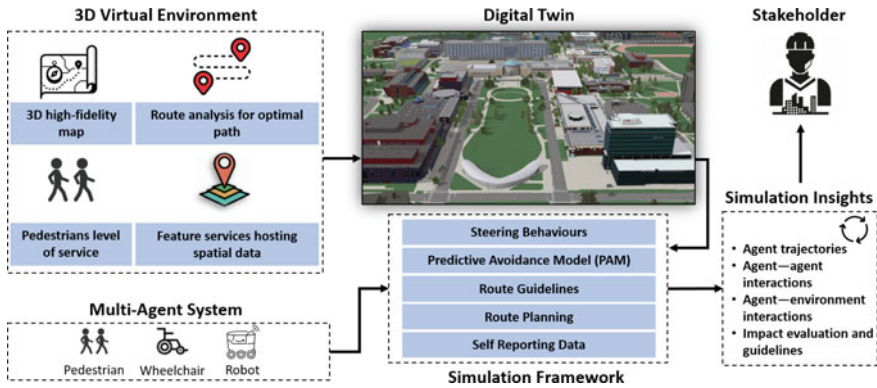


Fig. 4 An overview of the proposed framework, highlighting the crucial elements and fundamental operations in a digital twin-based simulator for pedestrians and side sidewalk delivery robots

4.1 *TwinWalk: Human-Robot Simulation System*

The digital twin-based sidewalk simulator, embedded with key functionalities, allows for the simulation of interactions between humans, robots, and their combined dynamics in a complex, realistic environment. This environment features terrain, vegetation, parking, sports facilities, and all campus buildings. Agents, comprising pedestrians and sidewalk robots, navigate the campus via sidewalks, guided by the ArcGIS network analysis routing API. Data from these interactions are examined to understand emergent patterns in human-robot interplay and pinpoint strategies to bolster safety, efficiency, and user comfort.

Figure 4 illustrates the multi-tool, multi-framework approach used to construct the sidewalk simulations, anchored by the detailed 3D model. The model includes GIS feature layers for trees, lighting, and street furniture, accurately depicted 3D buildings and sidewalks, and a point layer for all building access points. Agent-based modeling and intelligence are facilitated through the application of various steering behaviours (like follow, avoid, and seek), necessary for effective navigation. These behaviours, inspired by Reynolds (1987), are adapted to meet the specific demands of the simulation.

The simulation is rendered using ThreeSteer, a powerful open-sourced steering behaviours library based on THREE.js Marcon (2023). Although the THREE.js library provides an excellent base for implementing these behaviours, it has limitations in interacting with the 3D campus model which is hosted by ArcGIS 3D web scene services. To address this challenge, the interaction between ThreeSteer agents and ArcGIS was implemented using feature layers hosted on ArcGIS Online (Esri’s cloud-based software platform). Agents navigate towards their goals using sidewalks on an efficient path retrieved through ArcGIS routing analysis, conducted

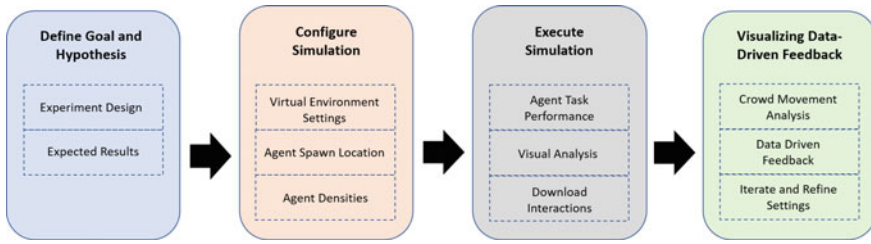


Fig. 5 Steps of simulation cycle: define the goal, set scenario, run and analyze, visualize feedback

on a feature layer of campus sidewalks. We selected Starship¹, as the basis for our robot model. Starship’s small self-driving robots, which weigh less than 35 kg and are designed to operate on sidewalks at a maximum speed of 6 km/h, are capable of delivering goods locally within 15–30 minutes and within a radius of up to 5 km.

Simulation cycle. The simulation cycle is a four-step process that begins with defining the goal and hypothesis of the expected results, see Fig. 5. This initial step is critical in determining the purpose and direction of the simulation, as it sets the tone for the entire process. The next step involves setting up the scenario, which includes determining the intensity of agents to simulate along with their origin and destination locations. This step requires careful consideration of the specific variables involved and should be tailored to the desired outcome of the simulation. Once the scenario has been set up, the simulation can be run and analyzed both visually and quantitatively. This step is crucial in determining the accuracy of the simulation and identifying any potential flaws in the design or execution. Finally, the data-driven feedback obtained from the simulation can be visualized, and sensitivity analysis can be performed. This step involves reviewing the results of the simulation and analyzing how changes to certain variables may affect the overall outcome. Overall, the simulation cycle is an iterative process that allows researchers to test hypotheses and gain insights into complex systems.

Route planning. Optimal paths are integral to routing applications, as they guarantee the selection of the most efficient and effective routes. Navigation and pathfinding are vital components of many simulation tools, such as Unity and Unreal Engine, which offer built-in services for these purposes. We employed Esri’s ArcGIS Indoors software tools to determine the best paths for routing applications. While ArcGIS Indoors tools are designed for creating indoor navigation networks, they can also be effectively utilized with sidewalk polygons representing building hallways to constrain the network lattice. To establish a precise and efficient routing system, the base pathway polygon was imported into the levels feature class of the indoor geodatabase, enabling the generation of a graph map for the base pathway. We used surveyed data points of entrance and exit points to thin the network by eliminating unnecessary preliminary network pathways. Further refinement and simplification of

¹ <https://www.starship.xyz/>.

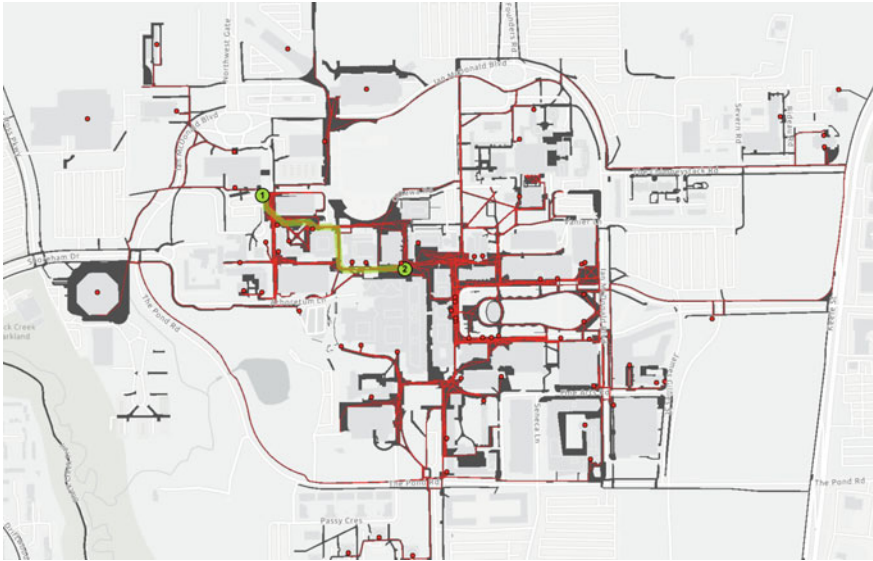


Fig. 6 Navigation analysis network graph: green-highlighted shortest path connects the starting point (1) to the destination (2), illustrating efficient routing. *Source* <https://storymaps.arcgis.com/stories/b694f3ae65b346e79437cc542fa7159d>

the thinned network were performed using a scripted geoprocessing tool in ArcGIS Pro (Fig. 6). By using these sophisticated tools and techniques, the routing system was optimized to deliver the most efficient and effective routes for pedestrians and sidewalk robots navigating through the campus.

Deployment and User Interface (UI). The web-based simulation platform (Fig. 7), hosted locally, can be accessed via any browser on a computer with sufficient GPU and CPU capabilities, eliminating the need for installing additional software or tools. As noted, the virtual model of the campus is hosted on ArcGIS Online and is accessible through the JavaScript API. This platform also employs feature services to host crucial spatial data, such as entrances and exits, street furniture, land-use, buildings, and other features. These data sets are retrieved and queried from ArcGIS Online using the same JavaScript API. The UI provides options for adjusting pedestrian and robot parameters, such as Route Control, which enables selecting starting and ending points for the route, and Directional Control, which allows for uni- or bidirectional movements between the designated points. Furthermore, the UI incorporates agent spawning control parameters, including uniform or random spawning, the number of pedestrian and robot agents, the interval between each spawn, and an option to enforce social distancing measures.

Simulation statistics. After configuring a desired scenario, users can initiate the simulation. Upon the simulation's conclusion, users have the option to download the logged data of agent trajectories as a CSV file. Alternatively, users can choose auto-

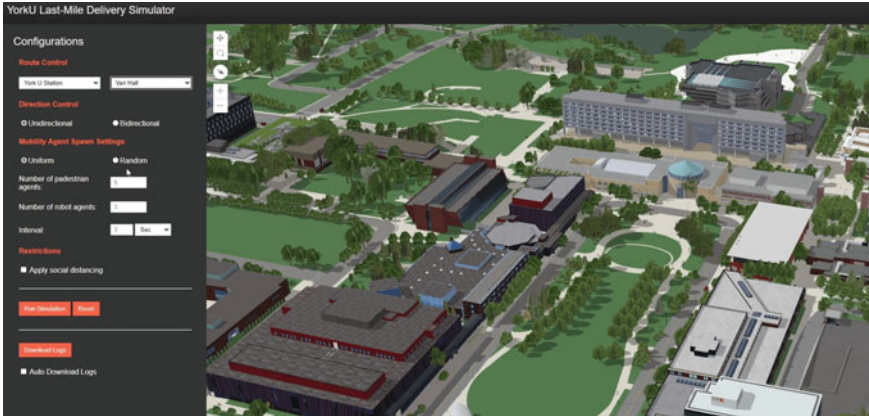


Fig. 7 The user interface (UI) of the web-based last-mile simulation platform, designed for conducting comprehensive campus-wide scenario analyses within a virtual representation of York University

matic downloading of the log files as soon as each agent completes its trajectory. This streamlined process allows for efficient data collection and analysis while maintaining user-friendly accessibility, making the platform ideal for studying various aspects of pedestrian and robot behaviour in a dynamic environment.

4.2 Pedestrians, Wheelchair Users, and Sidewalk Delivery-Robots

In our navigation approaches, pedestrians, wheelchair users, and robots are regarded as self-propelled entities traversing a continuous space. We utilized microscopic models as the basis for simulating pedestrian dynamics, which includes people walking and using motorized wheelchairs. We employed agent-based modeling for these groups to better capture the intricacies of their interactions and individual behaviours in navigating sidewalks. This approach allows for a more accurate representation of their individual behaviours and decision-making processes, which is critical for developing effective and socially compliant navigation strategies in shared environments. In particular, we modeled motorized wheelchair users as a part of our efforts to account for the full spectrum of entities that might be encountered in an urban sidewalk setting, ensuring that our simulations are inclusive of accessibility needs. Wheelchair users are an essential part of the pedestrian landscape and their inclusion is crucial for creating a realistic and comprehensive model of sidewalk dynamics.

By modeling individual pedestrian and wheelchair movements, we can derive macroscopic descriptions of pedestrian flow, enabling assessments of escape routes, the design of pedestrian facilities, and the examination of theoretical issues, such as



Fig. 8 Last mile delivery robot simulation tool: (a) and (b) show diverse mobility agents, including robots, pedestrians, and wheelchair users, seamlessly progressing to their destinations. In (c), agents adhere to the path created by the route analysis API

determining the safe proximity between robots, pedestrians, and wheelchair users or establishing appropriate robot movement speed Alonso-Mora et al. (2018); Nishino et al. (2017). This approach provides a holistic view of the interactions and dynamics on shared sidewalks, accounting for the varied behaviours and movements of pedestrians, wheelchair users, and robots as shown in Fig. 8.

Pedestrian simulation. Many pedestrian simulation models have been introduced, including force-based models such as Social Forces Helbing and Molnar (1995), which used attractive and repulsive forces to guide agents toward their goals and maintain safe distances between agents. This can be improved by adding a predictive aspect so that agents can trigger avoidance maneuvers before an oncoming collision. One such model is the Predictive Avoidance Model (PAM) Karamouzas et al. (2009), which uses a piecewise predictive function based on relative velocities between agents to estimate possible future collisions for a given time window. PAM has been shown to be able to model a variety of human behaviours such as distracted pedestrians Kremer et al. (2021), and is the simulation algorithm used in this work.

Delivery robot simulation. SADR offers an innovative solution to tackle last-mile delivery challenges by primarily operating on sidewalks, adhering to pedestrian regulations, and autonomously transporting packages Mintrom et al. (2022). The scientific rationale for this approach, deeply rooted in the literature on real-world robots, highlights the importance of SADR being specifically engineered to safely navigate pedestrian areas, minimizing their impact on vehicular traffic, and reducing congestion Morales et al. (2009); Srinivas et al. (2022). Equipped with advanced sensors, cameras, and navigation systems Kümmerle et al. (2015), these fully autonomous

delivery robots skillfully detect and avoid obstacles such as pedestrians, cyclists, pets, and other sidewalk users, complying with established pedestrian laws and regulations without the need for human intervention Beer et al. (2014).

To support further the idea of using pedestrian navigation regulations and models for robot navigation on sidewalks, Shiomi et al. Shiomi et al. (2014) developed a human-like collision avoidance system for mobile robots by applying a pedestrian model from human science literature, specifically the Social Force Model designed for shopping malls and similar pedestrian environments. The proposed system was tested through 2-hour field trials in real-world settings and demonstrated not only increased comfort for pedestrians but also safer navigation compared to traditional collision-free methods, as it better aligns with the behaviour of pedestrians in crowded situations. Based on the literature review and the promising results mentioned above, we used the PAM for our robot simulations to ensure effective and safe navigation in pedestrian environments.

5 Experiment Design and Evaluation Metrics

A controlled study using agent-based modeling, simulation, and the digital twin frameworks examines the safety and behaviour of pedestrian interactions with autonomous sidewalk delivery robots. The experimental design simulates various sidewalk interactions among pedestrians, wheelchair users, and delivery robots, with agent-based models representing each entity. Realistic attributes and behaviour, such as walking speed, dimensions, personal space requirements, and movement patterns, are incorporated. The study is structured into distinct phases with varying pedestrian, wheelchair user, and robot densities, and agents are introduced in a randomized sequence. Agents navigate the environment based on the optimal path derived from route analysis.

5.1 Experiment Design

As urban environments evolve, the impact of emerging technologies on pedestrian walkways and sidewalks must be considered. The traditional Fruin's Level of Service (LOS) model Fruin (2023) focuses on pedestrian density but does not account for the growing presence of sidewalk delivery robots. To address this, we propose a modified LOS model that considers both pedestrian and robot densities, providing a more accurate representation of dynamic interactions and space requirements on shared sidewalks. In the pedestrian density, we have included wheelchair users who form part of the varied pedestrian landscape in real-world settings. Wheelchair users were spawned with a randomness factor of 1:9, meaning for every 10 pedestrians, one was a wheelchair user. This was to ensure that our simulation closely mirrored the diverse range of entities one would encounter on a sidewalk. The proposed model

Table 2 Scenario codes, service labels, descriptions, and adjusted densities (LOS) for pedestrians and robots on sidewalks

| Scenario code | LOS | Description | Densities (ped/m ²) |
|---------------|------|---|---|
| S1 | A | Low-density pedestrian traffic | < 0.31 pedestrians |
| S2 | D | High-density pedestrian traffic | 1.08–0.72 pedestrians |
| S3 | A-R | Low-density pedestrian traffic with a low-density of robots | 0.24–0.28 pedestrians, 0.04–0.06 robots |
| S4 | D-R | High-density pedestrian traffic with a low-density of robots | 1.065–0.675 pedestrians, 0.04–0.06 robots |
| S5 | A-HR | Low-density pedestrian traffic with a high-density of robots | 0.21–0.24 pedestrians, 0.1–0.13 robots |
| S6 | D-HR | High-density pedestrian traffic with a high-density of robots | 1.005–0.57 pedestrians, 0.1–0.13 robots |

accounts for the physical dimensions of the robots and their potential impact on pedestrian flow and density. The robot dimensions are 27.4" (69.7 cm) in length, 22.4" (56.9 cm) in width, and 22.5" (57.1 cm) in height², the robots can influence the available space for pedestrians and result in lower pedestrian densities for a given LOS. In this study, we adjust Fruin's LOS model to include autonomous sidewalk delivery robots and present experimental scenarios with varying pedestrian and robot densities. This adjusted model offers a starting point for analyzing pedestrian and robot interactions on shared sidewalks, and specific densities can be refined and validated through data collection and analysis.

We introduce new labels for the adjusted LOS levels: A-R and D-R for low-density robots' presence, and A-HR and D-HR for the corresponding high-density robots' presence (Table 2). These labels represent scenarios where pedestrians share the sidewalk with robots, with densities adjusted for the space occupied by the robots. The updated LOS model aims to represent better pedestrian and robot dynamics on sidewalks, offering a more accurate framework for studying and planning shared walkways in the context of emerging urban technologies.

Eliminating cognitive memory from the agent-based models in this study ensures consistent agent behaviour and decision-making processes, allowing for the isolation of the effects of varying pedestrian, wheelchair user, and robot densities on interactions and navigation patterns. The study contributes to a deeper understanding of pedestrian interactions with autonomous sidewalk delivery robots under diverse densities and paves the way for future exploration of more complex adaptive behaviours and their potential implications for safety and efficiency in real-world scenarios.

Two discrete research zones were selected to scrutinize the influence of varying pedestrian and robot densities on sidewalk dynamics. As shown in Fig. 9, the first study area (a) is located between the university's Subway Station and the Lassonde Building, a bustling location with high pedestrian traffic and diverse demographics,

² <https://starshipdeliveries.com/industry/>.

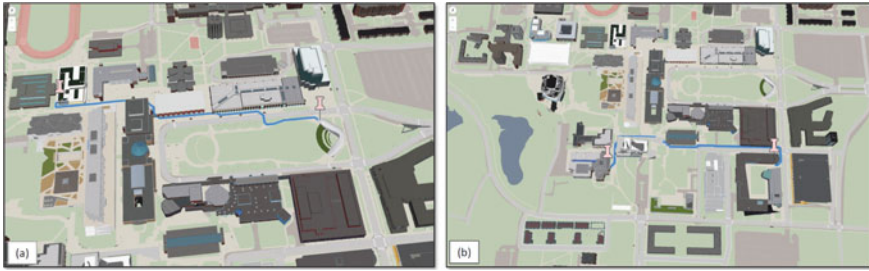


Fig. 9 Two study areas showcase pedestrian and robot routes, with walkway paths emphasized in blue, connecting origin and destination

ensuring a representative sample of the general population. The second study area (b) extends from Seneca at York to the Ignat Kaneff Building, another prominent location on campus with a significant pedestrian influx. These two study areas aim to capture the diversity of pedestrian behaviour and effectively evaluate the performance of the proposed scenarios under different conditions.

Data on individual agents and the system as a whole were gathered during each experimental scenario, including self-reported metrics such as position, speed, effort, kinetic energy, path length, and collision information. These data were collected at regular intervals and stored in a CSV file format for further analysis.

5.2 Evaluation Metrics and Their Significance in Sidewalk Coexistence of Pedestrians and Robots

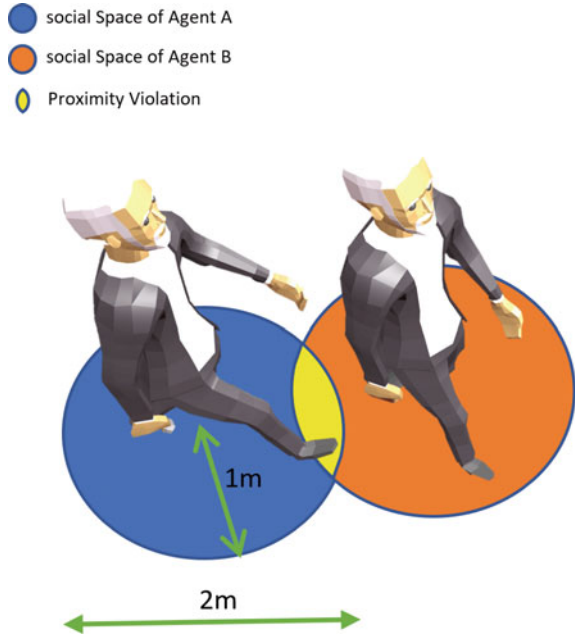
This section delineates the metrics selected for assessing the coexistence of pedestrians and robots on sidewalks. These metrics—personal space violations (2 m), collision points, and overall flow—have been judiciously chosen to holistically encapsulate the interaction dynamics.

Personal space violations operationalized as an infringement of the 2 m radius around each individual, serve as a litmus test of comfort and respect for social norms in pedestrian dynamics as illustrated in Fig. 10. Each occurrence of this 2 m space being violated by either another pedestrian or a robot is marked as a transgression. A comprehensive understanding of these transgressions is quintessential to calibrating the navigation algorithms of robots for seamless integration into pedestrian realms.

In the same vein, collision points are integral indicators of safety. A collision is defined as any physical contact between two entities—this could be between two pedestrians, a pedestrian, and a robot, or two robots. By monitoring these collision points, we can identify areas of concern and suggest alterations in robot navigation or sidewalk management to augment overall safety.

To gauge efficiency, we monitor the overall flow of pedestrians and robots on the sidewalk. This metric encapsulates the ease and speed with which pedestrians and

Fig. 10 An illustration of an agent's social space and its violation



robots navigate the sidewalk without causing substantial disruptions to one another. Enhanced flow optimization ensures that sidewalk robots execute their tasks without impinging on pedestrian movement, and reciprocally.

Thus, our analysis not only aims to decipher the existing dynamics of sidewalk robots' integration into pedestrian spaces but also endeavors to pinpoint areas for potential improvement. This will facilitate smoother navigation and minimal disruptions for all parties involved. By elucidating each variable and metric observed in this study, we strive to demystify the objectives of these simulation experiments and shed light on the rationale behind our selection of variables.

6 Results

The collected metrics help assess the efficiency, comfort, and safety of their coexistence on sidewalks, while also considering personal space violations (2m), collision points, and overall flow. The analysis aims to understand and optimize the integration of sidewalk robots into pedestrian environments, ensuring smooth navigation and minimal disruptions for both parties.

6.1 Proximity Violations and Collisions

We analyzed data (Fig. 11) collected from 12 different scenarios (i.e., repeating 6 scenarios from Table 2 for the two study areas as shown in Fig. 9), involving collisions and proximity violations among pedestrians, wheelchair users, and robots. The cumulative spawn count for all entities reached 698, consisting of 469 pedestrians, 186 robots, and 43 wheelchair users. There were a total of 475 collisions, with the majority involving robot-to-pedestrian interactions (47.6%), followed by pedestrian-to-pedestrian interactions (38.5%) and wheelchair users involved in 13.9% of the collisions. On the other hand, there were 40,674 recorded proximity violations, with pedestrian-to-pedestrian interactions being the most prevalent (70.0%). Robot-to-pedestrian violations accounted for 25.7% of the incidents, while wheelchair users were involved in a smaller percentage of proximity violations, constituting 4.3% of the total.

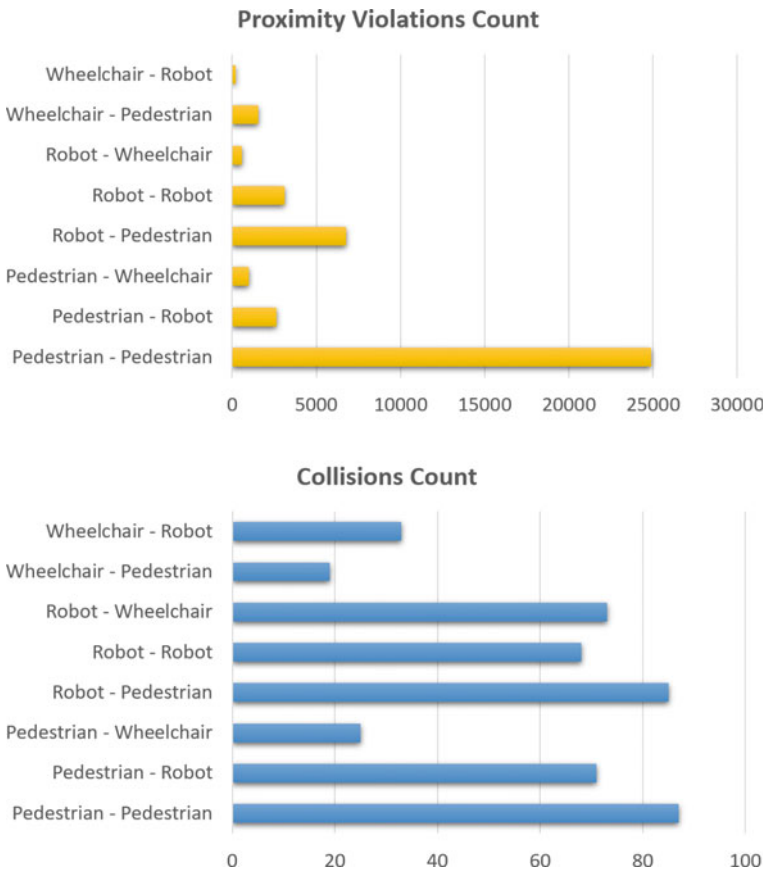


Fig. 11 Comparative analysis of collisions (top) and proximity violations (bottom): bar graphs depicting the prevalence and distribution of incidents among pedestrian, wheelchair user, and robot agent pairs

Table 3 Proximity violations and collisions between Pedestrian-to-Pedestrian (P-P) and Robot-to-Pedestrian (R-P) for the two study areas as shown in Fig. 9

| Scenario | Pedestrians | Robots | Study area (a) | | Study area (b) | |
|----------|-------------|--------|------------------------|--------------------|------------------------|--------------------|
| | | | Proximity violations | Collisions | Proximity violations | Collisions |
| S1 | 30 | 0 | P-P: 986 R-P: 0 | P-P: 0 R-P: 0 | P-P: 928 R-P: 0 | P-P: 0 R-P: 0 |
| S2 | 75 | 0 | P-P: 4805 R-P: 0 | P-P: 34 R-P: 0 | P-P: 4529 R-P: 0 | P-P: 13 R-P: 0 |
| S3 | 18 | 8 | P-P: 303 R-P: 211 | P-P: 0 R-P: 8 | P-P: 497 R-P: 257 | P-P: 2 R-P: 1 |
| S4 | 60 | 12 | P-P: 3560 R-P: 700 | P-P: 24 R-P: 13 | P-P: 3589 R-P: 961 | P-P: 3 R-P: 8 |
| S5 | 18 | 38 | P-P: 283 R-P: 821 | P-P: 0 R-P: 15 | P-P: 415 R-P: 831 | P-P: 2 R-P: 5 |
| S6 | 55 | 35 | P-P: 2818 R-P: 1622 | P-P: 22 R-P: 64 | P-P: 2783 R-P: 1945 | P-P: 12 R-P: 44 |

To understand better the overall share of each agent type in causing violations and collisions as per their population across all scenarios, we adjusted the above percentage calculations of proximity violations and collisions based on their spawn count. While pedestrians had the highest average of proximity violations (60.7), they experienced the fewest collisions (0.39 per pedestrian). Conversely, robots and wheelchair users had lower proximity violation averages but higher collision rates, with 1.2 collisions per robot and 1.5 per wheelchair user. The observed differences in proximity violations and collision rates among pedestrians, robots, and wheelchair users could also be attributed to factors such as the average moving speed and personal space occupied by robots on sidewalks. Robots, with their higher average speeds, may have a more challenging time adjusting to sudden changes in their environment or anticipating other agents' movements. This difficulty in adapting could result in increased proximity violations and collisions. Table 3 shows a snapshot of data collected for the given scenarios from the two study areas. Our findings suggest that urban planners and other stakeholders should take into consideration the population density (i.e., Level of Service) of sidewalks when permitting robots on the shared spaces. By accounting for population density, stakeholders can identify optimal times and routes for robots to operate, minimizing congestion and ensuring a safer coexistence with pedestrians and wheelchair users. This could involve implementing dedicated robot lanes, restricting robot usage or minimizing robot population density during peak hours, or mandating lower speed limits for robots in crowded areas.

6.2 *Flow Density*

Flow density is a crucial metric for assessing the movement and dynamics of pedestrians and sidewalk robots on sidewalks. It represents the concentration of agents in a given area and time interval, capturing the ease with which they can coexist and navigate the sidewalk. The observed differences in the average minimum distances between pedestrians and robots play a significant role in shaping the density flow patterns in the respective scenarios. The average minimum distance between robots and pedestrians is 0.599m, which is substantially larger than the 0.333m average minimum distance between pedestrians. This increased separation between robots and pedestrians suggests that robots maintain a greater buffer zone in their surroundings, contributing to a more evenly distributed density flow. This characteristic is particularly evident in scenarios with a higher proportion of robots, where the larger average minimum distances help to create a more dispersed flow and reduce congestion.

In contrast, the smaller average minimum distance between pedestrians indicates that individuals tend to maintain closer proximity to one another during violations than the robots. Consequently, scenarios with a higher number of pedestrians exhibit denser flows, as the closer interactions lead to a more compact and concentrated distribution of individuals. Figure 12 illustrates the difference in flow density, (a) depicting a scenario with only pedestrians (S1), while (b) showcasing a scenario featuring both pedestrians and robots (S3).

The findings suggest that, contrary to some reports Ban the robots (2023), robots may not inherently contribute to sidewalk congestion, potentially allowing for smoother co-existence with pedestrians. Hence, urban planners could consider incorporating robots sharing sidewalks as a design decision when designing shared urban spaces, provided they carefully account for the number of robots in the overall crowd density. This approach suggests a potential for smooth coexistence and navigation for both pedestrians and robots, contributing to a more harmonious and efficient urban environment.

6.3 *Robot Efficiency*

In scenario S4, we evaluated the trajectories of a pedestrian and a robot, each traveling from the same origin to the same destination over a given period. The evaluation was aimed at understanding the overall efficiency of navigation, focusing particularly on the robot's performance in its role of delivering parcels. While the term 'efficiency' might suggest a simple difference in travel speed, in the context of our study, it encompasses several other aspects of navigation. This includes the ability of the robot to effectively plan its route, make real-time decisions, adapt to environmental conditions, and avoid collisions, all of which collectively contribute to the completion of the task without unnecessary delays or interruptions.

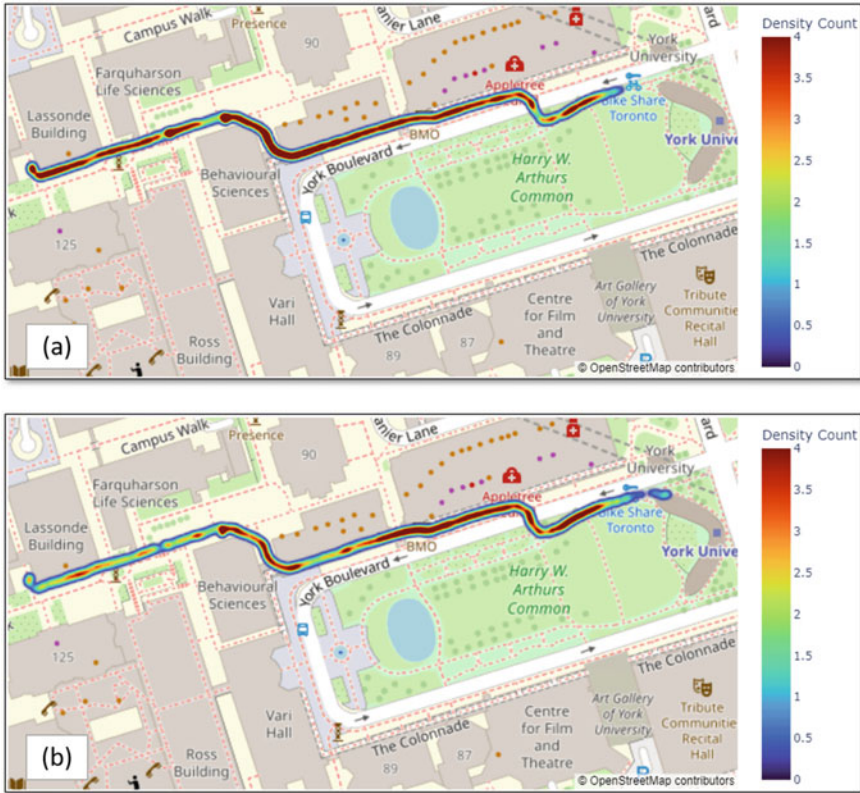


Fig. 12 Comparative analysis of flow density between two distinct scenarios: **a** scenario S1 representing low-density pedestrian traffic and **b** scenario S3 featuring comparable low-density pedestrian traffic but with an added low-density of robots

The trajectories, recorded in terms of latitude and longitude, allowed us to compute the total distance traveled and the total time taken by each agent. The pedestrian covered a total distance of 0.46 km, whereas the robot traveled slightly further, covering 0.47 km. However, the robot completed the trajectory in a shorter time span of 462.41 s (approximately 7.71 min), compared to the pedestrian who required 581.46 s (approximately 9.69 min). These findings highlight not just the differences in speed, but more importantly, the effectiveness of the robot’s navigation system in enhancing its efficiency in a shared environment. The robot’s capabilities for advanced collision avoidance and real-time decision-making played a key role in this context.

Figure 13 provides a visual representation of the trajectories of both the pedestrian and the robot, following the same route to the same destination. From the observed data, we noticed that the robot’s trajectories displayed wider turns compared to the pedestrian, suggesting that the robot maintained a greater interpersonal distance, especially in denser areas such as turns. This behaviour, which is in stark contrast to

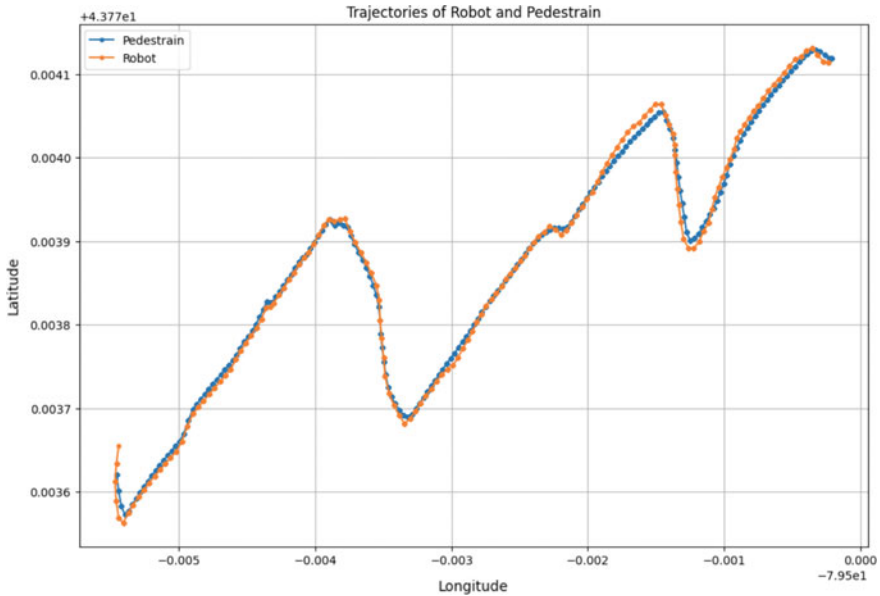


Fig. 13 Trajectories of pedestrian and robot following the same route to reach the same destination

pedestrians who tend to move in closer proximity to other agents, is a critical aspect of the robot’s design, enabling it to maintain a larger buffer zone for safety and efficiency. This capability ultimately contributes to the robot’s superior performance in shared sidewalk environments. Therefore, it is clear that the robot’s efficiency is a product of not just its speed, but also these added components that ensure safe, effective, and unobstructed sidewalk navigation.

7 Conclusion

This study has provided a comprehensive analysis of interactions among pedestrians, wheelchair users, and sidewalk robots, aiming to ensure their safe and comfortable coexistence in urban environments. Employing agent-based modeling, we simulated various scenarios to investigate the dynamics of these agents sharing sidewalk spaces.

The application of simulation-based experiments in this research yielded significant insights that are beneficial for formulating regulations governing the real-world operation of SADR in urban environments. The simulation outcomes, which include metrics such as pedestrian collisions, proximity infringements, and pedestrian traffic flow, offer substantial data. This information can guide decisions about suitable areas for SADR deployment, zones requiring a circumspect operation, and strategies to guarantee their safe and efficient functionality. Contrary to prevailing assump-

tions, our findings indicate that robots tend to preserve a larger safety margin and do not inherently exacerbate sidewalk congestion. These results suggest their potential for seamless integration with pedestrian traffic, fostering more harmonious and efficient urban ecosystems. The research also reveals a higher incidence of collisions involving robots, along with an increased rate of proximity violations experienced by pedestrians. These revelations underscore the necessity for additional research and the development of strategic measures to mitigate such incidents, particularly during interactions involving diverse agent types. The disparity in incident rates accentuates the importance of taking into account factors like the LOS of sidewalks when granting operational permits to sidewalk robots. These insights can empower urban planners to pinpoint the most favorable operating schedules and routes for robots, with the dual aim of minimizing congestion and enhancing safety for all entities involved. Potential interventions could involve the introduction of dedicated robot lanes, limiting robot operations during high-traffic periods, or imposing lower speed limits for robots in densely populated zones. By synthesizing our findings, we demonstrate the potential benefits of incorporating sidewalk robots into urban planning considerations. With careful planning and continued research, robots can share sidewalks safely and comfortably with pedestrians and wheelchair users, contributing to a more efficient and harmonious urban landscape.

References

- Alonso-Mora J, Beardsley P, Siegwart R (2018) Cooperative collision avoidance for nonholonomic robots. *IEEE Trans Robot* 34(2):404–420
- Ban the robots (2021) <https://www.thestar.com/opinion/contributors/2021/12/14/ban-the-robots-but-free-the-cargo-e-bikes-to-cut-traffic-congestion-on-toronto-streets.html>. Accessed 07 May 2023
- Beer JM, Fisk AD, Rogers WA (2014) Toward a framework for levels of robot autonomy in human-robot interaction. *Human-Robot Interact* 3(2):74
- Chugo D, Yamada S, Muramatsu S, Yokota S, She JH, Hashimoto H (2021) Mobile robot navigation considering how to move with a group of pedestrians in a crosswalk. In: *Human system interaction (HSI)*, pp 1–7
- Corominas Murtra A, Trulls E, Sandoval O, Pérez-Ibarz J, Vasquez D, Mirats-Tur JM, Ferrer M, Sanfeliu A (2010) Autonomous navigation for urban service mobile robots. In: 2010 IEEE/RSJ international conference on intelligent robots and systems, pp 4141–4146. <https://doi.org/10.1109/IROS.2010.5649151>
- Correa DSO, Sciotti DF, Prado MG, Sales DO, Wolf DF, Osorio FS (2012) Mobile robots navigation in indoor environments using kinect sensor. In: 2012 Second Brazilian conference on critical embedded systems, pp 36–41. <https://doi.org/10.1109/CBSEC.2012.18>
- Du Y, Hetherington NJ, Oon CL, Chan WP, Quintero CP, Croft E, Machiel Van der Loos H (2019) Group surfing: a pedestrian-based approach to sidewalk robot navigation. In: *Robotics and automation (ICRA)*, pp 6518–6524
- Fagnant DJ, Kockelman K (2015) Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transport Res* 77:167–181
- Figliozzi MA (2020) Carbon emissions reductions in last mile and grocery deliveries utilizing air and ground autonomous vehicles. *Transport Res* 85:102, 443

- Fruin JJ (2023) Pedestrian planning and design. Metropolitan Association of Urban Designers and Environmental Planners. New
- Helbing D, Molnar P (1995) Social force model for pedestrian dynamics. *Phys Rev E* 51(5):4282
- Hoffmann T, Prause G (2018) On the regulatory framework for last-mile delivery robots. *Machines* 6(3)
- Jennings D, Figliozzi M (2020) Study of road autonomous delivery robots and their potential effects on freight efficiency and travel. *Transport Res Rec* 2674(9):1019–1029
- Kapadia M, Pelechano N, Allbeck J, Badler N (2015) Virtual crowds: steps toward behavioral realism. *Synthesis Lect Visual Comput* 7(4):1–270
- Karamouzas I, Heil P, van Beek P, Overmars MH (2009) A predictive collision avoidance model for pedestrian simulation. *Motion in Games. Lecture notes in computer science*. Springer, Berlin, Germany, pp 41–52
- Kremer M, Haworth B, Kapadia M, Faloutsos P (2021) Modelling distracted agents in crowd simulations. *Visual Comp* 37:107–118
- Krüger J, Lien T, Verl A (2009) Cooperation of human and machines in assembly lines. *CIRP Annals* 58(2):628–646
- Kümmerle R, Ruhnke M, Steder B, Stachniss C, Burgard W (2015) Autonomous robot navigation in highly populated pedestrian zones. *Field Robot* 32(4):565–589
- Lei Z, Zhou H, Hu W, Liu GP, Guan S, Feng X (2022) Toward a web-based digital twin thermal power plant. *IEEE Trans Indust Inform* 18(3):1716–1725
- Liyanaage D, Pathum H, Madusanka A, Koggalage R (2021) Development of an automated delivery vehicle. *Engineering And Technology* 6(9):1012–1016
- Marcon E (2023) Three-steer: basic steering behaviors library based on THREE.js
- Mintrom M, Sumartojo S, Kulić D, Tian L, Carreno-Medrano P, Allen A (2022) Robots in public spaces: implications for policy design. *Policy Design Prac* 5:123–139
- Morales Y, Carballo A, Takeuchi E, Aburadani A, Tsubouchi T (2009) Autonomous robot navigation in outdoor cluttered pedestrian walkways. *Field Robot* 26(8):609–635
- Nguyen VT, Jayawardena C, Ardekani I (2018) A navigation model for side-by-side robotic wheelchairs for optimizing social comfort in crossing situations. *Robot Autonom Syst* 100:27–40
- Nishino N, Tsugita R, Chugo D, Yokota S, Muramatsu S, Hashimoto H (2017) Robot navigation according to the characteristics of pedestrian flow. In: Conference of the industrial electronics society (IECON). IEEE
- Pelechano N, Allbeck JM, Kapadia M, Badler NI (2016) Simulating heterogeneous crowds with interactive behaviors. CRC Press
- Ranieri L, Digiesi S, Silvestri B, Roccotelli M (2018) A review of last mile logistics innovations in an externalities cost reduction vision. *Sustainability* 10(3)
- Reynolds CW (1987) Flocks, herds and schools: a distributed behavioral model. *SIGGRAPH Comput Graph*
- Shiomi M, Zanlungo F, Hayashi K, Kanda T (2014) Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model. *Social Robot* 6:443–455
- Sorokin M, Tan J, Liu CK, Ha S (2022) Learning to navigate sidewalks in outdoor environments. *IEEE Robot Autom Lett* 7(2):3906–3913
- Srinivas S, Ramachandiran S, Rajendran S (2022) Autonomous robot-driven deliveries: a review of recent developments and future directions. *Transport Res* 165
- Tian Z, Shi W (2022) Design and implement an enhanced simulator for autonomous delivery robot. In: Connected and autonomous driving (MetroCAD), pp 21–29. IEEE
- Trautman P, Ma J, Murray RM, Krause A (2015) Robot navigation in dense human crowds: Statistical models and experimental studies of human-robot cooperation. *J Robot Res* 34(3):335–356
- Truong J, Zitkovich A, Chernova S, Batra D, Zhang T, Tan J, Yu W (2023) Indoorsim-to-outdoorreal: learning to navigate outdoors without any outdoor experience. arXiv preprint [arXiv:2305.01098](https://arxiv.org/abs/2305.01098)
- Usman M, Haworth B, Faloutsos P, Kapadia M (2021) Simulation-as-a-service: analyzing crowd movements in virtual environments. *Comput Animat Virt Worlds* 32(6):e1990

- Usman M, Lee TC, Moghe R, Zhang X, Faloutsos P, Kapadia M (2020) A social distancing index: evaluating navigational policies on human proximity using crowd simulations. In: ACM motion, interaction and games, pp 1–6
- Usman M, Schaumann D, Haworth B, Kapadia M, Faloutsos P (2019) Joint exploration and analysis of high-dimensional design–occupancy templates. In: ACM motion, interaction and games, pp 1–5
- Weckenborg C, Kieckhäfer K, Müller C, Grunewald M, Spengler TS (2020) Balancing of assembly lines with collaborative robots. *Business Res* 13(1):93–132
- Wen M, Dai Y, Chen T, Zhao C, Zhang J, Wang D (2022) A robust sidewalk navigation method for mobile robots based on sparse semantic point cloud. In: IEEE/RSJ intelligent robots and systems (IROS), pp 7841–7846
- Wen M, Zhang J, Chen T, Peng G, Chia T, Ma Y (2022) Vision based sidewalk navigation for last-mile delivery robot. In: Control, automation, robotics and vision (ICARCV), pp 249–254

Sensors and Dynamic Data in Urban Digital Twins

Dynamic Digital Twins: Challenges, Perspectives and Practical Implementation from a City's Perspective



Rico Richter , Frank Knospe, Matthias Trapp , and Jürgen Döllner 

Abstract Digital twins that serve as virtual representations of real-world objects and structures, are used in various applications for urban environments. Challenges for creating and maintaining digital twins involve data acquisition, fusion of heterogeneous data types, AI-based data analysis, and the integration into existing applications and workflows. In this paper, we present the concept and implementation of a dynamic digital twin from a city's perspective. The concept avoids explicit modeling to simplify the creation of a comprehensive data basis that can be easily updated frequently. 3D point clouds with semantics are used as representations for static objects and structures, such as buildings, infrastructure and subsurface structures. Dynamic aspects are represented through a time series of sensor data to enable real-time monitoring and change detection applications. A centralized data repository for applications, such as infrastructure monitoring, condition assessment, and inventory management represent a basis to support decisions. We present typical use cases and challenges from the perspective of a city and how the dynamic digital twins can create significant added value.

Keywords Digital twin · Mobile mapping · 3D point cloud · Georadar · Smart city

This article was selected based on the results of a double-blind review of the full paper.

R. Richter (✉)

Digital Engineering Faculty, University of Potsdam, Potsdam, Germany
e-mail: rico.richter.1@uni-potsdam.de

F. Knospe

Amt für Geoinformation, Vermessung und Kataster, Lindenallee 10, 45127 Essen, Germany
e-mail: frank.knospe@amt62.essen.de

M. Trapp · J. Döllner

Digital Engineering Faculty, University of Potsdam, Hasso Plattner Institute, Potsdam, Germany
e-mail: trapp@hpi.uni-potsdam.de

J. Döllner

e-mail: doellner@uni-potsdam.de

1 Introduction

Virtual representations of real-world objects and structures are essential to implement digital workflows for applications, such as infrastructure planning and management, environmental monitoring and sustainability, emergency response, and disaster management (Ferré-Bigorra et al. 2022). In this context, the concept of digital twins is well established today describing virtual replicas of physical objects, systems, or environments that incorporate real-time and historical data to create a digital representation of the physical world.

In the context of urban environments and cities, digital twins are being used to manage infrastructures (e.g., roads, transportation, energy, and water networks), support planning and decision making (Alva et al. 2022), as well as to provide up-to-date data for simulations (Broo and Schooling 2021; Schraven et al. 2011). The concept of digital twins is well presented and discussed for *Smart Cities* (Shahat et al. 2021; Wan et al. 2019) and *Industry 4.0* (Qi and Tao 2018). Selected examples of cities who presented digital twins in literature include Zurich (Schrotter and Hürzeler 2020), Vienna (Lehner and Dorffner 2020), and Helsinki (Hämäläinen 2021).

The derivation and generation of digital twins for new or planned structures is state-of-the-art and typically a straightforward task. The required data and specifications for the resulting virtual replica can be incorporated into the design, construction and maintenance process, and the necessary sensors and data collection methods can be implemented from scratch. Usually, explicitly modeled objects (e.g., BIM, IFC) and citywide 3D representations (e.g., CityGML models, textured meshes) are available to provide spatial data access.

However, creating digital twins for existing, large-scale environments, e.g., entire cities, is a challenging task. A comprehensive overview and rating of challenges based on an expert survey is given in Lei et al. (2023). Creating digital twins requires existing buildings, objects, infrastructure, and systems to be mapped into a virtual representation, e.g., using remote sensing technology (Volk et al. 2014), and (mobile) platforms equipped with sensors for monitoring tasks (Rao et al. 2022). Most digital twin applications require the combination of *static representations*, i.e., geometry including semantics and appearance, and *dynamic representations*, i.e., behavior and sensor data. Examples are movement monitoring of buildings, vegetation monitoring and condition assessment of the infrastructure. A fusion of different, heterogeneous sensor datasets (Volland and Asche 2017), captured at different frequencies (e.g., annually, over daily, up to every minute), is required and provides up-to-date information on-demand for a variety of applications if needed. Such a digital twin represents changes and behaviors with respect to various dynamic aspects and is therefore referred to as a *dynamic digital twin*.

In this paper, we present and discuss significant challenges, ranging from data acquisition, integration, analysis, and management to generate and maintain dynamic digital twins for entire cities. We show how 3D point clouds with semantics can be used for object representation to avoid time-consuming and error-prone derivation of explicitly modeled objects (e.g., 3D models). A fusion of such representation with

time-series of sensor data can be used to create a universal data basis for an ecosystem of applications. We present challenges, opportunities, and practical applications of dynamic digital twins, highlighting their potential to transform the way we cope with big data, enable real-time processing, and provide a platform for data-driven decision making. The expected potential and benefits are immense, including increased efficiency, sustainability, and resilience.

1.1 Digital Twins

In the scope of this paper, we refer to digital twins as a virtual representation of a physical object, system, or process that can be synchronized with its real-world counterpart in real-time or at various intervals (Barricelli et al. 2019; Jones et al. 2020).

Spatial digital twins are a specific type of digital twins that focus on the spatial aspects and characteristics of the physical object, system, or process. It is a 3D digital replica of the physical space, which is created by integrating data from various sources (e.g., sensors, cameras, and other devices), to provide a comprehensive coverage of space. In addition, spatial digital twins can include algorithms and analytics capabilities to observe and understand the behavior of the real-world objects and structures, as well as predict, optimize, and improve it.

The main challenge of constructing spatial digital twins is the immense amount of data required to describe, for example, an entire city, when using standard geometrical representation approaches; taking further into account the additional value added by automated semantic description of sensor data, due to the availability of high performance and cloud computing capabilities to process and analyze data. However, this requires a uniform data basis that is generated by heterogeneous sensor systems in the scale from earth observation to georadar. Since all semantic and dynamic objects have the spatial coordinate as an overall reference, this serves as a basis of data management. 3D point clouds obtained by different sensory systems already have the necessary sufficient density that enables a detailed and photorealistic visual representation.

Spatio-temporal digital twins represent data from different points in time (e.g., historical data). They can include and use real-time data from a number of sensors to capture and represent most of the spatio-temporal behavior of the physical system or object. Main objectives of spatio-temporal digital twins are to enable users to monitor and understand the behavior of the physical system to identify changes and anomalies (Minerva and Crespi 2021).

The behavior and representation of urban environments are highly dynamic. Hence, they require digital twins representing this by spatial, temporal, and semantic dimensions (Stoter et al. 2021). To highlight the importance of the dynamic characteristic, we introduce the term *dynamic digital twins*. These dynamic digital twins combine static, semantically rich models with a variety of sensors to cover even dynamic aspects to provide the ability to react to changes in the real-world system

in real-time and adapt accordingly. The dynamic aspect is used to classify a normal behavior or a movement tolerance of objects (e.g., bridge) or parts of objects (e.g., bearings of a bridge). This is especially important in the context of automated monitoring. This improves the performance, efficiency, and safety of physical systems and processes.

In the near future, this will comprehensively enable immersion, automated real-time analysis, and decision-making based on spatial digital twins—from space to infrastructural building data. With respect to this, the following requirements for dynamic digital twins can be identified (not limited to):

1. Virtual representation of any size physical objects and systems.
2. Synchronization with the real-world in real-time or at regular frequencies.
3. Efficient creation using (measurement) data, and algorithms (AI, modeling, statistics).
4. Ability to observe, understand, predict, optimize, and improve behavior of objects.
5. Dynamic and semantic description as fundamental part of a statistic model.
6. Real-time adaptability to changes in the physical system.
7. Integration of sensor data and real-time information for improved performance, efficiency, and safety.
8. Standardized and open interfaces for integration into a large ecosystem of applications.

Digital twins will continue to be researched and developed in the coming years. It is already evident that digital representations, functionalities, models, and interfaces must be considered as different aspects of the digital twin, rather than a monolithic data model.

1.2 Data Challenges

Major challenges for implementing digital twins are related to data (Lei et al. 2023). To use dynamic digital twins from a city's perspective in practice, it is essential to ensure that the data is accurate, reliable, and up-to-date. This requires the fusion of different data categories, such as 3D point clouds, images, videos, radar data, and other sensor data to support a possibly complete coverage of the physical environment. The focus of this paper is on challenges related to data. With respect to this, the following main challenges can be identified:

- **Data acquisition:** The data acquisition must be performed in regular time intervals with a high degree of automation and at different geometric scales. This may include, for example, the acquisition of road infrastructure and underground, terrain, vegetation and city furniture as well as the interior of individual constructions and buildings.

- **Sensor fusion:** Building a digital twin requires fusing data from different sensors. This is a crucial task, because sensors capture the environment from different perspectives (e.g., aerial or terrestrial), with different spatial characteristics (e.g., 2D or 3D), sampling rates, and accuracies. Spatial and temporal characteristics of data from each sensor must be aligned and synchronized to provide a consistent representation of the captured environment. A precise calibration of data acquisition systems (e.g., LiDAR and camera systems) and the accurate estimation of absolute and relative accuracy is necessary to ensure that the incoming data can be fused with historic and future data. Analysis methods that incorporate multiple data sources also require a reliable sensor fusion.
- **Data volume:** The size of the captured data is another challenge, especially for storage and compute infrastructure, which can be several hundred terabytes of data for an entire city. Hence, concepts and techniques for compression, level-of-detail, and change detection (Kharroubi et al. 2022) are required.
- **Date integration:** To enable application of existing workflows and systems, seamless data integration between the digital twin and systems used to manage the infrastructure is required. For example, the digital twin for a structure (e.g., bridge) should be integrated with the structure management system to enable real-time monitoring and control of the structure's performance. This requires a careful consideration of the data formats, protocols, and interfaces used by different systems and the need for standardization.
- **Compute infrastructure:** A compute infrastructure must be designed to handle massive data volumes and perform complex simulation and analysis tasks (Discher et al. 2019). Such infrastructure should be capable of processing real-time data streams to enable timely decision-making and response. Furthermore, the infrastructure must support advanced algorithms and machine learning techniques to extract meaningful insights from the data. This includes the ability to train and deploy machine learning models for object detection, predictive analytics, and anomaly detection.
- **Applicability to different scales:** Applicability at different geometric and spatial scales is a crucial aspect for digital twins. It should be able to handle objects and phenomena of various sizes, ranging from small-scale elements (e.g., cracks, potholes, and underground structures), to larger entities, such as buildings, neighborhoods, and even entire cities.
- **Automated analytics:** Classification, object detection, and semantic partitioning of the data are essential to enable dedicated and application-specific tasks. Therefore, automated analytics capabilities must be implemented and integrated into the digital twin (Xie et al. 2020).
- **Costs and investments:** These challenges are also a major consideration when implementing digital twins for cities. The creation of digital twins requires a significant investment for data acquisition (e.g., by service provider or own hardware), data management systems, and analytics tools. The cost of implementing digital twins varies depending on the scale of the infrastructure being monitored, the complexity of the sensors and models used, and the level-of-integration

required for existing systems. Therefore, careful cost–benefit analysis is required to determine the economic feasibility of implementing digital twins for cities.

The paper is structured as follows. In Sect. 2, we present our concept for a dynamic digital twin. The respective fundamental data is discussed in Sect. 3. Implemented use cases are presented in Sect. 4. Section 5 contains the conclusions and future work.

2 Concept of Dynamic Digital Twins for Cities

In this chapter, we present the concept and the continuous development of a dynamic digital twin exemplified by the city of Essen. The development of a digital twin for Essen began with the establishment of a geospatial data infrastructure back in 2001. The first comprehensive, semantically-enriched 3D representations at LOD2 were created by manual modeling approaches starting in 2006. It became apparent that this kind of spatial and static digital twin has limitations with respect to level-of-detail, frequent update possibilities, and costs. These limitations depend in particular on the available and used sensor technology (e.g., cost–benefit-aspect for business management action). In 2016, the City of Essen purchased its own multi-sensor vehicle to capture mobile mapping data regularly and cost-effectively for the entire city (Fig. 1). This was the starting point to establish a spatio-temporal digital twin. In 2020, IoT-driven sensor technology was established to monitor the environment based on sensor data and integrate behavioral and dynamic aspects to create the basis for a dynamic digital twin.



Fig. 1 Two mobile mapping vehicles of the city of Essen with image, video, LiDAR, and ground penetrating radar sensors. The left image shows a Trimble MX8 with GSSI ground penetrating radar horn antennas and the right image an MX9 with a Kontür radar array system

2.1 *Static Semantic Representation*

The static semantic representation of a city describes where and what is part of an infrastructure as well as its interactions with the environment. The data and visualizations represent a single time stamp and are available “offline”. For example, users can explore the spatial appearance, perform measurements, or count object instances based on the available information. The input data of different (surveying) sensors are fused via position description and classified via acquisition, evaluation, or analysis approaches. This results in a semantic digital twin that can describe objects, such as roads, as well as local features (e.g., potholes). However, in the case of modeling methods (such as Building Information Modeling (BIM)), without considering the real-world (Stojanovic et al. 2018b), models may differ from the existing as-built infrastructure. In this case, a comparison between the BIM planning model and the measured structure (change monitoring or change detection) was performed (Stojanovic et al. 2018a). Change detection is required to compare two time periods and to describe “spatial changes”. These can in turn be described in terms of rates-of-change (e.g., wear) using time-series analyses. Examples are clearance profile analysis (e.g., with a semantic point cloud), road marking evaluation, and vegetation monitoring.

2.2 *Dynamic Representation*

The dynamic representation describes how the infrastructure behaves. The properties of an infrastructure in combination with the semantic representation help to consider the range of possible characteristics in a differentiated way (As-Is). For example, a gauge on a stream describes the water level. If this data is recorded over a period of several years, the seasonal behavior pattern can be described on average or the extremes (Chatterjee and Ahmed 2022). Further, trend observations and forecasts (prediction), such as increase of flood peaks, can be derived and fed into planning processes or risk management. The database input can range from statistical values (e.g., energy efficiency from a BIM class), to the real consumption values (annual energy consumption), to the real-time values obtained from an IoT sensor system. Figure 2 shows how semantic and dynamic characteristics are described. Change detection occurs in static scenarios through new data acquisition, while in dynamic scenarios it involves analyzing time-series data and comparing it to a baseline measurement (outliers and seasonal effects need to be considered).

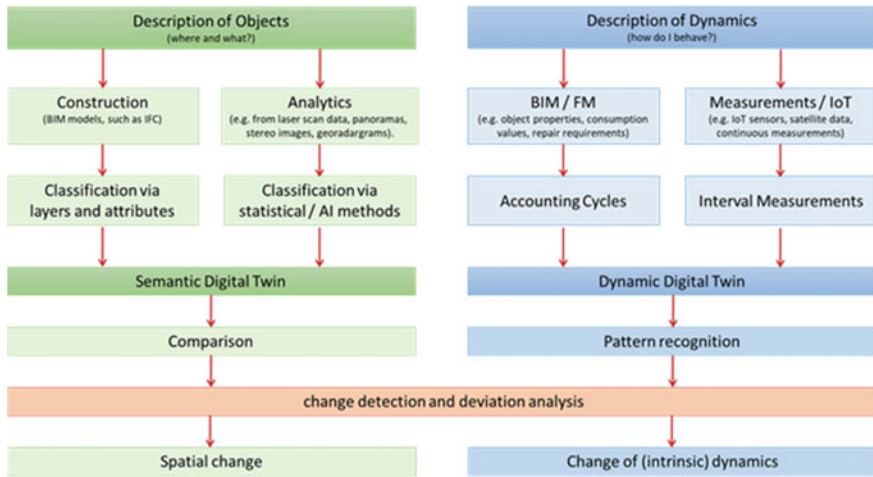


Fig. 2 Combination of semantic (green) and dynamic (blue) objects description for dynamic digital twins by the example of the city of Essen

3 Data for Dynamic Digital Twins

In this section, we present the relevant data collected to build dynamic digital twins as presented in Sect. 2.

3.1 3D Point Clouds

3D point clouds play the most important role in the creation of digital twins in Essen. A 3D point cloud is a collection of millions of 3D coordinates captured using LiDAR, photogrammetry, or ground-penetrating radar. Originally, this data was processed using software to create a digital object model of the scanned environment (e.g., building geometry). Today, the density of 3D point clouds is sufficiently high to omit explicit object modeling for many applications (Richter and Döllner 2014), such as visualization, inspection, measurements, object detection, change detection, or planning. Direct use of 3D point clouds in the context of digital twins offers a variety of benefits:

- **Accuracy:** Point clouds offer high accuracy and support level-of-detail, allowing for very precise digital twins to be created.
- **Completeness:** Point clouds capture all visible features of an object or system, enabling digital twins to be created with a high level-of-coverage.
- **Efficiency:** Point clouds can be created efficiently, allowing for digital twins to be synthesized faster than using traditional methods.



Fig. 3 Example of a 3D cloud captured with a mobile mapping vehicle. The gray-scale values represent reflected intensity

- **Integration:** Point clouds can easily be integrated into existing 3D modeling software, allowing for digital twins to be created efficiently.
- **Flexibility:** Point clouds can be used for a variety of applications, including the creation of digital twins, making digital twins flexible and adaptable for different purposes.

Until 2020, the city of Essen has used 3D modeling software exclusively for (manual/semi-automatic) evaluation of point clouds for object structure generation. This approach is systematically replaced by semantic classification and high-performance computing using statistical and AI models (Bello et al. 2020). This is feasible, as mobile laser scanners can now capture up to 1500 pts/m². The result is a sufficient visual representation of the physical environment (Fig. 3). The semantics attached to each point allow for filtering of objects/parts without having to model them as polygonal bodies or surfaces. Thus, the semantic digital twin is a virtual model of all real objects in the scanning area. While these scanning methods were previously used primarily for infrastructure, such as buildings, bridges, and roads, they are applicable to all physical objects today; ranging from trees to road markings, fences, and high-voltage wires.

3.2 *Raster-Image Data*

Mobile mapping vehicles are usually equipped with multiple camera systems that generate single images, videos, or panoramic images (Glira et al. 2022; Puente et al. 2013). All these raster-images are georeferenced and can be used directly for visualization and visual inspection tasks. Raster-image data can be used to add color information to 3D point clouds, providing an additional feature-layer for analysis.

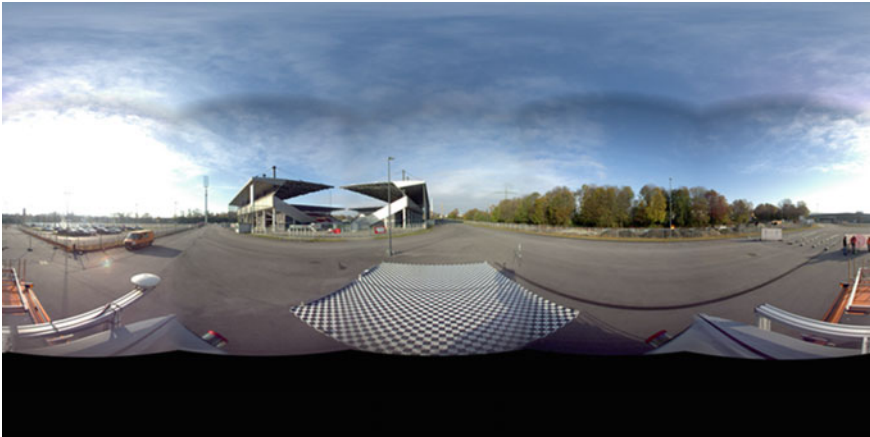


Fig. 4 Checkered carpet used for distortion registration of panoramic images

Information from 3D point clouds can be transferred to image space, allowing images to be enriched with depth and object information. A particular challenge is posed by 360° panoramic images, since the respective distortion must be accounted for the unwarping of coordinates. For this purpose, the city of Essen uses a checkered PVC carpet (Fig. 4). By driving over during acquisition, a transformation matrix is computed for any panoramic camera system. Additionally, deviations between laser scan and image can be calibrated in the vehicle close range.

In a survey of all roads in Essen, 240,000 panoramic images were acquired (1600 km of roads, 4500 km of lanes). In addition to this, the transformed close-up area in front of the vehicle is used to detect road damage (e.g., fractures, cracks, net cracks) as well as fixtures (e.g., gas slides, run-ins), and road markings. Due to the sometimes rapid changes in exposure, a shutter camera is used for detecting damages. Every 2 m, a 3 m² area in front of the vehicle is captured (2,250,000 shutter images). The resolution and exposure dynamics represent an optimal foundation for an AI-based processing. Finally, the panoramic and shutter images are matched to improve the results.

3.3 *Radar Data*

Ground penetrating radar (GPR) is an effective measurement method from geophysics that detects layer thicknesses and inhomogeneities in the road surface as well as in the subsurface non-destructively (Bell and Sengupta 2017; Bezerra and Benedetto 2019; Park et al. 2018). It emits high-frequency electromagnetic pulses that penetrate the ground and record reflections from objects below the surface. The measuring system consists of a transmitter and receiver unit. The transmitting

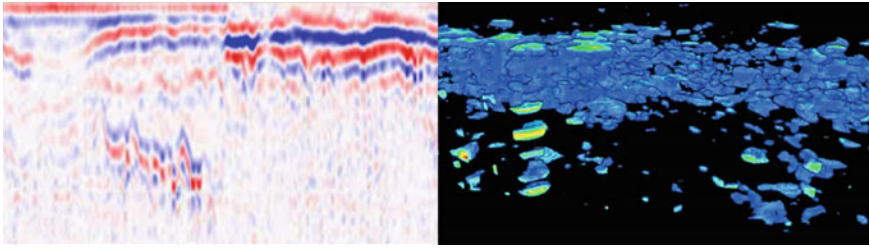


Fig. 5 Illustration of GPR data from mobile mapping vehicles captured as layer (left) or 3D point cloud (right)

antenna emits electromagnetic pulses controlled by pulse generators. Layer thicknesses with different electromagnetic properties (from 400 MHz to 2 GHz) and small-scale structures, such as cavities, reflect, or refract the waves emitted by the transmitting antenna. GPR data can be used to investigate subsurface structures and materials (Benedetto and Ciampoli 2019). GPR is usually used to synthesize a model of the subsurface in the form of cross-sectional images (e.g., with horn antennas), i.e., as a layer-by-layer structure (Loizos et al. 2016) of the subsurface (Fig. 5—left), or 3D point clouds, i.e., as a grid structure with density information (Reusch et al. 2019) (Fig. 5—right).

3.4 *Satellite Radar Data*

Satellites equipped with radar sensors such as TerraSAR-X, can be used to detect small changes and movements (a few millimeters based on an InSAR analysis) for discrete surface points. This data can be mapped to 3D point clouds to implement an object monitoring (e.g., for roofs, bridges, roads). It is used to describe and understand the behavior and typical dynamics of objects over time within digital twins. The objective is to detect a normal motion behavior and to describe deviations as discontinuities. Potential problems can be detected faster based on the analysis and topographic measurements can be performed more efficiently, e.g., on bridges that may pose a risk (Fig. 6).

3.5 *Sensor Data*

The term Internet of Things (IoT) refers to physical information sources that are part of a (global) information network, such as sensors and social networks. They represent the basis for the dynamics within a digital twin. In the surveying and geoinformation sector, IoT sensors are used to describe and understand the lifecycle of municipal infrastructures (e.g., trees, roads, buildings, and canals). As the location

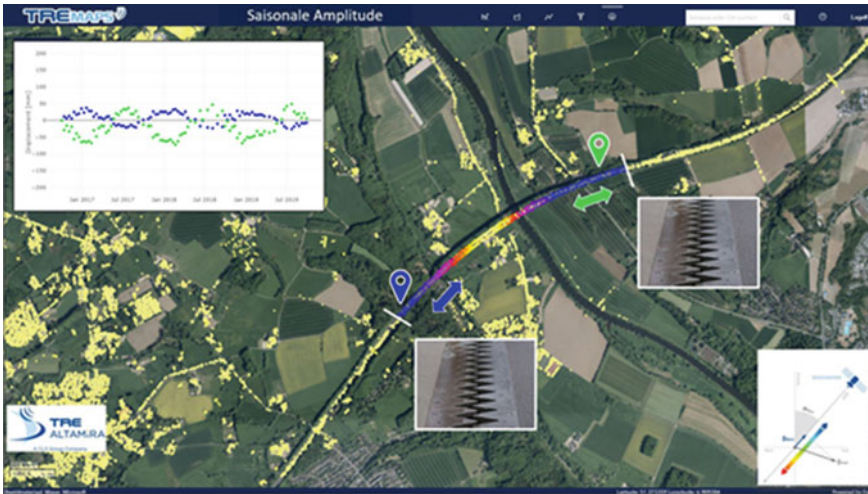


Fig. 6 Map application showing seasonality of bridge movements due to thermal dilatation (Walter and Anderssohn 2023). Movement of discrete surface points is colored by strength of movement

of the sensor is known (point coordinate), it can be spatially assigned to the semantic part of a 3D point cloud.

The development of low-cost sensors will be the key for the roll-out of applications. Topics such as bridge monitoring, traffic control, road condition assessment, or energy optimization are the economic fields in asset management.

3.6 Data Fusion

Digital twins offer a unique ability to represent various, sometimes heterogeneous, information in a unified form, thus generating knowledge from data. Although they are more than just data, the central challenge in implementing digital twins lies in effective data management. Additionally, accurate semantic and dynamic descriptions of their real-world counterparts require suitable algorithms. In Essen, the different components of digital twins do not function as standalone models, but are gradually being integrated to form a holistic, interconnected “digital ecosystem”. This indicates the significant economic, social, and ecological potentials of Essen’s digital twin. For instance, geodata modeling approaches and IoT sensors are already contributing to the analysis and optimization of municipal infrastructure in Essen using data acquired in real-time. For example, a spatial observation with control models, called Environmentally sensitive mobility and road infrastructure management, is currently being created with the three digital twins ground movement, road infrastructure and mobility. The data fusion is based on the spatial reference by linking point and surface information.

4 Monitoring Infrastructure Applications

In recent years, the concept of a digital twin of Essen has gained substantial momentum, driven by advancements in mobile sensing, IoT, and initiatives such as the Copernicus Satellite Program of the European Union. These developments have facilitated the widespread adoption of 3D scanning technologies, enabling the comprehensive acquisition of municipal infrastructures. Building components and systems are now equipped with sensors that can capture data in real-time.

The city of Essen currently uses dynamic digital twins in various fields. In the context of construction, it is used as a virtual representation of structures, such as buildings, bridges, roads, or other infrastructure projects for maintenance, inspection, and monitoring tasks. In addition, virtual worlds based on digital twins are emerging in the fields of urban green spaces, mobility, and air pollution for their description, analysis, and control. The dynamic digital twin is created using a multitude of data, sensors, and models that continuously monitor or simulate the state and dynamics of the physical object in the city. The digital transformation of infrastructure diagnostics enables the past to be tracked, the present to be controlled, and the future to be predicted. This facilitates fact-based decision-making to improve efficiency, minimize risks, and reduce maintenance costs. The following sections describe specific applications as part of the dynamic digital twin. These applications use the data presented in Sect. 3 and were chosen because they offer the greatest benefits in terms of time, cost, and social impact for the city of Essen.

4.1 Infrastructure Database for Road Environments by Semantic Digital Twin

Mobile measurement systems are used to collect road information using LiDAR scans, raster image data, and GPR data, which are uniformly stored as an attributed 3D point cloud. Both, the existing road-space inventory (e.g., traffic signals, traffic signs, markings, and vegetation) and the condition and substructure of the road are recorded. This data is evaluated by statistical analysis and AI methods and described semantically. AI models for point clouds use deep learning methods that operate directly on the point cloud (Bello et al. 2020). These models require training data in the form of labeled point clouds. The resulting semantic point cloud can be used as a “digital twin”, i.e., a digital representation of the road together with the surrounding road environment. Considering the amount of work, geometric modeling is dispensed or only created on specific project requests. Based on the digital twin, AI-based analysis methods can be developed and applied to enable regular and automated condition assessment of roads and their substructures (Mattes et al. 2023).

4.2 Road Condition Assessment by Semantic Digital Twin

Since 2018, the condition of road surfaces has been recorded and assessed (Giannopoulos et al. 2016). For this purpose, all roads in Essen are captured using mobile mapping and GPR at specified intervals. The transverse and longitudinal evenness is recorded using LiDAR data. Surface damages are recorded by different digital cameras (panoramic and shutter images). Physical state-values are computed from the data of the measuring systems (Hugenschmidt et al. 2014). The serviceability value is composed of the longitudinal and transverse evenness characteristics and primarily describes driving safety and ride comfort. In addition to the longitudinal and transverse evenness characteristics, the substance value also includes the substance characteristics (surface damage), which reflects the condition of a road surface. Finally, the total value is formed from the maximum of the use and substance value (Fig. 7).

In general, only the condition of the road surface is recorded in the existing regulations. The city of Essen additionally checks the road surface (layer structure, faults, cavities) and includes this within the assessment. The values determined are thus only an indicator of the renewal worthiness of a road section from the road surface and deeper damage. A poor road condition, and thus a need for action, can be indicated as a warning value and the threshold value at which the initiation of traffic-restricting or structural measures to maintain the road section must be examined.

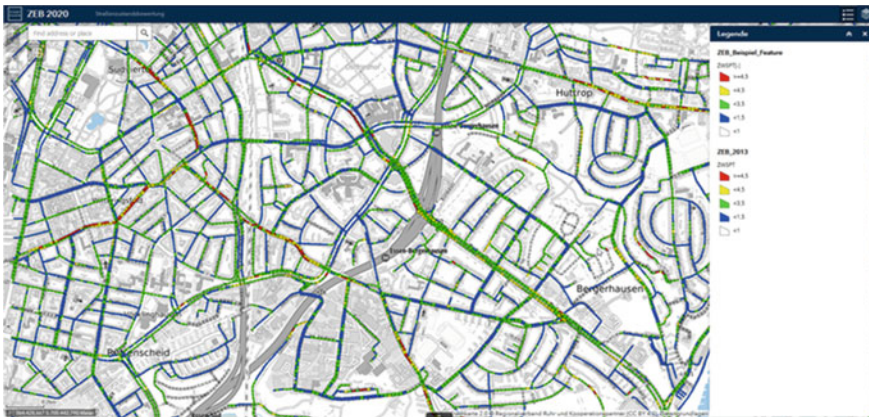


Fig. 7 Map application showing results of road condition assessment for road sections colored by condition categories

4.3 Ground Movement Monitoring and BIMSAR by Dynamic Digital Twin

Ground movements can pose a risk to the population and infrastructure. Therefore, movement processes in the area of underground use (e.g., induced by mining or geothermal processes) of relevant infrastructure (e.g., dikes, bridges) have long been monitored using terrestrial surveying methods (e.g., leveling, GPS). Modern remote sensing methods, such as satellite-based Interferometric Synthetic-Aperture Radar (InSAR), enable cost-effective measurements of ground motions with high precision and are available as a point cloud. Since 2017, the city of Essen has been using satellite-based SAR interferometry for a dynamic image of the earth’s surface. High-resolution TerraSAR-X (TSX) data are regularly acquired in the descending orbit of the satellite. The TerraSAR-X satellite operates in the X-band (wavelength = 3.11 cm). The ground resolution of the TSX data used, acquired in StripMap (SM) mode, is approximately 3 m². The repetition rate of the data is nominally 11 days. Figure 8 shows the map of the motion velocity for a defined area.

The project title BIMSAR already indicates that building information modeling and radar interferometry plays an essential role. However, in this case, it is not used here to detect surface movements, but to detect movements of buildings and traffic infrastructures, since unfavorable movements can affect the statics of buildings

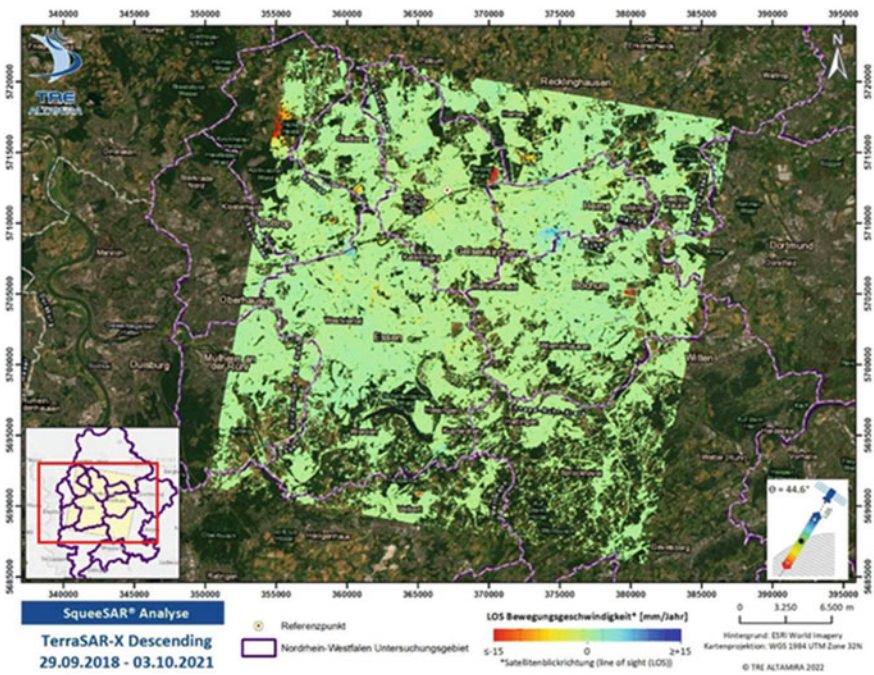


Fig. 8 Map of motion velocity (in mm per year) from TerraSAR-X descending data

and represent a risk. Further, dedicated building information derived from LiDAR data is applied for the analysis. Since the detection of building movements is a complex task, AI methods are used as a third important project component. With these three components, the BIMSAR project develops a cloud-based assistance system for monitoring buildings and traffic infrastructures. The data of the semantic digital twin is being used in Essen as in-situ data for SAR data analysis. The basis for the dynamic analysis are the so-called Persistent Scatterers (PS). PS refers to image regions (often single pixels) as back-scatterers, whose scattering properties vary little over time in a stack of coregistered radar images and can be well retrieved and analyzed over time, i.e., by mapping of PS motions to semantic infrastructure data from the point cloud. Especially for this step, AI methods assign registered movements more precisely to building components and better assess the nature of the change in terms of relevance. The points that are assigned to different parts of the structure are grouped and classified. This facilitates the assessment whether they are cyclically recurring, natural movements, or are continuous changes (trend). If a movement actually represents a danger for the observed object, experts can determine further actions.

4.4 Tree Monitoring with TreeCop by Dynamic Digital Twin

TreeCop (Synonym for the Copernicus satellite program and drought stress on trees) is supporting the forestry services of the city of Essen to assess the water supply of urban trees, identify priorities in site optimization, and watering trees dependent on artificial irrigation at an early stage and in a targeted manner in the future (Bell and Sengupta 2017). For this purpose, images from the European satellite program “Copernicus” are used for the dynamic digital twin. Every three days, a satellite records raster-data with a grid size of 10 m². By evaluating the color channels of the images, information about the chlorophyll content of the leaves and thus an indicator of tree health is obtained. Since the spectral signature of the tree canopy surface changes depending on its weather-related water supply, the satellite images can be used to identify the effects of drought stress, for large urban trees even with individual tree reference. To avoid drought stress, reliable data on current soil water supply is required early on to supplement weather forecast models. Therefore, sensor packages for soil moisture measurement have been installed in the root zone of selected urban trees at over 200 locations. The sensors are so-called vacuum meters, which measure the force required in a ceramic body to create the necessary negative pressure that the root must generate in order to extract water from the soil and absorb it with the appropriate nutrient solution. On each city tree, three such sensors measure this suction force at depths of 30, 60, and 100 cm. The sensors continuously transmit to a central database using the LoRaWAN radio data network. Based on the data interpretation, reliable statements about the water supply of the city tree locations, which are not equipped with sensors, are inferred. A monitoring dashboard is used to provide the data (Fig. 9).



Fig. 9 Monitoring dashboard for city trees

4.5 Environmentally and Socially Compatible Traffic Management by Dynamic Digital Twin

In the project “Digitization and environmentally compatible control of road traffic” in Essen, innovative technologies are being employed to digitize and optimize urban traffic. The project involves the use of sensor technology, cameras, and environmental sensors to collect real-time data on vehicle types, pedestrian movements, and environmental parameters, such as air quality. The data is managed and processed using cloud-based data management systems, a data hub, and high-performance computing. AI and deep learning techniques are utilized for data analysis and classification of vehicles (Fig. 10). Additionally, satellite data from the Sentinel 5P satellite is used for air quality monitoring. These technological advancements enable dynamic traffic analysis, environmentally sensitive traffic management, and data-driven decision-making for a more efficient and sustainable transport system in Essen.

4.6 Technological Challenges from a City’s Perspective

For summarization, the following technological challenges should be addressed when implementing dynamic digital twins:

- **Data privacy and data security:** A key issue in connection with digital twins is data privacy and data security. Digital twins collect and process large amounts of data that may contain personal information. In the simplest case, faces and license plates in images are rendered unrecognizable at runtime. At the same time, it is important to ensure that the data is protected from hacking and other threats,



Fig. 10 Traffic counting and visualization dashboard

especially if a municipal geodata infrastructure is integrated with external cloud applications.

- **Scalability of data storage and analysis platform (DataHub):** A further key issue in using digital twins is scalability on demand. This requires appropriate infrastructure and resources. Hardware resources to effectively process, analyze, and visualize data on a scale of several terabytes are generally not available within a local government. With the decision to build dynamic digital twins, the topic of a central DataHub should also be considered, if a powerful ecosystem of digital twins is to be created. A flexible and configurable ecosystem for cloud computing and storage is required.
- **Interoperability:** Another issue related to digital twins is interoperability. Municipalities should ensure that their digital twins are compatible with other (existing) systems and applications. This requires open standards and interfaces to ensure that data can be exchanged between these.
- **AI-based analytics:** AI methods for image, video, 3D, and sensor data are essential to achieve a high degree of automation for evaluation, classification, and prediction. This usually requires comprehensive training data, i.e., repositories of high quality training data should be built and maintained on a continuous basis.
- **Change detection and compression:** Change detection is important for updating data at regular time intervals. Further, historical data is important for applications such as predictive maintenance, however, representing all acquired datasets would result in huge amounts of data. Usually, only small parts of the urban area change, thus, storing only the changed parts would significantly reduce the required storage capacities.
- **Interactive visualization:** Scalable real-time rendering and visualization of dynamic digital twins and the corresponding multi-dimensional data layers pose significant challenges, especially when using web environments and mobile devices in the field. Optimizing rendering performance and developing intuitive

user interfaces are key to enable efficient exploration and analysis for a broad range of users, including those who are not tech-savvy.

- **Real-time simulation:** A further challenge related to dynamic digital twins is the need for supporting real-time simulations. These are important to ensure that the digital twin is able to mirror the behavior of the real-world object in real-time. Considering, for example, topics such as traffic control and regulatory situational awareness: if simulations are not performed in real-time, important data or events can be lost or delayed, leading to inaccurate forecasts and decisions. With respect to this, an important problem is the need to transfer and process large amounts of data from a variety of sensors and other sources (e.g., geodata infrastructure or social networks) to the database. This requires a powerful (transmission) infrastructure that is generally not yet in place (cf. broadband and 5G rollout) and specialized tools and technologies.

5 Conclusions and Future Work

In conclusion, the concept of digital twins has proven to be valuable in various domains, including infrastructure planning, environmental monitoring, and emergency response. While digital twins have been successfully implemented for new objects or structures, building digital twins for existing, large-scale infrastructures, such as cities, presents significant challenges. The integration of static and dynamic representations, combining geometry, semantics, appearance, behavior, and sensor data, is crucial for building dynamic digital twins that accurately represent the real world.

This paper has highlighted the major challenges involved in developing dynamic digital twins for entire cities, including data acquisition, integration, analysis, and management. We have discussed and demonstrated how 3D point clouds with additional semantics can serve as an efficient and reliable object representation, avoiding the time-consuming and error-prone process of explicit modeling. Nevertheless, semantic 3D point clouds allow the explicit derivation of geometry when applications require it (e.g. flood simulations). The fusion of static representations with time-series of sensor data enables the creation of a universal data basis that is capable of supporting a wide range of applications. Furthermore, we have explored the potential of dynamic digital twins to transform the way big data is handled, enabling real-time processing and facilitating data-driven decision making. By providing up-to-date information and insights, dynamic digital twins have the potential to significantly improve efficiency, sustainability, and resilience in urban environments.

With respect to the future, there are important areas of focus that require further exploration, both on conceptual and technical levels. These include, among others, the assurance of frequent data updates and effective handling of big data, and the incorporation of change detection mechanisms and timestamped data updates. Additionally, the choice of cloud providers and adherence to data governance frameworks,

such as GaiaX, will play a crucial role in ensuring data sovereignty and privacy. Interconnectivity between dynamic digital twins and the integration of various systems and infrastructures will also be key for a comprehensive future city management.

Overall, dynamic digital twins offer immense potential and benefits, paving the way for a new era of data-driven urban planning, management, and decision-making. By addressing the challenges outlined in this paper and leveraging the opportunities they present, we can unlock the full potential of dynamic digital twins and create smarter, more sustainable, and resilient cities.

Acknowledgements We thank the anonymous reviewers for their valuable feedback. This work was partially funded by the Bundesministerium für Digitales und Verkehr (BMDV) through grant 19F2210 (TWIN4ROAD) and the Bundesministerium für Bildung und Forschung (BMBF) through grant 01IS22062 (AI research group FFS-AI).

References

- Alva P, Biljecki F, Stouffs R (2022) Use cases for district-scale urban digital twins. *Int Arch Photogramm Remote Sens Spatial Inf Sci*, XLVIII-4/W4:5–12
- Barricelli BR, Casiraghi E, Fogli D (2019) A survey on digital twin: definitions, characteristics, applications, and design implications. *IEEE Access* 7:167653–167671
- Bell S, Sengupta A (2017) Ground-penetrating radar applications in transportation infrastructure. In: *Ground penetrating radar theory and applications*. Elsevier, pp. 457–488
- Bello SA, Yu S, Wang C, Adam JM, Li J (2020) Review: deep learning on 3d point clouds. *Remote Sens* 12(11):1729
- Benedetto A, Bianchini Ciampoli L (2019) Advances in ground penetrating radar surveys for road diagnostics. *Geosci* 9(6):258
- Bezerra J, Benedetto A (eds) (2019) *GPR in practice: a user's guide for road and bridge maintenance*. Springer
- Broo DG, Schooling J (2021) Digital twins in infrastructure: definitions, current practices, challenges and strategies. *Int J Construct Manag*
- Chatterjee A, Ahmed BS (2022) IoT anomaly detection methods and applications: a survey. *IoT 19*
- Discher S, Richter R, Trapp M, Döllner J (2019) Service-oriented processing and analysis of massive point clouds in geoinformation management. In: *Service oriented mapping: changing paradigm in map production and geoinformation management*. Springer International Publishing, Cham, pp. 43–61
- Ferré-Bigorra J, Casals M, Gangoles M (2022) The adoption of urban digital twins. *Cities* 131
- Giannopoulos A, Loizos A, Plati C (2016) Advances in GPR surveys for road infrastructure monitoring. In: *Pavement and asset management*. Springer, pp 389–415
- Glira P, Weidinger C, Kadiofsky T, Pointner W, Ölsböck K, Zinner C, Doostdar M (2022) 3D mobile mapping of the environment using imaging radar sensors. In: *Radar Conference*, 22
- Hugenschmidt J, Slob E, van der Kruk J, Vereecken H (2014) Monitoring time-lapse changes in the vadose zone using crosshole ground penetrating radar full-waveform inversion. *Water Resour Res* 50(5):4131–4152
- Hämäläinen M (2021) Urban development with dynamic digital twins in helsinki city. *IET Smart Cities* 3(4):201–210
- Jones D, Snider C, Nassehi A, Yon J, Hicks B (2020) Characterising the digital twin: a systematic literature review. *CIRP J Manuf Sci Technol* 29:36–52

- Kharroubi A, Poux F, Ballouch Z, Hajji R, Billen R (2022) Three dimensional change detection using point clouds: a review. *Geomatics* 2(4):457–485
- Lehner H, Dorffner L (2020) Digital geotwin vienna: towards a digital twin city as geodata hub. PFG—J Photogramm, Remote Sens Geoinform Sci 88(1):63–75
- Lei B, Janssen P, Stoter J, Biljecki F (2023) Challenges of urban digital twins: a systematic review and a Delphi expert survey. *Automat Construct* 147
- Loizos A, Plati C, Chryssanthopoulos MK (2016) Evaluation of pavement structures using ground penetrating radar: a review. *J Appl Geophys* 134:114–123
- Mattes P, Richter R, Döllner J (2023) Detecting road damages in mobile mapping point clouds using competitive reconstruction networks. *AGILE GIScience Ser* 4:7
- Minerva R, Crespi N (2021) Digital twins: properties, software frameworks, and application scenarios. *IT Professional* 23(1):51–55
- Park B, Kim J, Lee J, Kang M, An Y (2018) Underground object classification for urban roads using instantaneous phase analysis of ground-penetrating radar (GPR) data. *Remote Sens* 10(9):1417
- Puente I, González-Jorge H, Martínez-Sánchez J, Arias P (2013) Review of mobile mapping and surveying technologies. *Measurement* 46(7):2127–2145
- Qi Q, Tao F (2018) Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. *IEEE Access* 6:3585–3593
- Rao AS, Radanovic M, Liu Y, Hu S, Fang Y, Khoshelham K, Palaniswami M, Ngo T (2022) Real-time monitoring of construction sites: sensors, methods, and applications. *Automat Construct* 136
- Reusch A, Nolte E, Löhnert G (2019) Mapping of near-surface fractures in crystalline rocks using 3D GPR amplitude volumes. *Near Surface Geophys* 17(2):175–187
- Richter R, Döllner J (2014) Concepts and techniques for integration, analysis and visualization of massive 3d point clouds. *Comput Environ Urban Syst* 45:114–124
- Schraven D, Hartmann A, Dewulf G (2011) Effectiveness of infrastructure asset management: challenges for public agencies. *Built Environ Project Asset Manag* 1(1):61–74
- Schrotter G, Hürzeler C (2020) The digital twin of the city of zurich for urban planning. PFG—J Photogramm Remote Sens Geoinformat Sci 88(1):99–112
- Shahat E, Hyun CT, Yeom C (2021) City digital twin potentials: a review and research agenda. *Sustain* 13(6):3386
- Stojanovic V, Richter R, Döllner J, Trapp M (2018a) Comparative Visualization Of Bim Geometry And Corresponding Point Clouds. *Int J Sustain Dev Plan* 13(1):12–23
- Stojanovic V, Trapp M, Richter R, Hagedorn B, Döllner J (2018b) Towards the generation of digital twins for facility management based on 3d point clouds. In: 34th Annual ARCOM Conference, pp 270–279
- Stoter JE, Ohori GAKO, Noardo F (2021) Digital twins: a comprehensive solution or hopeful vision? *GIM Inter: Worldwide Magazine Geomat*
- TreeCop Project Essen, <https://unsere-smartcity-essen.de/projekte/treecop-sensoren-steuern-bew-uesserung-von-stadtbaeumen/>, last accessed 2023/05/15
- Voland P, Asche H (2017) Geospatial visualization of automotive sensor data: a conceptual and implementational framework for environment and traffic-related applications. *ICCSA* 2017:626–637
- Volk R, Stengel J, Schultmann F (2014) Building information modeling (BIM) for existing buildings—literature review and future needs. *Autom Constr* 38:109–127
- Walter D, Anderssohn J (2023) Nutzerzentrierte Weiterentwicklungen der InSAR-Messtechnik und deren Bereitstellungsdaten. *GeoMonitoring*
- Wan L, Nochta T, Schooling J (2019) Developing a city-level digital twin—propositions and a case study. In: International Conference on Smart Infrastructure and Construction ICSIC '19
- Xie Y, Tian J, Zhu X (2020) Linking points with labels in 3D: a review of point cloud semantic segmentation. *IEEE Geosci Remote Sens Mag* 8(4):38–59

Humans As Sensors in Urban Digital Twins



Binyu Lei, Yunlei Su, and Filip Biljecki

Abstract Digital twins have gained increasing attention as a tool to facilitate decision-making in the cities. However, the current discourse predominantly focuses on technical aspects while overlooking the human aspect in urban digital twins. This work proposes a conceptual framework that addresses the role of humans in relation to the urban environment, therefore highlighting the social value of urban digital twins. The proposed framework is subsequently implemented in a specific case study of outdoor walking comfort at National University of Singapore, validating its feasibility in practice. By incorporating human sensing data, such as participatory data, urban digital twins have the potential to represent the dynamic interaction between people and environments, generating a holistic physical-social-virtual system.

Keywords Urban planning · Participatory process · Human sensing data · Data integration · 3D city modelling

1 Introduction

Urban digital twins facilitate planning and construction in cities, as the means of modelling physical assets and implementing dynamic scenarios. By integrating data from various sources and utilising technical tools, digital twins enable an interactive loop between virtual and physical environments, thereby supporting efficient

This article was selected based on the results of a double-blind review of an extended abstract.

B. Lei · F. Biljecki (✉)

Department of Architecture, National University of Singapore, Singapore, Singapore
e-mail: filip@nus.edu.sg

B. Lei

e-mail: binyul@u.nus.edu

Y. Su

Department of Geography, National University of Singapore, Singapore, Singapore

F. Biljecki

Department of Real Estate, National University of Singapore, Singapore, Singapore

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
T. H. Kolbe et al. (eds.), *Recent Advances in 3D Geoinformation Science*, Lecture Notes
in Geoinformation and Cartography, https://doi.org/10.1007/978-3-031-43699-4_42

decision-making and cost-saving measures Fuller et al. (2020), Ketzler et al. (2020) and Charitonidou (2022). While the potential is well identified in the community, the social aspect is often overlooked in the whole lifecycle of urban digital twins. However, a shift towards a socio-technical perspective has been recently noticed in the current discourse surrounding urban digital twins, with increasing attention being paid to social and legal considerations Nochta et al. (2021) and Lei et al. (2023). We posit that urban digital twins, as the interdisciplinary concept, should provide insights into city planning and decision making, beyond the demonstration of technical functionality. However, there remains a research gap in developing a framework that considers the social aspect in urban digital twins and dynamically incorporates such information. As such, we identify two main aspects related to social and human value, addressing the socio-technical trend and enhancing the development of urban digital twins. First, since urban digital twins are built in various initiatives and regions, the accessibility of these models are scattered and unequally distributed around the world. It is implied that some digital twins are enclosed without public access, leading to a concern of lack of openness. Non-public models have insufficient evidence to demonstrate their purposes and capabilities to facilitate city management. In other words, an absence of public participation ignores the role of people in cities, resulting in an unclear position of digital twins in the urban realm. Second, collaboration among a variety of stakeholders (e.g. government, companies, research institutes and organisations) is an emerging topic in the field of urban digital twins, associated with, for example, data sharing and policy making in the local contexts. In relation to the first point, the human part is a significant component in urban digital twins, including but not limited to open accessibility and multistakeholder engagement. In this context, an inclusion of human aspect should be considered to enhance social value in urban digital twins, as well as improve the current socio-technical trend.

With the emergence of social networking, the role of human participation can be described as data sources from sensing information via explicit recruitment or implicitly by contributing on platforms Srivastava et al. (2012). Human sensing data presents an input source relying on collecting data by humans or by sensors acting on their behalf. Such human-centric sensing has been studied in strands of fields, from psychology to sociology, as a means to collect sensing data and gain insights into human behaviours Goodchild (2007), Jayathissa et al. (2020) and Galesic et al. (2021). Following the societal trend in the research communities, urban studies have increasingly embraced this concept to investigate the interaction between people and cities, e.g. leveraging human sensing data to understand urban changes Niu and Silva (2020) and Wang et al. (2023). As a new tool of the digital revolution in urban domains, digital twins have a strong demand for dynamic information (e.g. data collected from IoT infrastructure) to manage cities in the real world Bibri (2018) and Kuguoglu et al. (2021). While such sensors can support the twinning of the built environment in digital twins (e.g. gathering real-time weather data), they fail to capture the crucial aspect of the interaction between people and physical surroundings. An urban digital twin integrates the input of human sensing data (e.g. participatory and collaborative information), demonstrating the potential to reflect social activities and provide solutions for the uncertainties and complexities of urbanism Legacy

(2017) and Fotheringham (2023). In this sense, such a digital twin model indicates the capability to simulate urban scenarios, collect feedback from people, and facilitate human-environment interactions, eventually generating a dynamic loop between the physical, social, and virtual environments within the system. For example, a human-centred urban digital twin can be employed to examine political decisions regarding infrastructure, as an approach to enhancing resilience among people and communities, by integrating multiple sensing data from local citizens Ye et al. (2023).

This work raises two research questions: (1) How to incorporate the human aspect in urban digital twins? (2) What value can the concept of humans as sensors add to urban digital twins? These questions highlight the importance of considering the social and human dimensions in urban digital twins and exploring the potential benefits of using human sensors to gather valuable data for digital twin models. A conceptual framework is developed to integrate human sensing data (e.g. through a participatory process) in the lifecycle. Subsequently, the proposed framework is implemented in a specific area at National University of Singapore. In the case study, 15 participants recruited to collect sensing data, function as sensors, when performing outdoor walks, by wearing digital devices. Four data sources are gathered, which are comfort feedback, noise data, solar intensity and the percentage of urban objects in the surroundings. This case study aims to investigate how people interact with the built environment during walking, by analysing human sensing data. Meanwhile, it is proved as an example of validating the feasibility of our conceptual framework, namely examine the interaction between people and the surroundings with the aid of urban digital twins.

2 Related Work

For years, human sensing data, described as a collection of digital information generated by individuals to disseminate their feelings or experience on social platforms, have received wide attention in numerous studies Li et al. (2022), Dixon et al. (2021), Songchon et al. (2021), Li et al. (2021) and Liu et al. (2019). Large-volume human sensing data can convey a wealth of information based on human activities at various scales. Regardless of personal background and knowledge, such data offers valuable insights into how people interact with their environment and engage in different activities. In some cases, geospatial information embedded there, such as check-in data from platforms like Yelp and Tripadvisor, and geotagged images from services like Flickr, can be useful in understanding local contexts Senaratne et al. (2017). Through extracting spatial and temporal information, these human sensing data can support a wide range of research objectives, for example, disaster management, problem reporting, social behaviour and urban dynamics Horita et al. (2013), Feng et al. (2022) and Wang et al. (2023). For example, one of the most prominent applications is concluding man-land relationship by interpreting multiple human sensing data to recognise urban functions and land cover Johnson et al. (2017). Furthermore, Tu et al. (2020) conducted research on portraying urban vibrancy of Shenzhen by

analysing three types of crowdsourcing data. Ballatore and De Sabbata (2020) investigated place representation by comparing four crowdsourced content datasets and their context, forming a holistic understanding of urban life. Moreover, human generated data is also popular in characterising demographics (e.g. income, job-housing relationship) and then further inferring urban development Zhao et al. (2022) and Ballatore and De Sabbata (2020). Through extracting socio-economic features from such data, urban sprawl and urban deprivation are discussed in the scholar studies, revealing the city as a social system intertwined with human activities Cui et al. (2021) and Sabri et al. (2016). As such, this large-size amount of human sensing data has gained popularity as a supplementary source to demonstrate human activities associated with demographic, socio-economic and environmental factors Liu et al. (2021) and Tu et al. (2020).

Cities are an integrated system of elements, including human, assets, environment and their interaction, which are more complex than others Shahat et al. (2021). Emerging discussions expose an interest in socio-technical issues to address digital twins' functions beyond technical aspects. Batty (2018) identified one request in digital twins as merging socio-economic process into the built environment. A proposed method is to involve humans as sensors through their smartphones and integrate with physical information (e.g. traffic, building, energy), shaping a comprehensive data network. In the case of including people in urban digital twins, sensing data enable a glimpse of the nature of cities, e.g. how the city works, how human activities represent the city, and how their socio-economic characteristics affect the city. Applying urban digital twins to achieve such idealisation fully is illusory at present, however, some efforts have been carried out in the current research landscape by exploring mechanisms to include people in digital twins as a primary step. Dembski et al. (2020) presented an urban digital twin of Herrenberg as a means to engage people in the planning process. Such participation is conducted with virtual reality (VR) in the digital twins, supported by various dialogues between councillors, experts and citizen. People can share feedback and visions of future planning based on local knowledge. Urban digital twins support such participatory processes in a virtual environment and are easily accessible to different groups regardless of their backgrounds. Similarly, White et al. (2021) delivered a participatory digital twin for Dublin. The study applies the model to simulate flooding scenarios, aiming to design corresponding strategies for the city. Given the cited work's motivation, local citizens are invited to the planning process, providing feedback and detecting errors in such simulations. Considering the growing interest in citizen engagement, Abdeen and Sepasgozar (2022) proposed a framework with different layers to facilitate community participation in the urban digital twins. Besides involving public participation in different ways, another strand of research addresses socio-economic values by modelling human activities in urban digital twins. Klebanov et al. (2018) approached a method to digitalise humans as agents and simulate their dynamics depending on individual needs, intending to effectively implement strategic planning. Such a means enhances the position of people in the planning process and, at the same time, unfolds the potential to represent physical infrastructure and behaviour patterns in the digital twins. This consideration has also been adopted in simulating disaster scenarios. Fan

et al. (2020) proposed a framework to evaluate infrastructure disruptions based on analysing social sensing data in the enhanced digital twins.

Overall, a socio-technical trend has been emerging in the current discourse surrounding urban digital twins, addressing the consideration of humans. Motivated by the utilisation of human sensing data in other related fields, we intend to propose a conceptual framework that incorporates social and human dimensions into urban digital twins. To the best of our knowledge, this framework represents a novel approach by including non-technical values in urban digital twins and leveraging human sensing data to represent social activities within such models.

3 Conceptual Framework

The methodology is designed with two parts: data deployment and system management (Fig. 1). First, human-centred urban digital twins are built upon multiple data sources, such as from crowdsourcing, social sensing and participatory sensing, considering the features of people in social, cultural, and environmental dimensions. Taking participatory data as an instance, these data are directly contributed by participants that recruited for specific urban studies, such as information from wearable devices Miller et al. (2021), responses to questionnaires Liu et al. (2023), and feedback provided while interacting with urban digital twins White et al. (2021). By integrating such information into urban digital twins, they can represent individual perceptions of urban environment and simultaneously promote the role of people in the decision-making process. In this sense, deploying multiple types of data that indicate people factors has the potential to mirror how people perceive and behave in the virtual environment. With the consideration of such social and human aspects, digital twins facilitate the interaction between people and the urban environment while adding social value to their lifecycle.

The second part focuses on system management, including 3D construction, data update and maintenance. The core aspect involves automatically mapping dynamic data into a digital twin platform. We design our platform by utilising a browser-server architecture, ensuring maximum compatibility and interoperability with future input data and software. Our method incorporates the use of Cesium, an open-source platform, which supports data integration and online visualisation. The construction of 3D building models is achieved by extruding building footprints according to the height information, as well as enriching with semantic information, such as building types, the construction of year, roof types, building materials. Data collected from human is linked with geospatial location and then integrated into our platform in multiple formats, such as 3D Tiles and GeoJSON. Such mainstream data are represented using Cesium Markup Language (CZML), enabling real-time geospatial information exchange within the Cesium virtual earth Zhu et al. (2018). For instance, Cesium allows us to integrate mobility data that contains trajectory-related information, e.g. heart rate and walking speed. This information can then be integrated with urban phenomena such as urban shading, serving mobility analysis in the digital twins, e.g.

pushing real-time notifications to pedestrians to suggest a comfortable and walkable route. Overall, the interactive simulations have the potential to identify possible consequences of proposed urban strategies and facilitate decision making.

4 Case Study

4.1 Study Area

The conceptual framework is proposed to enhance the social value in digital twins at different urban scales. As an urban digital twin should represent the city ecosystem, various human activities and multi-functional buildings and spaces need to be included when implementing such a framework. That is, the features and coverage of the study area need to be reasonable to indicate the urban context. In this sense, this framework is applied to the Kent Ridge Campus at National University of Singapore (Fig. 2). The study area encompasses more than 300 buildings, covering a variety of uses, such as educational buildings, recreational facilities, and infrastructure. Such diversity can be considered to represent the campus as a suitable case study for a digital twin at the district or urban scale.

4.2 Data Collection

This case study includes two primary data sources: publicly available data and participatory data. Publicly available data, such as OpenStreetMap data, is used to generate 3D representations of buildings on the campus, including building footprints and

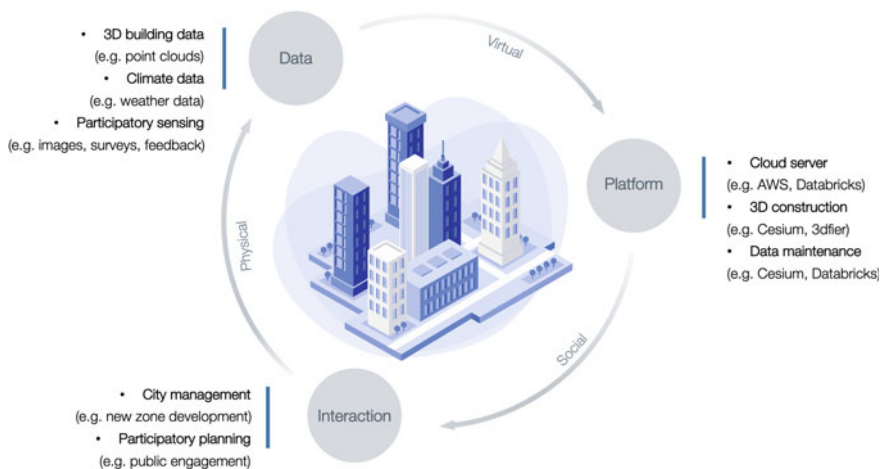


Fig. 1 The framework of human-centric urban digital twins



Fig. 2 Study area: Kent Ridge Campus, National University of Singapore. *Sources:* OpenStreetMap contributors and Google Street View

essential semantic information such as name, types, height, and number of floors Biljecki et al. (2023). Regarding participatory information, 15 participants (NUS students) were recruited to wear digital devices and collect data related to walking comfort and environmental factors. In detail, Cozie¹, an open-source iOS application built on the Apple Watch, was deployed to collect heart rates and comfort feedback when walking on the selected trajectory. Panoramic street view images were collected using GoPro cameras to represent the built environment at each location where participants submitted their comfort feedback. Noise data was recorded by the sound meter and solar intensity was monitored through Photometer application² on a smartphone. The experiments were performed thrice daily, with a single participant per walk, commencing at 10 am, 2 pm, and 5 pm, attempting to include multiple outdoor conditions. As such, participants function as sensors, collecting comfort perception data along with, e.g. location, timestamp and factors related to the built environment. A statistical plot summarises different values collected through the experiment in Fig. 3. Such data were subsequently integrated into the digital twin campus for showcases and analysis.

¹ <https://cozie-apple.app/>.

² <https://photometer.pro/>.

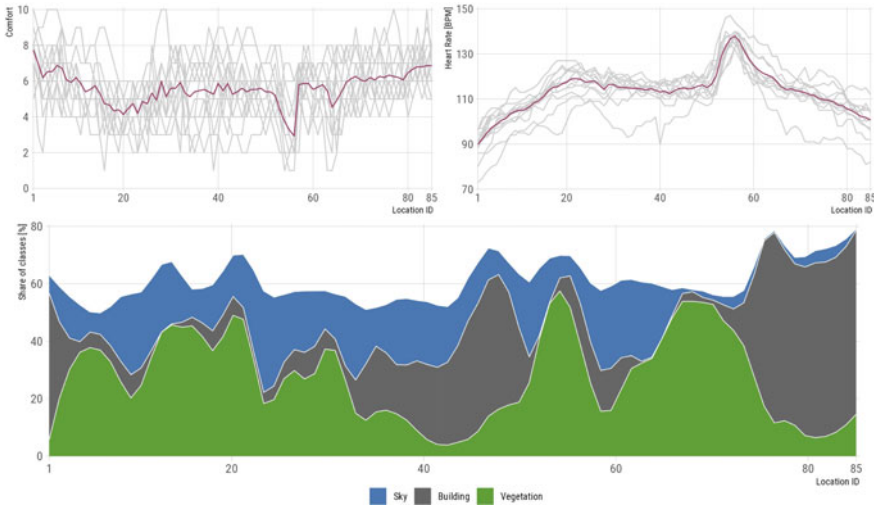


Fig. 3 Collecting participatory data from human sensing. The top plots show participants’ biometric values during walking along the selected path. The bottom plot demonstrates the built environment. Data courtesy of Liu et al. (2023).

4.3 Data Integration

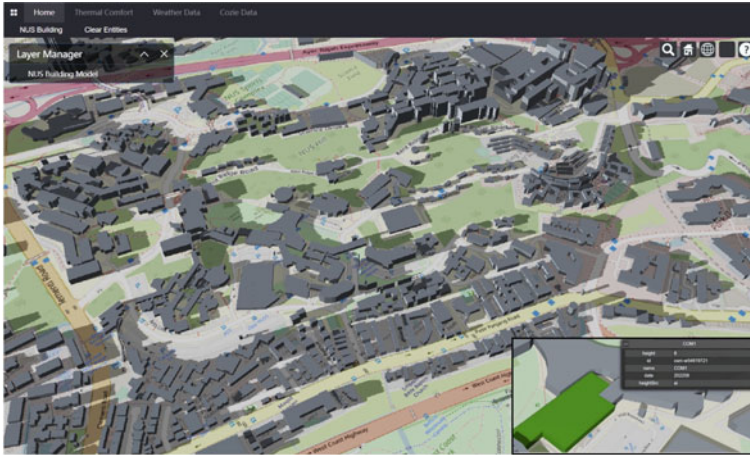
Integration is identified as a key challenge of urban digital twins Lei et al. (2023), involving system compatibility, format conversion, and other factors. In this case study, Cesium is designed as the platform and two types of data sources are included. Considering that the platform is browser-based and uses HTTP for visiting, all integrated data should be in a JSON-based format as input for Cesium. First, building data is downloaded from OpenStreetMap as an .osm file, and extracted through QGIS. FME (Feature Manipulation Engine), a robust tool for integrating and transforming spatial data, is applied to create 3D tilesets in JSON format, for extruding building footprints. Second, for the panoramic photos, they are segmented using DeepLabv3 Chen et al. (2017), to derive the percentage of urban objects visible from a viewpoint along the route. Four classes, namely *sky*, *building*, *vegetation* and *terrain*, are selected to represent the physical environment in this work. All participatory data is stored in a CSV format (15 CSV tables for 15 participants), and converted into JSON file in Python using Pandas library. A REST API is built to access and retrieve data via HTTP requests, which can be simultaneously parsed into the Cesium platform for further visualisation and analysis.

4.4 Implementation

3D buildings are generated from OpenStreetMap data according to heights or the number of floors. OpenStreetMap data in Singapore is of excellent quality (Biljecki, 2020), and thus, suitable for this research. Most buildings have semantic attributes such as name, height, levels, and type, while a small portion of them have additional roof-related information, such as roof type and roof colour. The 3D Tiles of buildings are visualised on Cesium, representing a simplified virtual model of campus. Participatory data is converted into CZML format as an input for online visualisation. Each walk performed by participants includes 85 locations around the College of Design and Engineering (CDE) of the National University of Singapore. The path in focus has a distance of approximately 1.7 km and includes a mixture of physical conditions, such as outdoor sidewalks, open spaces, and semi-open spaces (e.g. sheltered areas between two buildings), and heterogeneous scenes (e.g. varying greenery visible). Since the data is collected as location points, a walking trajectory denotes the connection of each point enriched with time-series data. A 3D human model is prepared to represent individual participant and simulate outdoor walks on the platform. When the human model moves along the trajectory, a figure displays dynamic changes in sensing data under different categories, e.g. heart rate and solar intensity (see Fig. 4).

5 Discussion

The concept of humans as sensors provides insight into representing social activities in urban digital twins. The lens of human sensing pushes forward the socio-technical shift in the current discourse, taking into account the social and human aspects. In this sense, the conceptual framework proposes the characteristics of social and human aspects in urban digital twins. Two features (i.e. openness and availability) are considered in the framework, regarding the encouragement of using publicly available data and open-source tools. In terms of added value from the social dimension, we deem that human plays a role in cities, offering opportunities for crowd-sourcing and sensing information. As such, the concept of humans as sensors is applied in the conceptual framework, aiming to integrate sensing data and enhance the dynamic interaction between people and the environment. Validating the capability of the proposed conceptual framework, we further implement it on a campus area at National University of Singapore. The generation of such a digital twin model relies on public data and open tools, which is based on an online browser, enabling a user-friendly interface. Meanwhile, through designing a participatory process, sensing data collected by participants is integrated into the platform, demonstrating the interaction between individual perception and the built environment. With the aid of deploying public data and utilising open-source tools, a human-centred digital twin offers increasing possibilities to encourage public participation and openness among different groups of stakeholders. The dynamic interaction enables dynamic



(a)



(b)

Fig. 4 The implementation of the conceptual framework at NUS campus. The top plot **a** presents 3D buildings in Kent Ridge Campus, National University of Singapore. The bottom one **b** demonstrates the dynamic interaction between outdoor walking and the surroundings. *Source:* OpenStreetMap contributors

data representation in urban digital twins and provides timely feedback based on the simulation of various scenarios. The proposed framework of human-centred digital twins takes into account significant characteristics of urban digital twins, such as dynamic information exchange and the human aspect. In the presented adoption in the campus of National University of Singapore, we incorporate thermal comfort and physiological data collected from participants into our digital twin model, along with public data (e.g. 3D buildings and climate data) with a representation of the physical environment. Through this proof-of-concept, we demonstrate the potential of integrating participatory sensing data to visualise and simulate the dynamic relationship between people and urban environments.

In terms of use cases, this proposed framework presents the scalability to serve multiple purposes in the urban realm. For example, the results of previous case study can be scaled to the entire campus. By extracting environmental factors from street-level images (e.g. volume of visible vegetation), a potential use case can be using information from the built environment to infer human comfort during walk. With the predicted comfort data, a relationship between environmental factors and individual perception can be examined in a large scale, which subsequently supports further studies. For instance, by integrating such information in the digital twin prototype, we can assess the greenery in the campus and simulate multiple walkable paths for pedestrians. A limitation should be elaborated in future studies is regarding data-related issues. First, considering participatory information, data privacy raises concern which hinders a large-scale deployment in urban digital twins. The access to such data is limited, as well as the recruitment and participation can be time-consuming and subject to cases. Moreover, the reliability is acknowledged as a key issue in urban digital twins when utilising such information for use cases. Besides data collection from participatory process, other types of human sensing data can also be considered as a representation of social activities in the proposed framework. Taking crowdsourced data as an example, OpenStreetMap data used for generating 3D buildings is contributed by volunteers who manually map or edit features, inevitable entailing data noise or accuracy-related barriers. In future research, we aim to explore possible use cases by adopting this conceptual framework, illustrating the inclusion of human and social values beyond data collection and visualisation. Furthermore, a detailed discussion of data privacy and reliability will be conducted as part of our long-term motivation, addressing the lifecycle of urban digital twins.

6 Conclusion

This work develops a conceptual framework of urban digital twins, considering social and human elements which are largely overlooked in the current discussions. Regarding the comprehensive architecture, this framework encourages openness and availability of urban digital twins, and subsequently involves multiple groups of stakeholders in the development process (e.g. government, companies and research institutes). In this context, data integration includes a variety of publicly available data, such as 3D buildings, crowdsourced information and environmental factors. Open-source tools are also recommended for 3D reconstruction and visualisation, for example, Cesium platform. Meanwhile, regarding the nature of urban data, management and maintenance service are considered to update dynamic changes between the physical and virtual environment. That is, enabling interactive simulations on the system. Such architecture of the proposed framework acknowledges the lifecycle of urban digital twins from data collection to system management. Human sensing data as an input highlights social and human dimension in the framework, for example, representing human activities and reflecting the interaction between people and the built environment in urban digital twins. This conceptual framework is

implemented on the Kent Ridge Campus at National University of Singapore. Leveraging participatory data to analyse how individuals are affected by the surroundings in the case study, our framework offers the feasibility for embedding social and human value in practice. We deem the proposed framework as an initial step to address socio-technical trend in urban digital twins, as a means of integrating multiple functionality beyond the technical aspect.

Acknowledgements We thank the members of the NUS Urban Analytics Lab, Building and Urban Data Science (BUDS) Lab, and Integrated Data, Energy Analysis + Simulation (IDEAS) Lab for the discussions. This research is part of the project Multi-scale Digital Twins for the Urban Environment: From Heartbeats to Cities, which is supported by the Singapore Ministry of Education Academic Research Fund Tier 1. This research is part of the project Large-scale 3D Geospatial Data for Urban Analytics, which is supported by the National University of Singapore under the Start Up Grant R-295-000-171-133.

References

- Abdeen FN, Sepasgozar SM (2022) City digital twin concepts: a vision for community participation. *Environ Sci Proc* 12(1):19. <https://doi.org/10.3390/environsciproc2021012019>
- Ballatore A, De Sabbata S (2020) Los Angeles as a digital place: the geographies of user-generated content. *Trans GIS* 24(4):880–902. <https://doi.org/10.1111/tgis.12600>
- Batty M (2018) Digital twins. *Environ Plan B Urban Anal City Sci* 45(5):817–820. <https://doi.org/10.1177/2399808318796416>
- Bibri SE (2018) The IoT for smart sustainable cities of the future: an analytical framework for sensor-based big data applications for environmental sustainability. *Sustain Cities Soc* 38:230–253. <https://doi.org/10.1016/j.scs.2017.12.034>
- Biljecki F (2020) Exploration of open data in Southeast Asia to generate 3D building models. *ISPRS Ann Photogram Rem Sens Spat Inf Sci* VI-4/W1-2020:37–44. <https://doi.org/10.5194/isprs-annals-vi-4-w1-2020-37-2020>
- Biljecki F, Chow YS, Lee K (2023) Quality of crowdsourced geospatial building information: a global assessment of openstreetmap attributes. *Build Environ* 237:110295
- Charitonidou M (2022) Urban scale digital twins in data-driven society: challenging digital universalism in urban planning decision-making. *Int J Arch Comput* 20:238–253. <https://doi.org/10.1177/14780771211070005>
- Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017) DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans Patt Anal Mach Intell* 40(4):834–848
- Cui N, Malleson N, Houlden V, Comber A (2021) Using vgi and social media data to understand urban green space: a narrative literature review. *ISPRS Int J Geo-Inf* 10(7):425. <https://doi.org/10.17645/ijgi.v1i2.620>
- Dembksi F, Wössner U, Letzgus M, Ruddat M, Yamu C (2020) Urban digital twins for smart cities and citizens: the case study of Herrenberg, Germany. *Sustainability* 12(6):2307. <https://doi.org/10.3390/su12062307>
- Dixon B, Johns R, Fernandez A (2021) The role of crowdsourced data, participatory decision-making and mapping of flood related events. *Appl Geogr* 128:102393. <https://doi.org/10.1016/j.apgeog.2021.102393>
- Fan C, Jiang Y, Mostafavi A (2020) Social sensing in disaster city digital twin: Integrated textual–visual–geo framework for situational awareness during built environment disruptions. *J Manage Eng* 36(3):04020,002. [https://doi.org/10.1061/\(ASCE\)ME.1943-5479.0000745](https://doi.org/10.1061/(ASCE)ME.1943-5479.0000745)

- Feng Y, Huang X, Sester M (2022) Extraction and analysis of natural disaster-related VGI from social media: review, opportunities and challenges. *Int J Geogr Inf Sci* 36(7):1275–1316. <https://doi.org/10.1080/13658816.2022.2048835>
- Fotheringham AS (2023) Digital twins: the current “krays” of urban analytics? *Environ Plan B Urban Anal City Sci* 23998083231169159
- Fuller A, Fan Z, Day C, Barlow C (2020) Digital twin: enabling technologies, challenges and open research. *IEEE Access* 8:108, 952–108, 971. <https://doi.org/10.1109/ACCESS.2020.2998358>
- Galesic M, Bruine de Bruin W, Dalege J, Feld SL, Kreuter F, Olsson H, Prelec D, Stein DL, van der Does T (2021) Human social sensing is an untapped resource for computational social science. *Nature* 595(7866):214–222. <https://doi.org/10.1038/s41586-021-03649-2>
- Goodchild MF (2007) Citizens as sensors: the world of volunteered geography. *Geo J* 69(4):211–221. <https://doi.org/10.1007/s10708-007-9111-y>
- Horita FEA, Degrossi LC, de Assis LFG, Zipf A, de Albuquerque JP (2013) The use of volunteered geographic information (VGI) and crowdsourcing in disaster management: a systematic literature review. In: *Proceedings of the nineteenth Americas conference on information systems*
- Jayathissa P, Quintana M, Abdelrahman M, Miller C (2020) Humans-as-a-sensor for buildings-intensive longitudinal indoor comfort models. *Buildings* 10(10):174
- Johnson BA, Iizuka K, Bragais MA, Endo I, Magcale-Macandog DB (2017) Employing crowdsourced geographic data and multi-temporal/multi-sensor satellite imagery to monitor land cover change: A case study in an urbanizing region of the Philippines. *Comp Environ Urban Syst* 64:184–193. <https://doi.org/10.1016/j.compenvurbsys.2017.02.002>
- Ketzler B, Naserentin V, Latino F, Zangelidis C, Thuvander L, Logg A (2020) Digital twins for cities: a state of the art review. *Built Environ* 46(4):547–573. <https://doi.org/10.2148/benv.46.4.547>
- Klebanov B, Nemtinov A, Zvereva O (2018) Simulation as an effective instrument for strategic planning and transformation of smart cities. *Int Multidiscip Sci GeoConf SGEM* 18(2.1):685–692. <https://doi.org/10.5593/sgem2018/2.1/S07.087>
- Kuguoglu BK, van der Voort H, Janssen M (2021) The giant leap for smart cities: scaling up smart city artificial intelligence of things (AIoT) initiatives. *Sustainability* 13(21):12,295. <https://doi.org/10.3390/su132112295>
- Legacy C (2017) Is there a crisis of participatory planning? *Plan Theory* 16(4):425–442
- Lei B, Janssen P, Stoter J, Biljecki F (2023) Challenges of urban digital twins: a systematic review and a delphi expert survey. *Autom Construct* 147:104716
- Li P, He X, Qiao M, Miao D, Cheng X, Song D, Chen M, Li J, Zhou T, Guo X, Yan X, Tian Z (2021) Exploring multiple crowdsourced data to learn deep convolutional neural networks for road extraction. *Int J Appl Earth Observ Geoinf* 104(102):544. <https://doi.org/10.1016/j.jag.2021.102544>
- Li X, Huang X, Li D, Xu Y (2022) Aggravated social segregation during the covid-19 pandemic: Evidence from crowdsourced mobility data in twelve most populated U.S. metropolitan areas. *Sustain Cities Soc* 81:103,869. <https://doi.org/10.1016/j.scs.2022.103869>, <https://www.sciencedirect.com/science/article/pii/S2210670722001962>
- Liu Y, Zhu AX, Wang J, Li W, Hu G, Hu Y (2019) Land-use decision support in brownfield redevelopment for urban renewal based on crowdsourced data and a presence-and-background learning (pbl) method. *Land Use Policy* 88:104188. <https://doi.org/10.1016/j.landusepol.2019.104188>
- Liu P, Zhao T, Luo J, Lei B, Frei M, Miller C, Biljecki F (2023) Towards human-centric digital twins: leveraging computer vision and graph models to predict outdoor comfort. *Sustain Cities Soc* 93:104480
- Liu Y, Gao S, Yuan Y, Zhang F, Kang C, Kang Y, Wang K (2021) Methods of social sensing for Urban studies. In: Yang X (ed) *Urban remote sensing*, 1st edn. Wiley, pp 71–89. <https://doi.org/10.1002/9781119625865.ch4>

- Miller C, Abdelrahman M, Chong A, Biljecki F, Quintana M, Frei M, Chew M, Wong D (2021) The internet-of-buildings (IoB)—digital twin convergence of wearable and IoT data with GIS/BIM. In: *Journal of physics: conference series*, IOP Publishing, vol 2042, p 012–041
- Niu H, Silva EA (2020) Crowdsourced data mining for urban activity: review of data sources, applications, and methods. *J Urban Plan Develop* 146(2):04020,007. [https://doi.org/10.1061/\(ASCE\)UP.1943-5444.0000566](https://doi.org/10.1061/(ASCE)UP.1943-5444.0000566)
- Nochta T, Wan L, Schooling JM, Parlikad AK (2021) A socio-technical perspective on urban analytics: the case of city-scale digital twins. *J Urban Technol* 28(1–2):263–287
- Sabri S, Rajabifard A, Ho S, Amirebrahimi S, Bishop I (2016) Leveraging VGI integrated with 3D spatial technology to support urban intensification in Melbourne, Australia. *Urban Plan 1*(2):32–48. <https://doi.org/10.17645/up.v1i2.623>
- Senaratne H, Mobasheri A, Ali AL, Capineri C, Haklay M (2017) A review of volunteered geographic information quality assessment methods. *Int J Geogr Inf Sci* 31(1):139–167. <https://doi.org/10.1080/13658816.2016.1189556>
- Shahat E, Hyun CT, Yeom C (2021) City digital twin potentials: a review and research agenda. *Sustainability* 13(6):3386. <https://doi.org/10.3390/su13063386>
- Songchon C, Wright G, Beevers L (2021) Quality assessment of crowdsourced social media data for urban flood management. *Comp Environ Urban Syst* 90:101690. <https://doi.org/10.1016/j.compenvurbsys.2021.101690>
- Srivastava M, Abdelzaher T, Szymanski B (2012) Human-centric sensing. *Philos Trans Royal Soc A Math Phys Eng Sci* 370(1958):176–197
- Tu W, Zhu T, Xia J, Zhou Y, Lai Y, Jiang J, Li Q (2020) Portraying the spatial dynamics of urban vibrancy using multisource urban big data. *Comp Environ Urban Syst* 80(101):428. <https://doi.org/10.1016/j.compenvurbsys.2019.101428>
- Wang J, Chow YS, Biljecki F (2023) Insights in a city through the eyes of Airbnb reviews: sensing urban characteristics from homestay guest experiences. *Cities* 140(104):399. <https://doi.org/10.1016/j.cities.2023.104399>
- White G, Zink A, Codecá L, Clarke S (2021) A digital twin smart city for citizen feedback. *Cities* 110:103064. <https://doi.org/10.1016/j.cities.2020.103064>
- Ye X, Du J, Han Y, Newman G, Retchless D, Zou L, Ham Y, Cai Z (2023) Developing human-centered urban digital twins for community infrastructure resilience: a research agenda. *J Plan Liter* 38(2):187–199
- Zhao X, Cui M, Levinson D (2022) Exploring temporal variability in travel patterns on public transit using big smart card data. *Environ Plan B Urban Anal City Sci* 23998083221089662. <https://doi.org/10.1177/2399808322108966>
- Zhu L, Wang Z, Li Z (2018) Representing time-dynamic geospatial objects on virtual globes using czml-part i: overview and key issues. *ISPRS Int J Geo-Inf* 7(3):97

Requirements for Web-Based 4D Visualisation of Integrated 3D City Models and Sensor Data in Urban Digital Twins



Joseph Gitahi and Thomas H. Kolbe

Abstract Urban Digital Twins (UDTs) have emerged as essential tools for managing city operations, forming the basis of smart city solutions. They offer a digital representation of the physical urban environment, which supports various city applications such as monitoring mobility, air quality, and modelling simulations. To accurately represent the physical world, UDTs need to be updated continuously to reflect the changes in the urban environment on time. The Internet of Things (IoT) enables real-time data collection to capture these changes. Combined with 3D city models, IoT allows the interactive visualisation of patterns and trends in UDTs. In this study, we conduct investigations on the requirements for the web visualisation of semantic 3D city models enriched with time-dependent properties from IoT and simulation data. We explore the 3D models and IoT data integration requirements, 4D web visualisation design considerations, and the technical implementation requirements for rendering dynamic properties for UDTs applications. The paper also presents a workflow and a web viewer prototype for the 4D visualisation of integrated 3D models and dynamic data.

Keywords IoT · 3D models · Urban digital twins · 4D visualisation

1 Introduction

The estimated global population in urban areas is 58% and is expected to rise to 68% by 2050. The rate is even higher in Europe at 76% and on an upwards trend to reach 83% by 2050 (UN-Habitat 2022). As a result, cities face complex urbanisation

This article was selected based on the results of a double-blind review of the full paper.

J. Gitahi (✉) · T. H. Kolbe
Chair of Geoinformatics, Technical University of Munich, 80333 Munich, Germany
e-mail: joseph.gitahi@tum.de

T. H. Kolbe
e-mail: thomas.kolbe@tum.de

challenges, especially concerning air quality, urban mobility, waste management and security. Smart City initiatives rely on digital technologies, especially the Internet of Things (IoT) and digital 3D models of city objects, to address these challenges and make city operations more efficient and liveable for the benefit of their inhabitants.

A digital twin represents a physical object, system, or process in a virtual world, created using models of the defined entities and dynamic data from sensors or simulations. Specifically, Urban Digital Twins (UDTs) are virtual representations of physical urban environments created by integrating 3D city models and IoT sensor data. 3D models describe the built city environment, while IoT sensors or simulation results feed the GIS-based 3D models with time-dependent data. IoT not only enables sensing the environment but also controlling the communication of the changes in the digital world to the physical world using actuators.

Integrating 3D city models and dynamic data establishes links between modelled physical entities and their respective dynamic data sources. The fourth temporal dimension from the integration enables the 4D visualisation of metrics from real-time and historical dynamic data using game engines or web browsers. Users primarily interact with UDTs through visual tools to understand dynamic urban environments in smart city applications such as monitoring of city operations, urban analytics, and simulations. Web visualisation has an edge over game engines since it requires no extra software installation or hardware by the user apart from a web browser. A large base of city stakeholders can thus interact with UDTs through 4D web visualisation.

This paper investigates the various requirements for effective 4D web visualisation of UDTs. We focus on (1) Requirements for efficient web visualisation of 3D semantic models. (2) 3D city models and dynamic data integration requirements. (3) Dynamic data harmonisation requirements. (4) The cartographic and user requirements to visualise dynamic data effectively and interactively in 4D web maps.

2 Background and Related Research

2.1 *Integration of 3D City Models and IoT Data*

The first steps towards integrating heterogenous datasets in UDTs are standardisation and harmonisation to standard formats and data models. Semantic 3D city models, which form the basis for developing UDTs, represent buildings, vegetation, city furniture, transport networks, and structures. The Open Geospatial Consortium (OGC) CityGML (Gröger et al. 2012; Kolbe 2009) is the most common, open standard for modelling, storing, and exchanging semantic 3D models in the geospatial domain. Currently, version 2.0 of the standard is in use, and development of the next-generation version 3.0 has been completed recently. Both commercial and open-source tools support the standard. The 3DCityDB is an open-source software suite with a database schema for relational database management systems (DBMS) and

a tool that facilitates loading the database with CityGML data and exporting it in various visualisation formats (Yao et al. 2018).

On the other hand, the OGC SensorThings API standard provides an open and unified way to interconnect IoT devices and data via the web (Liang et al. 2016). The standard defines a data model for managing IoT data and a RESTful API for integration with other web-based services. A vital API feature is the native geo-location support enabling the storage and retrieval of sensor data and the corresponding spatial properties. The API is highly scalable and ideal for large-scale IoT deployments like cities. In Hamburg, the API has been implemented in the urban data platform and supports over 2000 sensor stations (Fischer et al. 2021). FIWARE is another open-source platform that provides standardised APIs for managing IoT data in Smart City applications (Cirillo et al. 2019). The platform uses the Next Generation Service Interface (NGSI) data model and API to integrate and manage heterogeneous sensor data.

The need to create dynamic UDTs has led to research efforts on integrating semantic 3D city models and dynamic data from sensors and simulations. CityGML version 3.0 has introduced the Dynamizer module for integrating time-dependent properties with 3D city models using data from sensors and simulations (Chaturvedi 2021; Kutzner et al. 2020). The Dynamizer, defined as a feature type, stores time-series data and updates time-dependent CityGML feature properties. The module also supports modelling complex patterns that represent time variation of properties based on statistics or simulations. For highly variable dynamic data, such as those from sensors, the module supports linking features to sensor data streams using explicit connection information to sensor data services such as the OGC SensorThings API or FIWARE. Chatzinikolaou et al. (2020) used the Dynamizer concept as an Application Domain Extension (ADE) in CityGML 2.0 to enrich and visualise semantic 3D building models with simulation energy consumption data.

The CityThings approach (Santhanavanich and Coors 2021) has been proposed for integrating sensor data with 3D city models using OGC standards, OGC SensorThings API, and OGC CityGML. The approach links dynamic data into 3D models by extending SensorThings API's Things entities with the unique identifiers of target CityGML features. Since SensorThings API allows defining an arbitrary number of user-defined properties while creating a Thing entity, unique `gml_ids` of corresponding CityGML features were added as properties to things. This integration enables fetching sensor data from SensorThings API using the `gml_ids` as query filters for web-based visualisation.

2.2 4D Web Visualisation in UDTs

Developing 3D Web GIS applications requires converting 3D models into streaming formats and using Web mapping libraries. The CesiumJS open-source JavaScript library by Analytical Graphics Inc offers the capability to create 3D globes and interactive maps that run natively on modern browsers. It allows visualising multiple

2D and 3D spatial dataset formats on a virtual globe. In addition, Cesium has developed two data streaming formats, the OGC 3D Tiles (Cozzi et al. 2019) and Cesium Modelling Language (CZML). The 3D Tiles format is designed for streaming and rendering massive 3D geospatial data, while CZML is a JSON format for describing time-dynamic 3D scenes.

Although the 3D Tiles format has no temporal support, some studies have implemented dynamic styling of 3D tiles based on temporal data. One study developed a web application to visualise traffic and air quality data using CesiumJS by dynamically styling 3D Tiles based on user interactions or background sensor data updates (Ebrahim et al. 2021). Another study used CesiumJS and 3D Tiles to visualise real-time energy simulation results of buildings integrated into CityGML using EnergyADE (Mao et al. 2020). To enable end-users to visualise different simulation scenarios, they configured different 3D tiles styling strategies that users could change interactively. Additionally, the study combined colour and opacity visual variables to show multiple properties of a building simultaneously. The study identifies the need to use non-photorealistic visualisation of 3D city models for tasks such as simulations to speed up rendering on the web.

Another study developed dynamic web visualisation of traffic simulation results and 3D city models (Beil et al. 2022). The simulation results were derived from SUMO, a microscopic traffic simulation tool (Lopez et al. 2019) and are composed of movement data of different categories of traffic participants, pedestrians, and passenger cars. The study sought to create accurate visualisation of the simulation results by addressing three visualisation requirements: sufficient spatiotemporal resolution, models that match the physical environment, and smooth rendering of the visualisations on the web. The SUMO simulation results were converted into a CZML document for visualising traffic participants' movements in a CesiumJS-based web client alongside 3D city models. The study proposed a spatial and temporal tiling strategy for CZML data to improve the rendering performance on the web as an alternative to streaming CZML data incrementally.

2.3 Map Design Principles for 4D Visualisation

UDTs users interact with the physical urban environment through dashboards, 2D and 3D maps, and virtual and augmented reality user interfaces. The interfaces allow visualising the state of the UDTs, and in conjunction with actuators, they can control the physical world. 3D maps provide highly detailed visualisation and offer more insight into how objects relate to each other. Different UDTs users' needs should be considered when designing the interfaces above (Ferré-Bigorra et al. 2022).

By integrating a time component, 3D maps gain a fourth dimension that enables the dynamic visualisation of patterns and trends. The temporal dimension could lead to a high volume of data that might impede the user experience while interacting with the maps. The interaction could be affected by either the degradation of web performance due to large datasets or visual clutter from the extra time-dependent

information. The complexity caused by the large datasets could be solved using several strategies. One way is to filter features not required in a visualisation scene using attributes, spatial extent, or time property (Gaigg 2023). The visibility of data layers in a map could be toggled either by user controls or based on map scale or the view position.

In UDTs, non-photorealistic 3D maps are preferred to visualise thematic information using cartographic visual variables. The non-photorealistic visualisation removes non-required graphic details and thus highlights the intended information (Döllner et al. 2005; Li et al. 2022). The reduction of visual elements reduces the data sizes, which improves the real-time rendering of 3D maps.

Cartographic visual variables refer to how geographic features and data are represented in maps. Commonly used visual variables in 3D mapping include feature size, shape, colour, brightness, hue/saturation, transparency, texture, orientation, and position (Bertin 1981; Hardisty et al. 2001; Rautenbach et al. 2015). The variables can be modified according to thematic and dynamic data in 4D visualisations. The choice of visual variables and their modifications should conform to users' expectations by depicting the physical environment in a familiar presentation (Gaigg 2023; Döllner et al. 2006).

To visualise complex information in maps, including 3D scenes, a single visual variable often cannot represent complete information about a feature to the users. As discussed in Sect. 2.2, the study by Mao et al. (2020) used a combination of colour and opacity to visualise the simulated energy consumption and the ages of buildings, respectively. In another related qualitative study (Li et al. 2022), used compound visual variables to enhance the visualisation of LOD2 building models. The study introduced a primary variable to visualise attribute data of buildings and a secondary variable to investigate user cognition of the 3D maps. The main variables used were colour and transparency to represent building populations, while the light source and the background were applied as secondary variables. In the study, changing the background properties of the 3D scenes influenced users' spatial cognition, while changing the light source had little impact on how users interpreted the represented data.

3 Visualisation of Integrated 3D City Models and Dynamic Data in UDTs

3.1 Conceptual Framework

Mapping semantic 3D city models provides a more intuitive visualisation of space than 2D maps. On the other hand, dynamic data from sensors and simulations enable the visualisation and analysis of temporal trends. Integrating the two categories of datasets further enhances the visualisation of dynamic processes in the urban

environment. This paper identifies the following requirements for effective 4D web visualisation of integrated 3D city models and dynamic data.

R1: Efficient web representation of large 4D models

The OGC CityGML provides a standardised way to model, store, and exchange semantic 3D city models but cannot be visualised directly on the web. Therefore, the 3D models must be converted into web streaming formats such as OGC 3D Tiles or OGC Indexed 3d Scene Layer (I3S) (Reed and Belayneh 2022). These formats support the visualisation of massive geospatial datasets using 3D web mapping libraries such as CesiumJS and deck.gl.¹ However, these formats do not natively support temporal data. Some workarounds support 4D web visualisations by updating 3D Tiles' properties and styles on runtime using sensor or simulation data (Ebrahim et al. 2021; Mao et al. 2020).

CesiumJS supports dynamic visualisation using the CZML format for moving objects with time-varying geometries. While 3D Tiles and I3S have tiling strategies for optimising the rendering of massive 3D datasets, large CZML files significantly impact web performance. Optimisation measures such as the proposed spatial and temporal tiling strategy by Beil et al. (2022), asynchronous loading of CZML data and simplifying geometries should be used to improve web performance.

R2: 3D city models and dynamic data need to be connected

4D web visualisation applications require understanding the connection between semantic 3D models and dynamic data. The CityGML 3.0 Dynamizers module introduces a concept for linking 3D city objects and their time-dependent properties to dynamic data sources. The linking could be exploited to fetch dynamic data and update spatial, thematic and appearance attributes of 3D city objects in 4D web visualisations. Another option to establish links is using the CityThings (Santhanavanich and Coors 2021) approach, which links SensorThings API Things entities to their related 3D objects. With this approach, a 4D web application could check which sensor data in a scene links to city objects.

R3: Harmonised access to dynamic data

Dynamic datasets come from heterogeneous IoT sensors and simulation results. The various datasets must be harmonised to ease the integration in 4D visualisations. Standards like the OGC SensorThings API and FIWARE provide the data models and standardised APIs for managing and accessing the datasets. The SensorThings API offers standard REST API with spatial, temporal, and thematic filtering capabilities, which makes it suitable for harmonising heterogeneous data in a single repository. Another approach would be to harmonise heterogeneous dynamic datasets using the InterSensor service (Chaturvedi and Kolbe 2018), which performs on-the-fly encoding of the datasets to either OGC Sensor Observation Service or SensorThings API without storing the datasets.

¹ <https://deck.gl/>.

Based on requirements R2 and R3, streamlined integration of dynamic data into 3D models, for instance, could be performed in two steps using OGC standards: standardisation using SensorThings API and linking 3D models' dynamic properties to sensor data in the SensorThings API using CityGML Dynamizers. This integration, as shown in Fig. 1, would enable real-time updates of time-dependent spatial, thematic and appearance attributes of 3D models. By harmonising sensor and simulation datasets into the same data model, dynamic data integration into 3D models using Dynamizers is standardised. The 3D city models may contain zero to many dynamic properties; therefore, the Dynamizers must link each time-dependent property to the respective data stream in SensorThings API.

R4: 4D maps design and user requirements

4D web maps require selecting appropriate cartographic variables and styles that visualise the physical world and its dynamic processes in formats that are familiar to users and easy to interpret. The choice of visual variables depends on data characteristics and the amount to be visualised in a scene. A combination of variables

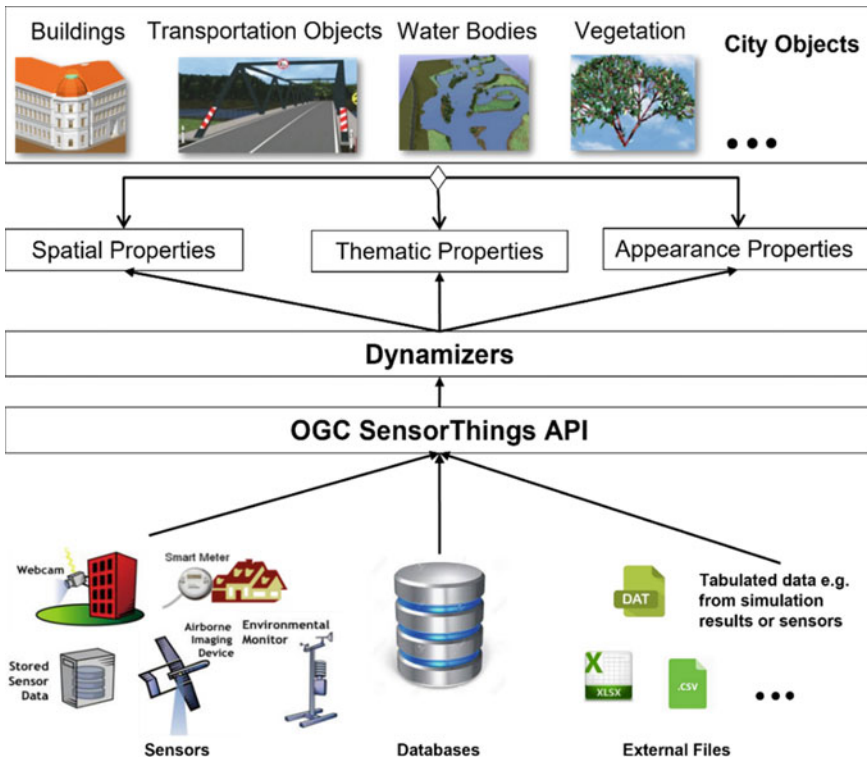


Fig. 1 A concept for standardised management of heterogeneous IoT data and simulation results using SensorThings API. The standardised dynamic data is further integrated with 3D city models using CityGML 3.0 Dynamizers. Adapted from Chaturvedi (2021)

may be required to visualise complex data or the same data at different levels. Visual variables for dynamic data could be applied by changing the appearance, geometry, or project information on 3D objects depending on the data type and 3D objects. Therefore, most visual styling must be determined either manually or automatically.

Web map design should consider different user and application needs to realise the benefits of integrated visualisation. For instance, domain experts require deep insights from the integrated datasets, while decision-makers will be interested in higher-level indicators. Visualisation design should thus consider the information load needed by each group of users and provide tools for data filtering, toggling the visibility of layers, and interacting with the datasets.

The following sections present a workflow and a web viewer prototype for the 4D visualisation of integrated 3D models and dynamic data. We conducted our experiment using Boulevard Sonnenstrasse, a street in the centre of Munich City, as the study area.

3.2 Datasets

GIS Datasets

We use three spatial datasets to represent the Boulevard Sonnenstrasse area on 3D web maps. First, we have LOD2 3D building models in CityGML format, provided as open data by the state mapping agency of Bavaria, Munich. The second dataset is a street space model which offers a polygonal representation of the different traffic lane classes and non-traffic areas. Additionally, there is a road markers dataset. The map in Fig. 2 shows the lane model and the road markings, which were acquired in shapefile format.

Sensor and Simulation Datasets

The second set of datasets comes from actual sensor data and simulation results. Traffic data from Inductive Loop Detectors (ILD) in Munich is obtained from Mobilitäts Daten Marktplatz (MDM)² platform, a German portal for sharing mobility data. The ILD dynamic data for each measured lane contains traffic speeds, flows and occupancy retrieved at 15-min intervals. Some of the ILDs in our study area are shown on the map in Fig. 2. We also fetch traffic lights switching data from the MDM platform, which we have not yet used in the visualisation experiments. The mobility department in Munich City shares the traffic datasets in DATEX II³ format, an XML-based standard for exchanging traffic data.

We also retrieve air quality data from measuring stations in Munich operated by (Bayerische Landesamt für Umwelt (LfU)). The stations provide hourly measurements of different air quality parameters concentrations; Air Quality Index (AQI),

² <https://www.mdm-portal.de/>.

³ <https://www.datex2.eu/>.



Fig. 2 A map of Boulevard Sonnestrasse showing the street space model, inductive loop detectors and the air quality monitoring station

Particulate Matter (PM10), Carbon Monoxide (CO), Ozone (O3), Sulphur Dioxide (SO2), Nitrogen Dioxide (NO2). We obtained the dataset from the German Environment Agency's (Umweltbundesamt)⁴ REST API. In the map shown in Fig. 2 the München/Stachus station that lies within the study area is shown.

In addition to the actual sensor data, we use traffic simulation results from the microscopic traffic simulation tool SUMO. The simulated data includes data from ILDs in the SUMO network, which contains the same information as actual ILDs in the city's road network, and the simulated vehicles' Floating Car Data (FCD).

Data Standardisation and Preparation

The spatial and sensor datasets discussed in the previous sections come from different sources and have various data models. We transformed the datasets to OGC standards to ensure data consistency, interoperability, and easy transferability to other cities. The traffic and air quality sensor datasets were integrated into the OGC SensorThings API, while the spatial datasets were converted into OGC CityGML format. The following sub-sections detail the transformation processes shown in Fig. 3.

⁴ <https://www.umweltbundesamt.de/api>.

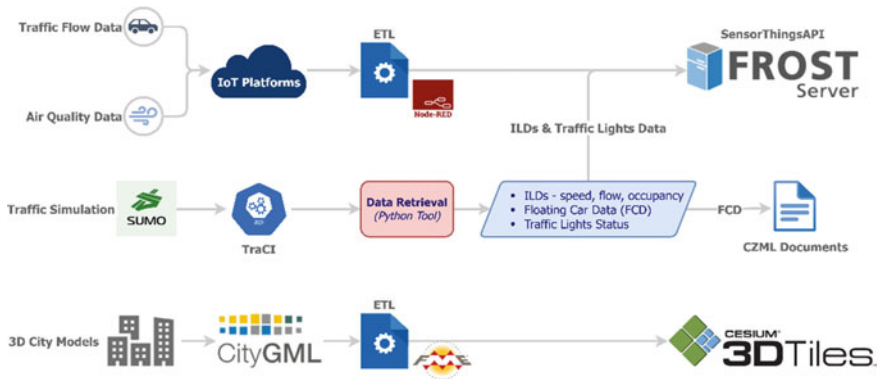


Fig. 3 Standardisation workflow for heterogenous 3D city models and sensor data using OGC standards

Sensor Data

The sensor datasets were transformed using Node-RED,⁵ an open-source visual programming tool for integrating IoT data and automating data processing tasks. For the ILD data, we get two sets of data. The first dataset contains the static list of all ILDs, with their IDs and location coordinates. From this dataset, we created entities following the SensorThings API data model. The entities for each ILD included a *Thing*, its *Location*, and a *Datastream* for each *ObservedProperty*. Before creating the *Datastreams*, we had created the three *Observed Properties*, speed, flow and occupancy, and their *Sensors*. The second dataset contains the temporal *Observations* from the ILDs. We used Node-RED to retrieve and store the data periodically by matching the *Observations* to their respective *Datastream* in the SensorThings API server.

For the air quality data, we created a *Thing*, *Location*, and *Datastreams* for each measurement station for the air quality parameters; *AQI*, *PM10*, *CO*, *O3*, *SO2*, and *NO2*. The sensor *Observations* were retrieved at hourly intervals and stored in the created *Datastreams*.

Simulation Data

We developed a Python tool to fetch SUMO simulation results using the Traffic Control Interface (TraCI) library (Wegener et al. 2008). The connection allows retrieval of movement data from all traffic participants, ILDs, and traffic lights. However, our experiments' target was the Floating Car Data (FCD) from the passenger vehicles category. Still, one could subscribe to all traffic participants, including bicycles, motorcycles, pedestrians, buses, trucks, and trams.

The next step was to define the specific variables we wanted to retrieve for each vehicle at each simulation step. These were the vehicle's position in geographic coordinates (x, y, z), angle, slope, time and speed.

⁵ <https://nodered.org/>.

The next step was to create a CZML file to store each vehicle's timestamps, positions, and orientations in the running simulation. The CZML file structure is made up of two main parts. The first part is a header object with an ID, name, CZML version, and clock property. The clock property is an object that specifies the period covered by the file and the starting time of the animation. In the second part, individual vehicles were added as objects called packets. Each packet contains the ID, name, description, optional 3D model path, position in geographic coordinates, and the orientation of the 3D model defined in unit quaternions at each time step. Unit quaternions are mathematical notations for representing spatial orientations in 3D space and are widely used in animation and simulation applications. A similar study by Beil et al. (2022) details the workflow required to convert SUMO simulation results into CZML format. We adopted the process in our Python tool to calculate the orientation of vehicle 3D models using positional coordinates, angles, and slopes. The CZML structure is extensible, with the option to add custom properties. We included the vehicle speed at each time step as a custom property. Figure 4 shows a snapshot of the created CZML dataset.

GIS Data

The LOD2 3D building models in CityGML 2.0 format were converted using FME into 3D Tiles format. For the semantic street space model and the road markings, we first converted the datasets from shapefile format into CityGML and then into 3D Tiles using the FME software. The street space model conversion follows the CityGML 3.0 Transportation Model, allowing a detailed representation of traffic spaces (Beil et al. 2020).

```

1 [
2   {"id": "simulation001",
3     "version": "1.0",
4     "name": "SUMOTrafficSimulationOutput",
5     "clock": {"interval": "2023-04-03T07:00:00Z/2023-03-31T07:10:00Z",
6               "currentTime": "2023-04-03T07:00:00Z",
7               "multiplier": 1}
8   },
9   {"id": "18444",
10    "name": "passenger",
11    "model": {"gltf": "./3Dmodels/CesiumMilkTruck.glb"},
12    "position": {"epoch": "2023-04-03T07:00:00Z",
13                "cartographicDegrees": [ 44.0, 11.56826, 48.140172, 0,
14                                          45.0, 11.568173, 48.140219, 0 ] },
15    "orientation": {"epoch": "2023-04-03T07:00:00Z",
16                    "unitQuaternion": [ 44.0, 0.3386128257985951, -0.1138048884613862, 0.8076963287179619, -0.4690591030376744,
17                                          45.0, 0.33195797886865874, -0.13196145270335846, 0.7811314654557384, -0.5120778348462516 ] },
18    "properties": { "speed": {"epoch": "2023-04-03T07:00:00Z",
19                             "number": [44.0, 22.5,
20                                         45.0, 21.2]}}
21  },
22  {"id": "18445", ...
23  }
24  ]
25 ]

```

Fig. 4 A snippet of the CZML document created from SUMO traffic simulation results

Dynamic 3D Models

Beyond visualising the static semantic properties of the 3D city models, our experiment explores adding time-dependent properties to the 3D models to show their temporal variation. The driving lanes from the street space model in shapefile format were selected and extracted for this experiment. The traffic states at each driving lane are dynamic and could be updated by fetching real-time data from ILDs installed in the lanes. We needed, therefore, to establish links between the dynamic properties of each lane and the sensor data measured by ILDs and stored in our SensorThings API server.

One limitation of our dataset is that not all lanes had ILDs installed. The lanes with no ILDs linked would lead to non-smooth visualisation of dynamic properties. For complete coverage of the traffic space with dynamic data, we currently make the simplifying assumption that the traffic state of a lane without a linked ILD is the same as that of a preceding lane.

The next step in the process was to add links to the ILDs data in SensorThings API. For this, we use a Python script to extract Datastream IDs for the ILDs that belong to the lanes and create a CSV file. Three new columns were added for each traffic state descriptor in the driving lanes datasets: speed, flow, and occupancy. Two more fields for the SensorThings API server HTTP and MQTT URL addresses were added and populated with the respective links. In the final step, we extended the driving lanes attribute by adding Datastream IDs from the CSV file through a spatial join based on the ILD ID field. Figure 5 shows a simplified view of the attributes table structure after processing. Once we extended the driving lanes dataset with the dynamic traffic state descriptor fields, we converted the dataset from the shapefile format into CityGML 3.0 and then to 3D Tiles format for visualisation.

3.3 Experiment Implementation

UDT Web Viewer Prototype

A UDT web viewer prototype was developed to visualise the semantic 3D models and the integrated dynamic data from traffic sensors. The application was built using *React*,⁶ a popular modern JavaScript framework for creating user interfaces. In addition to *CesiumJS*, the following JavaScript libraries were used:

1. *Resium*⁷—a library for creating React components for Cesium.
2. *MQTT.js*⁸—a client library for the MQTT protocol on browser-based environments.
3. *Redux*⁹—a state management library for JavaScript applications.

⁶ <https://react.dev/>.

⁷ <https://resium.reearth.io/>.

⁸ <https://github.com/mqttjs/mqtt.js>.

⁹ <https://react-redux.js.org/>.

Fig. 5 A simplified attribute data table structure of the driving lanes datasets. The fields in red contain SensorThings API HTTP and MQTT URLs. The blue fields indicate the dynamic properties, while the green fields store the SensorThings API *Datastream* IDs for each dynamic property

| Driving Lane |
|-----------------------------------|
| id |
| class |
| function |
| usage |
| <i>ild_id</i> |
| <i>sensorthingsapi_baseurl</i> |
| <i>sensorthingsapi_mqttserver</i> |
| speed |
| <i>speed_datastream</i> |
| flow |
| <i>flow_datastream</i> |
| occupancy |
| <i>occupancy_datastream</i> |

Datasets Configuration

In this step, we created a layer's configuration file that defined how different datasets would be loaded and handled in the application. For the 3D Tiles, we create an object array that holds the different datasets. Figure 6 shows an example of an object defining the Traffic layer created using the driving lanes data.

The properties of the object in Fig. 6 are described in Table 1. The only mandatory parameters in the configuration object are *layerID*, *URL*, and *featureId*.

The *dynamicProperties* configuration object links 3D feature properties to their associated SensorThings API *Datastreams* by implementing the CityGML Dynamizer concept in the web visualisation workflow. 3D Tiles, by default, do not support temporal data, but when loaded in *Cesium as Cesium3DTiles* layers, their properties can be updated using the *setProperty* function. We define the styling rules based on a dynamic property to visualise temporal data. The workflow for updating the dynamic properties is described in the following section.

Process Flow

Once the 3D Tiles are loaded as *Cesium3DTiles* objects in the web application, the individual features are iterated and stored in the application's state. The application fetches the sensor data from the SensorThings API server using the base URLs and Datastream IDs specified in the configuration file for features containing dynamic properties. Sensor observations are fetched for each feature using a dynamic URL containing time filters for a predefined period and stored in the application's state. The data is fetched asynchronously to allow other processes to run in parallel. If a

```

1  { layerID: 'Traffic',
2    url: process.env.PUBLIC_URL + "/geodata/TrafficFlow/tileset.json",
3    featureID: 'id',
4    icon: <TrafficRounded fontSize="medium" />,
5    show: true,
6    heightOffset: 0.3,
7    dynamicProperties: [
8      { property: "flow", datastream: "flow_datastream", description: "Traffic Flow", symbol: '',
9        sensorthingsapiBaseUrl: 'https://sta/frost/v1.1', sensorthingsapiMQTT: 'mqtt://sta:8883', },
10
11     { property: "occupancy", datastream: "occupancy_datastream", description: "Traffic Occupancy", symbol: '%',
12       sensorthingsapiBaseUrl: 'https://sta/frost/v1.1', sensorthingsapiMQTT: 'mqtt://sta:8883', },
13
14     { property: "speed", datastream: "speed_datastream", description: "Traffic Speed", symbol: 'km/h',
15       sensorthingsapiBaseUrl: 'https://sta/frost/v1.1', sensorthingsapiMQTT: 'mqtt://sta:8883', }
16   ],
17   style: new Cesium3DTileStyle({
18     defines: { occupancy: "${feature['occupancy']}"},
19     color: { conditions: [
20       ["${occupancy} > 15", "color('#FF0000', 0.2)"],
21       ["${occupancy} > 10", "color('#FFFF00', 0.2)"],
22       ["${occupancy} >= 0.1", "color('#00FF00', 0.2)"],
23       ["true", "color('#ffffff', 0.1)"],,],,
24     }},
25   },

```

Fig. 6 3D Tiles layer configuration object

Table 1 Description of the configuration layer parameters

| Property | Description |
|-------------------|--|
| LayerID | An ID used to manage the data layer in the application |
| URL | Path to the 3D Tiles data |
| FeatureID | Feature attribute used for selecting and picking |
| Icon | An icon for the layers’ navigation panel |
| Show | Indicates whether the layer will be displayed by default |
| HeightOffset | Applies a height offset to 3D Tiles in metres |
| DynamicProperties | A <i>dynamizer</i> object for layers with features containing dynamic properties |
| Style | Defines the styling rules based on Cesium’s 3D styling language |

height offset is specified in the configuration object, the 3D Tiles layer elevation is adjusted in metres to avoid overlap with other layers.

Once the data is stored, a custom function filters the data based on the current Cesium clock time. It updates the corresponding 3D tile feature with the dynamic property or properties from the filtered dataset. For each feature with a predefined style based on a dynamic property, the appearance updates according to the new property value. The diagram in Fig. 7 shows the process flow for visualising 3D Tiles with dynamic properties.

User Interface and Interaction

The web application’s user interface provides the functionalities required to interact with the datasets. The default *Cesium Timeline* widget controls and displays the

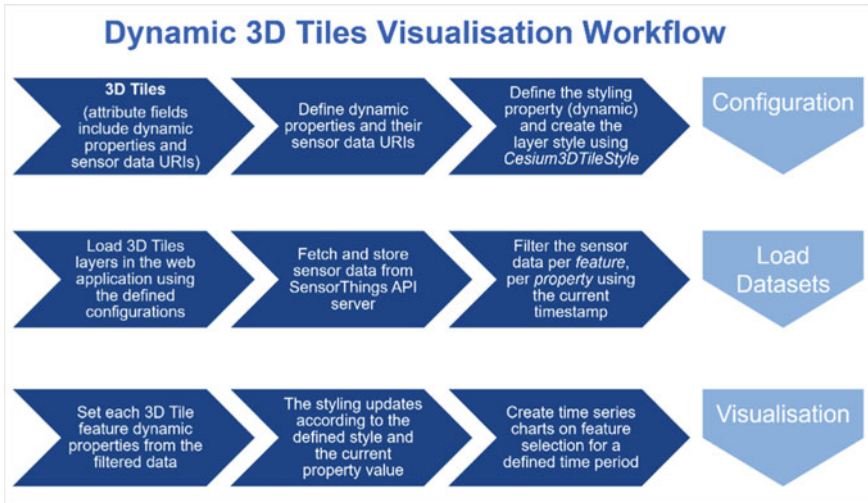


Fig. 7 Web application process flow for visualising 3D city models in 3D Tiles format and their dynamic properties

scene time. The widget connects to the application's state, so time selection triggers property updates on 3D features with dynamic properties.

A menu button opens a navigation panel that allows users to toggle layer visibility. In the same panel, users can select a time range for the scene by specifying the start and end timestamps. Updating the timestamps triggers an update of sensor data in the background, and the *Cesium Timeline* is updated, too, to reflect the changes.

The application shows a custom information box with a panel of two tabs for features with dynamic properties. The first tab shows a table with the current dynamic property values and time series charts for each property for the entire time range. The second tab shows all the semantic properties of a feature. For features with no dynamic properties, only one tab with a table of properties is displayed. We have used two visual variables for the driving lanes' traffic flow: colour and transparency. The colour variable indicates the level of occupancy levels, while the transparency, in combination with the height offset, allows visualising the underlying street model and road markings. The application demonstrates a 4D visualisation workflow by dynamically updating 3D Tiles properties. By using the *Cesium Timeline* to update the 3D Tiles' properties, additional datasets, such as CZML, could also be visualised simultaneously. Figure 8 shows a screenshot of the viewer.

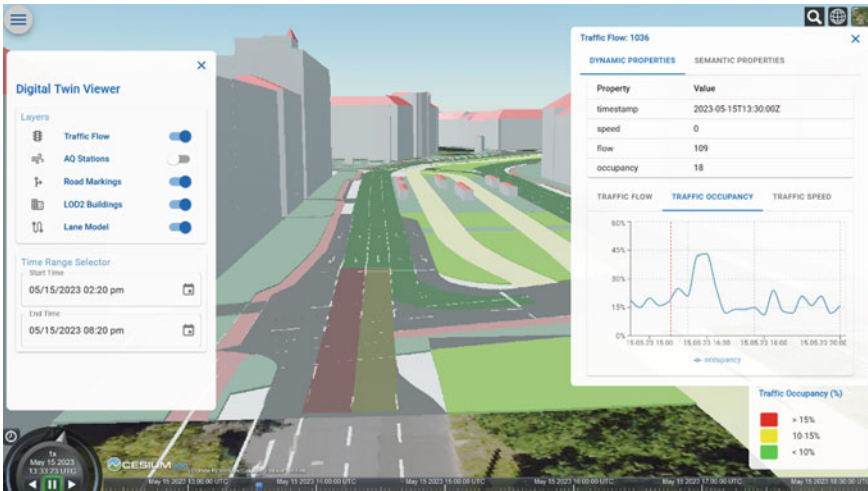


Fig. 8 The user interface for the prototyped UDT web viewer. The driving lanes’ traffic occupancy levels retrieved from the SensorThings API are visualised overlaying the street space model

4 Discussion

4.1 Data Integration Requirements

The first step towards visualising temporal changes in UDTs is integrating 3D city models with sensor and simulation data. Section 2.1. discusses the different approaches used for data integration. Sensor and simulation data could be integrated into 3D models as attributes or fetched dynamically from sensor data repositories. Combining sensor data into 3D models in advance is not feasible as the data is highly dynamic and often large depending on the observed period. Strategies that create explicit links between 3D models and sensor data are thus preferred. The CityGML 3.0Dynamizer module could further exploit the 3D city models and sensor relationships to determine the appropriate visual variables for 4D visualisations.

4.2 Technical Requirements

The data formats’ capabilities and limitations determine their applicability in visualising time-dependent properties in UDTs. Our experiment used the 3D Tiles format for stationary 3D objects and the CZML format to represent moving objects in the CesiumJS-based viewer. Further visualisation types may be required based on the 3D models and sensor data categories. Different chart types, text labels, map icons, and markers could be used to visualise the dynamic properties of 3D models.

Visualising 3D graphics on the web comes at a performance cost due to the high computing power and memory required to load 4D datasets. Adding sensor data increases the data load and computing resources demand. This could result to slow load times and lagging while navigating through a 3D scene. Data fetching, caching, and rendering optimisation strategies are thus critical to improving the performance of UDTs' web applications.

4.3 User Experience Requirements

The needs of end users who interact with UDTs through user interfaces must be met while developing dynamic visualisations. The user interfaces should be easy to navigate, highlight important information, and present data in a visualisation format consistent with user expectations. To further enhance the user experience when portraying 4D web scenes, the choice and placement of visual variables should consider the user's perspective and orientation of 3D models. Different visual variables could be used to display the same dynamic property of a 3D model, depending on the zoom level or camera position.

5 Conclusion and Future Work

This study has investigated different requirements for visualising dynamic UDTs and presented a workflow for developing a web viewer. As discussed in the previous chapter, 4D web visualisation design must consider the data integration, technical, and user requirements. While visualisation requirements and approach focus on semantic 3D city models and sensor data in this study, they also apply to visualising dynamic properties in Building Information Models (BIM).

In the future, more conceptual research is required to extend the data models of the 3D Tiles and I3S visualisation formats to support temporal data and spatiotemporal tiling. As web rendering performance is critical to smooth 4D visualisations, more research on 4D tiling strategies is required. Further technical development is needed to interpret CityGML Dynamizers contained within 3D city models and automatically derive corresponding data structures linking sensor services with the 3D objects used within the 4D web viewer. Above, more basic research will be required regarding the question of good ways to incorporate dynamic data visualisation into 3D scenes. How to leverage the semantic information provided in the 3D city model and different kinds of dynamic data streams? This information should be used to decide, e.g., the style, the usage of visual variables and the 3D/4D placement of visualisations of time series data.

Acknowledgements We thank the City of Munich for the cooperation in the Connected Urban Twins (CUT) project funded by the Federal Ministry for Housing, Urban Development and Building

of Germany. We also thank the city's Geodata Service and Mobility departments for providing datasets used in this study.

References

- Beil C, Ruhdorfer R, Coduro T, Kolbe TH (2020) Detailed streetspace modelling for multiple applications: discussions on the proposed CityGML 3.0 transportation model. *IJGI* 9(10):603. <https://doi.org/10.3390/ijgi9100603>
- Beil C, Kendir M, Ruhdorfer R, Kolbe TH (2022) Dynamic and web-based 4D visualization of streetspace activities derived from traffic simulations and semantic 3D city models. *ISPRS Ann Photogramm, Remote Sens Spatial Inform Sci*, Copernicus GmbH, 29–36. <https://doi.org/10.5194/isprs-annals-X-4-W2-2022-29-2022>
- Bertin J (1981) *Graphics and graphic information-processing*. de Gruyter
- Chaturvedi K (2021) Integration and management of time-dependent properties with semantic 3D city models. Doctoral dissertation, Technische Universität München [Online]. Available: <https://mediatum.ub.tum.de/?id=1542959>
- Chaturvedi K, Kolbe TH (2018) InterSensor service: establishing interoperability over heterogeneous sensor observations and platforms for smart cities. In: 2018 IEEE International Smart Cities Conference (ISC2), Sep, pp 1–8. <https://doi.org/10.1109/ISC2.2018.8656984>
- Chatziniolaou E, Pispidikis I, Dimopoulou E (2020) A semantically enriched and web-based 3d energy model visualization and retrieval for smart building implementation using CityGML and Dynamizer ADE. *ISPRS Annals Photogramm Remote Sens Spatial Inform Sci* VI-4/W1:53–60. <https://doi.org/10.5194/isprs-annals-VI-4-W1-2020-53-2020>
- Cirillo F, Solmaz G, Berz EL, Bauer M, Cheng B, Kovacs E (2019) A standard-based open source IoT platform: FIWARE. *IEEE Internet Things* 2(3):12–18. <https://doi.org/10.1109/IOTM.0001.1800022>
- Cozzi P, Lilley S, Getz G (2019) 3D tiles specification 1.0. Open Geospatial Consortium [Online]. Available: <https://docs.ogc.org/cs/18-053r2/18-053r2.html>
- Döllner J, Buchholz H, Nienhaus M, Kirsch F (2005) Illustrative visualization of 3D city models. Presented at the Electronic Imaging 2005, Erbacher RF, Roberts JC, Grohn MT, Borner K (eds), San Jose, CA, p 42. <https://doi.org/10.1117/12.587118>
- Döllner J, Baumann K, Buchholz H (2006) Virtual 3D city models as foundation of complex urban information spaces
- Ebrahim H, Santhanavanich T, Wuerstle P, Coors V (2021) Concept and evaluation of an urban platform for interactive visual analytics. *ISPRS Ann Photogramm Remote Sens Spatial Inf Sci* VIII-4/W1:33–40. <https://doi.org/10.5194/isprs-annals-VIII-4-W1-2021-33-2021>
- Ferré-Bigorra J, Casals M, Gangolells M (2022) The adoption of urban digital twins. *Cities* 131:103905. <https://doi.org/10.1016/j.cities.2022.103905>
- Fischer M, Gras P, Löwa S, Schuhart S (2021) Urban data platform hamburg: integration von Echtzeit IoT-Daten mittels SensorThings API. In: *ZfV—Zeitschrift für Geodäsie, Geoinformation und Landmanagement*, vol 1, p 47. <https://doi.org/10.12902/zfv-0330-2020>
- Gaigg M (2023) *Designing map interfaces: patterns for building effective map apps*, 1st edn. Esri Press, Redlands
- Gröger G, Kolbe TH, Nagel C, Häfele K-H (2012) OGC city geography markup language (CityGML) encoding standard
- Hardisty F, MacEachren A, Takatsuka M (2001) Cartographic animation in three dimensions: experimenting with the scene graph. Presented at the 20th International Cartographic Conference, Beijing, China: geovista.psu.edu [Online]. Available: https://icaci.org/files/documents/ICC_proceedings/ICC2001/icc2001/file/f17005.pdf

- Kolbe TH (2009) 'Representing and exchanging 3D city models with CityGML. In: Lee J, Zlatanova S (eds) 3D geo-information sciences. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 15–31. https://doi.org/10.1007/978-3-540-87395-2_2
- Kutzner T, Chaturvedi K, Kolbe TH (2020) CityGML 3.0: new functions open up new applications. PFG 88(1):43–61. <https://doi.org/10.1007/s41064-020-00095-z>
- Li B, Luo Z, Mao B (2022) Non-photorealistic visualization of 3D city models using visual variables in virtual reality environments. Procedia Comp Sci 214:1516–1521. <https://doi.org/10.1016/j.procs.2022.11.338>
- Liang S, Huang C-Y, Khalafbeigi T (2016) OGC SensorThings API part 1: sensing
- Lopez PA et al (2019) Microscopic traffic simulation using SUMO. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Maui, USA: IEEE, pp 2575–2582. Accessed: May 2, 2023 [Online]. Available: <https://www.itsc2019.org/>
- Mao B, Ban Y, Laumert B (2020) Dynamic online 3D visualization framework for real-time energy simulation based on 3D Tiles. ISPRS Int J Geo Inf 9(3):166. <https://doi.org/10.3390/ijgi9030166>
- Rautenbach V, Coetzee S, Schiewe J, Cöltekin A (2015) An assessment of visual variables for the cartographic design of 3D informal settlement models. Presented at the 27th International Cartographic Conference, Rio de Janeiro, Brazil: Maps Connecting the World, Aug. <https://doi.org/10.5167/UZH-117989>
- Reed C, Belayneh T (2022) OGC Indexed 3d Scene Layer (I3S) and scene layer package (*.slpk) format community standard. Open Geospatial Consortium. Accessed: Jul. 23, 2023 [Online]. Available: <https://docs.ogc.org/cs/17-014r9/17-014r9.html>
- Santhanavanich T, Coors V (2021) CityThings: an integration of the dynamic sensor data to the 3D city model. Environment Planning B: Urban Analyt City Sci 48(3):417–432. <https://doi.org/10.1177/2399808320983000>
- UN-Habitat (2022) World cities report 2022: envisaging the future of cities. United Nations Human Settlements Programme (UN-Habitat), Nairobi, Kenya [Online]. Available: https://unhabitat.org/sites/default/files/2022/06/wcr_2022.pdf
- Wegener A, Piórkowski M, Raya M, Hellbrück H, Fischer S, Hubaux J-P (2008) TraCI: an interface for coupling road traffic and network simulators. In: Proceedings of the 11th communications and networking simulation symposium, in CNS '08. New York, NY, USA: Association for Computing Machinery, Apr., pp 155–163. <https://doi.org/10.1145/1400713.1400740>
- Yao Z et al (2018) 3DCityDB—a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. Open Geospat Data, Soft Stand 3(1):5. <https://doi.org/10.1186/s40965-018-0046-7>

Investigation of CityGML 3.0 for Modelling Temporal Aspects in Underground Land Administration



**Bahram Saeidian, Abbas Rajabifard, Behnam Atazadeh,
and Mohsen Kalantari**

Abstract Rapid urbanisation and limited land availability have led to increased consideration of underground spaces. This increased utilisation of subterranean space has created a rise in its value, highlighting the significance of ownership of underground areas. A fully-integrated digital model that effectively represents underground space ownership is vital for communicating rights, restrictions, and responsibilities (RRRs) in underground spaces. Underground assets are often built and developed vertically at different time slots, resulting in changes to the legal ownership of underground spaces. Therefore, underground RRRs change over time due to factors such as new subdivisions, consolidations, boundary reconstructions, and land acquisitions. Current practices use 2D survey plans and property base maps to manage and communicate underground RRRs, with some textual notations as well as attributes providing temporal information. However, these methods have limitations, and studies have explored the use of 3D models to address these issues, though they often neglect the temporal aspects. Some studies have investigated 4D cadastre (3D cadastre + time) in different jurisdictions, but these studies mainly remain at a conceptual level for above-ground parcels and buildings. CityGML 3.0 is a data model that provides 3D geometries, topologies, semantics, and entities for modelling temporal aspects such as representing dynamic data and time series and maintaining spatial objects' history and versions. This paper explores the temporal aspects of underground land administration (ULA) and investigates CityGML 3.0 entities for modelling these temporal

This article was selected based on the results of a double-blind review of the full paper.

B. Saeidian (✉) · A. Rajabifard · B. Atazadeh

The Centre for Spatial Data Infrastructures and Land Administration, Department of Infrastructure Engineering, The University of Melbourne, Parkville, VIC, Australia
e-mail: bsaeidian@student.unimelb.edu.au

A. Rajabifard

e-mail: abbas.r@unimelb.edu.au

B. Atazadeh

e-mail: behnam.atazadeh@unimelb.edu.au

M. Kalantari

School of Civil and Environmental Engineering, UNSW, Sydney, NSW, Australia
e-mail: mohsen.kalantari@unsw.edu.au

aspects, with a synthetic prototype implemented as a proof of concept to demonstrate the applicability of a 4D ULA model. The result indicates that considering the temporal aspects of ULA using CityGML 3.0 entities can improve the functionality and capability of a land administration model in managing and communicating ownership information in underground areas.

Keywords Underground · Temporal · 4D cadastre · Land administration · CityGML

1 Introduction

Rapid urbanisation coupled with limited land availability has necessitated the use of underground areas. Infrastructure such as utilities and tunnels located underground is critically important for the efficient functioning of cities as well as essential urban amenities and facilities. As shown in Fig. 1, there exists a range of underground assets such as tunnels, utilities, train stations, walkways, and basements, with complex geometries and large spatial extents (DAUB 2019; Saeidian et al. 2021). The growing reliance on underground spaces has increased the value of such areas and has made the ownership of underground spaces increasingly important. Multi-dimensional (3D/4D) digital models can potentially provide an effective approach to manage and communicate ownership rights, restrictions, and responsibilities (RRRs) in subterranean spaces.

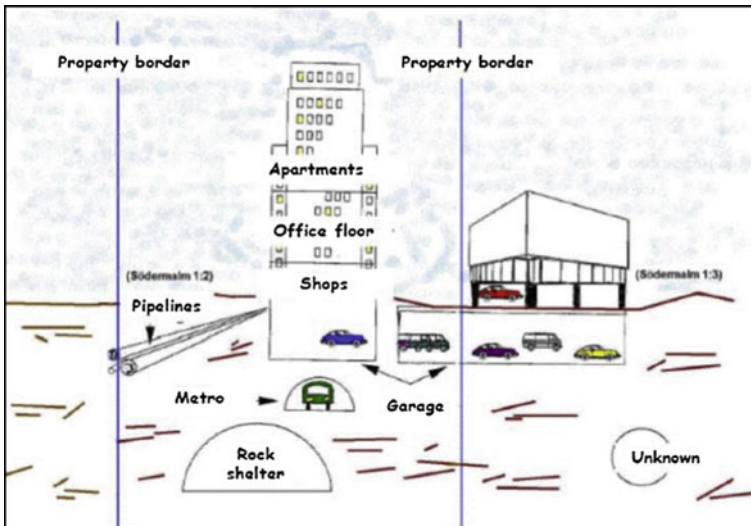


Fig. 1 Different underground assets with complex geometries compared to the above ground (Larsson et al. 2020)

As shown in Fig. 1, underground assets, such as pipelines, tunnels, or underground facilities, are often designed and developed in a vertical orientation, meaning they are constructed with multiple levels or layers stacked on top of each other, extending deeper into the ground. These assets are also built at different time slots, resulting in changes to the spatial extent of ownership RRRs when new infrastructure is developed. For example, establishing or removing underground utility networks over time changes the legal spaces associated with them or land acquisition for underground projects like tunnels changes the ownership interests from private to government-owned land. Therefore, underground RRRs change over time due to factors such as new subdivisions, consolidations, boundary reconstructions, and land and property acquisitions. Incorporating temporal data within a cadastral system would facilitate the monitoring of changes that take place over a period of time (Sürmeneli et al. 2022a). An integrated 4D model for underground land administration (ULA) is required to record not only the changes in the physical reality of underground assets but also to communicate how the underground legal ownership spaces associated with them are changed over time (Döner et al. 2010).

In the current practices, 2D survey plans and property base maps are used to manage and communicate the ownership RRRs in underground areas (Atazadeh et al. 2022). Survey plans include floor plans, cross-sectional diagrams, isometric representations, and textual notations to represent underground RRRs. These 2D survey plans also use textual notations and attributes to provide some temporal information such as easement creation/removal dates or subdivision/consolidation dates. This traditional approach provides a fragmented and siloed approach for describing the relationship between the time dimension and the 2D/3D spatial extent of underground RRRs (Döner et al. 2010).

The development and use of 3D models have been considered in different studies during the last two decades (Saeidian et al. 2021, 2022a, 2023a; Larsson et al. 2020; Atazadeh 2017), with less emphasis on the temporal aspects. A few studies investigated 4D cadastre (3D cadastre + time) in different jurisdictions. However, these studies are mostly at a conceptual level and investigate above-ground buildings, with less emphasis on underground cases. Effective management and utilisation of underground data depend on models and schemas to organise them (Lieberman and Roensdorf 2020; Saeidian et al. 2023b). A data model plays a fundamental role in developing an integrated model of underground assets and RRRs associated with them in a 4D environment, considering 3D spatial and temporal aspects (Saeidian et al. 2022a).

CityGML is a prominent city-scale data model that can represent different natural and artificial objects and logical spaces that have virtual boundaries like legal spaces in an integrated 3D environment. It also provides 3D geometries, topologies, semantics, and some entities for modelling temporal aspects such as representing dynamic data and time series and maintaining spatial objects' history and versions. This paper aims to explore the temporal aspects of ULA and investigates the CityGML 3.0 entities for modelling these temporal aspects, with a synthetic prototype implemented as a proof of concept to demonstrate the applicability of a 4D integrated ULA model.

2 Temporal Modelling in Literature

The topic of integrating time as an element in land administration systems has been addressed limitedly in the current literature, and it has been conceptually investigated in a few jurisdictions (Alberdi and Erba 2020).

Van Oosterom et al. (2006) investigated the necessity of a four-dimensional (4D) cadastral system by including time-related elements. This study represents an initial analysis and does not provide a conclusive solution. Nevertheless, the authors suggested that a 4D cadastre could be a desirable option. The authors reported that most land interests have a temporal component, and while separate spatial and time attributes can represent them, an integrated model of 3D space and time could provide benefits for future 4D cadastre development. The study concluded that the temporal aspects of land administration required in-details investigations.

Sürmeneli et al. (2022a, b) developed a 4D cadastral data model in the context of Turkey. In the first study, an LADM-based country profile was developed for 3D cadastral object registration with time attributes (3D + time). In the second study, the LADM-based country profile is linked with CityGML using an Application Domain Extension (ADE) for the representation of physical data along with the legal objects and their temporal aspects as time attributes. As the LADM standard only supports legal data and uses external sources for physical data, the first paper is limited to legal data. The second study proposes an integrated model in the context of Turkey, but it needs to be updated based on jurisdictional requirements and the latest version of CityGML. In these studies, the time information is defined using LADM entities. The *VersionObject* class of LADM defines time information (the start and end time/lifespan of objects as timestamps) and can be inherited by all LADM classes with a few exceptions.

Thompson and van Oosterom (2021) discussed event-based and state-based modelling of temporal aspects of 2D/3D land administration data. Since LADM partly provides bitemporal modelling, they proposed an extension for LADM to include bitemporal support, which allows for the representation of both the current and past states of land parcels as well as the events that led to changes in their status. The extension proposes a new way to model temporal aspects of land administration data that combines both event-based and state-based approaches. The concepts of bitemporal, event-based, and state-based modelling are explained in the next sections.

Döner et al. (2010, 2011) discussed the registration of utility networks in 3D cadastre considering temporal aspects in three jurisdictions (Queensland state of Australia, The Netherlands, and Turkey) in terms of legal, institutional, and technical aspects. The first work studied the conceptual basis of a 4D cadastre for utility networks. However, the second work developed a prototype in the context of the Netherlands. The conceptual bases of these studies are based on LADM. These studies concluded that the 4D registration of utility networks is possible and sufficient using 3D geometry data types and separate attributes for temporal aspects (changes).

Döner and Biyik (2013) investigated LADM for developing a data model for 3D/4D cadastre in Turkey. The paper discussed different underground cases such as

utilities and historical sites. However, the developed model is a country profile for Turkey and as authors reported, such 3D/4D models depend on the legal systems and technical requirements in jurisdictions. Babalola et al. (2015) also reviewed the studies that used LADM and its entities for modelling temporal aspects of land administration. This study concluded that although adding the fourth dimension (time) benefits land administration, governments, as well as land users, the works on the temporal aspects are limited, and further research are required.

Finally, Siejka et al. (2014) investigated the process of transition from 2D cadastres to 4D cadastres and the possibility of implementing a 4D cadastre (3D + time) in Poland. They concluded that adding the time dimension to a land administration model can enhance its efficiency, but it needs in-detail studies in all aspects of land administration, including legal, institutional, and technical aspects.

These studies mostly considered above-ground cases or specific underground assets. In addition, they were mostly at a conceptual level and considered only legal aspects using LADM entities. In this regard, this paper will investigate the development of a data model for 4D ULA by adding the fourth dimension (time) to a 3D integrated physical-legal model (3D integrated model + time). Saeidian et al. (2021) discussed three methods for developing an integrated data model for ULA and concluded that extending a leading data model that supports temporal aspects with legal information is a potential solution for a fully integrated 4D ULA. Wilhelm et al. (2021) compared different data models based on different criteria. Among the criteria considered, extensibility, 3D modelling, and temporal aspects are important for developing a data model for 4D ULA. Table 1 presents the assessment done by Wilhelm et al. (2021) for these parameters. CityGML and IFC are among the most prominent data models that cover not only temporal aspects but also 3D modelling. CityGML presents a semantic city-scale model, making it suitable to model large underground assets such as utility networks and tunnels. This data model is also more extensible compared to IFC which is important for defining legal spaces since these data models do not support legal spaces and their attributes and it is necessary to enrich them. Therefore, this research aims to investigate CityGML 3.0 to support the development of a data model for 4D ULA. In this regard, the Spatio-temporal aspects of ULA will be identified and CityGML 3.0 potentials and possible extensions to cover these aspects will be discussed.

Table 1 Comparison between data models to use for 4D ULA (Wilhelm et al. 2021)

| Criteria | CityGML | IFC | INSPIRE | LandInfra |
|---------------|---------|-----|---------|-----------|
| Time: static | + | ++ | + | + |
| Time: dynamic | ++ | + | – | – |
| 3D | ++ | ++ | + | ++ |
| Extensibility | ++ | + | + | + |

++ comprehensive

+ basic

– not included

3 Spatio-Temporal Aspects of ULA

This study will focus on the Victorian jurisdiction of Australia. However, the identified aspects and the investigation of CityGML to model these aspects can be replicated by other regions. In the current practice of Victoria, different types of underground data are required to be provided.

The first type is ownership interests (legal spaces) defined for underground assets. These legal spaces are classified into primary parcels and secondary interests (Saeidian et al. 2022b). Primary parcels include the fabric legal spaces and secondary interests provide benefits or pose restrictions on the primary parcels. The second type of legal data in the current practice is legal boundaries that delineate the spatial extent of legal spaces. In addition to the legal spaces and boundaries, the survey elements like survey points, observations and elevation/depth information are also provided. These elements are the result of a cadastral survey work used to connect legal spaces to a geodetic survey network.

These three types of data (legal spaces and boundaries and survey elements) are related to the legal aspects of underground areas. An integrated model needs to provide not only all legal objects but also physical assets in underground environments. Physical elements do not themselves have ownership status; however, these elements can be used to facilitate communicating and managing spatial extent and arrangements of ownership elements (Atazadeh 2017). For instance, walls and slabs are used to define ownership boundaries in basements. Physical objects can help the communication and interpretation of legal objects. They can also be used to check for any inconsistency. Finally, an integrated physical-legal model can expand the applications of a city-scale model for ULA.

Underground physical and legal data can also have different administrative information such as actors and relevant documents. Underground physical, legal, and administrative data need to be integrated into a 3D environment with relevant semantics, geometries, and properties (e.g., attributes and relationships). In addition to these components, each legal, physical, and administrative element can change over time. Thus, the last aspect of a 4D ULA model is temporal aspects.

Figure 2 shows the conceptual data model to create an integrated 4D underground model, considering the requirements of underground legal spaces and boundaries, survey elements, physical assets, administrative data, and temporal aspects. As shown, all legal, physical, and administrative feature classes inherit time information from the *AbstractTemporalElement* class.

All ULA objects presented in Fig. 2 can change over time. These changes can be described as discrete and continuous changes. Discrete changes can be thought of as events or transactions that happen at specific moments in time, while continuous changes are ongoing processes that unfold over time. For example, changes in the location of underground trains can be represented as a continuous change, whereas changes in the ownership of a property would be represented as a series of discrete events. Both types of changes can be important to consider when modelling temporal aspects of ULA. The continuous changes are more common for the physical assets in

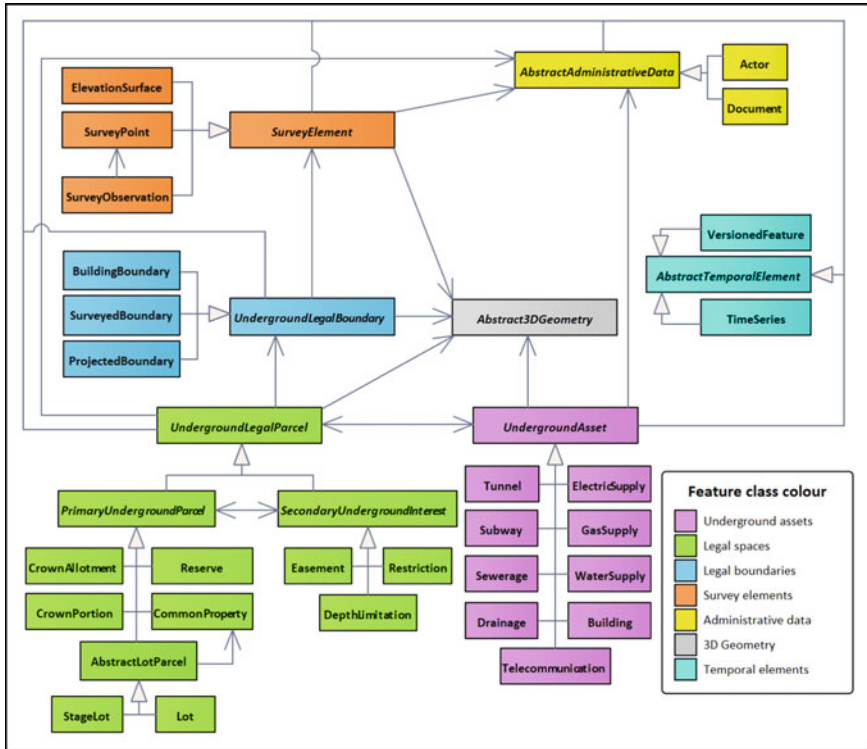


Fig. 2 The conceptual data model to create an integrated 4D underground model

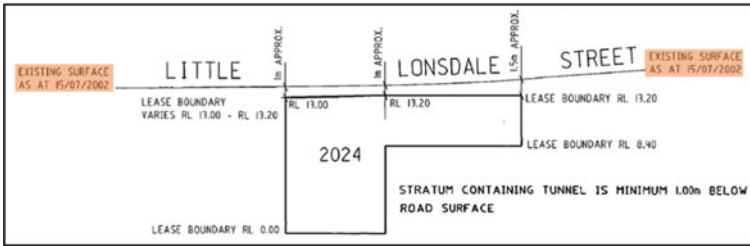
underground areas and can be modelled by time series. Legal data often have discrete changes that can be modelled through versioning.

The legal data changes can be related to the changes in ownership. For example, after establishing an underground utility network, the underground space occupied by the utility is defined as an easement, which is the legal right granted to an individual or entity, allowing them to use a portion of someone else’s land for a particular purpose. Removing an easement or expiring a restriction are other examples that change the ULA objects over time. Acquiring private underground spaces to construct an underground infrastructure such as a tunnel can also change the ownership from private land owned by an individual or private owner (defined as a lot in the context of the Victorian jurisdiction of Australia) to government-owned land (defined as a crown land in the context of Victoria). New subdivisions and consolidations also change the spatial extent and ownership information of underground areas over time. Changes can also be related to administrative information such as transactions on land (e.g., selling land). In addition, survey and legal data change over time because of boundary reconstructions. In the current practice, textual notations are used to specify temporal changes in 2D survey plans. Figure 3 shows some examples of temporal information in 2D survey plans.

| RECORD OF ALL ADDITIONS OR CHANGES TO THE PLAN | | | | | | | PLAN NUMBER PS523308E |
|---|--------------------------------|----------------------|----------------|----------|----------------|-------------------------------|--|
| WARNING: THE IMAGE OF THIS DOCUMENT OF THE REGISTER HAS BEEN DIGITALLY AMENDED. NO FURTHER AMENDMENTS ARE TO BE MADE TO THE ORIGINAL DOCUMENT OF THE REGISTER. | | | | | | | |
| AFFECTED LAND/PARCEL | LAND/PARCEL IDENTIFIER CREATED | MODIFICATION | DEALING NUMBER | DATE | EDITION NUMBER | ASSISTANT REGISTRAR OF TITLES | |
| LOT 2 | E-39 | CREATION OF EASEMENT | AK360651M | 23/05/13 | 2 | E.T.H | |

| RECORD OF ALL ADDITIONS OR CHANGES TO THE PLAN | | | | | | | PLAN NUMBER PC350024V |
|---|--------------------------------|---------------------|----------------|----------|----------------|-------------------------------|--|
| WARNING: THE IMAGE OF THIS DOCUMENT OF THE REGISTER HAS BEEN DIGITALLY AMENDED. NO FURTHER AMENDMENTS ARE TO BE MADE TO THE ORIGINAL DOCUMENT OF THE REGISTER. | | | | | | | |
| AFFECTED LAND/PARCEL | LAND/PARCEL IDENTIFIER CREATED | MODIFICATION | DEALING NUMBER | DATE | EDITION NUMBER | ASSISTANT REGISTRAR OF TITLES | |
| LAND HEREIN (E-2 & E-3) | - | REMOVAL OF EASEMENT | AQ726822W | 13/02/18 | 2 | G.LY | |

The creation and removal of underground easements for utilities over time

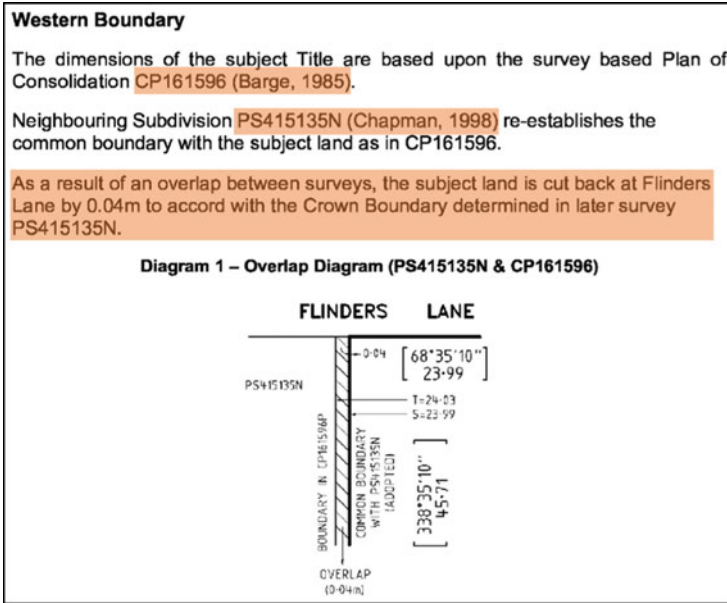


Defining underground legal boundaries and spaces using elements at a specific time

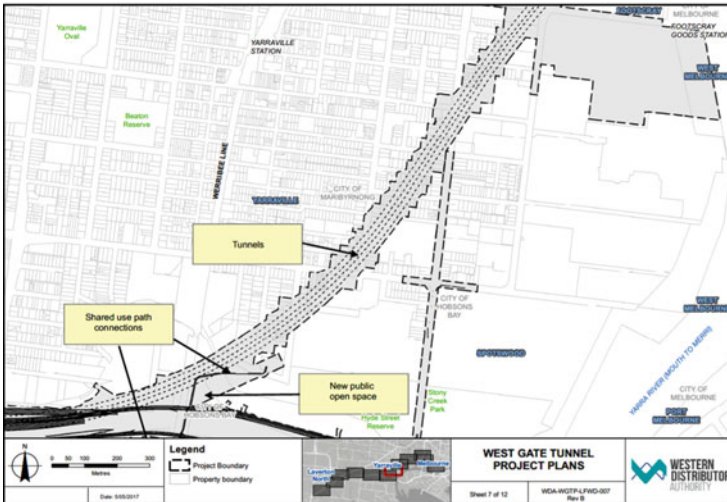
| |
|---|
| CREATION OF RESTRICTION No.2 |
| LAND TO BENEFIT: LOTS 1414 - 1419 (BOTH INCLUSIVE) |
| LAND TO BE BURDENED: LOTS 1414, 1415, 1418 AND 1419 |
| THE REGISTERED PROPRIETOR OR PROPRIETORS FOR THE TIME BEING OF ANY BURDENED LOT MUST NOT: |
| 1 BUILD OR ERECT OR PERMIT TO BE BUILT OR ERECTED ANY BUILDING THAT DOES NOT ALLOW AT LEAST 25m ² OF SECLUDED PRIVATE OPEN SPACE WITH DIMENSION OF AT LEAST 4m AT THE SIDE OR REAR OF THE BUILDING WITH CONVENIENT ACCESS FROM A LIVING AREA |
| 2 BUILD OR ERECT OR PERMIT TO BE BUILT OR ERECTED A GARAGE ON THE BURDENED LOT WITH A SETBACK FROM THE FRONT BOUNDARY OF LESS THAN 5.50m AND SAID GARAGE NOT BUILT TO EXCEED 6.5m IN LENGTH. |
| THIS RESTRICTION SHALL CEASE TO HAVE EFFECT 10 YEARS AFTER THE DATE OF REGISTRATION OF THIS PLAN |

The expiry date of a restriction

Fig. 3 Some examples of changes in legal information over time in 2D survey plans



Re-establishing legal boundaries over time



The underground land acquisition changes the ownership of spaces occupied by the tunnel project from private lands (lots) before the project to crown (government-owned) lands after the project (Department of Planning, 2017)

Fig. 3 (continued)

When modelling discrete temporal changes, two approaches are used: event-based modelling and state-based modelling. In event-based modelling, transactions/events are modelled as distinct entities within the system with their unique identity and attributes. It is possible to reconstruct every past state by traversing the whole chain of events when the start state and all events are known. It is also possible to represent the current state without keeping the start date and going back in time via the “reversal” of events. In contrast, state-based modelling explicitly models the states, i.e., the results. Each object has at least two dates/times that indicate the time interval during which the object is valid. Comparing two succeeding states makes it possible to reconstruct what happened as a result of one specific event. Obtaining the state at a given time is easy by selecting objects based on their time interval (ISO 2012).

Based on the examples provided in Fig. 3, different types of entities are required to include the temporal aspects of ULA objects in a 4D integrated underground model. The start date of each ULA object needs to define as the registration time of the object. In addition, some ULA objects such as restrictions or the original versions of parcels before subdivisions/consolidations have an end date. These cases can be modelled by either the start and end dates of the object or the start date and the period in which the object is valid like the restriction presented in Fig. 3 that is valid for 10 years. These entities define timestamps of objects which is state-based modelling. These timestamps can be used to manage the history of objects, execute temporal queries, and retrieve the ULA data at a specific time. The next section will investigate CityGML 3.0 to model ULA objects in an integrated 4D model and their discrete and continuous changes over time. The entities provided by CityGML 3.0 will also be investigated for event-based and state-based modelling of ULA discrete changes.

4 Investigation of CityGML 3.0 for 4D ULA

The developed CityGML-based data model for 4D ULA requires supporting all the aspects presented in the previous section (see Fig. 2). In this section, we will review the capabilities of CityGML 3.0 to support these aspects, especially the temporal elements.

CityGML can spatio-semantically represent a wide range of city objects and define their topologies, attributes, and appearances. The CityGML Conceptual Model (CM) includes various types of extension modules such as Tunnel, Building, Construction, and Transportation as well as the Core module. Therefore, CityGML CM provide several entities for modelling underground physical reality (physical aspects). In addition, there are some ADEs developed for underground physical assets such as utility networks. As these ADEs are mostly based on the previous versions, they need to be adapted to the new version of CityGML.

Nearly all geometries in CityGML datasets use 3D coordinate values (x , y , z). These values should be based on a coordinate system (CRS). As a result, every object is geo-referenced. Along with the primitive geometries such as point, curve, surface, and solid, CityGML provides geometry aggregations such as spatial aggregates

(MultiPoint, MultiCurve, MultiSurface, and MultiSolid) and composites (CompositeCurve, CompositeSurface, and CompositeSolid). This data model uses Boundary Representation (B-Rep) to define volumetric shapes (Kolbe et al. 2021). Therefore, CityGML 3.0 provides adequate geometry representations to define ULA objects.

CityGML does not have any specific module for defining legal spaces. However, the latest version, CityGML 3.0, provides logical spaces to model the spaces bounded by virtual boundaries. The underground legal spaces presented in Fig. 2 can be defined as logical spaces within CityGML 3.0. Therefore, this entity can be used to extend this data model to cover different types of underground legal spaces and their geometries, attributes, and relationships as used by Saeidian et al. (Saeidian et al. 2023a). In addition to 3D spaces, CityGML provides some entities to define different types of space boundaries. Some of these entities can be used to define legal boundaries defined in Fig. 2 such as wall surfaces as building boundaries. The ADE mechanism can also be used to extend these boundaries to support all types of underground legal boundaries such as projected and surveyed boundaries as discussed by Saeidian et al. (Saeidian et al. 2024). In addition, the “boundary” relationship is provided in CityGML 3.0 to define the relation between 3D spaces and 3D space boundaries that can be used for modelling the relationship between underground legal spaces and boundaries. Finally, CityGML does not fully support survey and administrative elements of ULA. Therefore, it requires to be extended to support these elements.

CityGML 3.0 provides some entities for the representation of dynamic (time-dependent) aspects of city objects and models. In CityGML, all objects can be represented with respect to their semantics, 3D geometry, as well as their changes over time (Kolbe et al. 2021). Three modules of CityGML 3.0 provides entities to define temporal aspects including the Core, Dynamizer, and Versioning modules. CityGML 3.0 also uses ISO standards such as ISO Temporal Schema (19108) and ISO Conceptual Schema Language (19103) to define temporal properties like datatypes (e.g., Time, Date, and DateTime).

The Dynamizer module defines entities to model changes in object properties that occur frequently or rapidly. It includes the concepts for representing the properties of city objects using time series data and connecting them to sensors, services that provide sensor data, or files from external sources (Kolbe et al. 2021). Therefore, the entities provided by this module can be used for underground physical assets with time series data. As explained in the previous section, legal data does not have continuous changes with time series data.

CityGML 3.0 introduces bitemporal timestamps as state-based modelling to support feature history with discrete changes. The bitemporal history model normally utilises two distinct dimensions of time: real-world time and the time when the information is entered into the database/model (Thompson and Oosterom 2021). The AbstractFeatureWithLifespan feature class of the CityGML Core module provides the bitemporal timestamps as presented in Table 2. The timestamps indicate the time frame during which a particular version of an object is included in a 3D city model and the duration that version exists in the real-world. By utilising these time intervals, it is possible to query a CityGML dataset for information on what the city looked

like at a particular moment in time and how the city model appeared at that time (Kolbe et al. 2021).

Temporal aspects of cadastral objects can be modelled by timestamp attributes (Sürmeneli et al. 2022a, b; Döner et al. 2010, 2011). Since underground legal spaces are modelled as logical spaces in CityGML 3.0, they are subsets of the feature class `AbstractLogicalSpace` of the Core module. In addition, the feature classes defining underground legal boundaries are subsets of the `AbstractSpaceBoundary` class. Therefore, feature classes defining underground legal spaces and boundaries are subclasses of `AbstractFeatureWithLifespan` and inherit its timestamps (Fig. 4). The timestamps provided by the feature class `AbstractFeatureWithLifespan` (Table 2) are adequate to model the history of underground legal spaces and boundaries. For example, the `validTo` attribute can be used to model the expiry date of a restriction. In the case of underground land acquisition for establishing underground infrastructures like tunnels, the `validTo` attribute can be used for acquired private lands (lots) to specify their end date (the date they are acquired), and the `validFrom` attribute can be used for the new government-owned lands (crown lands) to specify their start date (the date lots are acquired and converted to the new crown lands). These timestamps can also be used for private land subdivisions and consolidations to specify the start/end date of previous and new lots (the date of subdivisions or consolidations). In cases of underground land acquisition, subdivision, and consolidation, it is necessary to validate the timestamps (e.g., the end date of the previous parcels and the start date of new parcels must be the same).

The CityGML timestamps can be used to represent the evolution of ULA objects over time. In addition to these timestamps, the Versioning module of CityGML 3.0 also allows the representation of multiple, concurrent versions of city models by introducing two feature types: 1. `Version` that defines named states of city models as well as denotes the versions of objects belonging to such states; 2. `VersionTransition` to link versions and describes the reason for changes and applied modifications (Fig. 5). Transactions also represent modifications including creating, terminating, or replacing city objects (Kolbe et al. 2021). Therefore, this module provides entities required for event-based modelling of discrete changes of ULA objects. For a 4D ULA, the `TransactionTypeValue` and `TransitionTypeValue` enumerations should be extended to cover transactions and transitions based on jurisdictional requirements. In the context of Victoria, examples are subdivision and consolidation transactions.

Two approaches can be followed to model ULA temporal changes using CityGML entities (timestamps and the Versioning module). The first approach is to define all

Table 2 Timestamps provided by the Core module of CityGML 3.0 (Kolbe et al. 2021)

| Timestamps (attributes) | Description |
|------------------------------|--|
| <code>creationDate</code> | The date at which a CityGML feature was added to the city model |
| <code>terminationDate</code> | The date at which a CityGML feature was removed from the city model |
| <code>validFrom</code> | The date at which a CityGML feature started to exist in the real-world |
| <code>validTo</code> | The date at which a CityGML feature ended to exist in the real-world |

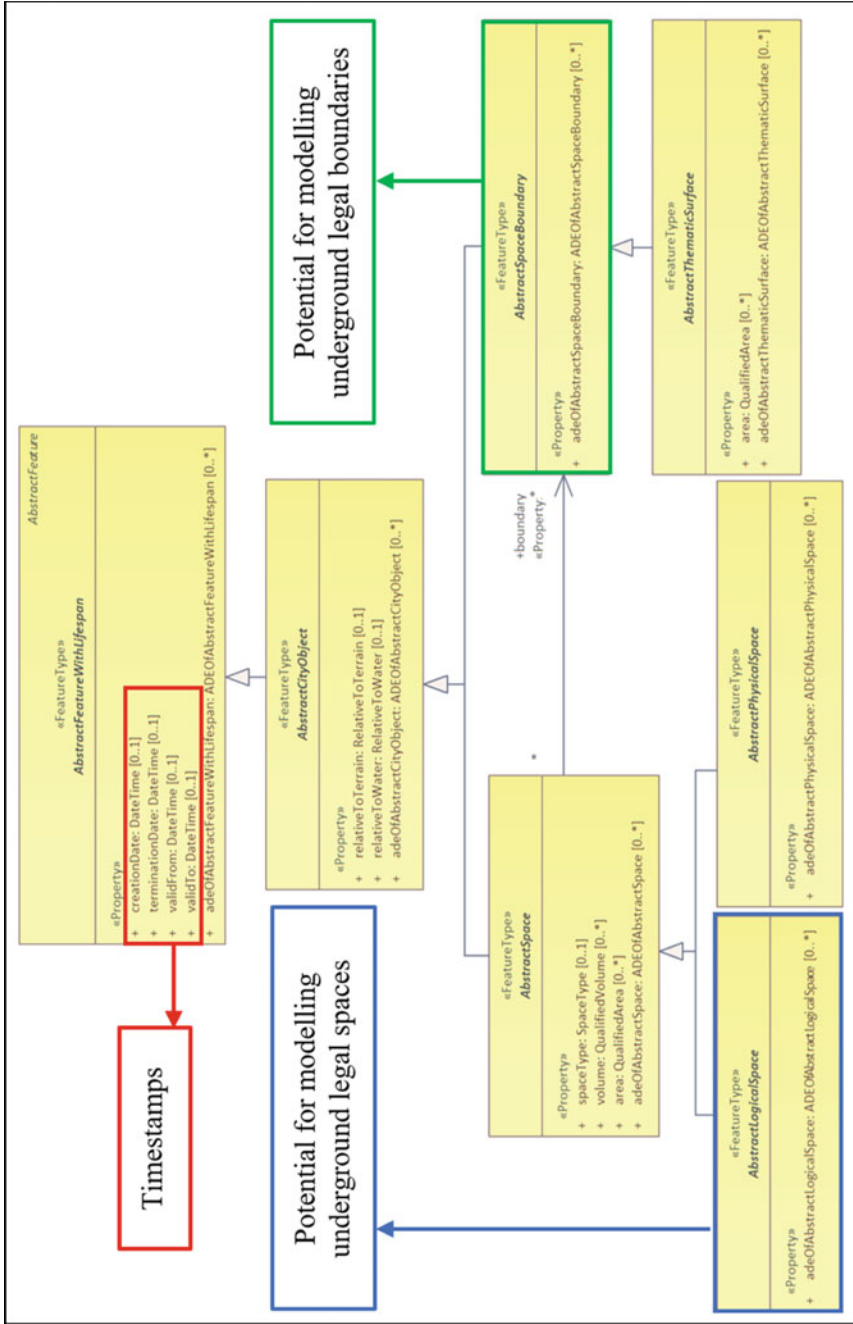


Fig. 4 The potential entities of the CityGML Core module to model underground legal spaces and boundaries and their timestamps (Kolbe et al. 2021)

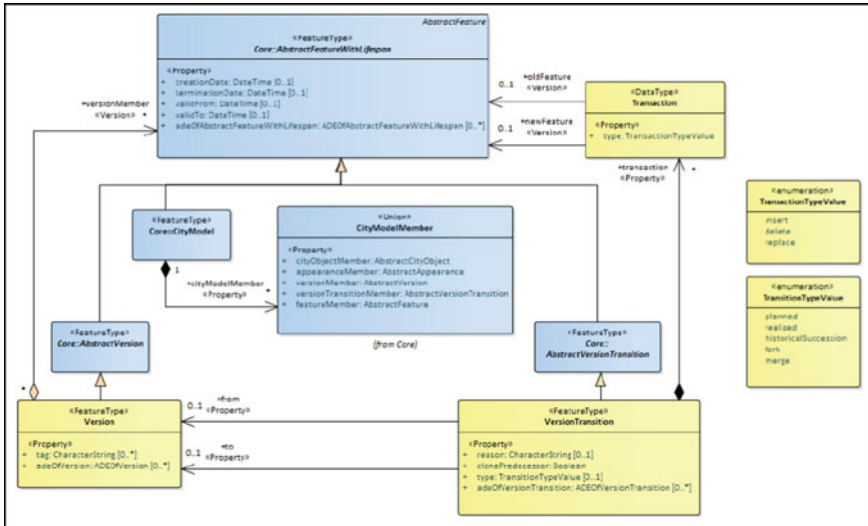


Fig. 5 The Versioning module of CityGML 3.0 (Kolbe et al. 2021)

underground cadastral objects over time in one version but specify their temporal aspects using timestamps. The second approach is to create a new city model version after a transaction like subdivisions on lands and connect the new version to the previous version using the Versioning module.

To demonstrate the concepts explained, we developed a synthetic model. Figure 6 shows the visualisation of this prototype in different years. As shown, this prototype defines several elements such as different legal spaces and physical assets in underground areas over time. Every element has its semantics, 3D geometry, and attributes in the context of Victoria. In addition, timestamps of all legal and physical objects are defined using the timestamps provided by the Core module of CityGML 3.0. Figure 6 shows the timestamps of an easement and a crown (government-owned) land as the legal spaces attached to the new utility and tunnel, respectively. This prototype can effectively show the changes over time in both physical assets and legal spaces.

5 Discussion and Conclusion

In this paper, we investigated the capabilities of the CityGML 3.0 conceptual model to develop a 4D model for underground cases. This standard can potentially be used to create an integrated 4D ULA model for a large area (city-scale) which is necessary for long underground utilities and tunnels. Moreover, by leveraging the CityGML ADE mechanism, the flexibility of the CityGML-based integrated 4D data model is enhanced, enabling future revisions, development, seamless data interoperability,

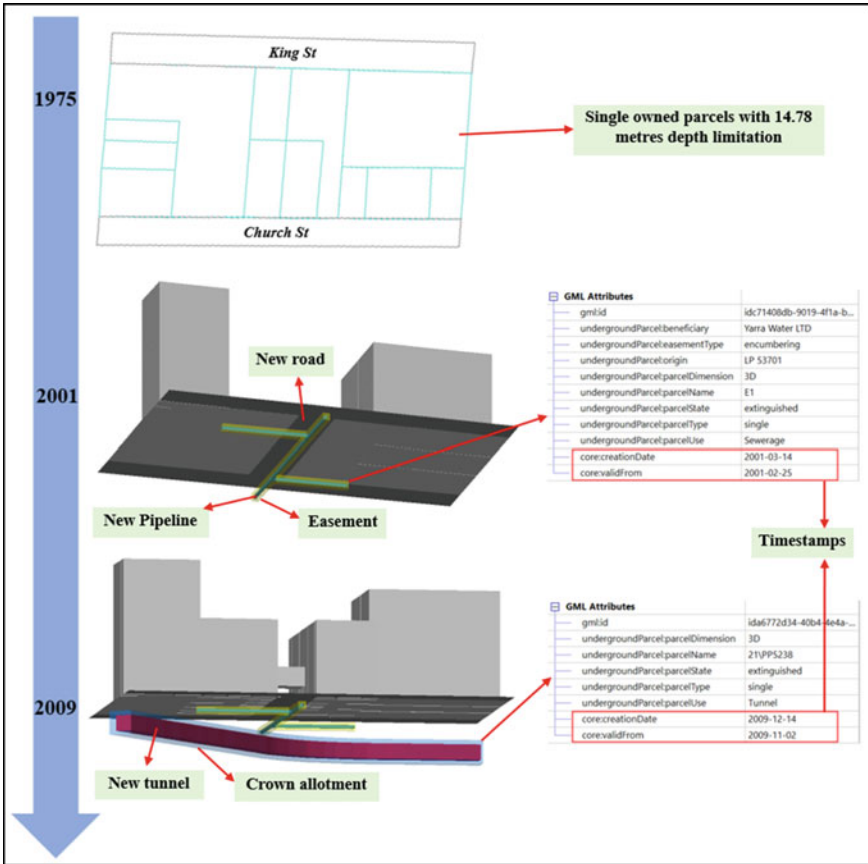


Fig. 6 The visualisation of the implemented prototype

and long-term maintenance. However, this standard requires to be extended to fully support 4D ULA:

- The CityGML CM provides entities to define a wide range of physical assets in underground space. There are also some ADEs to model underground assets. For example, the Utility Network ADE is used to model utility networks (Hijazi et al. 2018; Kutzner et al. 2018). However, these ADEs require adaptation to the latest version of CityGML and possible extensions,
- The logical space concept in the new version of CityGML can potentially be used to extend this data model to define different types of underground legal spaces,
- Although CityGML differentiates spaces and space boundaries, provides separate entities for them, and defines the relationships between them, it requires to be extended to define all types of underground legal boundaries such as surveyed and projected boundaries and the median face of physical structures like walls,

- CityGML does not fully support ULA survey and administrative elements. There are two approaches to incorporating these elements into a CityGML model. The first approach is to link CityGML with other standards that define these elements. For example, the Land Administration Domain Model (LADM) is rich in modelling legal and administrative elements and LandInfra is rich in modelling survey data. They can be integrated with CityGML. However, this integration requires addressing harmonisation implications. The second approach is the ADE mechanism of CityGML as a systematic and well-structured way to extend this standard to support modelling these elements. For example, Saeidian et al. (Saeidian et al. 2023c) used the ADE mechanism to enrich CityGML 3.0 with underground cadastral survey data.
- Finally, the entities provided by CityGML 3.0 can cover the temporal aspects of 4D ULA. The timestamps of the Core modules and the Versioning module provide the required entities to model discrete changes. These entities can model the changes in underground legal spaces and boundaries and ownership information over time. The entities provided by these modules enable storing and exchanging object versions within a single dataset, allowing software systems to work with different versions simultaneously and track changes to geometries and thematic properties over time. These entities enable both event-based and state-based modelling. The selection of the modelling approach depends on the jurisdictional requirements and preferences. In addition, some CityGML entities such as the enumeration for transaction types need to be customised based on the jurisdictional requirements (e.g., adding subdivision and consolidation transactions in the context of Victoria). Finally, the Dynamizer module can be used for modelling physical data with time series (dynamic or continuous changes).

There is no central registration system for underground assets. These asset owners have their own data and temporal changes are sometimes recorded by the owners. Land registration parties are also responsible for maintaining the legal information associated with these assets, which includes information on land ownership, boundaries, easements, and other details related to underground land. The temporal aspects of legal information are also available in the 2D plans provided by these parties. Integrating all legal and physical elements in an integrated model, considering the temporal aspects, can improve the communication and use of legal and physical data. It can also be used to check inconsistencies over time. For example, it can be checked if an easement is created for a newly established pipeline. The integrated 4D model not only improves the communication and interpretation of underground data over time but also expands the benefits of the model for different use cases. Compared with pure legal data models (e.g., LADM) and pure physical data models (e.g., MUDDI), the integrated 4D model can support physical and legal data elements of underground assets.

The synthetic model revealed the benefits of an integrated 4D model of underground assets and their ownership information over time. To implement a 4D integrated model for a real-world case study, it is crucial to examine the quality and

accessibility of 3D data and temporal information. Finally, when the fourth dimension is added to a 3D integrated ULA model, it is necessary to validate temporal data along with spatial and semantic data validation. For example, there must not be any gap in the time dimension for a primary parcel.

References

- Alberdi R, Erba DA (2020) Modeling legal land object for waterbodies in the context of 4D cadastre. *Land Use Pol* 98:104417
- Atazadeh B (2017) Building information modelling for urban land administration. PhD thesis. Department of Infrastructure Engineering, University of Melbourne, Australia
- Atazadeh B, Olfat H, Rajabifard A, Saeidian B (2022) Evaluation of the international 3D geospatial data models and IFC standard for implementing an LADM-based 3D digital cadastre. In: 10th International FIG workshop on the Land Administration Domain Model, Dubrovnik, Croatia
- Babalola S, Rahmanb AA, Choong T (2015) A brief review of land administration domain model and its temporal dimension. *J Adv Rev Scient Res* 6:1–15
- DAUB (2019) Digital design, building and operation of underground structures: BIM in tunnelling. Cologne
- Döner F, Thompson R, Stoter J, Lemmen C, Ploeger H, van Oosterom P, Zlatanova S (2010) 4D cadastres: first analysis of legal, organizational, and technical impact—with a case study on utility networks. *Land Use Pol* 27:1068–1081
- Döner F, Thompson R, Stoter J, Lemmen C, Ploeger H, van Oosterom P, Zlatanova S (2011) Solutions for 4D cadastre—with a case study on utility networks. *Int J Geogr Inf Sci* 25:1173–1189
- Döner F, Biyik C (2013) Conformity of LADM for modeling 3D/4D cadastre situations in Turkey
- Hijazi I, Ebert V, Donaubaer A, Kolbe TH (2018) Combining urban metabolism methods and semantic 3D city models. *Int Arch Photogramm Remote Sens Spat Inf Sci* 42:63–70
- Department of Planning (2017) West Gate Tunnel Project Section 3 Order and Maps. https://www.planning.vic.gov.au/__data/assets/pdf_file/0020/119261/Westgate-Tunnel-Project-Revised-Section-3-Order-and-maps.pdf
- ISO (2012) Geographic information—Land Administration Domain Model (LADM), vol. ISO19152. International Organization for Standardization (ISO), Geneva, Switzerland
- Kolbe TH, Kutzner T, Smyth CS, Nagel C, Roensdorf C, Heazel C (2021) OGC city geography markup language (CityGML) part 1: conceptual model standard. Open Geospatial Consortium, Wayland, MA, USA
- Kutzner T, Hijazi I, Kolbe TH (2018) Semantic modelling of 3D multi-utility networks for urban analyses and simulations: The CityGML utility network ADE. *Inter J 3D Info Model (IJ3DIM)* 7:1–34
- Larsson K, Paasch JM, Paulsson J (2020) Representation of 3D cadastral boundaries—From analogue to digital. *Land Use Policy* 98:104178
- Lieberman J, Roensdorf C (2020) Modular Approach to 3D Representation of Underground Infrastructure in the Model for Underground Data Definition and Integration (MUDDI). *Inter Archives Photogramm, Remote Sens Spatial Informat Sci* 44:75–81
- van Oosterom P, Ploeger H, Stoter J, Thompson R, Lemmen C (2006) Aspects of a 4D cadastre: a first exploration. In: *Shaping the change; XXIII international FIG congress*. FIG Office, pp 1/23–23/23
- Saeidian B, Rajabifard A, Atazadeh B, Kalantari M (2021) Underground land administration from 2D to 3D: critical challenges and future research directions. *Land* 10:1101
- Saeidian B, Rajabifard A, Atazadeh B, Kalantari M (2022a) Development of an LADM-based conceptual data model for 3D underground land administration in Victoria. In: 10th International FIG workshop on the Land Administration Domain Model, Dubrovnik, Croatia

- Saeidian B, Rajabifard A, Atazadeh B, Kalantari M (2022b) Extending CityGML 3.0 to support 3D underground land administration. *Int Arch Photogramm Remote Sens Spatial Inf Sci XLVIII-4/W4*, 125–132
- Saeidian B, Rajabifard A, Atazadeh B, Kalantari M (2023a) A semantic 3D city model for underground land administration: development and implementation of an ADE for CityGML 3.0. *Tunnelling and Underground Space Technology* 140, Article 105267
- Saeidian B, Rajabifard A, Atazadeh B, Kalantari M (2023b) Data lifecycle of underground land administration: a systematic literature review. *Survey Rev* 55:396–415
- Saeidian B, Rajabifard A, Atazadeh B, Kalantari M (2023c) Modelling underground cadastral survey data in CityGML. *Transactions in GIS* 27:1724–1747
- Saeidian B, Rajabifard A, Atazadeh B, Kalantari M (2024) Managing underground legal boundaries in 3D—extending the CityGML standard. *Underground Space*
- Sürmeneli HG, Koeva M, Alkan M (2022a) The application domain extension (ADE) 4D cadastral data model and its application in Turkey. *Land* 11:634
- Sürmeneli HG, Alkan M, Koeva M (2022b) Design and implementation of a 4D cadastral legal model for Turkish land administration infrastructure based on LADM. *Geocarto Inter*, 1–23
- Siejka M, Ślusarski M, Zygmunt M (2014) 3D+time Cadastre, possibility of implementation in Poland. *Surv Rev* 46:79–89
- Thompson R, van Oosterom P (2021) Bi-temporal foundation for LADM v2: Fusing event and state based modelling of Land administration data 2D and 3D. *Land Use Pol* 102:105246
- Wilhelm L, Donaubaauer A, Kolbe TH (2021) Integration of BIM and environmental planning: the CityGML EnvPlan ADE. *J Digit Landsc Archit* 6:323–324

Integrating Dynamic Data with 3D City Models via CityJSON Extension



Khawla Boumhidi, Gilles-Antoine Nys, and Rafika Hajji

Abstract Semantic 3D city models have been widely used to solve problems that affect the human-built environment. Due to the complexity of the city and its dynamic aspect, these models should allow studying the evolution of a phenomena in real time. Therefore, 3D city models are evolving towards Digital Twin of cities to allow handling the dynamic aspect of the city. For instance, such evolution requires real-time data to be collected using Internet of Things devices (IoT). This kind of dynamic data requires specific tools that should allow its particular exploitation and manipulation. To model the urban environment, CityGML and CityJSON for instance, as international 3D standards, allow creating, storing and exchanging 3D city models in an interoperable way. This paper aims to propose and implement a methodology for integrating real-time data in a city model by using the “Dynamizer” extension of CityGML based on CityJSON. We then have developed a web interface plugin to handle IoT data and their interactive visualization as a first step for future simulations. This solution allows the understanding of the 3D city model encoded with JSON, also the simplification and the easy comprehension of the Dynamizer extension compared to the CityGML solution which is difficult to decipher. The solution also makes it possible to establish the connection between the physical entities and their digital twins, to follow the evolution of their real characteristics over time, to represent them, analyze them and take the appropriate decision to preserve the components of the territory, to manage it correctly and ensure its resilience and sustainability.

Keywords CityJSON · CityGML · Dynamizers · Sensors · IoT · Web architecture

This article was selected based on the results of a double-blind review of the full paper.

K. Boumhidi (✉) · R. Hajji

College of Geomatic Sciences and Surveying Engineering, Hassan II Institute of Agronomy and Veterinary Medicine, 6202 Rabat, Morocco

e-mail: boumhidikhawla@iav.ac.ma

R. Hajji

e-mail: r.hajji@iav.ac.ma

G.-A. Nys

Geomatics Unit, Liège University, August 20 Square, 7B-4000 Liege, Belgium

e-mail: gansy@uliege.be

1 Introduction

3D city models are widely used for city management and planning. They associate spatial context to information related to the physical objects of cities (buildings, roads, network infrastructure, city furniture, etc.) and provide a means for interactive, spatial and semantic queries, and simulations. Besides of 3D geometries, 3D semantic city models cover thematic aspects, topology and appearances of the model environment. They are considered as major inputs of City Digital Twins (Chaturvedi and Kolbe 2016). A Digital Twin is a 3D virtual model connected with its physical twin and associated data in near-real time to allow simulating the state of the physical twin in real time. This concept is used in several applications such as real-time monitoring, design/planning, optimization, maintenance, remote access, etc. (Singh et al. 2021). The data flow between a physical object and its digital replica in both directions is required as a reference of digital twins (Fuller et al. 2020). The Digital Twin changes over time depending on the dynamics of data acquired and the changes occurred in the physical environment (Madni et al. 2019).

Sensors allow acquiring data on the state of the physical city model and give insight on how phenomena are influenced by it, but also how it affects the city elements. However, structuring dynamic data is a major challenge when it comes to make it available for immovable elements of the city. Sharing and understanding sensors data by a large number of users is an additional issue for their integrated management. In the context of the urban built environment, CityGML 3.0 improves the dynamic aspect of urban modeling through the concept of “Dynamizers” (Kutzner and Kolbe 2018). Dynamizers allow dynamizing a previously static attribute of city features by providing a smart access to a data source which is itself dynamic. However, due to the complexity of its XML-encoding, CityJSON was chosen as an alternative to CityGML for its ease of use. Indeed, CityJSON is a much simpler encoding based on JSON, easier to parse and lightweight alternative (Ledoux et al. 2019). However, even if many extensions have been proposed [Extensions | CityJSON (<https://www.cityjson.org/extensions/>)], the current version of this JSON-encoding solution does not yet support the “Dynamizers” provided in the new CityGML data model.

The objective of the present paper is to propose a solution of structuring near-real-time data from sensors to 3D CityJSON models. Their exploitation in a dedicated web application illustrates implementation of the proposed extended data model. The solution is based on two studies, (Nys and Billen 2021) and (Nys and Billen 2023) that propose a document-oriented database characterized by the flexibility of its schema and the support of JSON data. The solution allows the storage of 3D city models as well as the exploitation and updating of CityObjects.

The remainder of the paper is organized as follows: in Sect. 2, we present the literature background about the Dynamizers and give some use cases focusing on CityJSON. Section 3 exposes the adopted method for the integration of dynamic data via the Dynamizers. Section 4 is dedicated to the results analysis and discussion. In Sect. 5, we end with a conclusion and give future insights for improvements.

2 Background

3D city models have been used in various applications, especially in Smart-Cities projects. With the emergence of the Digital Twin concept, the semantics of these models is increasingly required to allow holding thematic management of the city. With regards to 3D standards, CityGML offers rich geometric and semantic modeling of the urban space in several domains. Recently, improvements of CityGML have been proposed in the last version (Kutzner et al. 2020). Among them the Dynamizers module allows integrating dynamic properties of city objects into the static model. It binds a static view of the city features with a dynamic data source. An attribute is then “dynamized” which opens possibilities coupling both static and dynamic data. In the next sub sections, we first present the specificities of CityGML Dynamizers module, its objective and how it works. We then present some use cases demonstrating the importance of integrating dynamic data into 3D city models in an urban context.

2.1 CityGML Dynamizers

The Dynamizers extension has been introduced as a core module of the CityGML standard in its third version (Kutzner et al. 2020). It allows modeling and supporting dynamic properties to follow the temporal variations of spatial, thematic properties and appearances of objects in 3D semantic city models. It allows integrating dynamic variations of the selected properties as time series without modifying the rest of the properties in the static model. The approach allows also real-time sensor observations to be integrated with city objects in a standardized way (Chaturvedi and Kolbe 2016). The Dynamizers replace the static value of the referenced object’s attribute with time-dependent values or a list of them. These dynamic values can be provided by tabulation of time/value pairs, by models of time/value pairs, by reference to an external file or by extraction of sensor observations. Their data structure defines the attributes linked to a dynamic source, which is updated at a quicker frequency, and a city model, which is itself slower to be updated. The Dynamizer class is defined by the following attributes (Chaturvedi et al. 2019):

- **AttributeRef**: Specific property of a static element of the city model whose value can be replaced with a dynamic one by using XPath which is a W3C recommendation used to navigate the elements and attributes of an XML document. It determines the position of the context element and replaces the attributes.
- **StartPOINT** and **EndPOINT**: they delimit the time interval for which the Dynamizer provides dynamic values, in other words, they are absolute extreme time positions for the dynamic values represented in the Dynamizer.

Dynamizers use the GML implementation consisting in mapping the time values (T_i) of the Temporal DomainSet, to their corresponding attribute values (V_i) of the RangeSet according to a CoverageFunction (Fig. 1) (Chaturvedi and Kolbe 2015).

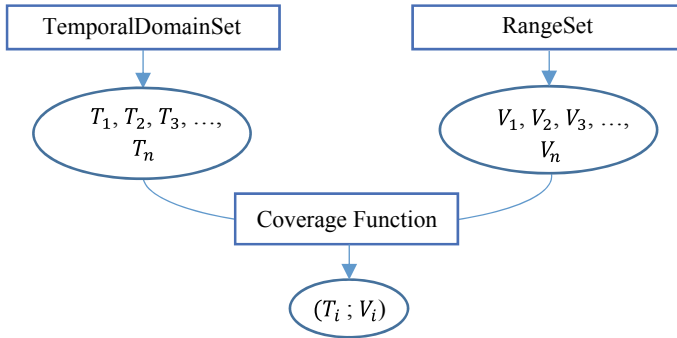


Fig. 1 GML implementation used for dynamizers

Dynamizers support the domain of time series whose values can be mapped to dynamic attributes. Dynamic data are modeled as AbstractTimeseries according to the SensorML standard [SensorML (<http://sensorml.com/index.html>)]. It consists of representing dynamic or time-varying values in a convenient and generic way.

2.2 Use Cases

Many authors have demonstrated the effectiveness of the Dynamizers in several domain applications. Chaturvedi et al. (2017a, b) used the Dynamizers concept to simulate solar energy production for roofs and building facades. In this work, Dynamizers allowed rendering results of dynamic simulations and visualization of suitable areas for photovoltaics installation.

Chaturvedi et al. (2019) used the concept of Dynamizers for energy simulations. Their project studied the energy consumption (electrical energy, heating energy, etc.) measured by a smart sensor installed in a residential building during various periods.

Santhanavanich and Coors (2021) presented the “CityThings” concept for integrating heterogeneous sensor data from the Open Geospatial Consortium’s SensorThings API into CityGML 3D city models for Smart Cities projects. They demonstrate the implementation of the CityThings concept in the Smart Villages project in the study area of Wüstenrot, Germany, by integrating dynamic sensor data from several systems, including solar panels, agro-thermal power plants and weather monitoring sensors to visualize the dynamic data injected into the 3D city model on the web platform.

On the other hand, CityJSON, the lightweight alternative to CityGML, has made modeling, structuring data, reading and interpreting it much easier for developers and users. Furthermore, some CityJSON datasets have been developed such as 3D BAG, “Basisregistratie Adressen en Gebouwen” in Dutch and “Register of Adresses and Buildings” in English, which is an up-to-date CityJSON dataset containing 10 million 3D building models of the Netherlands. These are used in urban applications. The

user can easily retrieve information about any object in the model either by displaying its attributes directly on the 3D viewer or by downloading the CityJSON file which is easy to read and understand [3D BAG Viewer (<https://3dbag.nl/en/viewer>)].

All these related works show the importance and at the same time the complexity of managing and manipulating dynamic data. Using CityJSON as an alternative encoding can be an efficient solution for handling 3D semantic city models, managing and visualizing both static and dynamic properties of CityObjects.

3 Methodology

Our approach consists in developing a Dynamizer extension for the CityJSON standard. The purpose is to allow linking a dynamic data source with a 3D city model encoded in the JSON alternative. In our case study, we opted for a dynamic data simulation using a fictitious application server simulating the production of sensors data. This server allows simulating sensor data, i.e. temperature (°C) and humidity (%) key/values pairs, according to the season and the time of day. The Timeseries of 30 days are produced at each call to the API. We consider that the sensor is mounted on top of the building to measure the outside air temperature and relative humidity.

The attributes and definitions of the Dynamizer are standardized in a conceptual model linked to the CityGML data model. The attributes allowing the connection to the data source are translated into the CityJSON specifications containing the mentioned core city model.

In this section, we first present the building blocks necessary to define the Dynamizer extension, then we present the process of its integration in the city model and finally their interactive visualization in the web (Fig. 2).

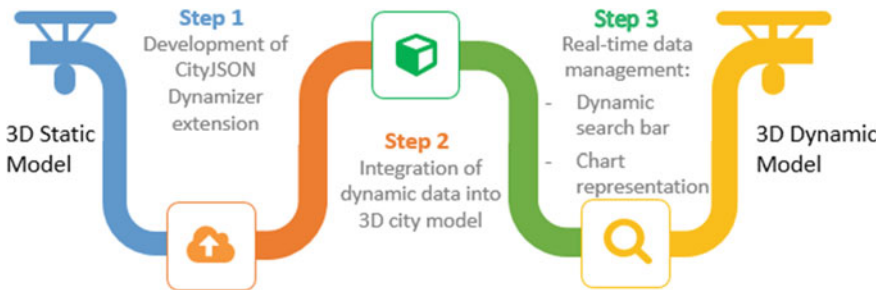


Fig. 2 The methodological workflow

3.1 Dynamizers JSON-Schema

The developed extension is based on the conceptual model of CityGML 3.0 and CityJSON 1.1.2 specifications. It includes the core extensions provided by the former in the schema of the later. Moreover, CityGML data model, in its simplified document-oriented database schema, allows handling Dynamizers in a much easier manner. In fact, both approaches: the NoSQL database schema and the Dynamizers are increasingly focused on developing web-related technologies for the expansion of information share and availability. It must be seen as part of a trend that goes towards the servicification of expertise and data management over the web.

In short, Dynamizers are defined within an additional class that consists of five attributes. All these information is structured in an additional class as a JSON object (Fig. 3):

- attributeRef: reference to the “dynamized” attribute;

```

"+Dynamizer": {
  "type": "object",
  "properties": {
    "attributeRef": {
      "description": "It refers to a specific attribute
of a specific city object by using an URI expression.
This is the static property that should overridden by
timeseries data",
      "uri": {
        "type": "string",
        "format": "uri-reference"
      }
    },
    "startTime": {
      "$ref": "#/definitions/TM_Position"
    },
    "endTime": {
      "$ref": "#/definitions/TM_Position"
    },
    "linkToSensor": {
      "$ref": "#/definitions/SensorConnection"
    },
    "dynamicData": {
      "$ref": "#/definitions/_AbstractTimeSeries"
    }
  }
}

```

Fig. 3 CityJSON Dynamizer extension

- `startTime`: time instant start of the time serie;
- `endTime`: time instant end of the time serie;
- `linkToSensor`: link to the sensor information;
- `DynamicData`: link to the time serie.

This JSON-schema consists of the core of the Dynamizers Extension for CityJSON v1.1.2.

Among these attributes, the dynamic data are linked to the dynamic data source by the `AbstractTimeSeries` class. `AbstractTimeSeries` represents data in the form of time series which can be either `AtomicTimeseries` or `CompositeTimeseries`. The former includes three types of data: `TimeseriesDomainRange`, `TimeseriesTimeValuePair` and `ObservationData`. While the later includes the `TimeseriesComponent` that defines the number of repetitions and the `additionalGap` attribute that allows two non-overlapping time series that were collected separately to connect to form a single time series.

The Dynamizers allow sensors retrieving collected observations through the `SensorConnection` class which has five attributes: `sensorID`, `serviceType`, `linkToObservation`, `linkToSensorML` and `sensorLocation` (Fig. 4).

3.2 Integration of Dynamic Data in the 3D City Model

An extension opens the city model core by providing new capabilities: new attributes, new `CityObjects`, new metadata, etc. A single CityJSON model can use several extensions at the same time. The “extensions” attribute of a CityJSON file binds the core model with these added capabilities. Attributes that are extending the original elements are then usable in the whole CityJSON file. For instance, an extended CityJSON file that has one “dynamized” Building but no geometries is structured as follows (Fig. 5).

3.3 Real-Time Data Management

Accessing to the data, structuring and finally visualizing it on a web application requires the development of specific components for 3D features but also dynamic elements. Usual components are used to render static features (features that are not evolving and/or updating on a quick frequency (hourly, daily, etc.)).

In a way that illustrates the new dedicated components, we extend the capabilities of CityJSON using the Measur3D architecture [<https://github.com/GANys/Measur3D>], a light and compact MERN application for the management of CityJSON city models developed by Nys and Billen (2021), part of which consists of a web application which purpose is to provide an easy–extendable solution for the development of specific applications and CityJSON extensions. It allows users retrieving

```

"SensorConnection": {
  "properties": {
    "SensorId": {
      "type": "string"
    },
    "serviceType": {
      "type": "string"
    },
    "linkToObservation": {
      "description": "link to observation response",
      "uri": {
        "type": "string",
        "format": "uri-reference"
      }
    },
    "linkToSensor": {
      "description": "link to the description/metadata of
the sensor device",
      "uri": {
        "type": "string",
        "format": "uri-reference"
      }
    },
    "sensorLocation": {
      "$ref": "cityobjects.schema.json#/_AbstractCityOb-
ject"
    }
  }
}

```

Fig. 4 Connection to sensor data

information about time-varying attributes, querying them and processing them. We propose a dynamic search bar, which follows a specific path to the new Dynamizers attribute introduced in the model. It is thus dedicated to the specific extension and parses temporal information using the ISO8601 standard. The purpose is to filter, to aggregate information and to provide generic statistical information on the filtered values (mean, median, trend, etc.). A second dynamic component plots dynamized phenomenon filtered by the search bar. For a better understanding, the data is modeled as time series on a chart.

Dynamic data changes asynchronously over time as new information becomes available. Handling JavaScript events provide solutions to update the newly created components in order to model the data dynamism. Two solutions are possible: a websocket that binds the web interface with a real time pipeline or aggregation APIs that would be managed by the provider and the data manager (i.e. not the urban model manager).

```
{
  "type": "CityJSON",
  "version": "1.1",
  "extensions": {
    "Dynamizer": {
      "url": "https://github.com/lkhawla/CityJSON-Dynamizer/blob/main/dynamizer.ext.json",
      "version": "1.0"
    }
  },
  "metadata": {
    "geographicalExtent": [
      300578.235,
      5041258.061,
      13.688,
      300618.138,
      5041289.394,
      29.45
    ]
  },
  "CityObjects": {
    "Building_1": {
      "attributes": {
        "color": "green",
        "temperature": 20,
        "+Dynamizer": {
          "attributeRef": "CityObjects/Building_1/attributes/temperature",
          "linkToSensor": {
            "SensorId": "50d9d53e-f303-11ec-b939-0242ac120002",
            "ServiceType": "Dummy sensor humidity",
            "linkToSensor": "http://localhost:3002/info",
            "linkToObservation": "http://localhost:3002/humidity"
          },
          "startTime": "2022-05-17T10:08:02.089Z",
          "endTime": "2022-07-16T20:18:26.614Z",
          "dynamicData": "AtomicTimeseries"
        }
      },
      "geometry": []
    }
  }
}
```

Fig. 5 CityJSON file structure for «dynamized» building

4 Results

This section illustrates the capabilities provided by the extension using simulated humidity and temperature sensors. We have created a mock-up server that simulates sensors serving information thanks to normative spaces and query accesses.

The Dynamizer extension developed according to the CityJSON standard is hosted on GitHub and is available for open access. The code and its documentation are available via the following link: <https://github.com/1khawla/CityJSON-Dynamizer>

For the reminder, this extension should be bind to the city model. In this example, some features of the model are extended to support temperature and humidity sensing devices (i.e. IoT sensors are linked to specific CityObjects documents).

For implementation, we used the Rotterdam model accessible online. We define the extension in “extensions”, then we choose an object of the model and we define the new attribute “+Dynamizer” and its properties among the existing attributes the same way as shown in Fig. 5.

Once the model is imported, we can access to information about CityObjects, retrieve information on dynamic attributes using the search bar and analyze easily the dynamic phenomena through the graphical representation (Fig. 6). City features that have had one of their attributes extended are depicted in a specific color: in blue, and the selected feature is depicted in yellow. Other features are not extended: they represent classic Buildings.

Selecting a feature queries the dynamic time series using the provided data source link: an URL to an API, a link to a file formatted as a Timeserie of values pairs, etc. Responses are stored globally on the application and populate the chart on the bottom. Depending on the data source capabilities, the data are then “frozen” until a different city feature has been selected. The selected feature can have more than one attribute but only one dynamic attribute can be represented at a time.

We can collect data on year, month, day, and a specific hour. The query must respect the form in which the data is encoded, otherwise the search will not return results. The search bar also allows the filtering of the graph. It allows you to filter the last month, the last day and the last hours of the last day as shown in the Figs. 7, 8 and 9.

When filtering the chart the data is not updated on the chart. It keeps the last data displayed by the last click on the city object concerned. The information displayed by the search bar is always updated and presents the information at the time of entering the query while that of the graph is extracted from the graph of Fig. 6 after filtering. The difference between the value displayed by the search bar and that on the chart is due to the generation time of each. This option allows you to zoom in to properly detect the variation of the phenomenon in a specific period. To update the graph, simply click on the object again, thus redo the filtering.

The examples presented in Figs. 7, 8 and 9 are taken as of August 14, 2022. They concern data for the period [July 15, 2022; August 14, 2022] since the server generates the data for a period of 30 days. This section will present the results on the “humidity” attribute expressed in %.

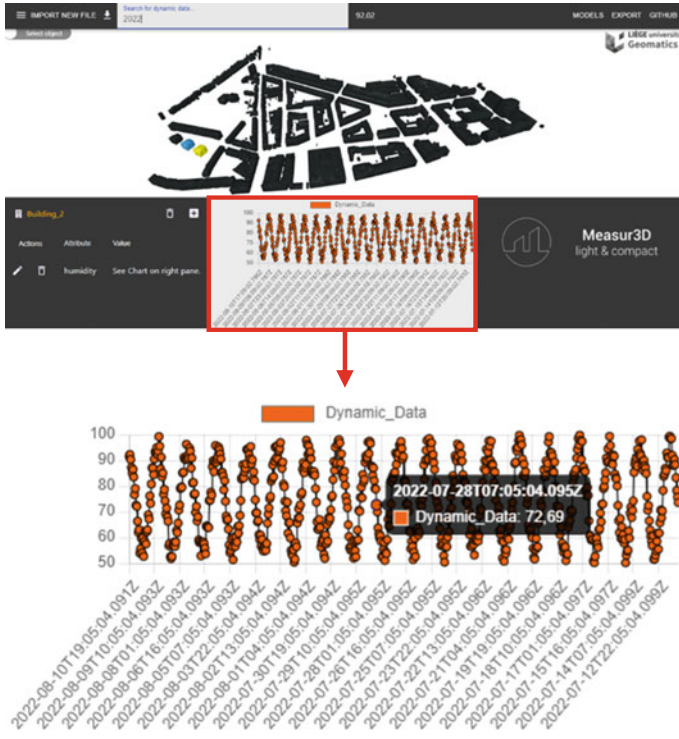


Fig. 6 Visualization of dynamic attributes (i.e. simulated humidity during 2022)

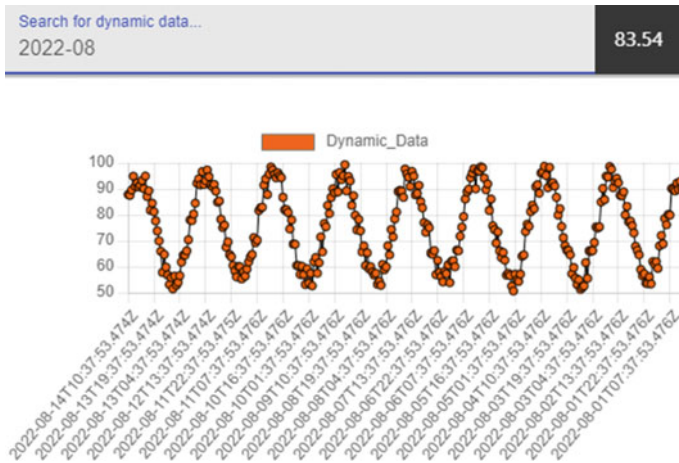


Fig. 7 Graphical representation of humidity data for the month of august

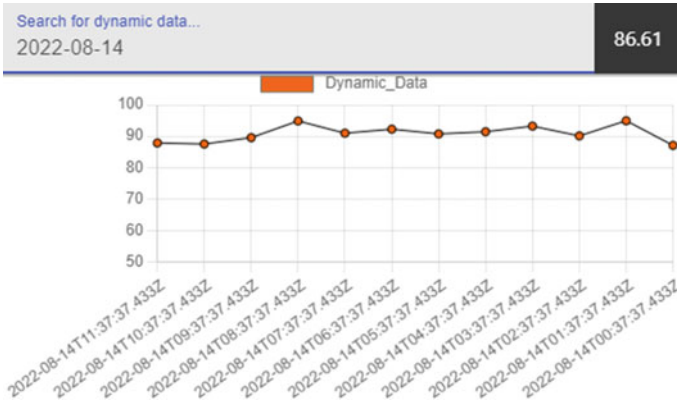


Fig. 8 Graphical representation of humidity data for August 14, 2022

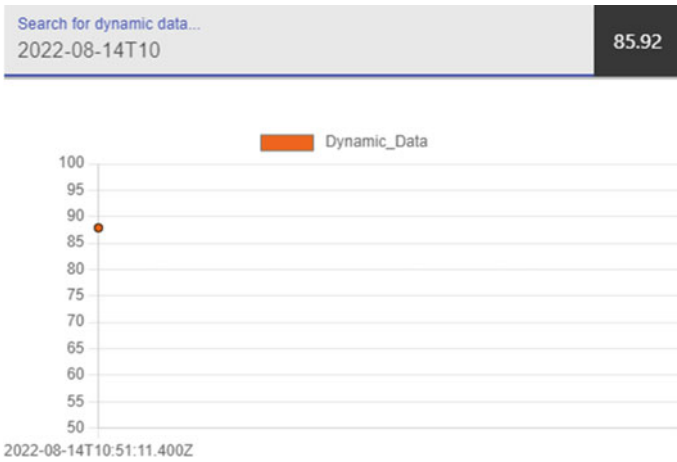


Fig. 9 Graphical representation of humidity data for a specific time of day

5 Discussion

The developed approach consists in creating a Dynamizers extension for the CityJSON standard. It allows integrating dynamic variations of some properties of city objects in the static 3D model. The dynamic data can be sensor observations or from another source. Besides, the extension was introduced into the 3D city model and the actual values of the different properties of the Dynamizers were defined. Finally, the visualization of the dynamic data could be done thanks to dedicated simple web components. For this purpose, we have developed a solution for retrieving and plotting the data in a sandbox application dedicated to this type of

development. It consists of a search bar that filters the displayed data and plot it in a chart. Some statistical values can be computed on the fly loading the data.

The advantage of our approach is that it can be used in several applications that require the monitoring of a phenomenon over time using the capabilities of CityJSON. The integration of real-time data allows implementing digital twins by allowing access to dynamic properties of city objects. Doing so, the city model become more interactive by reporting all the changes that the urban environment undergoes. It allows the interconnection of physical devices of the real urban environment with virtual city models. This interconnection allows the development of city models where everything is connected and well monitored: a city shadow (Kritzinger et al. 2018). Through these models, the user can manage and control his environment easily, make preventive studies to inhibit any phenomenon that could threaten its balance and contribute to the creation of resilient and sustainable cities.

Our approach addresses the issue of structuring and visualizing dynamic data in an urban context that can be applied to several urban use cases. However, further work is needed to enhance the visualization of data in order to extend it to different city objects such as trees, infrastructure installations, and to optimize visualization by using other visual 3D variables (texture or color that changes according to the evolution of a phenomenon). Another improvement would be to integrate multisource data in the digital twins and adapt the visualization solution to different data formats.

6 Conclusion

Semantic 3D city models have always been of great utility and importance in the management and development of the human environment. These models can be used to make real-time studies by tracking the change of dynamic data in the urban environment. In this perspective, we proposed an approach for the integration of dynamic data in 3D semantic city models based on the CityJSON standard and their visualization in a web interface (Measur 3D).

The Dynamizer extension developed in this study offers a method to enhance static city models with dynamic property values. The solution we designed and implemented shows the feasibility of integrating dynamic data into 3D city models for urban environment management. It allows visualizing in real time, access information and monitor the evolution of continuous phenomena.

References

- Chaturvedi K, Kolbe TH (2015) Dynamizers-modeling and implementing dynamic properties for semantic 3D city models. In: Eurographics workshop on urban data modelling and visualisation, pp 43–48

- Chaturvedi K, Kolbe TH (2016) Integrating dynamic data and sensors with semantic 3D city models in the context of smart cities. In: ISPRS annals of the photogrammetry, remote sensing and spatial information sciences, 11th 3D Geoinfo conference, Athens, Greece, vol IV-2/W1, pp 31–38
- Chaturvedi K, Smyth CS, Gesquière G, Kutzner T, Kolbe TH (2017a) Managing versions and history within semantic 3D city models for the next generation of CityGML. In: Advances in 3D geoinformation. Lecture notes in geoinformation and cartography. Springer, Cham, pp 191–206
- Chaturvedi K, Willenborg B, Sindram M, Kolbe TH (2017b) Solar potential analysis and integration of the time-dependent simulation results for semantic 3D city models using dynamizers. In: ISPRS annals of the photogrammetry, remote sensing and spatial information sciences, 12th 3D Geoinfo conference 2017, Melbourne, Australia, vol IV-4/W5, pp 25–32
- Chaturvedi K, Yao Z, Kolbe TH (2019) Integrated management and visualization of static and dynamic properties of semantic 3D city models. In: The international archives of the photogrammetry, remote sensing and spatial information sciences, 4th international conference on smart data and smart cities, Kuala Lumpur, Malaysia, vol XLII-4/W17, pp 7–14
- Fuller A, Fan Z, Day C, Barlow C (2020) Digital twin: enabling technologies, challenges and open research. *IEEE Access* 8:108952–108971
- Kritzinger W, Karner M, Traar G, Henjes J, Sihn W (2018) Digital twin in manufacturing: a categorical literature review and classification. *IFAC-PapersOnLine* 51(11):1016–1022
- Kutzner T, Kolbe TH (2018) CityGML 3.0: sneak preview. In: PFGK18-Photogrammetrie-Fernerkundung-Geoinformatik-Kartographie, 37, vol 27. Jahrestagung in München, pp 835–839
- Kutzner T, Chaturvedi K, Kolbe TH (2020) CityGML 3.0: new functions open up new applications. *PFG* 88:43–61
- Ledoux H, Ohori KA, Kumar K, Dukai B, Labetski A, Vitalis S (2019) CityJSON: a compact and easy-to-use encoding of the CityGML data model. *Open Geospatial Data Softw Stan* 4(4):1–12
- Madni AM, Madni CC, Lucero SD (2019) Leveraging digital twin technology in model-based systems engineering. *Systems* 7(7):1–13
- Nys G-A, Billen R (2021) From consistency to flexibility: a simplified database schema for the management of CityJSON 3D city models. *Trans GIS* 25(6):3048–3066
- Nys G-A, Billen R (2023) From consistency to flexibility: handling spatial information schema thanks to a middleware in a 3D city modeling context. *Trans GIS* 27(1):115–133
- Santhanavanich T, Coors V (2021) CityThings: an integration of the dynamic sensor data to the 3D city model. *Environ Plann B Urban Analytics City Sci* 48(3):417–432
- Singh M, Fuenmayor E, Hinchy EP, Qiao Y, Murray N, Devine D (2021) Digital twin: origin to future. *Appl Syst Innov* 4(36):1–19

Software and Tools for Spatial Data

OGC Data Exchange Toolkit: Interoperable and Reusable 3D Data at the End of the OGC Rainbow



Francesca Noardo, Rob Atkinson, Ingo Simonis, Alejandro Villar,
and Piotr Zaborowski

Abstract Standard data models are powerful to enable data interoperability. However, due to their ambition to comprehensively represent a domain, they tend to be over-specified. On the other hand, they allow too much flexibility in implementation and modelling details, to adapt to several use cases needs. The OGC Data Exchange Toolkit intends to overcome the two issues by means of a methodology to define profiles through linked data technologies, operated via the OGC Rainbow. Moreover, attributes are defined to specify the desired object representation requirements, solving the under-specification issue. A template is provided, intended to support manual input of data requirements, generating in turn a human-readable description and a machine-readable input for the profiling tool itself. The profile allows automatic datasets validation against the defined requirements. The prototype is developed and tested on a sample CityJSON dataset.

Keywords Standard data models · Data model profile · Data requirements · Data validation · Ontologies · OGC Rainbow

This article was selected based on the results of a double-blind review of an extended abstract.

F. Noardo (✉) · R. Atkinson · I. Simonis · A. Villar · P. Zaborowski
Open Geospatial Consortium Europe, Technologielaan 3, B-3001, Heverlee, Belgium
e-mail: fnoardo@ogc.org
URL: <https://www.ogc.org>

R. Atkinson
e-mail: ratkinson@ogc.org

I. Simonis
e-mail: isimonis@ogc.org

A. Villar
e-mail: avillar@ogc.org

P. Zaborowski
e-mail: pzaborowski@ogc.org

1 Introduction

FAIR principles (Findable, Accessible, Interoperable, Reusable) (Wilkinson et al. 2016) are essential to support an optimised and efficient production and use of data in distributed, multi-user environments. Distributed ecosystems such as *Data Spaces*, *Spatial Data Infrastructures*, *Digital Twins*, etc., are increasingly being conceptualised and realised to support reuse of data and system capabilities to deliver on policy and operational objectives. The rapid emergence of 3D, 4D and AI capabilities has resulted in more forms, and larger sets, of data with different structural characteristics. Hence the need to utilise very large quantities of data across many dimensions, with varying structures, has resulted in the renewed recognition attention on semantic interoperability across heterogeneous systems, rather than trying to enforce structural similarity.

Open standards are essential to facilitate this. Standard data models are intended to describe domains' information through agreed schemata in order to produce consistent and unambiguous data. However, open standard data models are either intended to cover a huge number of use cases or are closely tied to a specific purpose. General models are complex to use and yet still unable to deal with the full spectrum of requirements for related use cases.

In practice, comprehensive data models with a fixed schema simultaneously both over-specify and under-specify the information needed to describe a use case's needs. Over-specification occurs when many aspects are represented even though they may not be relevant to a particular application, in an attempt to be a "one-size-fits-all" solution. Under-specification occurs when aspects of the model are oversimplified, or left to different implementation and presentation decisions. Under-specification leads to implementation-specific choices that hinder the reusability and safe interpretation of standards-compliant data. In fact, several datasets may represent a different selection of features with different modelling and filling choices but still remain data model-compliant in the same way. For example, different approaches may choose to use either geometric solids or component surfaces, and have a range of different conventions regarding how such geometric elements are identified and topologically related. Different selection of attributes to characterise objects may be used, with similar or ambiguous naming and rules for content standardisation. Consequently, this makes it difficult for users to understand these datasets, and the use of datasets that conform to a specific data schema does not *per se* guarantees consistent or valid results in every application (Wagner et al. 2012; Noardo 2022).

For example, many spatial systems describe their own profile of the underlying ISO 19107 Geometry model¹ by using some words in the introduction of a document. For example the GeoJSON specification² declares "*GeoJSON supports the following geometry types: Point, LineString, Polygon, MultiPoint, MultiLineString, and MultiPolygon*". When extrapolating to 3D, 4D geometries, and support for topology

¹ <https://www.iso.org/standard/66175.html>.

² <https://geojson.org/>.

in 2D and 3D the variability between specifications becomes greater still, and there will obviously be great difficulty in finding, accessing and interpreting such informal wording to determine the level of compatibility and the transformations possible for data based on these specifications.

Defining a set of formal profiles can capture the intent in a simplified way and support data processing pipelines that can exploit these relationships.

Software developers can implement tools for standard data models (such as importers, exporters, analyzers, etc.) that consider the various options for different functions (Noardo et al. 2021). However, the problem always arises here that any leeway drives up implementation costs. If these cannot be borne, they lead to restrictions in the functional space of the respective tool. Even if these restrictions are clearly documented, the acceptance of such tools in the user groups is limited. Users and practitioners may have reservations about using automated tools, even if they are based on standardization of data sets that conform to the data model, because they find potential inaccuracies that make them not 100% reliable. Often the barrier is simply discovering the relevant documentation to allow interpretation of how an application has used an underlying standard, even if the standard itself is well understood.

Ultimately, these problems affect trust in data and data processing. As a result, the power of open standards is not fully exploited by supporting automation and interoperability.

The solution to this problem lies in the combination of modularity of data models and dynamic, unambiguous definition of specific constraints and assignment of the degrees of freedom available to the model. This allows users to use only the elements necessary for their data and not be burdened with unnecessary complexity by the overall model. Specific applications in the domain can formally profile the overall model, which unambiguously defines the parts used as well as corresponding implementation decisions for a particular community of practice. By dynamically providing such application context-dependent definitions, data providers can provide users with any information necessary for unambiguous processing and interpretation of the data at runtime. Crucially, the provision of this information must be directly processable by both machines and human users.

The direct implication of this hard-won experience with standardisation data models is that we need a more flexible and dynamic ecosystem of standards components, with simplified general conceptual models, multiple schema options, application profiles and the support for the validation and vocabulary aspects—a wide spectrum of resources that are not easily contained in the existing approach of an ever-growing set of heterogeneous document formats distributed across the Web.

The same issue was addressed by the ISO standard 19131.2 on Geographic information—Data Product Specifications (ISO/TC 211 2022). It provides a data model to support the conceptual formalisation of semantics and data structure related to specific requirements of use cases, for the same reasons of supporting data requirements definition and following datasets' description. Because of this last aim, the direct relation with the ISO 19115 standard on Metadata (ISO/TC 211 2014) is foreseen in the model.

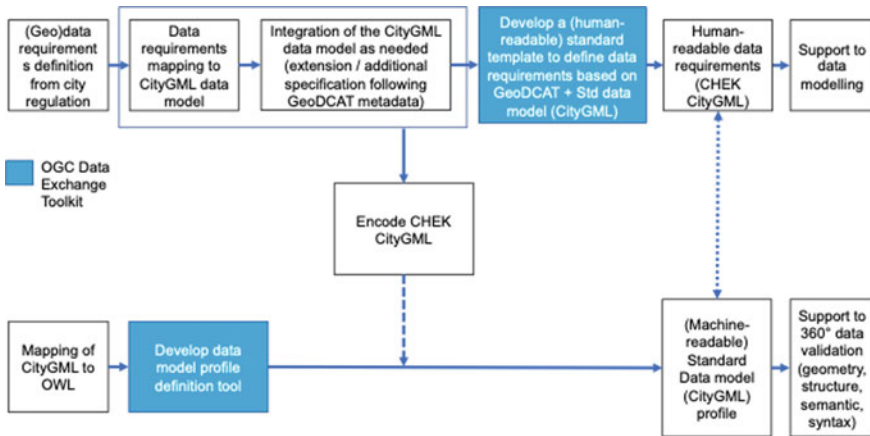


Fig. 1 Methodology workflow

However, the use of standardised data models to formally describe the data semantics and structure is not foreseen within the ISO 19131 standard, with consequent loss of the potential interoperability advantage given by the use of standard data models.

To overcome this gap, in this study, a prototype of the *OGC Data Exchange Toolkit* is developed starting from a use case, in order to support the formal specification of a standard data model profile, including detailed definition of semantics and geometry of the single contained objects.

A template for data requirements definition was defined starting from a use case (i.e., the data requirements to check a building permit regulation in the municipality of Ascoli Piceno, in Italy). The data requirements defined for the test refer to the CityJSON v.1.1 standard data model (Ledoux et al. 2019), which was mapped to OWL and identifiers were assigned. These were organised in a template, useful to support similar operations, as well as facilitating the following development of a compliant standard data model profile (Sect. 3.1).

After cross-referencing the requirements in the template to the model, a profiling tool was configured to extract the same profile defined in the template into a machine readable (semantic) model and use it for validation purposes (Sect. 3.3).

A sample data is mapped to an equivalent semantic model, and the validation is performed using standard tools ensuring transparency, openness and trust of the process (Sect. 3.4).

2 Methodology

A bottom-up approach starting from a use case was adopted to iteratively build and test the necessary profile and supporting tools. Within the Horizon Europe project

'Change toolkit for Digital Building Permit' (CHEK)³, on the digitalisation of building permits, an initial CityJSON profile was defined, useful to support automatic checks of a city regulation (Fig. 1). Diverse information is involved in the digitalisation of building permits, including Building Information Models (BIM) and Geographical Information Systems (GIS). In this paper the focus is on the specification of requirements for the GIS data, as a 3D city model compliant to the CityJSON standard data model.

2.1 *The Definition of a Template for Data Requirements Specification*

Starting from the formalisation of a selected city regulation as a use case, the information requirements were defined and mapped on the chosen standard data model, in this case the OGC Community standard CityJSON⁴.

The mapping of regulatory information to a CityJSON profile was effective to cover the representation of the objects necessary for the building permit analysis. However, a more detailed description of both some general characteristics of the dataset itself (e.g. spatial and temporal coverage, coordinate reference system, format, etc.) and specific representation details of the single objects (e.g. buildings' level of detail, accuracies, etc.) could not be properly covered by the CityJSON data model.

Therefore, to comprehensively represent the data requirements, a subset of the relevant attributes of metadata schemas, such as ISO 19115 and GeoDCAT (Raes et al. 2019) (other relevant metadata standard for geospatial data) was selected and used in the toolkit. Finally, additional attributes (such as 'accuracy') were considered, as relevant to support interoperability, re-use and possibly integration with different sources Noardo (2022). The obtained information was compared and properly mapped to the contents of ISO 19131. An extension to the ISO 19131 was defined to allow direct reference to standard data models in the Application Schema Information description.

The result of such mapping was gathered in tables that describe the reference template for the *OGC Data Exchange Toolkit*. In addition, they will be implemented as an interactive tool facilitating the filling of the necessary information in a more user-friendly interface. The tool will facilitate, for example, the mapping of the selected standard data model or related class to their respective URIs. The filled requirements will be later suitably exported in both a human readable data requirements specification, supporting data modelling, and in a more formal format able to support the linked data-based profiling tool.

³ <https://chekdbp.eu>.

⁴ <https://www.cityjson.org>.

2.2 *The Profiling Tool Development*

For the development of the profiling tool, to facilitate the mapping, first an Ontology for CityJSON was developed to provide a unique and unambiguous identifier (in the form of a Universal Resource Identifier—URI) for each Class and Property in the model, and to capture relationships. The OWL, RDFS and SHACL languages can then be used to describe each of these within a knowledge graph so that profiles can access underlying concepts when required. These URIs then allow us to define mappings in a way that can be published in a FAIR fashion, using the OGC Registry for Accessible Identifiers of Names and Basic Ontologies for the Web (Rainbow).⁵

The OGC Rainbow is a “full-spectrum” interoperability environment that provides for publication and cross-referencing any type of resource used in creating or using a spatio-temporal dataset. OGC Rainbow enables model-driven standards with continuous integration and testing, data model and building block registries, and many other tools that ultimately enable semantic interoperability, the long-awaited next interoperability level.

The OGC Rainbow supports communities to publish profiles of existing baseline standards.

At this stage the focus is on the machine-readability of such profiles, with an emphasis on provision of validators, and forms of the profile that are directly validatable. Exploiting semantic technologies to model the standards baseline provides a set of identifier about which statements can be made.

Technologies such as SHACL⁶ allow “shapes” to be defined that control how elements of such models may link to related data with its own data models, and rules defined for validation of data against such linked models. The “shape” refers to the pathways that are present in an interconnected graph of related data, and it can be seen that such an approach is naturally suited to the integration of related systems with their own preferred data models.

The profiling toolkit supports cataloguing of such profiles using the Profiles Vocabulary⁷ which allows profiles to be incrementally developed on top of more general profiles.

The OGC Rainbow uses such a profile catalogue to manage the heterogeneous forms of resources from many different sources—each is validated against a range of requirements for the specific resource type, but also further augmented and even transformed using specific profiles describing the input data source and the final form that can be integrated into the linked knowledge graph.

With this powerful decomposition of complex specification environments into standardised pieces, specific application challenges can then be supported by a common toolkit. As we shall describe below, this toolkit allows for a best practice “Con-

⁵ <https://www.ogc.org/resources/rainbow/>.

⁶ <https://www.w3.org/TR/shacl/>.

⁷ <https://www.w3.org/TR/dx-profl/>.

tinuous Integration/Continuous Testing” (CI/CT) approach to defining the specific profiles of standards.

The required subset of CityJSON can then be defined as a formal profile of CityJSON, or a set of related profiles for different levels of detail.

2.3 Validation

We test the template and the standard data model profile in conjunction with each other to ensure the tool deliver useful forms of profiles, and that the profiles meet the business (use case) objectives. The profile itself is tested using sample data⁸, and the testing process is validated through feedback from data producers as well as software developers who benefit from the clarity and flexibility of the tested examples.

3 Results

The prototyped toolkit includes a template to specify data requirements (i.e., complying to the same data model profile) in human-readable format, supporting the production and modelling of data.

Moreover, a standard data model profiling tool allows the definition of the specific part of a standard data model and the specific implementation requirements in a machine-readable format, leveraging the OGC Rainbow semantic capabilities. Profiles rely on underlying specifications having unambiguous identifiers for the component elements, which is a natural consequence of using semantic technologies as the underlying expression language, since these are based on links (predicates) between identifiable “subjects” and other objects or literal data values. A generated machine-readable profile can support the automatic validation of datasets as well as software development and configuration to exploit the data. The extensible profile model used can even include specific implementation resources such as code functions if required.

A part of the urban regulation related to the minimum distances of the new building, submitted to the municipality for approval, in the municipality of Ascoli Piceno (art. 61), in Italy, was chosen as a starting point. The data requirements for the representation of the context as a 3D city model were considered and extracted following the RASE methodology (Hjelseth and Nisbet 2011).

The considered text of the regulation is (translated from Italian): “2—[...] *between windowed walls of facing buildings, a minimum distance is prescribed, equal to the height of the tallest building and no lower than 10 m; if the facing facades overlap for more than 12 m, the rule applies also when only one of them has windows. [...]* In case of setback storeys, the rule applies per each floor. [...] 4—*Minimum distance between*

⁸ <https://www.cityjson.org/tutorials/getting-started/>.

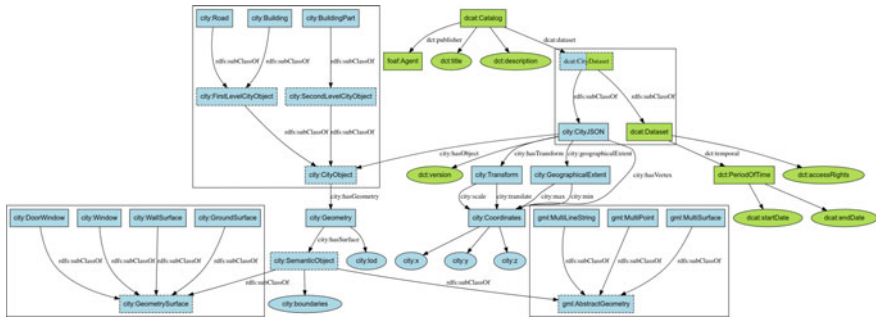


Fig. 2 Profile of CityJSON v.1.1 needed for the prototype test (in blue), integrated with the additional attributes intended to further specify the data, as represented in RDF. Rectangles are classes; the labels for the arcs define the relationship type (object property); abstract classes have dashed outlines; data properties are ellipses

buildings with roads in-between, excluding cul-de-sac roads serving single buildings or settlements, must be equal to the road width plus: (1) 5.00m per side, in case of streets width lower than 7.00m, (2) 7.50m per side, in case of streets width between 7.00 and 15.00m, (3) 10.00m per side, in case of streets width higher than 15.00m”.

From the analysis of the text, the relevant information was defined, and later mapped to the CityJSON v.1.1 data model. Additional requirements, for the overall dataset or the single objects were then defined and mapped, whether possible to the existing standards, in particular to metadata standards such as GeoDCAT or ISO 19115, or to the data product specification standard ISO 19131.

The resulting model, as represented in Fig. 2 was on the one hand described in tables defining a data requirement definition template intended to receive users input (Sect. 3.1), and then converted into RDF format to support the profiling tool developments (Sect. 3.3).

3.1 Data Requirement Definition Template

A template for the definition of the required information was developed. The goal of such a template is, on the one hand, to facilitate the municipalities officers, or the professionals commissioned by them, to define and publish the data requirements related to the geoinformation, as part of the digitalisation process of regulations and building permit overall. On the other hand, the result of the data requirement definition through the template will be made available to the data modellers and producers as a clear specification of the data to be produced. The format of such export should therefore be easily understandable by the modellers. In case the production of the 3D city model would use automatic modelling algorithms, a different machine-readable format able to input the useful parameters to such algorithm could be developed in future.

Table 1 Description of the dataset in the OGC Data Exchange tool template.

| Ref | GeoDCAT | | | | | | | - | ISO19131 | | | |
|--------------------|--|--|---|--|---|-----|----|----|--|-----------------|--|-------------------------|
| Field | Title | Spatial / geographic coverage | Temporal coverage | Description | Access rights / rights holder (Licence) | | | | Reference system | Unit of measure | Reference (standard) data model | Delivery Format |
| | | | | | BY | SA | NC | ND | | | | |
| Prescribed content | Title of the dataset, or topic, intended to identify the scope | Spatial extent of the dataset, defined in the chosen CRS | Temporal framework of the data representation | Short description of the dataset | BY | SA | NC | ND | Coordinate Reference System EPSG - OGC ID in Rainbow | Unit of measure | Reference data model and version URI in Rainbow (to format of be selected via a codelist name – URI) | Encoding of the dataset |
| Example | Example Building Permit 3D city model | surrounding of the new building e.g. parcel in which the new building is designed + 1km buffer | Updated with latest and approved new works | 3D city model LoD 3.3 intended to support automatic checks of urban regulation for new buildings within a GIS environment. | Yes | Yes | No | No | Local projected CRS | m | CityJSON v.1.1 URI | JSON |

In addition, the same information will be exported in a suitable format to be automatic input for the profiling tool explained in Sect. 3.3.

Table 1 summarises the information to be defined for the dataset to be modelled, including parameters useful to describe the necessary information and referring to metadata description standards, mostly GeoDCAT⁹ and ISO19131.¹⁰ This alignment will facilitate the storage of the same values as initial metadata set for the dataset publication, increasing its findability and re-usability later.

The field ‘Access rights/rights holder’ is intended to store the kind of license of the data. At the moment it is structured according to four criteria, compliant to the Creative Commons licenses model¹¹, based on which the user should indicate if it is acceptable that data license requires: “Attribution” (BY), “Share alike” (SA), “Noncommercial use” (NC) or “No Derivative works” (ND).

The field ‘Reference (standard) data model’ is intended to store the reference to the used (and profiled or extended) standard data model and specific version. It is supposed to provide the semantics and the structure for the application schema¹² being defined as a profile of it. The reference to the schema should be an official unique identifier. The defined profile will later specify the required application schema in detail (Table 2).

Table 2 explains the needed features of each object which should be included in the dataset, properly mapped to the suitable part of the standard data model (yellow part) and further specified as useful for the use case.

‘Reference data model subset’ attribute is used to indicate standard data models’ modules or any other kind of segmentation, as foreseen by the single standards. Application Domain Extensions, as foreseen in CityGML or other kinds of exten-

⁹ <https://semiceu.github.io/GeoDCAT-AP/releases/2.0.0/>.

¹⁰ <https://www.iso.org/standard/85092.html>.

¹¹ <https://creativecommons.org/about/cclicenses/>.

¹² The ‘application schema’, according to the definition from ISO 19101-1:2014, 4.1.2, is the conceptual schema, i.e. the formal description of a conceptual model, for data required by one or more applications or use cases.

Table 2 Detailed description of the dataset’s objects in the OGC Data Exchange tool template. The yellow fields are intended to define the needed profile, while the others help further specifying the needed features. In the examples, the purple fields are referring to the CityJSON v.1.1 schema, while the green ones refer to CityGML v.3, showing the applicability of the tool to different standards.

| Ref | - | Profile of standard data model | | | | | - | GeoDCAT | - |
|--------------------|---|---|--|--|--|--|--|------------------------------------|---|
| Field | Object | Reference data model subset | Class | SuperClass with property | Property | Geometry object | Accuracy | Spatial Resolution / LoD | Additional notes on the spatial representation |
| Prescribed content | Mapped city object | URI in Rainbow (to be selected via a codelist name - URI) | Selected from drop-down list based on the standard data model (with URIs in Rainbow) | Selected from drop-down list based on the standard data model (with URIs in Rainbow) | Selected from drop-down list based on the standard data model (with URIs in Rainbow) | Selected from drop-down list based on the standard data model (with URIs in Rainbow) | Based from the use case requirements defined with stakeholders | According to Biljecki et al., 2016 | Further specifying the kind of representation needed, based on the use case requirements |
| Example | Existing buildings | Building | Building | Building | Geometry.type | MultiSurface | 0.2 m | LoD 3.3 | To be included: external walls with windows, all protruding parts of the building, height = max height of the building at the top of the roof |
| | | Building | Window | Building | Geometry.type | MultiSurface | 0.2 m | | |
| | | Building | WindowSurface | AbstractThematicSurface | boundary | GM_MultiSurface | 0.2 m | | |
| | | Building | WallSurface | AbstractThematicSurface | boundary | GM_MultiSurface | 0.2 m | | |
| | | Building | WallSurface | AbstractThematicSurface | lod3MultiSurface | GM_MultiSurface | 0.2 m | | |
| | Road, including sidewalks, flowerbed parkings | Road | Road | Road | Geometry.type | MultiSurface | 0.2 m | LoD0 | |
| | Road | TrafficArea | AbstractThematicSurface | lod0MultiSurface | GM_MultiSurface | 0.2 m | LoD0 | | |
| Road | AuxiliaryTrafficArea | AbstractThematicSurface | lod0MultiSurface | GM_MultiSurface | 0.2 m | LoD0 | | | |

The yellow fields are intended to define the needed profile, while the others help further specifying the needed features. In the examples, the purple fields are referring to the CityJSON v.1.1 schema, while the green ones refer to CityGML v.3, showing the applicability of the tool to different standards

sions, such as the CityJSON ones can be referred here as well. ‘Class’, in case of CityJSON, refers to the kind of semantics associated to the object (e.g. ‘semantic surface’).

Whenever possible, the filling of each field should refer to uniquely identified resources, which can be automatically integrated with related semantic resources in the OGC Rainbow or other shared repository. For example, the fields to be filled through URI links can preferably be: the kind of license and access rights, the coordinate reference system, the standard data model which is the main reference for the semantic and structure, data format, as well as all the aspects of the profile defined (i.e., yellow fields in Fig. 2). Note that URI links may be in a convenient short form by identification of the “namespace”—URI base—for terms that form part of the canonical identifier—thus a simple term “Building” may be interpreted as “namespaceForCityGMLBuilding” provided the lexical form of the term is identical in each case—hence the value of the template derived from the underlying ontology to ensure this is robust.

To facilitate the filling of URIs as well as the consistent selection of standard data model features, the tables can be filled through an online interface allowing the selection of the possible contents of each cell from a drop-down list, in the cases in which the content should be restricted to specific values, including all the information related to the standard data model prescribed aspects (classes, properties, model version, etc.) and related URIs.

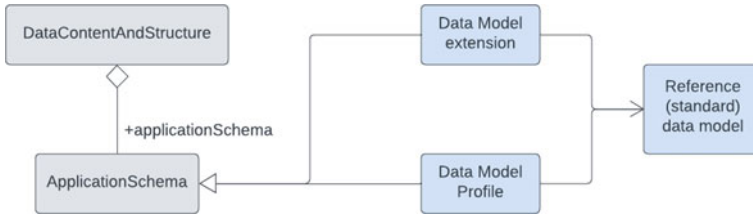


Fig. 3 Extension provided by the OGC data exchange toolkit presented in this paper (in blue) to the ISO19131 DataContentAndStructure section (in grey)

3.2 The OGC Data Exchange Toolkit as an Extension of ISO 19131

In the *DataContentAndStructureSection* of the *DataProductSpecification* standard ISO 19131.2, it is foreseen that the *DataContentAndStructure* might be represented by either Application schema information (MD_ApplicationSchemaInformation) or content information description (MD_ContentInformation), both borrowed from the metadata standard ISO 19115.

Application schema information as described in ISO 19115 supports the storage of information related to the application schema used, but it does not foresee explicitly the use of standardised domain data models or vocabularies, nor the use and definition of a profile of them.

Figure 3 depicts how the OGC Data Exchange Toolkit extends and further specify the ISO 19131 on ‘Data Product Specification’ (in grey) to allow a consistent representation of the application schema, referring to standardised data models and respective profiles or extensions (in blue).

3.3 The Profiling Toolkit

The OGC Profiling Toolkit is a suite of open source tools and resources for CI/CT/CD (Continuous Integration, Testing, Deployment) using GIT repositories.

Each tool is managed in separate repository and published as python packages to support reuse. Github actions are used to automatically apply these tools to a repository containing the profile being developed.

The tools leverage, and may publish outputs to **registers** of reusable assets, which are published as Linked Data using the Profiles vocabulary.

Thus the tools themselves embody the FAIR principles, the profile specifications developed using these can be published as FAIR Linked Data, and finally any service or dataset implementing the profiles generated can, by referencing this Linked Data describing the data, itself be published using FAIR principles. The ubiquitous nature of the URI identifiers and publication of the meta-models for profiles, specifications

Table 3 Components of the OGC profiling toolkit

| Component | Purpose | Interoperability foundations |
|--------------------------|---|-----------------------------------|
| Profiles register | Source and target for re-use of profiles | PROF vocabulary |
| Profile repository | Development of profiles | GIT |
| Validators | Rules defining a profile | e.g. SHACL |
| Examples | Illustration and testing of validators | e.g. JSON-LD |
| Description | Additional documentation | Source + rendered HTML |
| Source models | Authoritative standards | UML or encoding schema (e.g. XSD) |
| Source model OWL version | integration of model | OWL, SHACL |
| Controlled vocabularies | Profile data rules | SKOS |
| Validation pipeline | CI/CT/CD | GITHub actions, docker etc. |
| Semantic uplift | Link data to profile | JSON-LD |
| Semantic transformers | Data commonality (isomorphism) validation | SHACL-AF |

and profiles using the same semantic technologies ensures interoperability of all these different aspects.

Reusability is further enhanced by the provision of validated examples, allowing developers and users of specifications, implementing software or data using this structure to be confident they can easily understand derived profiles.

Table 3 identifies the elements of the Profiling Toolkit and the standards or common infrastructure that underpin these components.

The result of this methodology is a spectrum of related resources, each adhering to a simple suite of standards, with a set of validatable profiles for each standard. The overall architecture and the full set of interoperability requirements for each component are beyond the scope of this description, however using Linked Data principles each resource can be linked to standards and validation processes, and the data management history (provenance) or any other relevant metadata.

3.4 Test and Validation

To validate this approach a simple experiment was undertaken in advance of city-scale testing planned in the next phase.

The validation focuses on several questions:

- Which parts (profile) of CityJSON need to be present?
- Which specific additional requirements the dataset and data objects should comply to?
- Can such a profile be described and used to validate data meets these requirements?

Figure 4 shows a snippet of the CityJSON test case (with verbose details such as coordinates and vertex references collapsed for readability). The “semantics” element

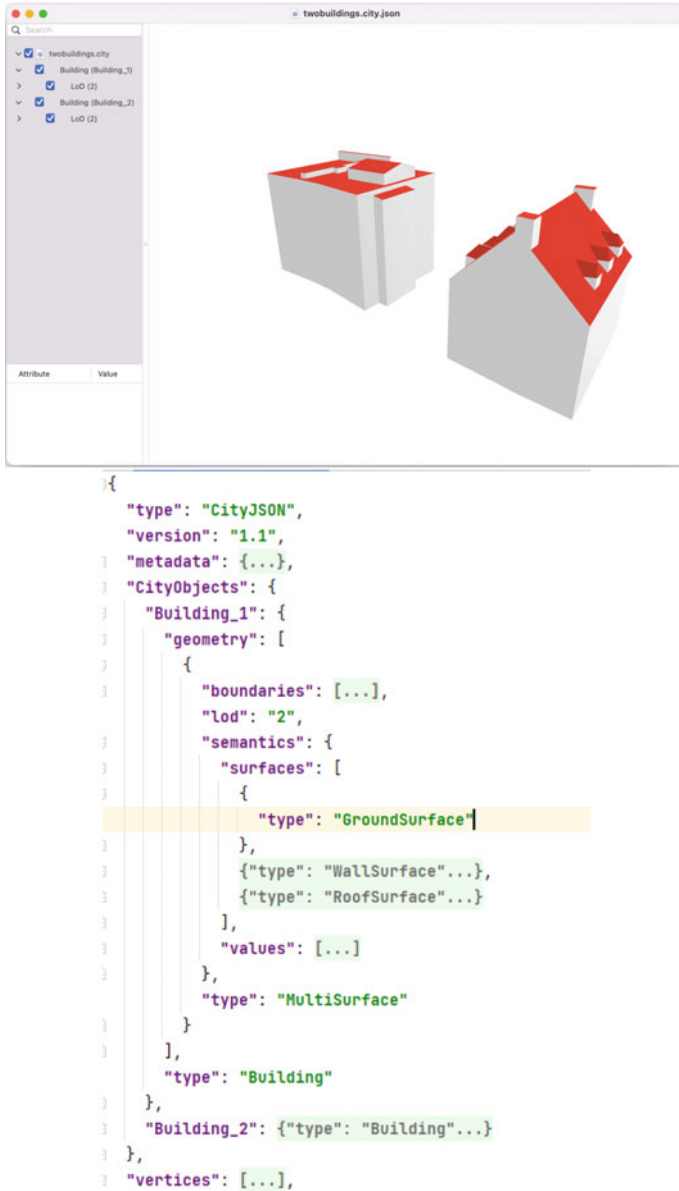


Fig. 4 Sample CityJSON data for a simple test case. Visualization in azul (Arroyo Ohori 2020) and JSON encoding (with some details collapsed)

(e.g. “WallSurface” provides a sub-typing approach that is equivalent to the Class specialisation in the CityGML UML model.

```

"@context": {
  "@version": 1.1,
  "@base": "#",
  "city": "http://example.com/vocab/city/",
  "dct": "http://purl.org/dc/terms/",
  "xsd": "http://www.w3.org/2001/XMLSchema#",
  "CityJSON": "city:CityJson",
  "type": "@type",
  "version": "city:version",
  "CityObjects": {
    "@id": "city:hasObject",
    "@context": {
      "Bridge": "city:Bridge",
      "BridgeConstructiveElement": "city:BridgeConstructiveElement",
      "BridgeFurniture": "city:BridgeFurniture",
      "BridgeInstallation": "city:BridgeInstallation",
      ...
    }
  }
}

```

Fig. 5 Example section of JSON-LD context that maps JSON structural elements to the CityGML semantic model

Using the OGC Profiling Toolkit a JSON-LD context document (Fig. 5) can be used to transform the JSON into an explicit implementation of the CityGML model. The complexity of this process is beyond the scope of this experiment, however the tools to perform linked uplift and transformation pipelines are an important component of the OGC Profile Toolkit necessary to deal with the myriad of alternative ways of structuring information. The general approach is to map the data structure to the semantic concepts early, which then allows alternative representations to co-exist in the same Linked Data environment if necessary.

Figure 6 shows a snippet of the resulting form of this data matching the target ontology (CityGML). This RDF syntax (Turtle) represents the data in a canonical form that can be directly validated against the model and profile using standardised validation rules and toolkits. Figure 7 shows an example validation rule in the SHACL (Shapes Constraint Language) standard.

The defined rules and profile were successfully tested against the ‘2buildings’ sample dataset (Fig. 4). The resulting validation report (Fig. 8) shows three different types of validation that can be made. The first relates to general requirements—such as data licenses, date (currency) of information etc. The second violation relates to whether the particular dataset has the right level of detail—this is optional in the underlying CityJSON (and CityGML) models, but a requirement to meet the CHEK business rules.

```

<#city-objects-Building_1> a city:Building ;
  city:hasGeometry [
    city:hasSurface [
      a city:GroundSurface, gml:MultiSurface ;
      city:boundaries ( ( ( <#vertices-0> <#vertices-1> <#vertices-2> ) )
    ] , [
      a city:WallSurface, gml:MultiSurface ;
      city:boundaries ( ( ( <#vertices-11> <#vertices-9> <#vertices-12> )
    ] , [
      a city:RoofSurface, gml:MultiSurface ;
      city:boundaries ( ( ( <#vertices-20> <#vertices-19> <#vertices-21> )
    ] ;
    city:lod "3.3" ;
  ] .|

```

Fig. 6 Example canonical form of data ready for validation

```

<#Lod3.3>
  a sh:NodeShape ;
  sh:targetClass city:Building, city:BuildingPart, city:BuildingRoom, city:BuildingStorey,
  city:BuildingUnit, city:BuildingInstallation;
  sh:not [
    sh:sparql [
      sh:prefixes ex: ;
      sh:select """
        SELECT $this (city:lod as ?path) (?lod as ?value) WHERE {
          $this city:hasGeometry/city:lod ?lod .
          FILTER regex(str(?lod), '^3.3|4.*$')
        } LIMIT 1
        """ ;
    ] ;
  ];
  sh:message "Building has LoD < 3.3" ;
  sh:severity sh:Violation
.

```

Fig. 7 Example SHACL rule (LOD present)

Thus the OGC Profiling Toolkit can support the full process of bringing one or more different data forms into a common semantic framework where explicit rules can be executed to validate the data. The profiles allow sophisticated and fine-grained control over exact data requirements.

4 Discussion

This initial development allowed testing the feasibility of the approach, which will enable the development of a more complete profiling tool for more complex cases.

```

Validation Report
Conforms: False
Results (6):
Constraint Violation in NotConstraintComponent (http://www.w3.org/ns/shacl#NotConstraintComponent):
  Severity: sh:Violation
  Source Shape: <file:///github/workspace/ogc/chek-profiles/cityjson/paper/shapes.ttl#License>
  Focus Node: <http://example.com/twobuildings.ttl#city>
  Value Node: <http://example.com/twobuildings.ttl#city>
  Message: License does not support commercial usage
Constraint Violation in NotConstraintComponent (http://www.w3.org/ns/shacl#NotConstraintComponent):
  Severity: sh:Violation
  Source Shape: <file:///github/workspace/ogc/chek-profiles/cityjson/paper/shapes.ttl#Lod3.3>
  Focus Node: <http://example.com/twobuildings.ttl#city-objects-Building_2>
  Value Node: <http://example.com/twobuildings.ttl#city-objects-Building_2>
  Message: Building has LoD < 3.3
Constraint Violation in NotConstraintComponent (http://www.w3.org/ns/shacl#NotConstraintComponent):
  Severity: sh:Violation
  Source Shape: <file:///github/workspace/ogc/chek-profiles/cityjson/paper/shapes.ttl#RoadsRequired>
  Focus Node: [ ]
  Value Node: [ ]
  Message: Dataset contains no Road objects

```

Fig. 8 Validation report from the testing of the developed profile over the ‘2buildings’ sample dataset.

A significant challenge, was to choose a data encoding for sample data. In this case CityJSON was chosen to represent the CityGML model. The exact degree of conformance of CityJSON to the full CityGML model is an open question, however the working hypothesis is that for the subset required (the CHEK profile) CityJSON and CityGML terminology and semantics are equivalent, even though for the case of CityJSON the mapping of concepts to data structures for elements such as geometry and topology are not formally defined.

Some limitations and obstacles to be overcome were pointed out, to support the future efforts. A key obstacle, which is directly addressed by this approach, is the lack of formal descriptions of how existing data formats and datasets relate to common underlying concepts. The CHEK project will demonstrate that every aspect does not need to be resolved to demonstrate that for a specific profile of each standard sufficient interoperability can be established to automate data transformation and business rule evaluation.

A key outcome of this is the potential to characterise the commonality, differences, inter-relationships and transformation options between different 3D geometry and topology regimes, which in turn becomes a building block for describing integrations between different application domain models in the 3D environment. This outcome arises in part from the unique approach CityJSON uses to define a set of vertices and then reference them in nested array structures carrying implicit semantic information (e.g. the first level of nesting of a solid is the set of faces of that solid).

The inclusion of more detailed specification of the single objects in a structured format allows removing the ambiguity still remaining in many datasets, even though they may conform to various structural data models. Extending specifications to the specific semantic information required to support transformation to other well-known standards enabling real interoperability and re-usability of datasets. Datasets that

declare conformance to specific profiles can be reliably validated and transformed using automated and robust pathways in a transparent and reproducible fashion, using tested reusable tools to reduce effort, cost and risk of invalid outcomes. This is a step further towards the compliance to the FAIR principles, for which simple compliance to standard data model is often not sufficient.

Creating profiles of complex standards such as OGC CityGML and buildingSMART Industry Foundation Classes (IFC)¹³ for Building Information Models (BIM) allow for data sharing, including supporting exchanges between Geographical Information Systems and engineering environments by focusing on the “Goldilocks zone”. Furthermore, the flexible nature of semantic expressions of this simplified views means these can be extended for specific purposes later, leaving a simplified core neither over-specific nor under-specified for the practical purposes of integrating separate systems for specific purposes.

Vocabularies, syntactical rules for data, inferencing rules for related data and consistency checking and transformations between related data models can all then be defined against the profile, without requiring the full standard data model to be treated in the same way. Example data objects can be used to test these models (and create useful, validated examples in documentation), using the same profiling toolkit used for each part of the overarching spectrum of resources.

5 Conclusions

A prototype of a data exchange toolkit for 3D geoinformation was developed in form of a data specification template intended to be filled (for the moment) manually, according to the use case needs, and an OWL-based profiling tool to support automatic extraction and validation of datasets against the defined profile.

In future work the tool will be further tested with more extended data requirements.

In addition, it will be completed to include the whole CityGML v.3 standard model, as well as extending the functionalities, by following the same methodology, to other standard data models from OGC and possibly external ones (e.g. the INSPIRE thematic model¹⁴). Finally, the management of Application Domain Extensions or similar mechanisms to extend the standard data models will be added and tested in future releases of the tool.

An additional attribute, relevant for the dataset description and possible data requirement could be ‘Language’, as also proposed within the GeoDCAT model. However, for the moment we only consider datasets in English as a prescription. Future works could be intended to extend the tool to foresee the use of different languages. Other integrations and extensions may be considered within the future developments.

¹³ <https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/>.

¹⁴ <https://inspire.ec.europa.eu/data-model/approved/r4618-ir/html/>.

Moreover, the toolkit will be refined within the CHEK project as well as in other research projects in which the OGC is involved (e.g. USAGE¹⁵, AD4GD¹⁶, ACCORD¹⁷, ILIAD¹⁸) as well as becoming part of the OGC developments and discussions with OGC members, to improve the performances and test within multiple use cases.

Using the same approach, further checking can be performed on the objects themselves against the business rules (permit checking in this case). Such rules are not limited to SHACL, however by expressing these against the canonical (semantic) model such business rules can be applied to many forms of original source data, and can be more transparent because specialised code to apply to specialised data structures is avoided.

The results of the initial experimentation has confirmed positive answer to the feasibility questions and show a mechanism to define a profile can be based on a spectrum of standardised components, linked by open and transparent data transformation and validation pipelines.

The complexity of the structural transformations (such as geometry and sub-typing classifications) highlights how important it is to isolate these aspects from the business rules. The business rules can be defined against the single semantic model, and any complex structural transformations between semantically equivalent encodings can be done using reusable tools.

Acknowledgements This work was funded by the ‘Change toolkit for Digital Building Permit’ (CHEK) project, funded by the European Union’s Horizon Europe programme under Grant Agreement No.101058559.

References

- Arroyo Ohori K (2020) azul: A fast and efficient 3d city model viewer for macOs. *Trans GIS* 24(5):1165–1184
- Hjelseth E, Nisbet N (2011) Capturing normative constraints by use of the semantic mark-up RASE methodology. In: *Proceedings of CIB W78-W102 conference*, pp 1–10
- ISO/TC 211 (2014) Iso 19115-1:2014, geographic information—metadata—part 1: fundamentals. Standard ISO 19115-1:2014, International Organization for Standardization, <https://www.iso.org/standard/53798.html>
- ISO/TC 211 (2022) Iso 19131:2022, geographic information—data product specifications. Standard ISO 19131:2022, International Organization for Standardization, <https://www.iso.org/standard/85092.html>
- Ledoux H, Arroyo Ohori K, Kumar K, Dukai B, Labetski A, Vitalis S (2019) CityJSON: a compact and easy-to-use encoding of the CityGML data model. *Open Geospatial Data Softw Stand* 4(1):1–12

¹⁵ <http://usage-project.eu>.

¹⁶ <https://ad4gd.eu>.

¹⁷ <https://accordproject.eu>.

¹⁸ <https://www.ocean-twin.eu>.

- Noardo F (2022) Multisource spatial data integration for use cases applications. *Trans GIS* 26(7):2874–2913
- Noardo F, Arroyo Ohori K, Biljecki F, Ellul C, Harrie L, Krijnen T, Eriksson H, van Liempt J, Pla M, Ruiz A et al (2021) Reference study of CityGML software support: the GeoBIM benchmark 2019-part ii. *Trans GIS* 25(2):842–868
- Raes L, Vandenbroucke D, Reznik T (2019) Geodcat-ap. OGC Discussion paper 18-001r1, Open Geospatial Consortium, https://portal.ogc.org/files/?artifact_id=82475
- Wagner D, Wewetzer M, Bogdahn J, Alam N, Pries M, Coors V (2012) Geometric-semantical consistency validation of CityGML models. In: *Progress and new trends in 3D geoinformation sciences*, Springer, pp 171–192
- Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten JW, da Silva Santos LB, Bourne PE et al (2016) The fair guiding principles for scientific data management and stewardship. *Sci Data* 3(1):1–9

cjdb: A Simple, Fast, and Lean Database Solution for the CityGML Data Model



Leon Powalka, Chris Poon, Yitong Xia, Siebren Meines, Lan Yan, Yudian Cai, Gina Stavropoulou, Balázs Dukai, and Hugo Ledoux

Abstract When it comes to storing 3D city models in a database, the implementation of the CityGML data model can be quite demanding and often results in complicated schemas. As an example, 3DCityDB, a widely used solution, depends on a schema having 66 tables, mapping closely the CityGML architecture. In this paper, we propose an alternative (called ‘cjdb’) for storing CityGML models efficiently in PostgreSQL with a much simpler table structure and data model design (only 3 tables are necessary). This is achieved by storing the attributes and geometries of the objects directly in JSON. In the case of the geometries we thus adopt the *Simple Feature* paradigm and we use the structure of CityJSON. We compare our solution against 3DCityDB with large real-world 3D city models, and we find that cjdb has significantly lower demands in storage space (around a factor of 10), allows for faster import/export of data, and has a comparable data retrieval speed with some queries being faster and some slower. The accompanying software (importer and exporter) is available at <https://github.com/cityjson/cjdb/> under a permissive open-source license.

Keywords CityGML · 3DCityDB · 3D modelling · DBMS · CityJSON

1 Introduction

The international standard *City Geography Markup Language* (CityGML) is a data model designed for the storage of digital 3D models representing urban areas and landscapes (Kutzner et al. 2020; Gröger and Plümer 2012; OGC 2021b). It allows us to define and store the majority of commonplace 3D objects within cities, such as

This article was selected based on the results of a double-blind review of the full paper.

L. Powalka · C. Poon · Y. Xia · S. Meines · L. Yan · Y. Cai · G. Stavropoulou · H. Ledoux (✉)
Delft University of Technology, Delft, The Netherlands
e-mail: h.ledoux@tudelft.nl

B. Dukai
3DGI, Delft, the Netherlands

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
T. H. Kolbe et al. (eds.), *Recent Advances in 3D Geoinformation Science*, Lecture Notes
in Geoinformation and Cartography, https://doi.org/10.1007/978-3-031-43699-4_47

781

buildings, roads, rivers, bridges, vegetation, and city furniture. Additionally, it supports various levels of detail (LoDs) for the 3D objects, which enables and facilitates complex applications and use-cases (Biljecki et al. 2015).

The CityGML data model, currently at version 3.0, has three known encodings (more details in Sect. 2):

XML/GML encoding: The XML/GML encoding (built upon GML (OGC 2007)) was initially the only standardised encoding for CityGML, which explains the—rather confusing—name choice for the data model. The latest official release of the XML/GML encoding supports CityGML version 2.0 (OGC 2012); however a release planned for 2023 will also support CityGML v3.0.

CityJSON: Its version 1.0 is standardised by the OGC (OGC 2021a), and its version 1.1¹ implements a subset of the CityGML v3.0 data model. Its flat hierarchy and simple structure make it around 6 times more compact than the XML/GML encoding, thus allowing for easier manipulation and exchange on the web (Ledoux et al. 2019).

3DCityDB: The *3D City Database* is a “geo-database solution” (schema and accompanying software) supporting three different relational DBMSs (database management systems). It implements a mapping of the CityGML data model (currently for v1.0 and v2.0 only) to the database schema to allow for a fast implementation (Yao et al. 2018). It is not standardised by the OGC.

DBMSs can greatly simplify the management of large 3D city models: they are arguably the best tool to store and manage very large datasets (of any kind), are already part of the ecosystem of many organisations, and offer several advantages over file-based systems, eg security, versioning, scalability, etc. (Ramakrishnan and Gehrke 2001). This makes 3DCityDB a popular solution, especially for handling country-level data and for offering access to multiple users. In most cases, the data owners store the data with 3DCityDB on a remote server and allow the users to access the data through a website, filter it by objects/LoDs/areas and obtain a subset of the 3D city model, in various different formats, for instance KML, COLLADA, and gITF.

However, while the 3DCityDB is widely used, Pantelios (2022) argues that its use can be somewhat complex and difficult for end-users. The main culprit is the fact that datasets are split over 66 tables and the *Simple Feature* paradigm OGC (2006) is not used (geometries are stored across different tables, not in a column of an object), which translates to very complex queries that necessitate several joins. Pantelios (2022) solution is to create extra views for attributes and geometries, and to offer simplified access to them through a graphical interface (built upon QGIS). However, this comes at the cost of increasing the size of the database.

We present in this paper an alternative to 3DCityDB, which we name ‘cjdb’. It is composed of a database schema (containing only 3 tables) and accompanying software for import and export of CityJSON v1.1 files (thus the CityGML v3.0 core model is supported). As further explained in Sect. 3, our data model is inspired

¹ <https://cityjson.org>.

by the *Simple Feature* paradigm (each row has the geometries of the object stored in one column), but instead of using PostGIS geometry types, we exploit the fact that PostgreSQL can store JSON objects directly in binary format with the `jsonb` type. The reason for this choice is that PostGIS geometry types (notice that to use 3D types the SFCGAL extension (Borne et al. 2023) would be required), would not allow the storage of appearances (textures and/or materials) and of semantic information on the surfaces. Therefore, the geometries of a given city object (eg a building, a tree, a lamppost) are stored together with the object in JSON format, as defined by CityJSON. Our simple structure allows us to compress by an order of around 10 the typical size of a database as stored with 3DCityDB (taking into account data and (spatial) indexes), and, as shown in Sect. 4, this size reduction does not come with a penalty for the speed of the data retrieval. Our data model is at the moment only for PostgreSQL, but because it is so simple (only 3 tables are necessary), it could surely be ported to other databases.

2 Related Work

2.1 CityGML Data Model

To represent a region in 3D, CityGML recursively decomposes it into semantic objects (Gröger and Plümer 2012). It defines the classes most commonly found in an urban or a regional context, and the hierarchical relationships between them (eg a building is composed of parts, which are formed of walls, which have windows). Also, the CityGML semantic classes are structured into several modules, eg Building, Land Use, Water Bodies, and Transportation.

The geometry of the objects is realised with a subset of the geometry definitions in ISO19107 (ISO 2003) (only linear and planar primitives are however allowed), which also allows aggregations of geometries: a single building can for instance be modelled with a `CompositeSolid`. Furthermore, it is possible to attach textures, materials, and semantics to each of the surfaces of a 3D geometry. The geometry types of most geo-DBMSs do not allow us to represent such complex 3D geometries.

One of the main characteristics of CityGML is that it supports different levels of detail (LoDs) for each of the classes, which means that in theory for a single building, or a single tree, several geometries could be stored.

2.2 CityJSON + CityJSONFeature

CityJSON, with its latest version 1.1, is a JSON-based exchange format for the CityGML data model. It implements all the core modules of CityGML v3.0, and some

other modules are supported.² As explained in Ledoux et al. (2019), it was designed to improve the weaknesses of the XML-encoding of CityGML: large filesize, complex structure to manipulate, several ways to store a given characteristic, unfit for the web, etc.

Its geometry structure is similar to that of computer graphics formats (eg OBJ and STL), and allows us to compress by a factor of around 6 XML-encoded CityGML files. A thorough comparison shows that it is nearly as compact as formats that do not allow semantics, complex attributes, and coordinate reference systems (Praschl and Pointner 2022).

While the original files for CityJSON v1.0 were compact, one weakness was that files for large areas were not suitable for streaming. That is, to be able to process one object in a file, the client had to have in memory the whole file. Version 1.1 solved this issue by introducing a new type: `CityJSONFeature`, which represents *independently* one city object in a CityJSON file (eg a ‘Building’ or a ‘Bridge’). The idea is to decompose a region into its many features, create several JSON objects of type `CityJSONFeature`, and stream them sequentially or store them in a JSON text sequence (IETF 2015).³ This is conceptually the same as the *GeoJSON Text Sequences*⁴ used for processing and exchanging large 2D GIS datasets.

We exploit this type to store independently each feature in one row of the database, although, as explained below, we modify the JSON structure slightly, split it over a few columns, and use indexes to accelerate performance.

2.3 NoSQL Databases

Nys and Billen (2021) developed a non-relational data model for CityJSON, and implement it in a JSON document database (MongoDB). They tested with one dataset (containing around 3500 city objects), and managed to reduce the size by a factor of 40% when compared to 3DCityDB. Only one query was benchmarked (retrieval of a random building with its attributes and single geometry), and their solution performed better than 3DCityDB (again around 40% faster). There is no information about how their solution performs with other queries that practitioners expect from a DBMS solution (eg the ones from Section 4.3).

2.4 3DCityDB

While the data model of CityGML could have been automatically mapped to relational tables, Yao et al. (2018) mentions that for 3DCityDB a semi-automatic method

² see details at: <https://www.cityjson.org/citygml/v30/>.

³ *JSON Lines text file* is one possibility: <https://jsonlines.org/>.

⁴ <https://datatracker.ietf.org/doc/html/rfc7946#appendix-C>.

was used to reduced the number of tables and the number of joins to perform queries (which will also typically reduce query times). The result is nonetheless, for v4.4, a total of 66 tables, many of which remain empty if, for instance, only buildings without appearances are stored.

As Pantelios (2022) mentions, the attributes for a given type are stored in different tables, depending on whether an attribute is prescribed by the CityGML data model or not. This complicates greatly the retrieval of information with SQL queries.

Interestingly, the 3D geometries are decomposed into their (semantic) surfaces, and each surface is stored in a separately row in a single table. Different flags are used to indicate whether a 3D geometry is a solid/watertight or a surface/not. While this approach allows to compactly store 3D volumetric geometries, in practice several joins are necessary to retrieve all the surfaces of a given feature. The volumetric 3D types available in PostGIS-SFCGAL (Borne et al. 2023) are also used (thus duplication of data).

Interestingly, PostGIS geometries for the footprint are not stored, only the 2D bounding box. This is in our opinion an odd choice because many queries on 3D city models are 2D queries (all buildings inside a given area, within a given distance, etc.).

3 Data Model, Software, and Engineering Decisions

3.1 Data Model

As shown in Fig. 1,

the cjdb data model is simple and akin to using the Simple Feature paradigm (OGC 2006), as PostGIS does. Each row in the table `city_object` stores a CityJSON city object (for instance a 'Building', a 'BuildingPart', a 'SolitaryVegetationObject', etc.) and the 'geometry' column stores a JSON array of geometries (Fig. 2 shows an example).

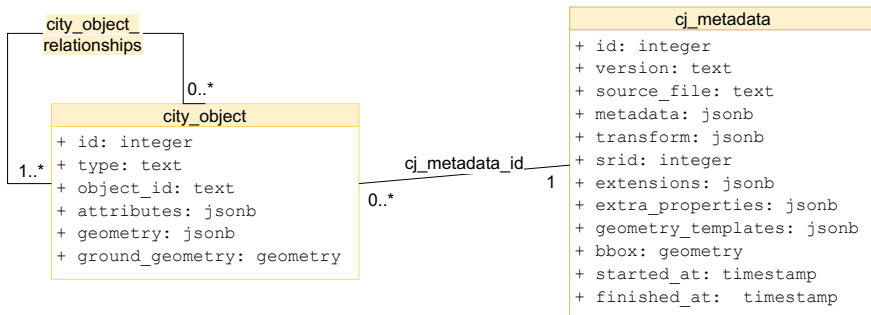


Fig. 1 UML diagram of cjdb

```

1  [
2  {
3    "type": "Solid",
4    "lod": "2.2",
5    "boundaries": [
6      [ [ [11.1, 22.6, 9.9], [16.21, 42.8, 19.9], ... ]
7    ],
8    "semantics": {
9      "surfaces": [
10     { "type": "RoofSurface" },
11     { "type": "WallSurface" },
12     ...
13     ],
14     "values": [ [0, 1, ...] ]
15   }
16 }
17 ]
18

```

Fig. 2 Example snippet stored in the ‘geometry’ column: an array of CityJSON geometries

The array is necessary because a given feature can have more than one geometry (eg the most obvious case is when several LoDs are stored), and it should be noticed that the JSON stored in the database is different from CityJSON: vertices are not stored separately and we replace the vertex identifiers in the “boundaries” property by their real-world coordinates.

For each row, we also store the identifier of the object, its type, the attributes (stored as `jsonb`), and we also store the ground geometry as a PostgreSQL 2D type (to be able to spatially index them in 2D, see below). Notice that the children objects are stored in separate rows (and not with their parent object), ie a ‘Building’ does not have its potential children in the same row, and each of them (eg a ‘BuildingPart’) is stored in a separate row.

The implemented version adds one more table: a table named `city_object_relationships` to store the relations between ‘parents’ and ‘children’ city objects (CityJSON has a flat structure and stores for instance a ‘Building’ at the same level as its ‘BuildingPart’ or ‘BuildingInstallation’ and links them). This table is added to improve query speed since we often need to process a parent with its children.

Finally, the table `cyj_metadata` has one entry for each CityJSON file imported to the database and stores the following properties of CityJSON:

- the coordinate reference system (CRS);
- the precision used for storing the coordinates, used mostly when exporting data;
- the geometry templates (used for trees or bus stops);
- the CityJSON Extensions (to extend the base data model of CityJSON with application-specific types/attributes/semantics);
- the bounding box of the file.

3.2 *Importer*

To facilitate the usage of the cjdb data model we have implemented an importer and released it open-source under the MIT license: <https://github.com/cityjson/cjdb>.⁵ The tool is developed in Python and has a command-line interface. Observe that because the data model is feature-centric, the importer will read JSONL files (JSON Sequences) and not CityJSON files. A CityJSON file can however be automatically converted into a list of features with the accompanying software *cjio*.⁶

The importer creates the 3 necessary tables, and populates them by parsing and modifying the CityJSON features according to the cjdb data model, as explained above.

Ground geometry extraction: As many queries on a 3D city model are typically performed in 2D, such as retrieving all objects within a certain area or selecting the object clicked upon in a 2D view, we have chosen to store the ground surface of each object as a 2D PostGIS geometry. This is achieved by iterating over all of the object's surfaces and selecting the horizontal ones with the lowest elevation. If multiple levels of detail (LoDs) are available, we select the lowest LoD. This addition enables us to perform rapid 2D spatial queries on the data without any joins (see Sect. 4). In comparison, performing the same spatial queries in 3DCityDB requires multiple joins. Alternatively, the enveloping bounding box, which in 3DCityDB is stored together with the object, can be used instead of the actual object geometry in order to perform spatial queries when the accuracy is not important.

Indexes: The data in the `city_object` table is expected to be retrieved mostly through spatial queries. Therefore, we decided to add a GiST index on the 'ground_geometry' column and cluster the table based on that index. In order to improve query performance on the JSON columns we added a Generalized Inverted index (GIN), which is specialised for items with composite values. Additional full or partial BTree indexes can be applied during import, on specific attributes of the city objects, if the user expects that the table will be queried based on those attributes.

3.3 *Exporter*

The Python implementation of cjdb also offers an exporter. As is the case for 3DCityDB, a SQL query is used to filter which objects in the database should be exported (the identifiers in the table `city_object`). The output is a CityJSONL file (a sequence of `CityJSONFeatures`), which can automatically be converted to a CityJSON file with *cjio*.

⁵ The version described in this article is v1.3.

⁶ <https://github.com/cityjson/cjio>.

Table 1 The 3 datasets used for the benchmark

| | # Building | # BuildingPart | LoDs present | # attributes |
|--------|------------|----------------|---------------|--------------|
| 3DBAG | 112673 | 110387 | 0/1.2/1.3/2.2 | 30 |
| NYC | 23777 | 0 | 2 | 3 |
| Vienna | 307 | 1015 | 2 | 7 |

Table 2 Import and export times, from/to CityJSONL

| | 3DCityDB | | cjdb | |
|--------|----------|--------|--------|--------|
| | Import | Export | Import | Export |
| 3DBAG | 6780 | 721 | 1260 | 412 |
| NYC | 273 | 161 | 23 | 25 |
| Vienna | 12 | 7 | 2 | 2.5 |

All times in seconds

4 Benchmark

To compare the performance of the cjdb data model against that of 3DCityDB, we created a benchmark dataset using data from 3 different countries; the Netherlands (3DBAG), Austria (Vienna), and USA (NYC). The 3DBAG dataset is composed of 100 tiles from the 3DBAG⁷ (Peters et al. 2022); we randomly chose tiles from 3 cities in the Netherlands (Delft, Amersfoort, and Zwolle). The Vienna dataset covers the Austrian city whereas the NYC dataset covers a small part of central New York City; both datasets can be downloaded in CityJSON format from <https://www.cityjson.org/datasets/>. As can be seen in Table 1, all the datasets are building-centric, as is often the case with 3D city models, but here they are modelled differently and have different LoDs/sizes.

We imported each dataset in two different databases, one created with 3DCityDB and one cjdb, and below we compare them in terms of import/export time, data size, and data retrieval. Since cjdb is only available for PostgreSQL, we did not perform any tests on Oracle or PolarDB.

4.1 Import and Export Times

We compared the import time for all 3 datasets and we found that cjdb is considerably faster than 3DCityDB. As an example, the 100 tiles of the 3DBAG were imported in 21 min in cjdb whereas it took 113 min with 3DCityDB; see Table 2 for all details. This is expected, since the storage in the cjdb database is very close to the data model

⁷ www.3dbag.nl.

of CityJSONL, while for 3DCityDB the geometries need to be processed and each surface to be stored separately. However, it is worth noting that in 3DCityDB the creation of the necessary tables is performed separately before the import, whereas cjdb creates the necessary tables at the time of import.

For the export, we exported each dataset to a CityJSON file (for 3DCityDB) and to a CityJSONL file (for cjdb). As shown in Table 2, cjdb is also generally faster.

Observe that those values are somewhat unfair to compare because the output is to a different format (and some time would be needed to convert from a CityJSONL to a CityJSON), because different languages are used for the importer/exporter (Java for 3DCityDB, Python for cjdb), and because 3DCityDB exporter is multi-threaded.

4.2 Database Size

One significant difference between 3DCityDB and cjdb is the database size they occupy. After importing the datasets, we measured the total size of each database, including the indexes and the TOAST tables (The Oversized Attribute Storage Technique), as shown in Table 3. We notice that cjdb occupies significantly less space than what 3DCityDB demands for the same data.

The main reason for this size difference lies in the different approaches to storing the features. In 3DCityDB, the different semantic surfaces (wall/roof/ground/etc) are considered separate city objects, contributing to the total number of rows in the `cityobject` table. In cjdb, the walls, roof and surfaces are considered intrinsic properties of each city object and thus remain in the `jsonb` format stored at the geometry column of each object. This decision has several advantages: reduction of the size of the `city_objects` table and faster and simpler queries where only city objects are concerned (eg Q1 and Q4 in Table 4). But there is also a major drawback: we cannot perform spatial queries on the geometries of specific semantic surfaces.

Another reason for the different database sizes is the different amount of indexes. We notice that cjdb implements significantly less indexes and as a result requires way less storage for them. In 3DCityDB, indexes account for almost half of the database's size.

One more noticeable difference is that, since cjdb stores attributes and geometries in `jsonb` format, it requires more space for TOAST tables. TOAST (*The Oversize Attribute Storage Technique*) is a PostgreSQL mechanism which controls the size of the data stored in a field. If the data exceeds the maximum allowed limit, TOAST breaks the too-wide field values down into smaller pieces, and stores them out-of-line in a TOAST table. Columns of type `jsonb` tend to carry quite wide values and often utilise the TOAST tables. Thus when measuring the cjdb database size we need to take the extra toast tables into account. However, even with the extra TOAST tables, the total database size remains around ten times smaller than that of 3DCityDB.

Finally, it should be noticed that the 3DCityDB size should in reality be larger than the number we obtain: since its data model allows only but one geometry per LoD and that the refined LoDs by Biljecki et al. (2016) are not supported, the 3DCityDB

Table 3 Database size comparison for the 3 datasets, all values in MB

| | 3DCityDB | | | | cjdb | | | |
|---------------|----------|---------|-------|-------|--------|---------|-------|-------|
| | Tables | Indexes | TOAST | Total | Tables | Indexes | TOAST | Total |
| 3DBAG | 5463 | 4322 | 112 | 9898 | 257 | 57 | 755 | 1070 |
| NYC | 590 | 735 | 0.5 | 1326 | 26 | 4 | 25 | 54 |
| Vienna | 30 | 42 | 0.5 | 73 | 1.5 | 0.5 | 4 | 6 |

Table 4 The 8 queries we used for the benchmark

| | |
|----|---|
| Q1 | Retrieve the ids of all buildings based on one attribute (roof height higher than 20 m) |
| Q2 | Retrieve all buildings within a 2D bounding box |
| Q3 | Retrieve building intersecting with a 2D point |
| Q4 | Retrieve the number of parts for each building |
| Q5 | Retrieve all buildings having a specific LoD geometry |
| Q6 | Add new ‘footprint_area’ attribute |
| Q7 | Update ‘footprint_area’ attribute by adding 10 m |
| Q8 | Delete ‘footprint_area’ attribute |

importer selects either the LoD1.2 or LoD1.3 from the inputs, and thus one LoD is missing. In cjdb, all available LoDs are stored together in the ‘geometry’ column.

4.3 Data Retrieval

The data retrieval comparison was performed based on the execution time of SQL queries which aim to retrieve the same data from both databases. PostgreSQL heavily relies on caching, therefore the queries below were run several times to ensure the cache was warm.

We performed 8 queries that we believe are representative of what a typical practitioner (or server hosting data to be downloaded) would need.

Those are listed in Table 4. The exact SQL queries we used for the 3DBAG dataset are listed in Appendix 6; similar queries were used for the other 2 datasets.

Q1. Query based on attributes: 3DCityDB offers a list of predefined building attributes within the `building` table, which include ‘year_of_construction’ and ‘roof_type’—attributes that are not in this list are stored in the table `cityobject_genericattrib`. Cjdb on the other hand offers more flexibility since all the attributes remain in JSON format in the `attributes` column, regardless of the attribute name.

Since none of our datasets have attributes from the 3DCityDB’s predefined list, we decided to compare the attribute-based data retrieval for both databases based on non-listed attributes. In this specific example, we queried all the buildings with

roof height ('h_dak_max' for BAG 'HoeheDach' for Vienna) higher than 20 m. The New York dataset was not taken into account for this query, since there is no specific attribute about the roof height.

For cjdb no join is necessary since the attributes are stored together with the city object but the equivalent in 3DCityDB requires a join between the `city_object` and the `cityobject_genericattrib` tables. As shown in Table 5, cjdb is faster than 3DCityDB for Vienna but performs almost the same as 3DCityDB for the 3DBAG dataset. This is related to both the size of the dataset and the size of the attributes column in cjdb; the bigger the `jsonb` column, the slower the query, since the information stored in the TOAST tables will need to be retrieved and decompressed.

Q2 & Q3. 2D Spatial queries: The spatial queries in 3DCityDB tend to be complicated since the geometries of the objects are stored in other tables and require joins to retrieve them. As an example, in order to find all buildings within a certain 2D bounding box, their ground surfaces had to be retrieved from the `surface_geometry` table. An alternative but less accurate solution would be to use the bounding box geometry of the object, which is stored in the city objects' table. Cjdb does not require any join to retrieve the same data, since the `city_objects` table contains the ground geometries of the objects. As a result the cjdb query is significantly faster than the equivalent of 3DCityDB, as shown in Table 5. We observe similar speed differences with other spatial queries, such as retrieving the building intersecting with a given point in 2D (Q3).

Q4. Number of parts query: We compared how the two databases perform with the retrieval of the number of building parts per building. For simplicity we considered only first level children for each building and we also require the buildings without any parts to be part of the result. Both queries require a single join and their execution times varies depending on the size of the dataset. When the 3DBAG dataset is examined, cjdb seems to be slower than 3DCityDB but for the other datasets, cjdb is faster. However it is worth mentioning that for cjdb the join with the city object table could be skipped and the number of parts could be retrieved with a single aggregation query on the `city_object_relationships` table, but only for the objects which have parts.

Q5. LoD-based query: While one strength of CityGML is that many LoDs of one city object can be stored with the object, exporting a given one to perform an analysis is useful in practice. We therefore tested a query to obtain all building ids with LoD1 (in 3DCityDB) and LoD1.2 (in cjdb) for the 3DBAG dataset and LoD2 for the other 2 datasets. Cjdb performs slower for all datasets; this is probably due to the large size of the 'geometry' column which is in `jsonb` format and therefore requires joining to the TOAST tables.

It should be noticed that if the geometry of the building needs also to be retrieved, the equivalent query for 3DCityDB requires joins which significantly lower the speed.

Q6, Q7 & Q8. INSERT/UPDATE/DELETE attribute queries: We also compared how the databases perform when adding, modifying or deleting an attribute of a building. When it comes to adding new attributes to the database (Q6), cjdb performs considerably slower than 3DCityDB. This can be attributed to the different structure

Table 5 Average execution times for the benchmark queries (all times in ms)

| | 3DBAG | | NYC | | Vienna | |
|----|----------|--------|----------|------|----------|------|
| | 3DCityDB | cjdb | 3DCityDB | cjdb | 3DCityDB | cjdb |
| Q1 | 290 | 285 | – | – | 22 | 2 |
| Q2 | 172 | 1.4 | 59 | 0.3 | 14 | 0.4 |
| Q3 | 1.0 | 0.2 | 0.4 | 0.1 | 0.2 | 0.1 |
| Q4 | 418 | 478 | 240 | 46 | 15 | 2 |
| Q5 | 343 | 1877 | 217 | 392 | 15 | 34 |
| Q6 | 3981 | 11 660 | 1135 | 1425 | 15 | 10 |
| Q7 | 10 796 | 10 903 | 2333 | 1161 | 15 | 9 |
| Q8 | 4393 | 10 984 | 1040 | 682 | 59 | 8 |

of the databases: new attributes in 3DCityDB can simply be inserted as rows in the relevant table, whereas in cjdb the `jsonb` must be modified. However, when it comes to updating existing attributes (Q7), the speed of cjdb is similar for datasets having many attributes, and faster for datasets having few attributes. Deleting attributes (Q8) follows a similar behaviour. Generally, it can be noticed that the number of attributes and the size of the dataset can significantly affect queries on the `jsonb` columns.

5 Conclusions and Future Work

The cjdb project started with the goal of creating a simpler and leaner alternative to 3DCityDB for web servers, allowing users to efficiently store and retrieve 3D city models. Our data model follows the Simple Feature paradigm and has only 3 tables (instead of 66 for 3DCityDB). This is achieved by maintaining the structure of CityJSON and storing JSON directly in the database, using the PostgreSQL type `jsonb`. Because the structure of the data model is close to that of CityJSON files, we can significantly improve the import/export times to/from a database. Furthermore, as we have shown, our simple model is around ten times more compact and it offers retrieval speed comparable to those of 3DCityDB. More specifically we notice that the cjdb performs better when it comes to 2D spatial queries but it performs slower when the `jsonb` columns need to be altered (up to 3 times slower). We also notice that the query speed on the `jsonb` columns is significantly affected by the size of the dataset; the more the attributes in the `jsonb` columns get, more time is required to parse them.

Figure 3 shows the 8 queries, each time have been normalised (we divided the time of cjdb by that of 3DCityDB).

It should be mentioned that the data model of 3DCityDB is more generic and can be implemented with 3 different DBMSs (and not only PostgreSQL). It also allows us to query semantic surfaces, something that is currently not possible with the cjdb model

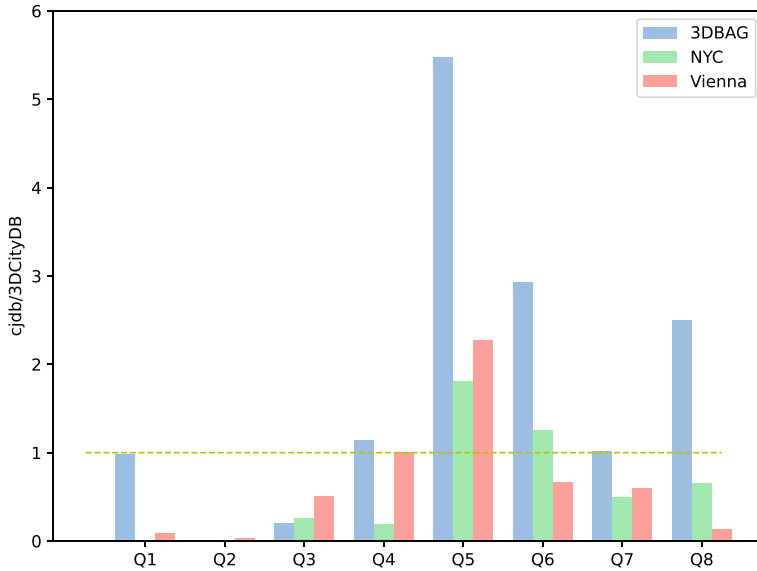


Fig. 3 Data retrieval comparison between cjdb and 3DCityDB for the 8 queries using the 3DBAG dataset. The y-axis corresponds to (cjdb/3DCityDB); a bar lower than the yellow line (located at 1.0) means that cjdb has a faster query time, a bar higher than 1.0 means a slower query time

because it stores the semantic surfaces in `jsonb` format in the ‘geometry’ column of each city object. However, we plan to remedy to the situation by implementing database functions to extract the geometries from semantic surfaces. Furthermore, at the moment, retrieving data from cjdb requires the user to parse the `jsonb` and extract the necessary information, something that is far from being optimal. We plan in the near future to implement some helper plugins in mainstream open-source products (eg QGIS) to be able to view and query cjdb, similar to what is currently being built for 3DCityDB.⁸

While this was not previously discussed, it should be mentioned that CityJSON Extensions are supported by the cjdb data model. This means that extra functions to dynamically extend the data model (as required for 3DCityDB, see Yao and Kolbe (2017)) are not necessary. This is because CityJSON Extensions, unlike CityGML Application Domain Extensions (ADEs), are constrained to follow the structure and rules of other city objects (see Ledoux et al. (2019) for more details).

We also plan to add support for textures and material, at the moment the information is simply stored in the JSON of each geometry, but as is the case for semantic surfaces, we will add database functions to allow users to query and update those.

⁸ <https://github.com/tudelft3d/3DCityDB-Tools-for-QGIS>.

6 The 8 SQL Queries Used for the Benchmark for the 3DBAG Dataset

| | 3DCityDB | cjdb |
|----|--|---|
| Q1 | <pre>SELECT gmlid FROM 3dcitydb.cityobject co JOIN 3dcitydb.cityobject_genericattrib cogaa ON co.id = cogaa.cityobject_id WHERE cogaa.attrname = 'h_dak_max' AND cogaa.realval > 20;</pre> | <pre>SELECT object_id FROM cjdb.city_object WHERE (attributes -> 'h_dak_max') :: float > 20;</pre> |
| Q2 | <pre>SELECT c.gmlid, sg.geometry FROM 3dcitydb.thematic_surface AS ts LEFT JOIN 3dcitydb.surface_geometry AS sg ON ts.lod2_multi_surface_id = sg.parent_id LEFT JOIN 3dcitydb.cityobject AS c ON ts.building_id = c.id WHERE ST_Contains(ST_MakeEnvelope(85400, 446900, 85600, 447100, 7415), sg.geometry) AND ts.objectclass_id = 35;</pre> | <pre>SELECT object_id, ground_geometry FROM cjdb.city_object WHERE TYPE = 'Building' AND ST_Contains(ST_MakeEnvelope(85400, 446900, 85600, 447100, 7415), ground_geometry);</pre> |
| Q3 | <pre>SELECT c.gmlid, sg.geometry FROM 3dcitydb.thematic_surface AS ts LEFT JOIN 3dcitydb.surface_geometry AS sg ON ts.lod2_multi_surface_id = sg.parent_id LEFT JOIN 3dcitydb.cityobject AS c ON ts.building_id = c.id WHERE sg.geometry && st_setsrid(ST_MakePoint(85200, 446900), 7415) AND ts.objectclass_id = 35;</pre> | <pre>SELECT object_id, ground_geometry FROM cjdb.city_object WHERE "type" = 'Building' AND ground_geometry && st_setsrid(ST_MakePoint(85200, 446900), 7415);</pre> |
| Q4 | <pre>SELECT cj.gmlid AS building_id, count(b.id) AS number_of_parts FROM 3dcitydb.cityobject cj LEFT JOIN 3dcitydb.building b ON cj.id = b.building_parent_id WHERE cj.objectclass_id = 26 GROUP BY cj.id;</pre> | <pre>SELECT co.object_id AS building_id, COUNT(cor.child_id) AS number_of_parts FROM cjdb.city_object co LEFT JOIN cjdb.city_object_relationships cor ON co.object_id = cor.parent_id WHERE co.type = 'Building' GROUP BY co.object_id;</pre> |

Q5

```

SELECT                                SELECT
  co.gmlid                             object_id
FROM                                    FROM
  3dcitydb.cityobject co              cjdb.city_object
JOIN 3dcitydb.building b              WHERE
  ON co.id = b.id                      geometry::jsonb @> '{"lod": 1.2}'::jsonb;
WHERE
  b.lod1_solid_id IS NOT NULL;
    
```

Q6

```

INSERT INTO                            UPDATE
  3dcitydb.cityobject_genericattrib   cjdb.city_object
(attrname, datatype, realval, cityobject_id)
SELECT                                SET
  'footprint_area',                  attributes = jsonb_set(
  3,                                  attributes :: jsonb,
  ST_Area(cityobject.envelope),      '{footprint_area}',
  cityobject.id                       to_jsonb(ST_Area(ground_geometry))
FROM                                    ) :: json
  3dcitydb.cityobject                WHERE
WHERE                                  TYPE = 'Building';
  cityobject.objectclass_id = 26;
    
```

Q7

```

UPDATE                                UPDATE
  3dcitydb.cityobject_genericattrib   cjdb.city_object
SET                                    SET
  realval = realval + 10              attributes = jsonb_set(
WHERE                                  attributes :: jsonb,
  attrname = 'footprint_area';        '{footprint_area}',
                                       to_jsonb(
                                       CAST (
                                       jsonb_path_query_first(
                                       attributes :: jsonb,
                                       '$.footprint_area'
                                       ) AS real
                                       ) + 10.0
                                       )
                                       ) :: json
WHERE                                  WHERE
  TYPE = 'Building';
    
```

Q8

```

DELETE FROM                            UPDATE
  3dcitydb.cityobject_genericattrib   cjdb.city_object
WHERE                                    SET
  attrname = 'footprint_area';        attributes = jsonb_set_lax(
                                       attributes :: jsonb,
                                       '{footprint_area}',
                                       NULL,
                                       TRUE,
                                       'delete_key'
                                       ) :: json
WHERE                                  WHERE
  TYPE = 'Building';
    
```

References

Biljecki F, Stoter J, Ledoux H, Zlatanova S, Çöltekin A (2015) Applications of 3D city models: state of the art review. *ISPRS Int J Geo-Inf* 4(4):2220–9964. <https://doi.org/10.3390/ijgi4042842>

Biljecki F, Ledoux H, Stoter J (2016) An improved LOD specification for 3D building models. In: *Computers, environment and Urban systems*, pp 25–37. <https://doi.org/10.1016/j.compenurbsys.2016.04.005>

Borne M, Mercier H, Mora V, Courtin O (2023) SFCGAL. <https://oslandia.gitlab.io/SFCGAL/>, last visited: 01 May 2023

Gröger G, Plümer L (2012) CityGML—interoperable semantic 3D city models. *ISPRS J Photogram Rem Sens* 71:12–33

- IETF (2015) JavaScript Object notation (JSON) text sequences. <https://datatracker.ietf.org/doc/html/rfc7464>
- ISO (2003) ISO 19107:2003: geographic information—spatial schema. International Organization for Standardization
- Kutzner T, Chaturvedi K, Kolbe TH (2020) CityGML 3.0: new functions open up new applications. *Photogram Rem Sens Geoinf Sci* 88(1):43–61
- Ledoux H, Ohori KA, Kumar K, Dukai B, Labetski A, Vitalis S (2019) CityJSON: a compact and easy-to-use encoding of the CityGML data model. *Open Geospatial Data Softw Stand* 4(4). <https://doi.org/10.1186/s40965-019-0064-0>
- Nys GA, Billen R (2021) From consistency to flexibility: a simplified database schema for the management of CityJSON 3D city models. *Trans GIS*. <https://doi.org/10.1111/tgis.12807>
- OGC (2006) OpenGIS implementation specification for geographic information—simple feature access. Open Geospatial Consortium inc., document 06-103r3
- OGC (2007) Geography markup language (GML) encoding standard. Open Geospatial Consortium inc., document 07-036, version 3.2.1
- OGC (2012) OGC city geography markup language (CityGML) encoding standard. Open Geospatial Consortium inc., document 12-019, version 2.0.0
- OGC (2021a) CityJSON community standard 1.0. Open Geospatial Consortium inc., document 20-072r2, version 1.0
- OGC (2021b) OGC city geography markup language (CityGML) part 1: conceptual model standard. Open Geospatial Consortium inc., document 20-010, version 3.0.0, available at <https://docs.ogc.org/is/20-010/20-010.html>
- Pantelios K (2022) Development of a QGIS plugin for the CityGML 3D city database. Master's thesis, MSc thesis in Geomatics, Delft University of Technology
- Peters R, Dukai B, Vitalis S, van Liempt J, Stoter J (2022) Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the Netherlands. *Photogram Eng Rem Sens* 88(3):165–170
- Praschl C, Pointner A (2022) Evaluation of file formats in the context of geo-referenced 3d geometries. GitHub repository at <https://github.com/FHOOEAIST/geofiles>, <https://doi.org/10.5281/zenodo.5376368>
- Ramakrishnan R, Gehrke J (2001) Database management systems. McGraw-Hill Science/Engineering/Math
- Yao Z, Kolbe TH (2017) Dynamically extending spatial databases to support CityGML application domain extensions using graph transformations. In: Kersten TP (ed) *Kulturelles Erbe erfassen und bewahren—Von der Dokumentation zum virtuellen Rundgang*, 37, vol 26. Wissenschaftlich-Technische Jahrestagung der DGPF, DGPF, Würzburg, Germany, pp 316–331
- Yao Z, Nagel C, Kunde F, Hudra G, Willkomm P, Donaubaue A, Adolphi T, Kolbe TH (2018) 3DCityDB—a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. *Open Geospatial Data Softw Stand* 3(2)

Introducing the 3DCityDB-Tools Plug-In for QGIS



Giorgio Agugiaro , Konstantinos Pantelios, Camilo León-Sánchez ,
Zhihang Yao , and Claus Nagel

Abstract This paper introduces a new plug-in for QGIS that allows to connect to the free and open-source 3D City Database to load CityGML data, structured as classic GIS layers, into QGIS. The user is therefore not required to be a CityGML specialist, or a SQL expert, as the plug-in takes care of hiding from the user most of the complexity in terms of underlying data model and database schema implementation. The user can therefore load CityGML thematic “layers” (e.g. for buildings, bridges, vegetation, terrain, etc.), explore their geometries in 2D and 3D and access and edit the associated attributes. At the same time, depending on the user privileges, it is possible to delete features from the database using either normal QGIS editing tools, or a “bulk delete” tool, also included. The plug-in is composed of two parts, a server-side one, which must be installed in the 3D City Database instance, and the client-side one, which runs as a QGIS plug-in in strict sense. A GUI-based tool is also provided for database administrators in order to install/uninstall the database-side part of the plug-in, and manage users and their privileges. All in all, the 3DCityDB-Tools plug-in facilitates the access to CityGML data for GIS practitioners from heterogeneous fields and expertise with the common denominator being the well-known QGIS environment.

This article was selected based on the results of a double-blind review of the full paper

G. Agugiaro (✉) · C. León-Sánchez
3D Geoinformation Group, Faculty of Architecture and the Built Environment, Department of Urbanism, Delft University of Technology, Julianalaan 134, 2628BL Delft, The Netherlands
e-mail: g.agugiaro@tudelft.nl

C. León-Sánchez
e-mail: c.a.leonsanchez@tudelft.nl

K. Pantelios
Coöperatieve Rabobank U.A. RNT DP Consumer Banking, Croeselaan 18, 3521CB Utrecht, The Netherlands

Z. Yao · C. Nagel
Virtualcitysystems GmbH, 7 B/C, 10789 Tauentzienstr, Berlin, Germany
e-mail: zyao@vc.systems

C. Nagel
e-mail: cnagel@vc.systems

Keywords 3D city database · QGIS · CityGML · CityJSON · Plug-in

1 Introduction

Semantic 3D city models are being used more and more in a wide variety of applications (Biljecki et al. 2015) like energy (León-Sánchez et al. 2021; Monteiro et al. 2017; Rossknecht and Airaksinen 2020), flood simulation (Kilsedar et al. 2019), integration with BIM (Kolbe and Donaubaauer 2021; Noardo et al. 2020), micro-climate simulation (Chen et al. 2020), urban planning (Agugiaro et al. 2020), visibility analyses (Virtanen et al. 2021), traffic simulation (Ruhdorfer et al. 2018), etc.

Given the number and heterogeneity of urban objects to represent and the heterogeneity of data to describe them, adoption of standards is beneficial to facilitate usage and exchange of information depending on the application and the involved stakeholders. For this purpose, the Open Geospatial Consortium (OGC) has adopted CityGML (Gröger and Plümer 2012) as an international standard to store and exchange spatial and non-spatial data related to semantic 3D city models. CityGML comes as a UML-based conceptual data model and several encodings: two are file-based and rely on XML or JSON—the latter commonly known as CityJSON (Ledoux et al. 2019)—, while a third one is based on SQL and is called the 3D City Database or, for short, 3DCityDB (3DCityDB, 3D City Database Documentation 2023; Yao et al. 2018). The benefit of using a database encoding is that databases are built to handle and organise large amounts of data, which semantic 3D city models usually consist of.

The 3DCityDB is an open-source project currently developed for PostgreSQL, Oracle and PolarDB/Ganos databases. The database schema implements the CityGML standard with semantically rich and multi-scale urban objects. The 3DCityDB has been used in production and commercial environments for more than a decade already, as well as in academia and in several research projects related to 3D city models. Besides the database schema, the 3DCityDB normally ships with a suite of additional tools, collectively packaged as “3DCityDB Suite” (2023). The suite contains—among the rest—a Java-based Importer/Exporter, which allows to import and export XML-based CityGML and CityJSON files from/to the database, as well as to further export data in KML, COLLADA, and glTF formats for the visualisation for example in Google Earth and CesiumJS.

On the one hand, importing CityGML data into the 3DCityDB has the advantage to avoid direct XML or JSON file parsing by allowing the user to interact directly with data via “standard” tables and SQL commands. On the other hand, the structure of the database schema is rather complex and requires sometimes long SQL queries in order to extract data properly. For example, the current version 4.x of the 3DCityDB consists of a set of 66 tables used mostly to store feature data, but also to handle relations between them. Attributes referring to the same CityObject (e.g. a Building, a Bridge, a Road, etc.) are often stored in multiple linked tables. Additionally, CityGML allows for nested features (e.g. a Room is part of a Building), and

one feature can have multiple representations in terms of geometry: for any given CityObject there are multiple LoDs. In addition, also within the same LoD there can be different possibilities. For example, a building in LoD2 can be represented by means of a solid geometry, a multi-surface geometry, or by means of thematic surfaces (e.g. an aggregation of WallSurfaces, RoofSurfaces, GroundSurfaces, etc.). On top of that, in the same 3DCityDB instance there can co-exist several “copies” of a 3D city model. For the sake of simplicity, we will refer to them as “scenarios” in this paper, but in reality they are stored in different database schemas. The resulting complexity of the 3DCityDB reflects the rich structure of CityGML, but, eventually, the level of SQL knowledge required to interact with it might be indeed beyond that of a common GIS practitioner, de facto limiting access to the data stored therein. The query shown in Fig. 1 provides a simple example. It retrieves all building roofs built since 2015 from the 3DCityDB. As it can be seen, several tables must be joined and geometries must be collected. This implies that a rather advanced knowledge of SQL and, above all, of the 3DCityDB structure is required to write and successfully run such query.

```
1 SELECT
2   ts.id AS roof_id,
3   co_ts.gmlid AS roof_gmlid,
4   b.id AS building_id,
5   co.gmlid AS building_gmlid,
6   b.year_of_construction,
7   ST_Collect(sg.geometry) AS roof_geom
8 FROM
9   citydb.thematic_surface AS ts
10  INNER JOIN citydb.cityobject AS co_ts
11    ON (co_ts.id = ts.id)
12  INNER JOIN citydb.surface_geometry AS sg
13    ON (ts.lod2_multi_surface_id = sg.root_id)
14  INNER JOIN citydb.building AS b
15    ON (b.id = ts.building_id)
16  INNER JOIN citydb.cityobject AS co
17    ON (co.id = b.id)
18 WHERE
19   ts.objectclass_id = 33 AND -- roofsurfaces
20   b.objectclass_id = 26 AND -- buildings
21   b.year_of_construction >= '2015-01-01'::date
22 GROUP BY
23   ts.id,
24   co_ts.gmlid,
25   b.id,
26   co.gmlid,
27   b.year_of_construction
28 ORDER BY
29   b.id,
30   ts.id;
```

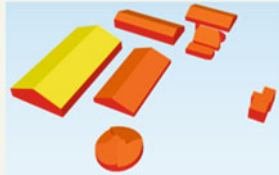


Fig. 1 Example of query to the 3DCityDB to extract the buildings roofs built since 2015. A simple visual example of the result is given in the image in the lower bottom

These limitations appear even stronger if one recalls the fact that QGIS (formerly: Quantum GIS) was originally developed in the early 2000s with the primary goal of accessing data stored in PostgreSQL/PostGIS.

Overcoming the above-mentioned limitations has been one of the main reasons behind the development of the 3DCityDB-Tools plug-in: the purpose is to let the plug-in “hide” and handle the complexity of the 3DCityDB schema(s) in the background, while providing a user-friendly, GUI-based interface directly from within QGIS.

Just two conceptually similar plug-ins were found during the preliminary research work. They were tested and analysed before starting with the development of the 3DCityDB-Tools plug-in, in order to collect information and learn from previous experiences. In general, both are designed to connect to PostgreSQL/PostGIS and both are still rather limited in the type and extents of offered functionalities, not documented or apparently not in active development anymore.

The 3DCityDB Explorer plug-in (2023), for example, is available in GitHub, but last updated in 2021. It allows to load data from the 3D City Database and modify the CityGML generic Attributes of the underlying geometries. Data are loaded into QGIS considering a combination between a maximum number of features (set by the user) and the current extents of the map. However, it only works for “Building” features represented in LoD2 geometries, and no dependent objects (e.g. BuildingInstallation, Room) can be loaded. Additionally, the layers do not contain the attributes. Finally, only one “scenario” can be accessed. Lastly, the plug-in doesn’t seem to account for cases of multiple database users with different privileges.

The 3DCityDB Viewer plug-in 3DCityDB Viewer (2023) is also available in GitHub, although it seems that the project is not maintained anymore, the last commit being in 2021, too. The plug-in allows to load data from a 3DCityDB instance. The user can load “Building” features based on all LoDs, but no children objects. Features are however loaded only in terms of geometry, so no attributes are available at all. As with the “3DCityDB Explorer” mentioned before, only tables from the default citydb schema can be accessed. There is no check on the amount of data to load, which means that the plug-in attempts to import the entire database. This means that, in case of huge amounts of data, QGIS might crash. Finally, similarly to the previous plug-in, it doesn’t seem to support multiple database users with different access privileges.

Although it does not interact with the 3DCityDB, the CityJSON-Loader plug-in (2023); Vitalis et al. 2020) was also evaluated, as it allows to load into QGIS and interact with CityGML data encoded as CityJSON files. The user can select which features to import, and this applies also to the different LoDs. It is important to note that the plug-in loads CityJSON files by converting the data into vector layers stored temporarily in memory. This means that changes happening in the QGIS environment (both in geometries or attributes) are not saved to the original CityJSON file. In order to save changes, users need to export the layers as a new file, however QGIS does not support a CityJSON driver for writing files. Consequently, the plug-in focuses mostly on loading data and does not allow for direct data modifications. Lastly, it is not possible to load data just for a particular area. For big CityJSON files, this could cause performance and stability issues in QGIS.

The results of the preliminary analyses were collected in order to help define the overall goal, the user and software requirements of the “3DCityDB-Tools” plug-in. In the remainder of the paper, the main characteristics and functionalities of the plug-in will be presented and described, followed by a discussion on the current development status, the current limitations and the planned future improvements.

2 User and Software Requirements

From the user point of view, the plug-in should allow to load data from the 3DCityDB as “layers”, ideally following the Simple Features for SQL Model (2023). In simple words, this means that each layer contains a number of objects having each one a single geometry and one tuple of attributes. Probably, the most well-known example is represented by a shapefile. In the case of CityGML, this means that for each class—for example in the case of a Building—there might exist different layers in which the associated attributes are the same, but the geometries vary depending on the chosen LoD. As a matter of fact, the number of possible layers that can be generated following this approach is close to 600, therefore, from the software point of view, this requires some logic to deal with these numbers.

Just to exemplify this concept, a CityGML building object might be equivalent to several layers that follow the Simple Features for SQL Model. If the attributes can be considered to be always the same, this does not apply however to its geometries. A building could be represented by means of its footprint (i.e. as LoD0), or as a prismatic geometry (i.e. as LoD1), or as a geometry that includes the simplified shape of the roof (i.e. as LoD2), or a more detailed one (i.e. LoD3), etc. Additionally, a building might also be decomposed into its component surfaces, e.g. as a set of Ground-, Wall- and RoofSurfaces. Again, each thematic surface can be associated with geometries at different levels of detail. Allowing the user to easily choose which geometry to use and work with (also depending on the available ones in the database), has been one of the main challenges (and goals) of the 3DCityDB-Tools plug-in.

Additionally, a 3D city model can be extremely large and therefore it may be not possible to load it completely in memory without crashing the client. This implies that some logic is also required at software level to limit or control the amount of data that the user can select and load into QGIS. As a result, this has been another goal to pursue when developing the plug-in.

Furthermore, since data are already stored in the database, the user should be allowed to perform typical database operations, i.e. inserts, updates, and deletions. In the case of the 3DCityDB, and depending on the type of privileges granted to the user, it was decided that:

- No insert operations are allowed at all. Nevertheless, inserting new CityGML data can still be carried out as usual and at any time by means of the existing Importer/Exporter tool;

- Attributes can be edited and updates are then stored back to the database, but geometries cannot be modified;
- Features can be deleted.

Finally, unlike other existing plug-ins, the user should be given the possibility to access different citydb schemas (aka “scenarios”).

This set of above-mentioned user requirements has contributed to the definition of the software requirements, as well as the overall structure of the plug-in. First and foremost, the plug-in has been developed based on the 3DCityDB version 4.x for PostgreSQL/PostGIS, but avoiding—as far as possible—technological lock-ins in case it might be extended in future to support Oracle or other databases. The 3DCityDB 4.x supports CityGML v. 1.0 and 2.0, but not the recently released CityGML 3.0 (Kutzner et al. 2020).

Regarding QGIS, the Long Term Release v. 3.22 (first released in autumn 2021) has been chosen, given its maturity—although the plug-in works also on the latest QGIS LTR v. 3.28 (as of summer 2023) without problems. The reason for choosing QGIS as the target front-end of the plugin is due to the fact that it is a well-known and established open-source software, widely used by a heterogeneous and steadily growing community. In addition, since it’s very first releases in the early 2000s it has a native support for PostgreSQL/PostGIS. Furthermore, it has a strong 2D and some 3D visualisation functionalities, although the latter ones are still a bit unstable. Last but not least, it can be extended with Python-based plug-ins, for which several freely available examples and a fairly good documentation already exist.

3 The 3DCityDB-Tools Plug-In

In general, the 3DCityDB-Tools plug-in is composed of two main parts. The first is the server-side one (also called “QGIS Package”) and it is written completely in PL/pgSQL, the procedural language of PostgreSQL. The “QGIS Package” must be installed on top of a 3DCityDB database instance in order to enable the communication and data exchange between the 3DCityDB and QGIS.

The second, client-side part is written in Python 3.9 and uses the QGIS Python-based API in order to communicate with the host application. The Qt library is used for all user interface elements, while the dialogs are designed in Qt Creator (Qt 2023), which is also shipped with QGIS. Except for the server-side part, there are no additional software requirements for the client-side to work, as all aforementioned libraries are provided by the QGIS installation. The overall structure is represented in Fig. 2.

More in detail, the “QGIS Package” is responsible to create and manage the layers, the users and their privileges. In particular, each layer consists in a database view that links all necessary tables containing the feature attributes with a materialized view containing the geometries for the selected LoD. A simplified example is given in Fig. 3, based on the class “Building”. In the 3DCityDB, the attributes are stored in

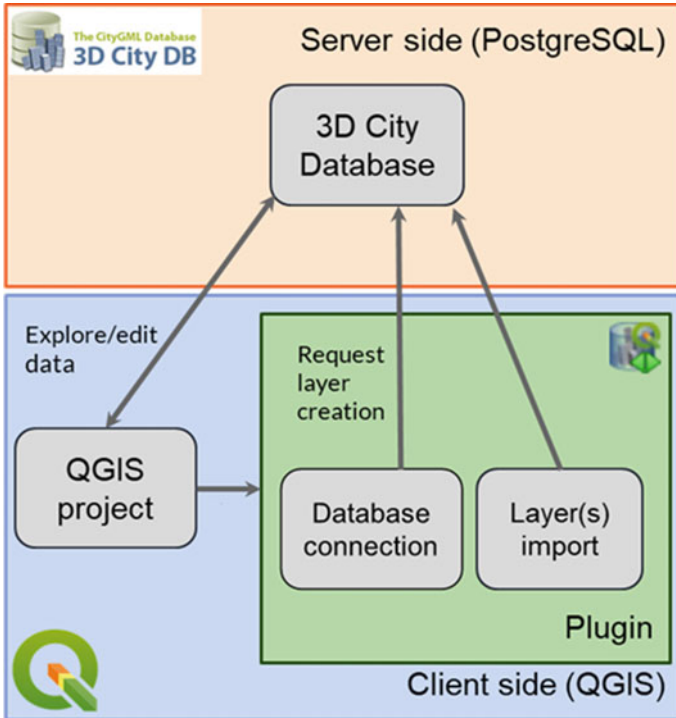


Fig. 2 Overall structure of the 3DCityDB-Tools plug-in, with a server-side part for PostgreSQL and a client-side one for QGIS

tables CITYOBJECT and BUILDING. These two tables are linked together using as primary/foreign key the ID of the “Building” object and then linked to the respective materialized view containing a specific LoD geometry. A set of naming conventions for prefixes and suffixes has been defined and implemented that allows to identify each layer uniquely within the same 3DCityDB instance. At the same time, triggers and trigger functions have been developed to make each view updatable (as far as the attributes are concerned).

The reason for choosing materialized views for the geometries is due to the complexity of how geometries are decomposed and stored in the 3DCityDB. A detailed explanation of how the CityGML geometry model is mapped to the 3DCityDB is beyond the scope of this paper, so the reader is invited to refer to the online documentation for a better understanding. In short, regardless of the nature of the original GML geometry (e.g. solid or multi-surface), all geometries are decomposed and stored as simple 3D polygons, whereas information about their hierarchy and aggregation is also preserved. Preliminary performance tests have shown that querying (and thus aggregating again) the 3D polygons directly from the corresponding 3DCityDB table is rather time-consuming. As a consequence, the materialized views offer instead a good compromise to allow for a better user experience

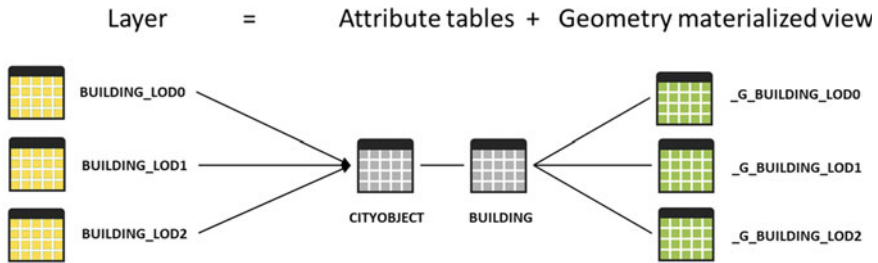


Fig. 3 Simplified representation of how layers are composed for the 3DCityDB-Tools plug-in in the case of Building objects. Tables containing building attributes (here: CITYOBJECT and BUILDING) are linked to the corresponding materialized views with specific LoD geometries

at the cost of some storage space and the time needed to generate/refresh them upon layer creation. In addition, another advantage of relying on materialized views is that also implicit geometries can be created in advance and used as “normal” geometries. In CityGML, implicit geometries can be used to represent features (e.g. a street lamp, a traffic sign, a bus stop, etc.) by means of “template geometries” that must then be instantiated, roto-translated and scaled by means of a 3D affine transformation.

In order to cope with the large number of layers that may result from all possible combinations, a number of checks has been added and implemented. First, layer creation functions are grouped according to the CityGML modules (Building, Bridge, Vegetation, Transportation, Terrain, etc.) and can be invoked individually. Therefore, if a user is only interested in working with building data, only those layers will be created. Second, during the layer creation process, a check is performed to count the number of existing features for that layer. If, for example, there are no data at all regarding Rooms or BuildingInstallations in the database, then those layers will not be generated. The same goes also for LoDs: layers are created only if data (here: geometries) are available in a specific LoD.

Finally, the user can define the size of the area for which the layers will be generated. This means that, especially with very large city models, it is not necessary to generate materialized views of the whole city model, but only within a user-selected, smaller area. This has the benefit of reducing the storage space consumed by the materialized view and, more importantly, the time needed to refresh the materialized views.

The client-side part of the plug-in allows the user to interact with the “QGIS Package” on the server via a set of GUI-based dialogs, and to interact with the data in QGIS itself. Currently, as of version 0.8.2, the client-side part offers three GUI-based tools:

- The “**QGIS Package Administration**” is used to install the server-side part of the plug-in, as well as to set up database user access and user privileges;
- The “**Layer Loader**” allows the user to load and interact with layers in the 3D City Database directly from QGIS

- The “**Bulk Deleter**” can be used to bulk delete features from the database, either all at once, or by means of spatial and feature-related filters.

The “QGIS Package Administration” is meant to be used only by a database administrator. Figures 4, 5 and 6 serve as a visual reference as the following text will reference certain parts of the GUI dialog indicated by letters. Once the connection is set up (a), the database administrator can perform different operations. The first time, the installation of the “QGIS Package” must be carried out (b). For every action, status information is provided in the Connection status box (c). Once the server-side part is installed, the user installation box is activated. It is now possible to choose which database users are allowed to connect to the selected 3DCityDB database instance from the plug-in. This is achieved by adding users to a specific database user group (d). For each group member, the database administrator has to set up the server-side configuration by creating a user schema (e) which will contain the user’s layers and settings. Finally, database privileges (read-only or read and write) (f) can be granted or revoked for each user and for each existing citydb schema (or “scenario”) (g). Once the database administrator has completed the setup, the GUI dialog can be closed. The client-side plug-in can now be used—provided the user is entitled to.

The “Layer Loader” GUI dialog can be loaded by any user. Again, Figs. 7 and 8 will serve as a visual reference. Once the user has connected to a 3DCityDB instance, the list of citydb schemas (or “scenarios”) is shown, and information is provided regarding the access privileges (read-only: “ro”, or read and write: “rw”) (h). Once the user selects the citydb schema to work with, the extents of the whole dataset are displayed in the map canvas. It is possible to set the area extents for which the layers will be generated (i), those are then displayed by means of the blue bounding box. Additionally, the user can choose whether the layers have to be created for all CityGML modules, or only for some of them (j). Afterwards, the next operation is the creation of the layers. Materialized views and the updatable views needed for each layer are set up, and materialized views can be subsequently refreshed. If all requirements are met (as indicated in the Connection status box at the bottom of the dialog), the user can move on to the next tab, where layers will be eventually loaded into QGIS.

For the Layers tab, Fig. 9 will provide visual guidance to the reader when it comes to the principal operation to be carried out. First, the user is given the chance to further reduce the extents of the data to be imported into QGIS (l). These extents are represented by means of a green bounding box. By default, the green bounding box coincides with the blue one defined before, but the user is given the option to change it. Successively, the user filters the layers to be selected by first defining the CityGML module and the LoD (m). These drop down menus are updated depending on the actual data available within the green bounding box. Finally, the filtered list of available layers is generated, from which the user can check which layer(s) will be eventually loaded into QGIS (n).

Once the layers are imported into the QGIS main window, the user can interact with them as “normal” GIS layers and perform the usual set of operations. A hierarchical

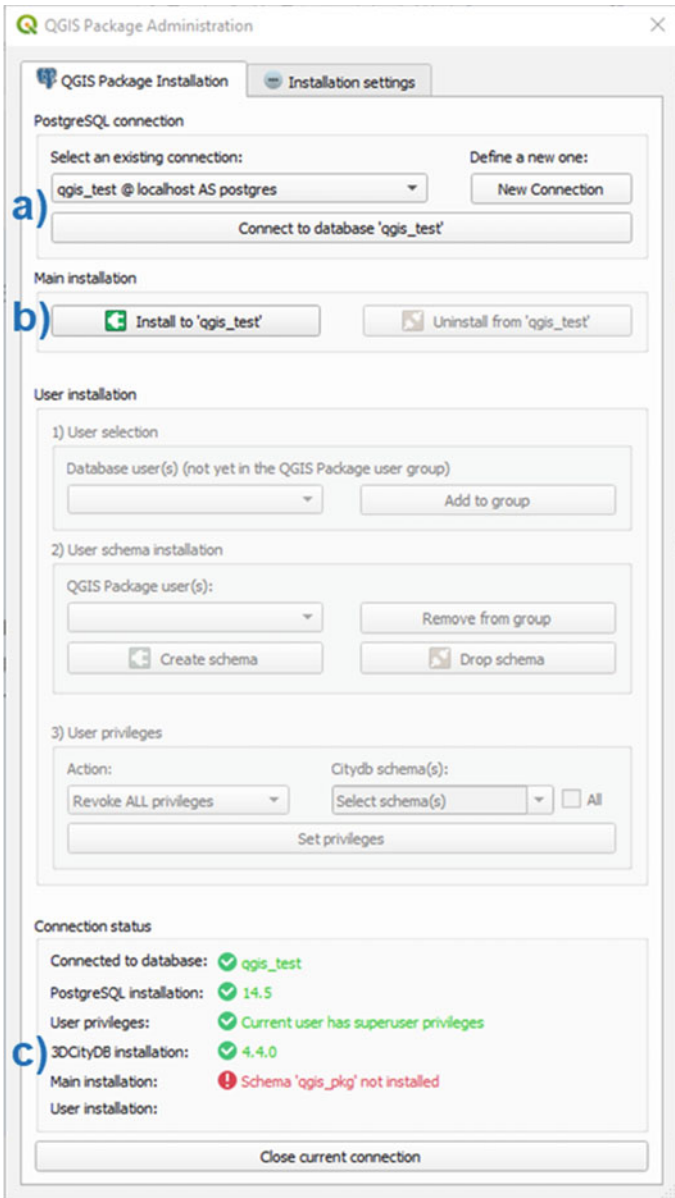


Fig. 4 Overview of the connection and installation operations within the “QGIS Package Administration” GUI dialog in the client-side part of the plug-in

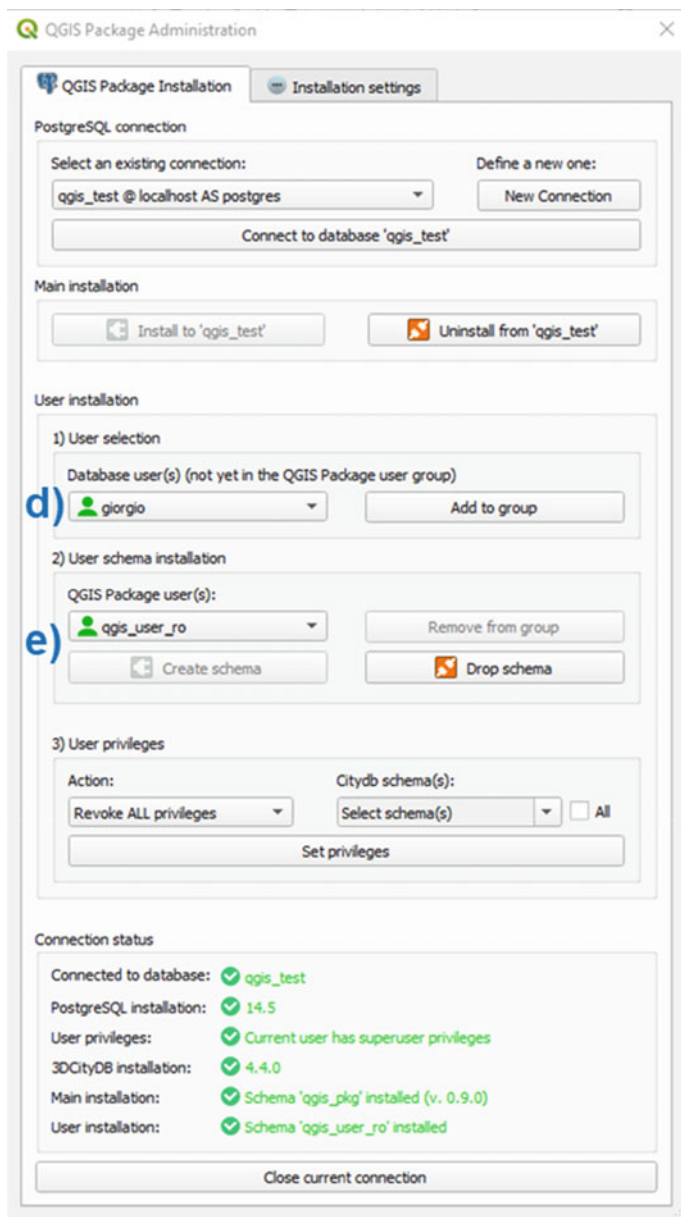


Fig. 5 Overview of the user management operations within the “QGIS Package Administration” GUI dialog in the client-side part of the plug-in

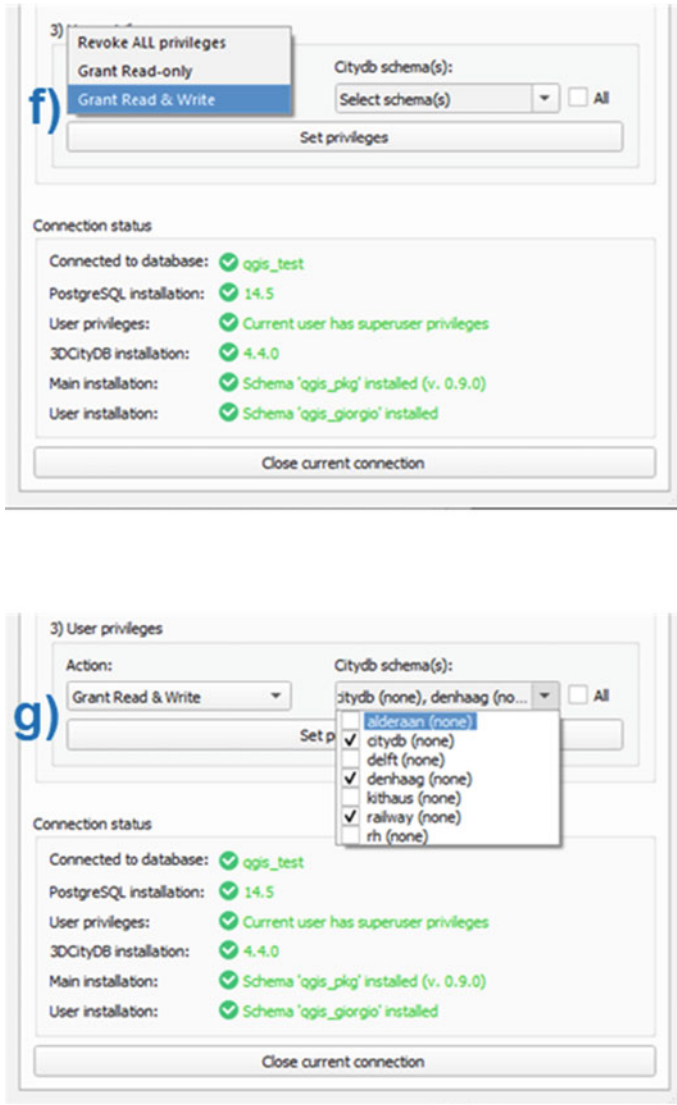


Fig. 6 Overview of the user privileges operations within the “QGIS Package Administration” GUI dialog in the client-side part of the plug-in

Table of Contents (Layers tab) is generated and updated upon each layer import, it offers an overview of the loaded layers ordered according to the CityGML module and LoD. An example is provided in Fig. 10. The user can select features, and access their attributes, either via the table view, or by means of customised attribute forms that present the feature’s attributes, but also contain nested tables related to CityGML generic attributes, addresses and external references. If the user is allowed to edit

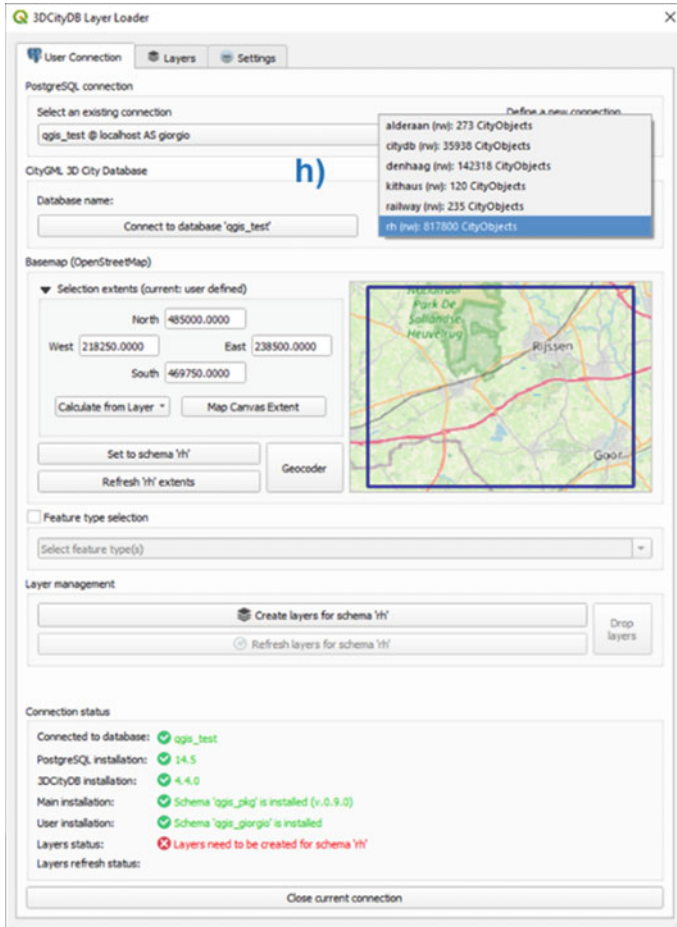


Fig. 7 Overview of the connection operations within the “User connection” tab of the “Layer Loader” GUI dialog in the client-side part of the plug-in

data, the attribute forms perform a series of checks to reduce errors during data entry. For example, enumeration values are converted to drop down menus and, in case of invalid input, the user is informed also visually. For example, in Fig. 11 an error is shown for the number of storeys below ground, as the value must be a positive integer, and in the case of the storey height value, the units of measure are missing. Finally, the user can visualise the layers in 3D using either the QGIS 3D Map or the plug-in Qgis2threejs Exporter (Qgis2threejs 2023), as shown in Fig. 12. It must be said however that 3D visualisation support is still rather unstable in QGIS, especially in the case of the 3D Map which might crash while loading the 3D layers and rendering the scene. The Qgis2threejs Exporter is a bit more stable, however

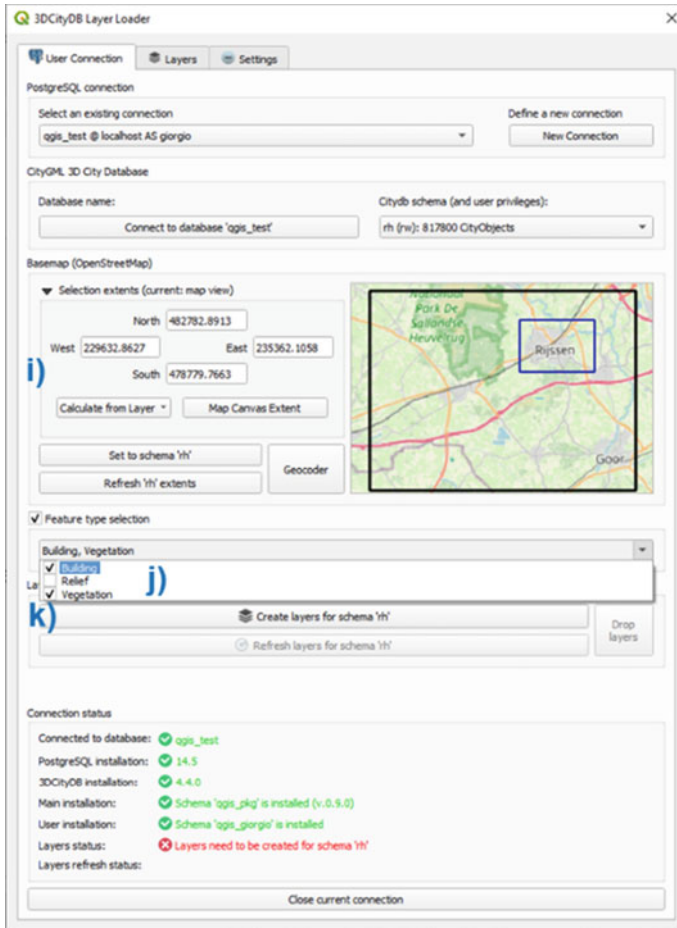


Fig. 8 Overview of the layer generation operations within the “User connection” tab of the “Layer Loader” GUI dialog in the client-side part of the plug-in

artefacts are sometimes still visible with some 3D geometries. These limitations will be discussed later on in the paper.

The third and last available GUI dialog of the 3DCityDB-Tools plug-in is the “Bulk Deleter”. It is meant to be used by a user having at least read-and-write privileges. Figures 13 and 14 provide a visual reference of the following text. Once the user has connected to a 3DCityDB instance, the list of available citydb schemas (or “scenarios”) is shown in a similar way as seen before, however only those citydb schemas are listed for which read-and-write privileges are granted. Once the citydb schema is selected (o), the user can delete features in two ways. The first is to clean up the whole citydb schema (p). In other words all tables are truncated and the schema is completely emptied. Alternatively, the user can perform a selection on the type

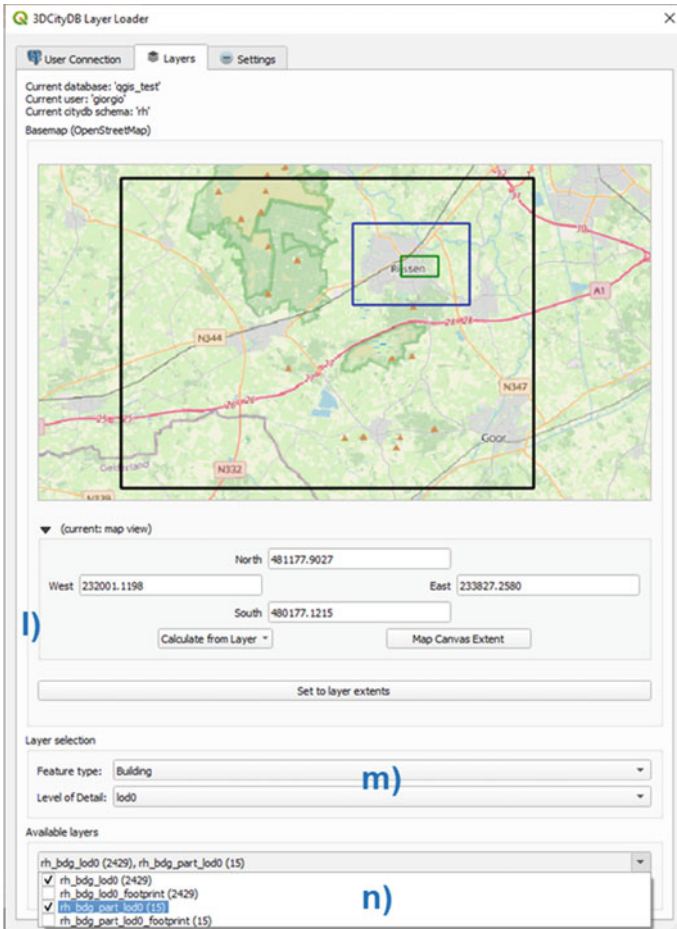


Fig. 9 Overview of the operations within the “Layers” tab of the “Layer Loader” GUI dialog in the client-side part of the plug-in

of features to delete, choosing either from a list of available CityObject modules, or from a list of available CityGML top-class features (q). Additionally, it is possible to set the area extents inside which features will be deleted. Such extents are represented by a red bounding box. The information presented in the drop down menus is updated dynamically.

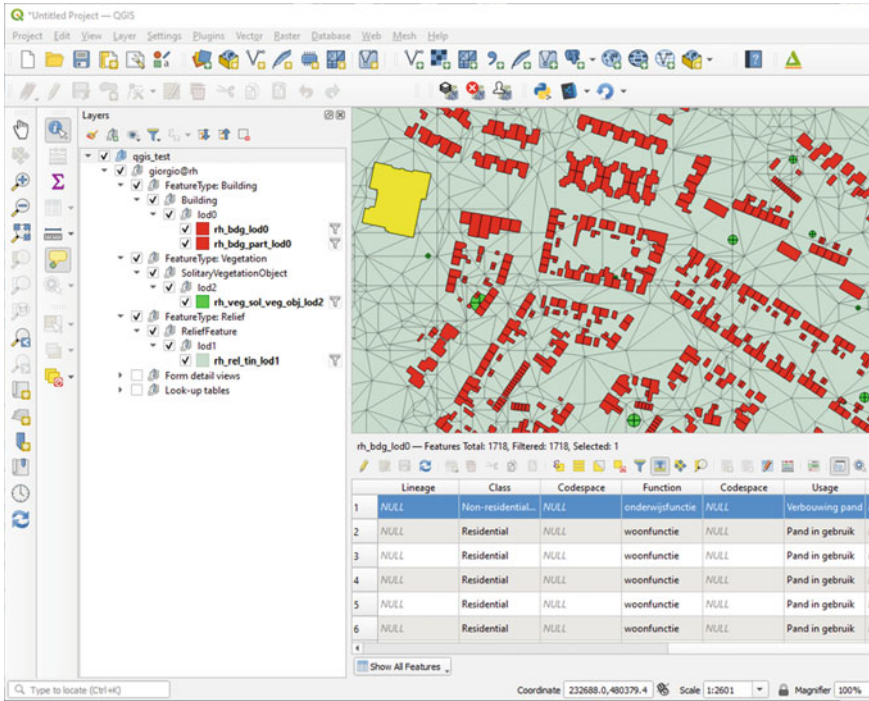


Fig. 10 Example of layers loaded in QGIS. The hierarchical Table of Contents (Layers tab on the left) is generated and updated automatically upon layer import. Attributes can be accessed via the standard attribute table view

4 Tests and Current Limitations

The plug-in has been tested with different datasets varying both in terms of size (geographical extents and number of features) and in terms of available CityGML modules and classes. Table 1 contains an overview of the main characteristics of such test datasets. All datasets have been previously imported into a 3DCityDB instance using the Importer/Exporter. Tests have been conducted to check whether layers can be generated correctly, the attributes can be retrieved, displayed and updated, the geometries can be correctly visualised in 2D and in 3D, the features can be deleted. At the same time, the plug-in was tested on a variety of software configurations (Windows, Linux, MacOS) and different PostgreSQL/PostGIS versions.

Tests conducted so far show that, in general, the back-end part of the plug-in works as expected with all datasets used. Of course, the larger the area is for which the layers are created, the longer the time will be to refresh the materialized views associated to the layers. To test the most time-consuming use case, all layers were generated for all datasets for the whole dataset (e.g. maximum extents). As results show in Table 2, both operations to generate the layers and to refresh the underlying

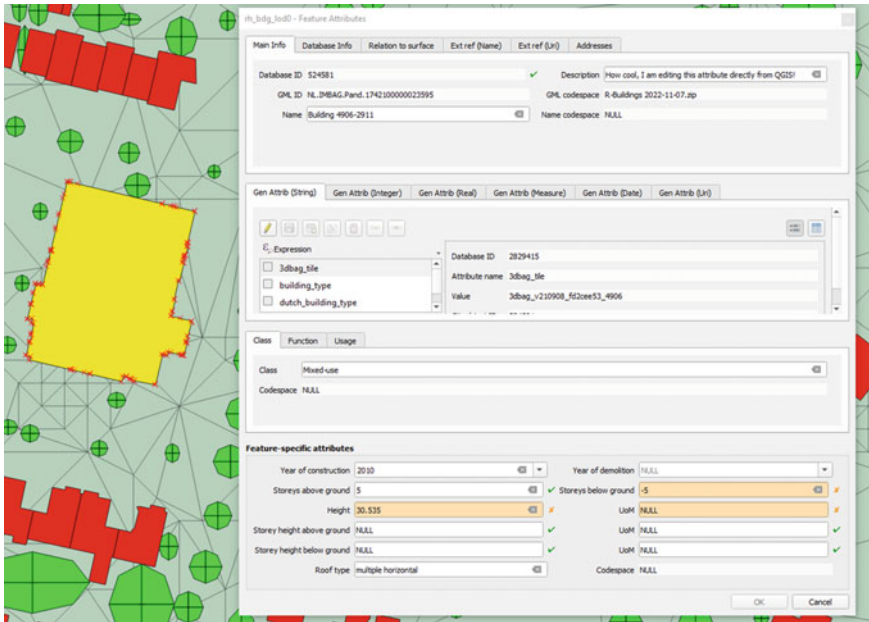


Fig. 11 Example of a customised attribute form used in QGIS to present all attributes of a feature. If the user is allowed to edit the attributes, checks are carried out interactively to avoid errors during data input. The user is notified also visually. External references, addresses and generic attributes are also supported and can be added, deleted or updated

materialized views could always be carried out in acceptable times from the user perspective. Obviously, selecting smaller extents leads to shorter times.

When it comes to the front-end, regardless of the extent of the layers (i.e. the blue bounding box), the ability to further reduce the size of the data to load into QGIS (i.e. the green bounding box) is useful to guarantee a good user experience. Using the front-end part of the plug-in in QGIS, the user can therefore:

- Visualize and interactively define an area of interest for which layers will be created;
- Choose and select for which CityGML module(s) the layers will be created;
- Further select for which features and for which LoD layers can be selected and loaded into QGIS;
- Visualize data in 2D and 3D using QGIS (or the Qgis2threejs plug-in);
- Access feature attributes and eventually edit them, if sufficient privileges are granted.

Usability tests have been carried out with heterogeneous beta testers and the feedback collected has been so far positive, proving that the user interaction with the 3DCityDB has become indeed easier than before. Some feedback suggestions have also been used to improve the design of the GUI dialogs. There are however still

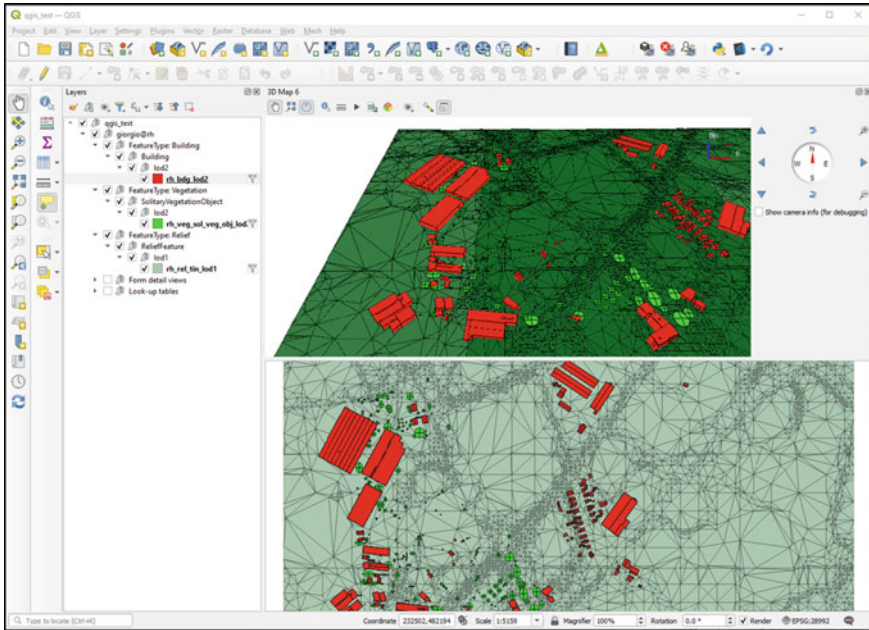


Fig. 12 Example of 2D and 3D visualisation (using QGIS 3D Map View)

some limitations that currently apply to the 3DCityDB-Tools plug-in. For example no raster-based layers from the 3DCityDB are supported. This applies basically to the Relief module, and more specifically to the RasterRelief class. Additionally, the CityGML module CityObjectGroup is not supported by the “Layer Loader”, although it is supported by the “Bulk Deleter”. Currently, no CityGML appearances are supported, i.e. neither colours nor textures can be read from the 3DCityDB and applied to the loaded features. This is however in part due to the limitations imposed by the capabilities of QGIS, which only supports simple or rule-based coloured surfaces for polygon layers. Finally, no CityGML Application Domain Extensions (ADEs) are supported at the moment.

Although it is not a limitation related to the plug-in itself in a strict sense, a major drawback experienced so far by the users is due to some 3D visualisation problems in QGIS which affect the possibility to visualize CityGML data in 3D. Therefore—for the sake of completeness—they will be briefly reported here. As a matter of fact, trying to visualise layers in 3D may lead, from time to time, to visualisation artefacts (e.g. geometries showing spikes) or to crashes of QGIS altogether. This happens when using both the default QGIS 3D Map and—less frequently—the Qgis2threejs plug-in. Some tests were carried out to check the validity of the geometries. As shown for example in Fig. 15, while QGIS shows artefacts (1) (please note the spikes in the WallSurface geometries, shown in the upper part of the image; the roofs are shown only in the lower part), the same (valid) WallSurface geometries can be successfully

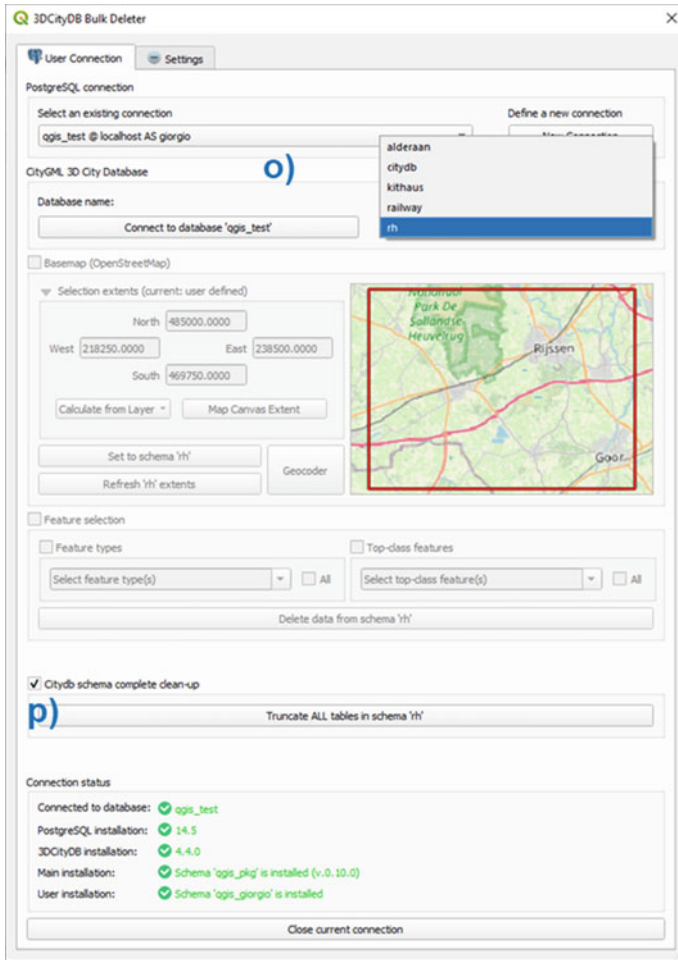


Fig. 13 Overview of the connection and database clean-up operations within the “Bulk Deleter” GUI dialog in the client-side part of the plug-in

rendered in Safe Software’s FME Data Inspector (2) or Google Earth (3). Further tests were conducted exporting the datasets to other formats, such as GeoPackage. In QGIS, the same issues still apply, no matter whether data come from PostGIS or from a GeoPackage file. The reason for such problems is not clear at the moment and needs therefore further investigation, however a first hypothesis is that the reason could be how geometries stored as well-known binary are then converted before triangulation and rendering, with possible loss of precision in the geometry coordinates happening during the process.

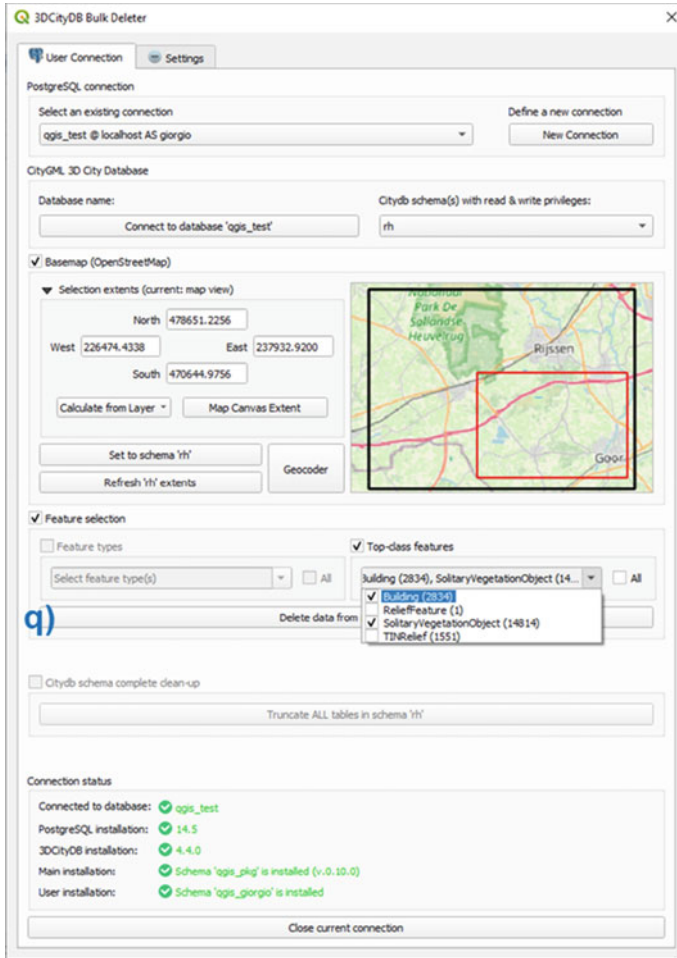


Fig. 14 Overview of the feature selection operations within the “Bulk Deleter” GUI dialog in the client-side part of the plug-in

5 Conclusions and Outlook

This article has presented and described the design and the main functionalities of the 3DCityDB-Tools plug-in for QGIS. The plug-in is already available on GitHub (2023) and on the official QGIS plug-ins repository (2023). The plug-in was conceived and developed to facilitate management and visualisation of data stored in the 3D City Database, which currently supports CityGML v. 1.0 and 2.0. As semantic 3D city models tend to be huge datasets and are generally best managed in spatial databases, the main idea behind the development of this plug-in is to facilitate access and use of CityGML/CityJSON data for those practitioners that lack a

Table 1 List of datasets used to test the “3DCityDB-Tools” plug-in, and their main properties

| Dataset [source] | Number of CityObjects | Database schema size | CityGML modules | LoDs |
|---|-----------------------|----------------------|--|---------------|
| FZK Haus (2023) | 120 | 5 MB | Building | 0, 1, 2, 3, 4 |
| Railway (2023) | 235 | 23 MB | Bridge, building, CityFurniture, generics, relief, transportation, tunnel, vegetation, WaterBody | 3 |
| Delft (2023) | 287,242 | 2.4 GB | Building | 0, 1, 2 |
| Amsterdam (2023) | 319,117 | 4 GB | Bridge, buildings, generics, LandUse, transportation, vegetation, WaterBody | 1 |
| Rijssen-Holten (2023); León-Sánchez et al. (2022) | 827,105 | 1.8 GB | Building, relief, vegetation | 2 |
| Den Haag | 3,143,353 | 4 GB | Building | 1, 2 |
| Vienna (2023); Agugiaro (2016) | 7,512,795 | 23 GB | Building, generics, LandUse, vegetation, relief | 1, 2 |

Table 2 Time required to generate the layers and to refresh them, respectively. Times shown in this table are measured on a Dell Latitude 7490 compute with Core i7-8650U @ 1.90 GHz CPU, 32 GB RAM and 2 TB SSD, and Windows 10 21H2, QGIS v. 3.22 LTR, PostgreSQL v. 14 and PostGIS v. 3.2. Each time value is the average of three consecutive measurements

| Dataset | Layer generation (s.s) | Layer refresh (mm:ss.s) |
|----------------|------------------------|-------------------------|
| FZK Haus | 2.3 | 00:00.4 |
| Railway | 10.4 | 00:01.4 |
| Delft | 6.5 | 01:12.1 |
| Amsterdam | 11.4 | 01:51.4 |
| Rijssen-Holten | 11.3 | 01:15.2 |
| Den Haag | 14.3 | 02:05.9 |
| Vienna | 33.5 | 13:38.7 |

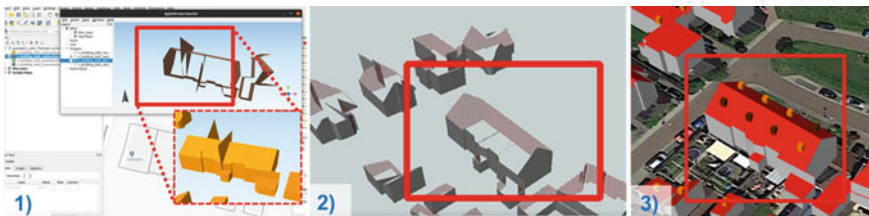


Fig. 15 Examples of artefacts generated during 3D visualisation in QGIS (via Qgis2threejs plug-in). The WallSurfaces of the building highlighted in (1) are shown with and without RoofSurfaces. Some artefacts while rendering the WallSurfaces can be noticed. Such artefacts are however not present when visualising the same WallSurface geometries in (2) (FME Data Inspector) and in (3) (Google Earth, after exporting to KML using the Importer/Exporter)

deep knowledge of the international standard OGC CityGML data model, and/or have limited experience with SQL/Spatial-RDBMSs in general. For this reason, a view/layer-based concept was conceived and implemented in order to bridge common GIS practitioners and complex 3D city models. At the same time, an approach to allow different users to access the 3D city model database was realised.

As a result, the plug-in allows to connect to local or remote instances of 3DCityDB for PostgreSQL/PostGIS and to load data as “classical” layers into QGIS. The user can then interact with them as usual, i.e. perform analyses, work with associated attributes, edit and update them, explore and visualise the data. Additionally, data in the database can be deleted, either using classical QGIS editing tools, or bulk-wise.

From the internal tests carried out so far, and from the feedback received from some early adopters and external testers, the plug-in seems to work well and, most importantly, it can be used not only by experts but also by the “target user” it was originally developed for: GIS practitioners without deep knowledge of CityGML or databases. The complexity of CityGML, and by reflection of the 3DCityDB structure, is sufficiently “hidden” to leverage access to CityGML/CityJSON data to a wider number of users interested in spatial urban analytics.

A simple visual example of the main contribution provided by the 3DCityDB-Tools plug-in is presented in Figs. 16 and 17: the same query shown in Fig. 1 is now run either using the “QGIS Package” (i.e. with circa 50% less SQL code lines), or directly from QGIS using only GUI tools (i.e. without any need to write SQL).

More in general, the added value of choosing QGIS to develop this plug-in is manifold. First and foremost, it is a free and open-source platform which is used by a large and steadily growing amount of heterogeneous users. If, on the one hand and as described in the previous sections, there exist still some issues mainly regarding the 3D visualisation and support for textures, on the other hand QGIS comes with a plethora of processing and data conversion tools “out of the box”—without counting the number of already available plug-ins developed by the community. For example,

```

1 SELECT
2     rs.id AS roof_id,
3     rs.gmlid AS roof_gmlid,
4     rs.building_id AS bdg_id,
5     b.gmlid AS bdg_gmlid,
6     b.year_of_construction,
7     rs.geom AS roof_geom
8 FROM
9     qqgis_user_ro.citydb_bdg_lod2_roofsurf AS rs
10    INNER JOIN qqgis_user_ro.citydb_bdg_lod2 AS b
11              ON b.id = rs.building_id
12 WHERE
13     b.year_of_construction >= '2015-01-01'::date
14 ORDER BY
15     b.id,
16     rs.id;

```




Fig. 16 Using the “QGIS Package”, i.e. the server-side part of the 3DCityDB-Tools plug-in, the same query as in Fig. 1 can now be written in 16 lines (instead of 30)

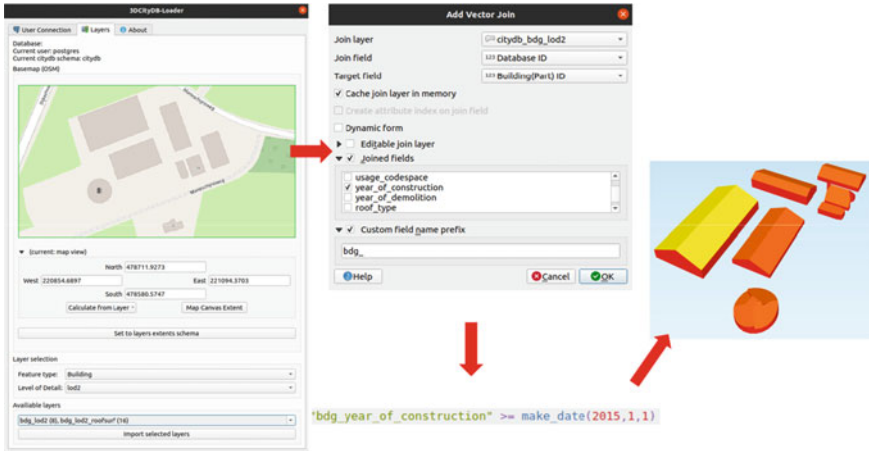


Fig. 17 Using the client-side part of the 3DCityDB-Tools plug-in, the same query as in Fig. 1 can now be constructed and run using the GUI of QGIS without writing any SQL

QGIS already incorporates SAGA GIS (2023) and GRASS GIS (2023) functions and the ability to further transform data formats through GDAL (2023).

Although the current development status of the plug-in already covers most of the originally envisioned characteristics, there are still a number of further improvements that are planned for the future. Besides further testing, we are looking forward to more user feedback so that the overall plug-in structure (functionalities, GUI dialogs, etc.) can be possibly improved. Looking further, inclusion of CityGML Application Domain Extension (ADE) support is planned despite probably being among the major challenging ones, as it will require the revision and extension of the current implementation for both the server-side and the client-side part in order to cope with new ADE classes, data types, and relations. This is however still subject of research at TU Delft with preliminary results expected during the second half of 2023. Finally, the upcoming version 5.0 of the 3DCityDB will support CityGML 3.0. Therefore it will be a valuable addition to investigate—and possibly integrate—support for it also in the “3DCityDB-Tools” plug-in.

Acknowledgements The authors would like to thank María Sánchez Aparicio for the thorough testing of the plug-in and the feedback provided to improve its usability, and Tendai Mbwanda for the contribution to the initial development of the “Bulk Deleter”.

References

3DCityDB, 3D City Database Documentation. Retrieved from <https://3dcitydb-docs.readthedocs.io>. Accessed on 1 Jul 2023

3DCityDB Explorer. Retrieved from <https://github.com/3dcitydb/3dcitydb-qgis-explorer>. Accessed on 1 Jul 2023

- 3DCityDB Suite. Retrieved from <https://github.com/3dcitydb/3dcitydb-suite>. Accessed on 1 Jul 2023
- 3DCityDB-Tools. Retrieved from <https://github.com/tudelft3d/3DCityDB-Tools-for-QGIS>. Accessed on 1 Jul 2023
- 3DCityDB Viewer. Retrieved from <https://github.com/aberhamchristomus/3DCityDB-Viewer>. Accessed on 1 Jul 2023
- Aguiaro G (2016) First steps towards an integrated CityGML-based 3D model of Vienna. In: ISPRS annals of the photogrammetry, remote sensing and spatial information sciences, 2016 XXIII ISPRS congress, Prague, Czech Republic, vol III-4, pp 139–146
- Aguiaro G, González FG, Cavallo R (2020) The city of tomorrow from... the data of today. *ISPRS Int J Geo-Inf* 9(9), 554:42
- Amsterdam Dataset. Retrieved from <https://3d.bk.tudelft.nl/projects/geobim-benchmark/amsterdamgml.html>. Accessed on 1 Jul 2023
- Biljecki F, Stoter J, Ledoux H, Zlatanova S, Çöltekin A (2015) Applications of 3D city models: state of the art review. *ISPRS Int J Geo-inf* 4:2842–2889
- Chen S, Zhang W, Wong NH, Ignatius M (2020) Combining CityGML files and data-driven models for microclimate simulations in a tropical city. *Build Environ* 185:107314. <https://doi.org/10.1016/j.buildenv.2020.107314>
- CityJSON Loader. Retrieved from <https://github.com/cityjson/cityjson-qgis-plugin>. Accessed on 1 Jul 2023
- Delft Dataset. Retrieved from <https://3dbag.nl/en/download>. Accessed on 1 Jul 2023
- Den Haag Dataset. Retrieved from <https://ckan.dataplatform.nl/dataset/3d-stadsmodel-den-haag-2021-citygml>
- FZK Haus Dataset. Retrieved from https://www.citygmlwiki.org/index.php?title=KIT_CityGML_Examples. Accessed on 1 Jul 2023
- GDAL. Retrieved from <https://gdal.org>. Accessed on 1 Jul 2023
- GRASS. Retrieved from <https://grass.osgeo.org>. Accessed on 1 Jul 2023
- Gröger G, Plümer L (2012) CityGML—interoperable semantic 3D city models. *ISPRS J Photogramm Remote Sens* 71:12–33. <https://doi.org/10.1016/j.isprsjprs.2012.04.004>
- Kilsedar CE, Fissore F, Pirotti F, Brovelli MA (2019) Extraction and visualization of 3D building models in urban areas for flood simulation. In: The international archives of the photogrammetry, remote sensing and spatial information sciences, 2019 GEORES 2019—2nd international conference of geomatics and restoration, Milan, Italy, vol XLII-2/W11, pp 669–673. <https://doi.org/10.5194/isprs-archives-XLII-2-W11-669-2019>
- Kolbe TH, Donaubaauer A (2021) Semantic 3D city modeling and BIM. In: Shi W, Goodchild MF, Batty M, Kwan MP, Zhang A (eds) *Urban informatics*. The Urban Book Series. Springer, Singapore, pp 609–639. https://doi.org/10.1007/978-981-15-8983-6_34
- Kutzner T, Chaturvedi K, Kolbe TH (2020) CityGML 3.0: new functions open up new applications. *PFG* 88:43–61. <https://doi.org/10.1007/s41064-020-00095-z>
- Ledoux H, Ohori KA, Kumar K, Dukai B, Labetski A, Vitalis S (2019) CityJSON: a compact and easy-to-use encoding of the CityGML data model. *Open Geospatial Data Softw Stand* 4(4):1–12
- León-Sánchez C, Giannelli D, Aguiaro G, Stoter J (2021) Testing the new 3D bag dataset for energy demand estimation of residential buildings. In: The international archives of the photogrammetry, remote sensing and spatial information sciences, 6th international conference on smart data and smart cities, Stuttgart, Germany, vol XLVI-4/W1-2021, pp 69–76
- León-Sánchez C, Aguiaro G, Stoter J (2022) Creation of a CityGML-based 3D city model testbed for energy-related applications. In: The international archives of the photogrammetry, remote sensing and spatial information sciences, 7th international conference on smart data and smart cities (SDSC), Sydney, Australia, vol XLVIII-4/W5-2022, pp 97–103
- Noardo F, Ellul C, Harrie L, Overland I, Shariat M, Ohori KA, Stoter J (2020) Opportunities and challenges for GeoBIM in Europe: developing a building permits use-case to raise awareness and examine technical interoperability challenges. *J Spat Sci* 65(2):209–233. <https://doi.org/10.1080/14498596.2019.1627253>

- QGIS Plugins. Retrieved from <https://plugins.qgis.org/plugins/citydb-tools/>. Accessed on 1 Jul 2023
- Qgis2threejs. Retrieved from <https://github.com/minorua/Qgis2threejs>. Accessed on 1 Jul 2023
- Qt. Retrieved from <https://www.qt.io/product/development-tools>. Accessed on 1 Jul 2023
- Railway Dataset. Retrieved from <https://nervous-ptolemy-d29bcd.netlify.app/samplefiles/>. Accessed on 1 Jul 2023
- Rijssen-Holten Dataset. Retrieved from <https://github.com/tudelft3d/Testbed4UBEM>. Accessed on 1 Jul 2023
- Rossknecht M, Airaksinen E (2020) Concept and evaluation of heating demand prediction based on 3D city models and the CityGML energy ADE—case study Helsinki. *ISPRS Int J Geo-inf* 9(602):1–19. <https://doi.org/10.3390/ijgi9100602>
- Ruhdorfer R, Willenborg B, Sindram M (2018) Coupling of traffic simulations and semantic 3D city models. *gis.Science*, vol 3
- SAGA. Retrieved from <https://saga-gis.sourceforge.io>. Accessed on 1 Jul 2023
- SFS, Simple Feature for SQL Model. Retrieved from <https://www.ogc.org/standard/sfs/>. Accessed on 1 Jul 2023
- Monteiro CS, Pina A, Cerezo C, Reinhart C, Ferrão P (2017) The use of multi-detail building archetypes in urban energy modelling. *Energy Procedia* 111:817–825. ISSN 1876–6102
- Vienna dataset. Retrieved from <https://www.wien.gv.at/stadtentwicklung/stadtvermessung/geodaten/viewer/geodatendownload.html>. Accessed on 1 Jul 2023
- Virtanen J-P, Jaalama K, Puustinen T, Julin A, Hyyppä J, Hyyppä H (2021) Near real-time semantic view analysis of 3D city models in web browser. *ISPRS Int J Geo-Inf* 10(138):1–23. <https://doi.org/10.3390/ijgi10030138>
- Vitalis S, Ohori KA, Stoter J (2020) CityJSON in QGIS: development of an open-source plugin. *Trans GIS* 24(5):1147–1164. <https://doi.org/10.1111/tgis.12657>
- Yao Z, Nagel C, Kunde F, Hudra G, Willkomm P, Donaubaauer A, Adolphi T, Kolbe TH (2018) 3DCityDB—a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. *Open Geospatial Data Softw Stand* 3(5):1–26. <https://doi.org/10.1186/s40965-018-0046-7>

Challenges and Steps Toward Implementing 3D Cadastral Database—Physical Data Model of LADM



Javad Shahidinejad, Mohsen Kalantari, and Abbas Rajabifard

Abstract The management and storage of 3D cadastral data presents significant challenges due to the complexity of the data and the need for effective storing, manipulating, retrieving, managing, querying, and analysing data. The current version of Land Administration Domain Model (LADM) is limited to a conceptual model and lacks specific guidelines for practical implementation, leading to challenges in converting it into a physical model within production systems. In addition, LADM requires specialised expertise for comprehensive understanding and implementation, hindering its user-friendliness. In addition, the literature in this field paid less attention to the principles and fundamentals of designing a 3D cadastral database. In order to address this knowledge gap, this research aims to investigate design steps for a 3D cadastral database based on LADM as well as challenges associated with the design process. The paper reviews the conceptual model of LADM and investigates converting this conceptual model to a logical model by identifying code lists as multivalued attributes, defining integrity constraints, and considering normalization. Additionally, challenges associated with designing physical models, such as defining different data types are discussed. The current study signified the importance of a well-defined database schema and proposed initial steps to achieve a robust and comprehensive 3D cadastral database. The result is based on an ongoing research project on developing a 3D cadastral database.

Keywords 3D cadastre · LADM · Physical data model · Database management system (DBMS)

This article was selected based on the results of a double-blind review of an extended abstract.

J. Shahidinejad (✉) · A. Rajabifard

The Centre for Spatial Data Infrastructures and Land Administration, Department of Infrastructure Engineering, The University of Melbourne, Parkville, VIC, Australia
e-mail: javad.shahidinejad@student.unimelb.edu.au

M. Kalantari

School of Civil and Environmental Engineering, UNSW, Sydney, Australia

1 Introduction

The Land Administration Domain Model (LADM) as an international standard supports interoperability and provides a formal language for describing current land administration systems (ISO 19152 2012). However, the current version of LADM is limited to a conceptual model and lacks specific guidelines for practical implementation (Rajabifard et al. 2021). It provides a common ontology that can facilitate the creation of a physical data model for 3D cadastre. Nevertheless, converting the conceptual model into a physical model within a production system is challenging. Apart from the need of comprehensive understanding of LADM, system designers must address additional requirements not explicitly covered by the standard, leading to limited implementation of LADM-based country profiles in real production systems (Bar-Maor and Smyth 2022). Additionally, the surveying and representation package within LADM introduces the concept of geometry, but it does not define a standardised format for exchanging geometry data (Oldfield et al. 2017). Consequently, converting data from this standard to another and vice versa may encounter various issues, including information loss such as attributes, textures, and identifications, improper conversions affecting face orientation and data types, violations of topological validity like intersections or gaps, and the risk of losing relationships (Zlatanova et al. 2012).

In addition, it is worth noting that LADM is a complex data model that has been presented through a well-structured framework divided into three comprehensive packages: Party, Administrative, and Spatial Unit, as well as Surveying and Representation as a subpackage of the Spatial Unit package. As a result of its intrinsic complexity, the system appears simple at a higher level, but is complex in the details. Gaining a comprehensive understanding of the classes, attributes, and relationships within LADM requires specialised expertise, along with the ability to effectively align them with the specific requirements of local land administration systems (Bar-Maor and Smyth 2022). Standards should provide clear and unambiguous guidance to ensure consistent interpretation and implementation. Despite efforts made to provide clarity through instance level and different country profiles, the absence of implementation details and comprehensive explanations within the LADM can pose challenges for users in comprehending its intended meaning and ensuring accurate application. This issue is further compounded by the requirement for a significant level of expertise and training to effectively understand and implement the model. Using practical use cases with implementation guidelines is highly required by users as this guideline significantly enhances the comprehension and practical application of standards in real-world scenarios. The LADM lacks such practical guidance, making it difficult for users to bridge the gap between theoretical concepts and implementation steps.

As a broader concept, 3D cadastre involves experts and knowledge regarding legal aspects (how to define and register a 3D parcel), institutional support to establish relationships between parties, and technical aspects to realise it (data acquisition methods, modelling, storage, and visualisation techniques) (Aien et al. 2011; Kalojianni et al. 2020). In addition, a cadastral system must be capable of storing legal

as well as physical data so it should have the ability to store various 3D spatial datasets and relationships including topological and geometrical relationships. The connection between legal and physical data models is crucial for a comprehensive understanding of a land administration system. Physical models play a vital role in defining the semantic and spatial information for various physical objects, including buildings, roads, tunnels, and bridges in an urban environment (Olfat et al. 2021). In the design process of 3D cadastre, a 3D physical model is created based on 2D plans and the shape and texture of buildings. However, the legal spaces defined by legal boundaries are currently accessible in a 2D view, limiting spatial analysis. To achieve a fully functional 3D model, the legal spaces need to be accessible in a 3D environment, with the legal boundaries assigned to physical objects in the model (Barzegar et al. 2020). Therefore, to extend the flexible and modular basis of LADM, which best represents legal boundaries and Rights, Restrictions, and Responsibilities (RRRs), some integrated models using technical standards such as CityGML (Kutzner and Kolbe 2018), Industry Foundation Classes (IFC) standard (ISO 16739-1 2018) for Building Information Modelling (BIM) models, IndoorGML (Lee et al. 2014) for indoor navigation systems, and LandInfra/InfraGML standard (Scarponcini 2016) for modelling and representing land and infrastructure features have emerged to link legal and physical aspects and represent the full 3D cadastre. However, most of the proposed integrated data models are designed at the conceptual level without developing a physical data model. As a result, there is a significant knowledge gap in this area, which must be addressed in order to achieve a fully integrated 3D data model (Sürmeneli et al. 2022a).

A considerable number of challenges encountered in the implementation of a conceptual model, particularly in the context of 3D cadastre, can be overcome by converting it into a physical model. Despite this, there has been limited research on the feasibility of using 3D cadastral databases to store and manage cadastral data (Olfat et al. 2021). Unlike the conceptual model, which can have different definitions depending on various perspectives, a physical model provides a unique and meaningful representation that is less prone to confusion (Elmasri and Navathe 2016). Moreover, the presence of a well-defined database schema would facilitate the implementation process and enable the automation of the 3D cadastre data lifecycle (Olfat et al. 2021). Thus, the purpose of this research is to emphasise the challenges involved and outline the essential steps for designing a 3D cadastre database schema at the physical model level, addressing the need for a more comprehensive and efficient implementation of LADM.

The next Sect. 2 reviews the current approaches for designing a 3D cadastral database. Then, Sect. 3 explains the steps for designing a 3D cadastral database based on the principles of database design and investigates the challenges in each step. Finally, the last Sect. 4 of this paper discusses research gaps and provides some directions for future studies.

2 Literature Review

The field of 3D cadastral database has been relatively underexplored, with a few studies primarily focusing on the basic aspect of 3D data storage within databases. However, the studies ignored explaining the principles and foundational steps involved in designing a database for 3D cadastre. Furthermore, the papers mainly focused on data storage in databases rather than specifically addressing the design of a database for 3D cadastre purposes. The database has been considered as a component of the overall work, utilised to store data, or integrated during the project's workflow. For example, (Sürmeneli et al. 2022b) used PostgreSQL database to store 4D (3D + t) cadastral data and executed temporal querying. (Cemellini et al. 2020) developed a web-based prototype that combined 2D and 3D cadastre with a time component. Their primary focus was on visualisation, however, they designed a database schema by mapping Queensland Digital Cadastral Database (DCDB) to LADM. Some papers worked on the lifecycle of 3D cadastral data and utilised database as part of their prototype or framework (Kalogianni et al. 2020; Olfat et al. 2021). Barzegar et al. (2021) developed a spatial query technique to identify different types of property boundaries such as exterior, median, and interior boundaries. (Tekavec and Lisec 2020a, 2020b) used PostgreSQL/PostGIS and simple features computational geometry algorithms (SFCGAL) extension for 3D indoor modelling. Moreover, FME as an extract, transform, and load (ETL) software was employed for data transformation. There are also some attempts of using database in 3D underground data (Janečka and Bobíková 2018), 3D indoor modelling (Tekavec and Lisec 2020a), 3D crowdsourced cadastre (Gkeli et al. 2020), registration strata objects in database (Nasorudin et al. 2016) and 3D hybrid cadastral model (Spirou-Sioula et al. 2013). In addition, Koeva et al. (2019) investigated the change detection based on point clouds in support of 3D indoor cadastre. They linked the result of change detection to a 3D cadastre database. In addition, several papers explored related fields such as BIM and the use of IFC standard (Barzegar et al. 2020, 2021; Olfat et al. 2021; Solihin et al. 2017), as well as Digital Twin (Sammartano et al. 2021). Some researchers have also utilised CityGML and 3D City Database (3DCityDB) (Guntel and Aydinoglu 2021; Halim et al. 2021; Li et al. 2019; Liamis and Mimis 2022). Although the majority of these papers employed relational databases, a few have used NoSQL databases (da Purificação et al. 2022; Višnjevac et al. 2017, 2019).

A few studies have discussed all three main stages of database design for 3D cadastre. For instance, (Aien 2013) discussed the conceptual, logical, and physical data models and developed an XML schema for 3D cadastre. (Barzegar et al. 2020, 2021) designed a 3D database schema and proposed a methodology for transforming BIM data into the designed schema which consider legal and physical aspects of 3D cadastre (Halim et al. 2021; Rajabifard et al. 2019, 2021) proposed a methodology for storing 3D data in National Digital Cadastral Database (3D-NDCDB). To ensure database normalisation and minimisation of data redundancy, they merged some entities within the LADM standard into one class in PostgreSQL. Additionally, "look-up" tables were created to facilitate the management of code lists associated with

specific attributes in LADM. Normalisation was also discussed by (Sürmeneli et al. 2022a, 2022b), but the specific tables and extent of normalisation were not clearly specified.

In summary, most studies have not provided significant attention to the fundamental design steps of a 3D cadastral database. Instead, they have mostly focused on data storing and querying through databases. Accordingly, a comprehensive study is needed to determine the principles and steps involved in the design of 3D cadastral databases.

3 Steps and Challenges

The efficient data management typically requires the use of a database (Coronel and Morris 2017). A database is a collection of data and metadata that is organised and stored in a way that makes it easy to share, access and manipulate. A database is managed by a software system known as Database Management System (DBMS), enabling users to perform tasks such as data creation, updates, deletions, and queries using query Languages. The DBMS simplifies these tasks by storing data in a structured way, allowing for robust and efficient data management (Ramakrishnan and Gehrke 2003). Additionally, the DBMS is a collection of programs that control accesses to the data, and manage the database structure. The database development lifecycle, as depicted in Fig. 1, represents the step-by-step process of creating and managing a database system. This lifecycle typically involves several stages including initial study, database design, implementation and loading, testing and evaluation, operation, and maintenance and evolution (Coronel and Morris 2017).

In the initial step of database development for 3D cadastre, the operational components, requirements, problems, objectives, scope of the design, and the data to be stored and managed are defined. The database design phase consists of three fundamental steps: conceptual, logical, and physical design. In this research, our focus is on adopting LADM conceptual model into the fundamental principles of database design phase.

Conceptual data models represent the highest level of abstraction in data modelling, offering concepts that align with how users perceive data. High-level data models provide an overview of core concepts with entities and relationships, encompassing operational processes, events, and in the case of object-oriented designs, the associated methods that can be invoked. Logical data modelling contains further details about entities and relationships. Its main goal is to facilitate information exchange within a specific context, prioritising relevant information while omitting unnecessary complexity. Physical data modelling contains the necessary level of detail to implement entities into a real database using Data Definition Language (DDL) (Coronel and Morris 2017). The following outlines the sequential process for designing the 3D cadastral database (Alattas et al. 2018; Rajabifard et al. 2019; Shojaei et al. 2023; Sürmeneli et al. 2022a):

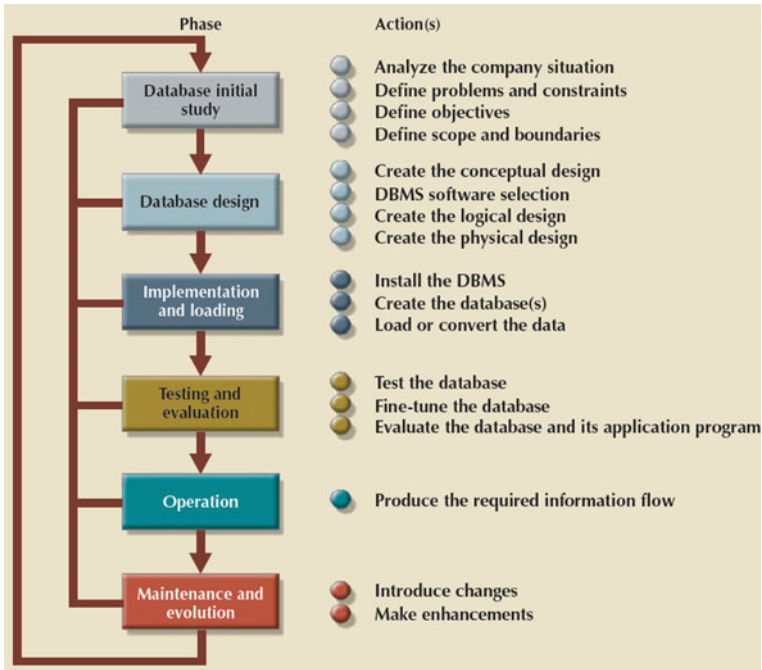


Fig. 1 Database lifecycle (Coronel and Morris 2017)

Conceptual Design Phase Since LADM is a conceptual data model, we need to modify the design to follow the conceptual design phase in the database to implement this standard in a database, similar to designing Entity-Relationship (ER) models. The process of designing conceptual model of LADM involves several essential steps. Firstly, it requires the identification of entities, represented as classes in LADM. Secondly, relationships between these entities must be established to capture the associations and dependencies within the system. Subsequently, constraints such as connectivity, participation, and cardinality need to be applied to these relationships to ensure data integrity and consistency. Additionally, attributes describing the entities are incorporated to provide more detailed information.

Logical Design Phase There are some technical challenges while converting LADM into a logical model. In order to maintain data integrity within the LADM, it is crucial to incorporate primary keys (PK) and foreign keys (FK) in the database design. This ensures referential integrity, where data consistency is preserved through proper associations between related entities. In instances where the creation of a new table is necessary, specifying PKs is an important step to identify unique records within the table. The selection of these unique keys may vary based on the jurisdictional requirements. By considering these principles, the integrity of data storage and efficiency of the data retrieval and insertion can be preserved in the LADM. In addition, LADM incorporates certain code lists that can be considered as multivalued attributes in

the context of database design. Attributes in database categorised into single-valued and multivalued. A multivalued attribute is an attribute that can have many values for a single entity occurrence. It is essential to unpack these multivalued attributes, while transitioning to the logical design phase. To design a database, one solution is to add new tables for multivalued attributes, resulting new tables for different code lists based on LADM. As an example, Fig. 2 illustrates the content of Spatial Unit Package, showing its associations with other fundamental classes and the inclusion of code lists. The LADM consists of 24 code list tables in total including three tables of code lists in the Party package, seven in the Administrative package, ten in the Spatial Unit package, and four in the Surveying and Representation subpackage. Creating a new table for these code lists increases the design complexity and makes query execution times longer because it requires joining multiple tables in the query. For instance, Fig. 3 illustrates the new table for the LA_VolumeType code list. Furthermore, the process of normalising the data is essential to address different anomalies that may arise from denormalised data. Moreover, there are some functions in LADM, such as `ComputeVolume()`, that must be checked whether they should be recorded in the database or not. For example, in some cases, these functions or attributes are considered as derived attributes that do not require to be stored in a separated column.

Physical Design Phase After establishing a logical model, the next step is to implement the model in a database. One of the challenges in this phase is related to data types. As shown in Fig. 2, the data type of volume is defined as `LA_VolumeValue`, including two attributes: `volumeSize` and `type`. However, this specific data type is not available in the databases. There are various solutions to address this issue, such as merging the two attributes into a single attribute and considering them as text or string values. Alternatively, creating a new table for these data types is another option which is demonstrated in Fig. 3. The first solution is not recommended as it causes inconsistency in the database. On the other hand, creating a new table would increase number of tables, leading to more complex queries. Such queries require JOIN operations, which could potentially decrease the performance due to the complexity of query analysis. Furthermore, it is necessary to explore some data types from other standards. Notably, `CI_ResponsibleParty` class of LADM uses ISO 19115, Geographic information–Metadata (ISO 19115–1 2014) to establish data quality element. Furthermore, ISO 19156, Geographic information–Observations and measurements (ISO 19156 2011) is used within LADM as a guidance for the determination of survey elements, process, and observations entities. The spatial feature classes in LADM such as `BoundaryFace` and `BoundaryFaceString` use the ISO 19107, Geographic Information–Spatial schema (ISO 19107 2003) to define geometry representations such as `GM_Point`, `GM_Multicurve`, and `GM_Multisurface`. This type of linking one standard to another standard can lead some challenges. First of all, it needs understanding of other standards. In the other hand, it enables the use of international language and vocabulary which increase data interoperability and easier data sharing. It should be noted that the issues discussed for data types in LADM may exist for data types of new standards used in LADM. In addition, it is necessary to model 3D spatial data and volume into the database. However, most databases do not support 3D spatial

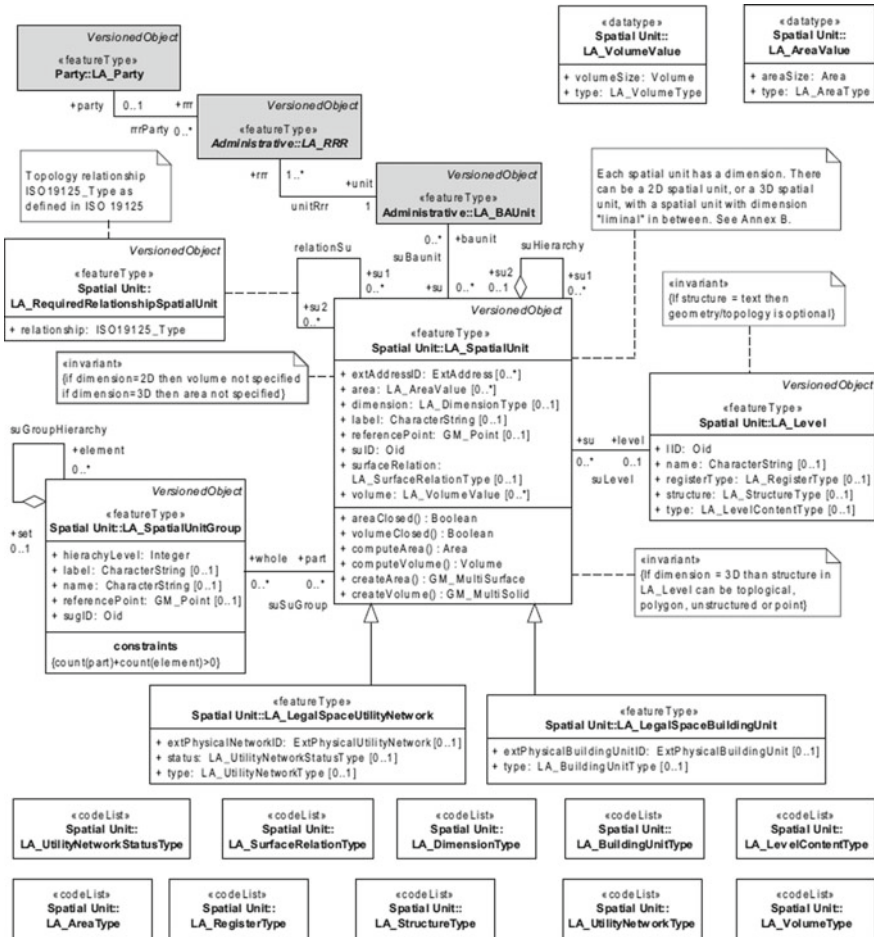


Fig. 2 Content of spatial unit package with associations to other basic classes (ISO 19152 2012)

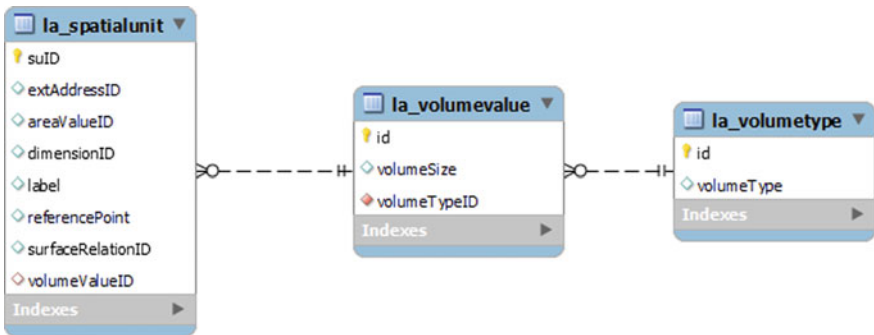


Fig. 3 New tables created for LA_VolumeValue data type and LA_VolumeType code list

data types. In this regard, there is a need to extend spatial data types in DBMSs to support 3D volumetric data types (Stoter and van Oosterom 2002).

4 Conclusion

In recent years, the need for an effective data management system has become increasingly crucial due to rapid technological advancements, leading to a significant increase in data volume, diversity, and complexity. Despite the importance of a 3D cadastral database management system, some jurisdictions still rely on using file formats or 2D databases to store and manage their cadastral data. This research is part of an ongoing project focused on developing 3D cadastral databases. The paper addressed the challenges associated with managing and storing 3D cadastral data and proposes a step-by-step approach to design a robust database schema at the physical model level. A key issue highlighted is the lack of practical implementation guidelines for LADM. Furthermore, the existing research on 3D cadastral database design is limited and primarily concentrates on data storage rather than the crucial aspect of database design. To bridge this gap, the paper emphasised the importance of addressing the database design phases including conceptual, logical, and physical design levels. Various challenges were discussed, such as handling multivalued attributes, implementing normalisation to ensure data consistency, and establishing data integrity. Additionally, the research explored the challenges of using data types from other standards. By addressing these challenges in the proposed steps, a comprehensive and efficient 3D cadastral database schema can be developed, enabling more effective land administration and data management. For future studies, it is required to explore further how LADM can be made more user-friendly and accessible to users without specialised expertise. Additionally, it would be investigated how practical guidance in the form of examples, use cases, and implementation guidelines can be incorporated into LADM to facilitate its understanding and application in real-world scenarios.

References

- Aien A, Rajabifard A, Kalantari M, Williamson I (2011) Aspects of 3D cadastre—a case study in Victoria. In: FIG working week 2011, pp 1–15
- Aien A (2013) 3D cadastral data modelling. PhD thesis, The University of Melbourne, Victoria, Australia
- Alattas A, van Oosterom P, Zlatanova S, Diakit  A, Yan J (2018) Developing a database for the LADM-IndoorGML model. In: 6th international FIG 3D cadastre workshop, Delft, The Netherlands, pp 261–278
- Bar-Maor A, Smyth K (2022) ArcGIS and LADM (ISO 19152): from conceptual to implementation. In: FIG Congress 2022, volunteering for the future—geospatial excellence for a better living, Warsaw, Poland, pp 1–15

- Barzegar M, Rajabifard A, Kalantari M, Atazadeh B (2020) 3D BIM-enabled spatial query for retrieving property boundaries: a case study in Victoria, Australia. *Int J Geogr Inf Sci* 34(2):251–271
- Barzegar M, Rajabifard A, Kalantari M, Atazadeh B (2021) Identification of property boundaries using an IFC-based cadastral database. *Land* 10(300):1–36
- Cemellini B, van Oosterom P, Thompson R, de Vries M (2020) Design, development and usability testing of an LADM compliant 3D cadastral prototype system. *Land Use Policy* 98:104418
- Coronel C, Morris S (2017) *Database systems: design, implementation and management*, 13th edn. Cengage Learning
- da Purificação NRS, Henrique VB, Amorim A, Carneiro A, de Souza GHB (2022) Reconstruction and storage of a low-cost three-dimensional model for a cadastre of historical and artistic heritage. *Int J Build Pathol Adapt*
- Elmasri R, Navathe SB (2016) *Fundamentals of database systems*. Pearson Education
- Gkeli M, Potsiou C, Ioannidis C (2020) A technical solution for 3D crowdsourced cadastral surveys. *Land Use Policy* 98:104419
- Guntel A, Aydinoglu AC (2021) Producing and visualizing 3D building geodatabase as a part of 3D cadastre project. In: *The international archives of the photogrammetry, remote sensing and spatial information sciences, the 6th international conference on smart city applications, Karabuk University, Virtual Safranbolu, Turkey, vol XLVI-4/W5-2021*, pp 239–243
- Halim NZA, Lim CK, Karim H (2021) Developing a 3D city model database beyond cadastral purposes. In: *IOP conference series: earth and environmental science*, vol 767. IOP Publishing, pp 1–8
- ISO 19107 (2003) *Geographic information–spatial schema*
- ISO 19156 (2011) *Geographic information–observations and measurements*
- ISO 19152 (2012) *Geographic information–land administration domain model (LADM)*
- ISO 19115–1 (2014) *Geographic information–metadata–part 1: fundamentals*
- ISO 16739–1 (2018) *Industry foundation classes (IFC) for data sharing in the construction and facility management industries—part 1: data schema*
- Janečka K, Bobíková D (2018) Registering the underground objects in the 3D cadastre: a case study of wine cellar located in the vineyard area Tokaj. *Acta Montanist Slovaca* 23(3):260–270
- Kalogianni E, van Oosterom P, Dimopoulou E, Lemmen C (2020) 3D land administration: a review and a future vision in the context of the spatial development lifecycle. *ISPRS Int J Geo Inf* 9(107):1–25
- Koeva M, Nikoohemat S, Elberink SO, Morales J, Lemmen C, Zevenbergen J (2019) Towards 3D indoor cadastre based on change detection from point clouds. *Remote Sens* 11(1972):1–21
- Kutzner T, Kolbe TH (2018) CityGML 3.0: sneak preview. In: *PFGK18-photogrammetrie-fernerkundung-geoinformatik-kartographie*, 37, vol 27. Jahrestagung, München, pp 835–839
- Lee J, Becker T, Nagel C, Kolbe TH, Sisi Z, Li K-J (2014) OGC® IndoorGML, version 1.0
- Li W, Zlatanova S, Yan JJ, Diakite A, Aleksandrov M (2019) A geo-database solution for the management and analysis of building model with multi-source data fusion. In: *The international archives of the photogrammetry, remote sensing and spatial information sciences, ISPRS and GEO workshop on geospatially-enabled SDGs monitoring for the 2030 Agenda, Changsha, China, vol XLII-4/W20*, pp 55–63
- Liamis T, Mimis A (2022) Establishing semantic 3D city models by GRextADE: the case of the greece. *J Geovis Spat Anal* 6(15):1–18
- Nasorudin NN, Hassan MI, Zulkifli NA, Rahman AA (2016) Geospatial database for strata objects based on land administration domain model (LADM). In: *The international archives of the photogrammetry, remote sensing and spatial information sciences, 2016 international conference on geomatic and geospatial technology (GGT), Kuala Lumpur, Malaysia, vol XLII-4/W1*, pp 329–334
- Oldfield J, Van Oosterom P, Beetz J, Krijnen TF (2017) Working with open BIM standards to source legal spaces for a 3D cadastre. *ISPRS Int J Geo Inf* 6(351):1–19

- Olfat H, Atazadeh B, Badiie F, Chen Y, Shojaei D, Rajabifard A (2021) A proposal for streamlining 3D digital cadastral data lifecycle. *Land* 10(642):1–16
- Rajabifard A, Atazadeh B, Yip KM, Kalantari M, Anaraki MR, Olfat H, Badiie F, Shojaei D, Lim CK, Zain MAM (2019) Design and implementation of a 3D national digital cadastral database based on land administration domain model: lessons learned from a 3D cadastral project in Malaysia. In: 8th land administration domain model workshop (LADM2019)
- Rajabifard A, Atazadeh B, Kalantari M, Olfat H, Shojaei D, Badiie F (2021) Design and development of an LADM-driven 3D land administration system: lessons learned in Malaysia. *Land Use Policy* 102(105252):1–12
- Ramakrishnan R, Gehrke J (2003) *Database management systems*, 3rd edn. McGraw-Hill, New York
- Sammartano G, Avena M, Cappellazzo M, Spanò A (2021) Hybrid GIS-BIM approach for the Torino digital-twin: the implementation of a floor-level 3D city geodatabase. In: *The international archives of the photogrammetry, remote sensing and spatial information sciences, XXIV ISPRS Congress*, vol XLIII-B4-2021, pp 423–430
- Scarponcini P (2016) OGC® Land and infrastructure conceptual model standard (LandInfra), version 1.0
- Shojaei D, Badiie F, Olfat H, Rajabifard A, Atazadeh B (2023) Requirements of a data storage infrastructure for effective land administration systems: case study of Victoria, Australia. *J Spat Sci* 68(3):431–449
- Solihin W, Eastman C, Lee Y-C (2017) Multiple representation approach to achieve high-performance spatial queries of 3D BIM data using a relational database. *Autom Constr* 81:369–388
- Spirou-Sioula K, Ioannidis C, Potsiou C (2013) Technical aspects for 3D hybrid cadastral model. *Surv Rev* 45(333):419–427
- Stoter J, van Oosterom P (2002) Incorporating 3D geo-objects into a 2D geo-DBMS. In: *ACSM-ASPRS 2002 annual conference proceedings*, pp 1–12
- Sürmeneli HG, Koeva M, Alkan M (2022a) The application domain extension (ADE) 4D cadastral data model and its application in Turkey. *Land* 11(634):1–16
- Sürmeneli HG, Alkan M, Koeva M (2022b) Design and implementation of a 4D cadastral legal model for Turkish land administration infrastructure based on LADM. *Geocarto Int* 37(26):12096–12118
- Tekavec J, Lisec A (2020a) Cadastral data as a source for 3D indoor modelling. *Land Use Policy* 98:104322
- Tekavec J, Lisec A (2020b) 3D geometry-based indoor network extraction for navigation applications using SFCGAL. *ISPRS Int J Geo-inf* 9(417):1–15
- Višnjevac N, Mihajlović R, Šoškić M, Cvijetinović Ž, Bajat B (2017) Using NoSQL databases in the 3D cadastre domain. *Geodetski Vestnik* 61(3):412–426
- Višnjevac N, Mihajlović R, Šoškić M, Cvijetinović Ž, Bajat B (2019) Prototype of the 3D cadastral system based on a NoSQL database and a JavaScript visualization application. *ISPRS Int J Geo-inf* 8(227):1–18
- Zlatanova S, Stoter J, Isikdag U (2012) Standards for exchange and storage of 3D information: challenges and opportunities for emergency response. In: *Proceedings of the 4th international conference on cartography and GIS, Albena*, vol 2. International Cartographic Association, pp 17–28

Optim3D: Efficient and Scalable Generation of Large-Scale 3D Building Models



Anass Yarroudh, Abderrazzaq Kharroubi, and Roland Billen

Abstract 3D building reconstruction in the urban environment is a key step in developing urban digital twins. So far, the existing researches focus on the modeling process and overlook the scalability issues related to the extensive data and iterative calculations. In this paper, we propose an approach for efficient and scalable 3D generation of large-scale building models from aerial LiDAR data. Our method involves data indexing and tiling for enhanced efficiency, along with parallel processing to accelerate the modeling process. We also correct ground-floor elevation inaccuracies in the final model to ensure reliable applications and analysis. The results show that, for large datasets, the indexing overcomes the memory saturation and speeds up the tiling and the reconstruction processes. The parallel processing accelerates the reconstruction process by nearly five times compared to an iterative approach. Additionally, using K-nearest neighbors instead of radius search resulted in more accurate elevation values. Our source code is available at: <https://github.com/Yarroudh/Optim3D>.

Keywords Reconstruction · LiDAR · Indexing · Tiling · Parallel processing · 3D Building models

This article was selected based on the results of a double-blind review of an extended abstract.

A. Yarroudh (✉) · A. Kharroubi · R. Billen
Geomatics Unit, University of Liège, Bâtiment B5a, Allée du six Août 19, 4000 Liège, Belgique
e-mail: ayarroudh@uliege.be

A. Kharroubi
e-mail: akharroubi@uliege.be

R. Billen
e-mail: rbillen@uliege.be

1 Introduction

Buildings are one of the most important urban objects to represent in 3D city models due to their prominent role in the urban tissue (Biljecki et al. 2015). They are usually represented by various types of data, such as 2D footprints and 3D LiDAR (Light Detection and Ranging) data. LiDAR provides fast acquisition of abundant high-resolution topographical data (Tomljenovic et al. 2015; Venugopal and Kannan 2013), making it valuable for reconstructing detailed building models. However, in an urban context, buildings modeling using three-dimensional point clouds can be problematic due to several reasons. The process presents challenges not only in creating detailed and accurate models but also in the computational demands. The processing of massive LiDAR data is time consuming due to the huge amount of records and the iterative calculations Venugopal and Kannan (2013).

In this paper, we propose an approach for efficient and scalable 3D generation of large-scale detailed building models from aerial LiDAR data. Our method involves data tiling and indexing for enhanced efficiency, along with parallel processing that accelerates the reconstruction process by nearly five times. This overcomes limitations of existing methods that are computationally expensive and lack scalability, especially with large datasets. Additionally, we correct elevation inaccuracies in the final model to ensure precise and accurate results, leading to reliable applications.

To briefly summarize, the main contributions of this work are twofold:

- An efficient and scalable approach for generating large-scale 3D building models.
- An accurate method to correct elevation inaccuracies in the final models.

The remainder of the paper follows this structure: Sect. 2 presents an overview of the existing 3D modeling of buildings, while Sect. 3 delves into a comprehensive explanation of the proposed method. In Sect. 4, we present and discuss the results, and finally, Sect. 5 contains the conclusion of the paper.

2 Related Works

The concept of Level of Detail (LoD) aims to distinguish various representations of semantic 3D city models at different scales. It is primarily used to express the level of geometric detail within a model. The authors in (Biljecki et al. 2016) proposed a series of 16 refined LoDs for buildings, with 4 refined LoDs for each of the LOD0-3 (Fig. 1). These refined levels were developed based on a literature review and an inventory of existing models, identifying their main similarities and common aspects. Therefore, they provide practitioners with a standardized way to define the complexity of their models.

Managing large urban data poses significant challenges. This includes software and hardware limitations in handling large spatial datasets of buildings. Therefore,

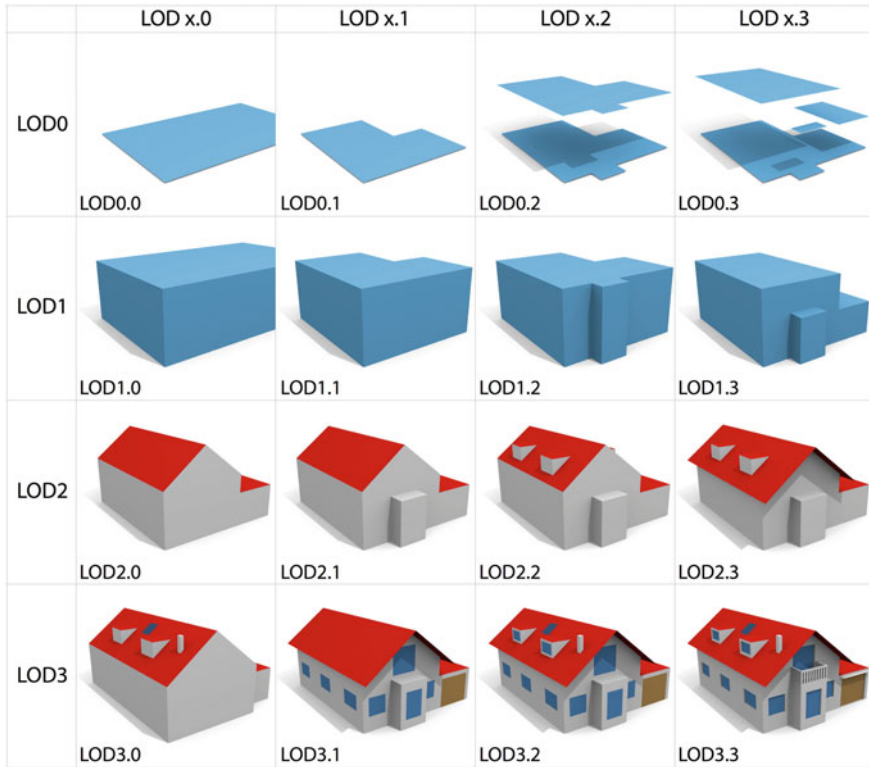


Fig. 1 Visual example of the refined LoDs for a residential building (Biljecki et al. 2016)

efficient spatial data indexing techniques are necessary for improved data retrieval and management in urban contexts Azri et al. (2013). These techniques optimize query response time and improve data retrieval efficiency.

Kd-trees and quad-trees are two commonly used space partitioning and indexing techniques in spatial data management. A kd-tree is a binary search tree used to index k-dimensional point data distributed in a k-dimensional space Sayar et al. (2015). As shown in Fig. 2, it recursively divides a dataset into smaller regions by splitting it along the median of one of its dimensions. The quad-tree is a data structure used to subdivide the 2D plane into four quadrants until each subdivision contains a small number of features or reaches a specific level of detail Azri et al. (2013) (Fig. 2). In 3D space, an octree is used as an extension of the quad-tree data structure. Similar to a quad-tree, which subdivides the 2D plane into four quadrants, an octree subdivides the 3D space into eight octants.

Besides using spatial indexing techniques, the selection of appropriate three-dimensional reconstruction approach is crucial for achieving detailed, accurate models and ensuring an efficient and time-saving modeling process. The authors in Li et al. (2016) proposed a fully automatic approach for reconstructing Manhattan-world

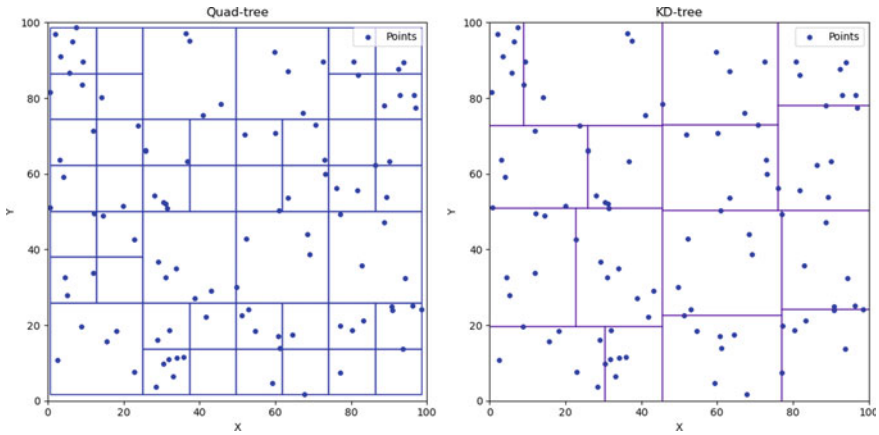


Fig. 2 Comparison between quad-tree and kd-tree indexing of a set of points

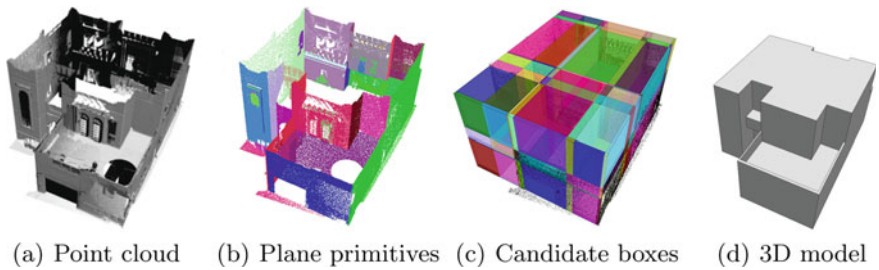


Fig. 3 An overview of Manhattan-world Urban reconstruction approach Li et al. (2016)

urban scenes. Their key idea centers around representing the buildings’ geometry using a set of well-aligned boxes (Fig. 3). The process starts by extracting planar primitives from the point cloud data, followed by an iterative refinement step. Subsequently, candidate boxes are derived by partitioning the point cloud space into a non-uniform grid. From this pool of candidates, an optimal subset is selected to best approximate the buildings’ geometry. The key contribution of their work is the innovative transformation of scene reconstruction into a labeling problem, which they solve using a novel Markov Random Field formulation. Differing from previous approaches tailored to specific types of input point clouds, Li et al. (2016) claims that their method can achieve accurate reconstructions from a variety of data sources.

Another framework was proposed by Nan and Wonka (2017) for reconstructing lightweight polygonal surfaces from point clouds. Unlike conventional methods that prioritize extracting geometric primitives or arranging them appropriately, PolyFit focuses on intersecting plane primitives and identifying appropriate combinations to create a manifold polygonal surface model without boundary. As illustrated in Fig. 4, the process can be divided into two key steps: candidate face generation and face selection. The first step consists of extracting the planar primitives from the

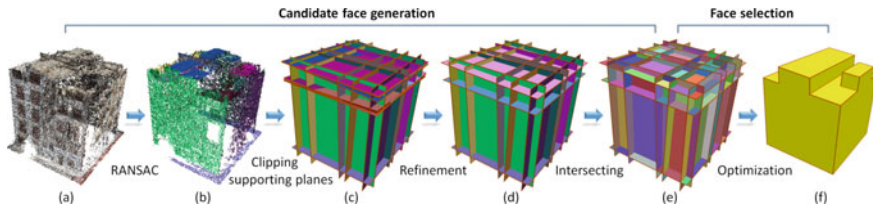


Fig. 4 PolyFit pipeline: **a** input point cloud **b** planar segments **c** supporting planes of the initial planar segments **d** supporting planes of the refined planar segments **e** candidate faces **f** reconstructed model. All planar segments and faces are randomly colored Nan and Wonka (2017)

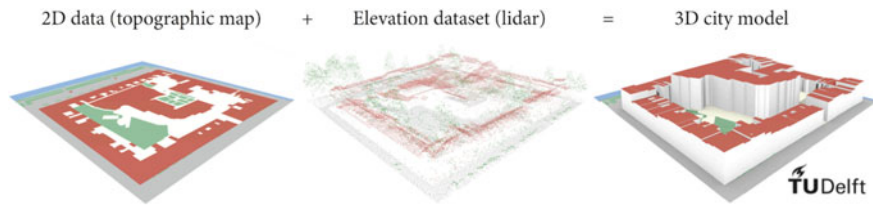


Fig. 5 Overview of 3dfier Ledoux et al. (2021)

point cloud using RANSAC (RANdom SAMple Consensus). An optimal subset of candidate faces is selected through an optimization process based on binary linear programming formulation, where hard constraints are applied to ensure the final polygonal surface model remains manifold and watertight. According to Nan and Wonka (2017), their approach exhibits proficiency in recovering sharp features and robustness against noise, outliers and missing data.

Based on Polyfit for single-building reconstruction, City3D Huang et al. (2022) proposes an extended hypothesis-and-selection-based polygonal surface reconstruction for large-scale cases. The approach addresses challenges by separating individual buildings using vectorized building footprints, inferring vertical planes based on prior knowledge of building structures, and incorporating these inferred planes as constraints in the surface reconstruction process. It produced a level of detail 2 (LOD2.2) and utilized planar primitives which can reconstruct most building types. However, for cases with curved surfaces, the framework cannot handle them. Hence, an extension to include primitives like spheres, cylinders, and parametric surfaces would be necessary.

In addition to hypothesizing and selection strategy approaches, Ledoux et al. (2021) proposed a software called 3dfier that serves as a solution to simplify the creation of 3D city models by transforming 2D GIS (Geographical Information System) datasets, such as topographical data, into three-dimensional representations (Fig. 5). The elevation is obtained from a point cloud and the software uses the semantics of each polygon to perform elevation adjustments, resulting in a comprehensive and accurate three-dimensional representation. Among other objects, buildings are generated as LoD1 blocks using a rule-based structure to facilitate the extrusion.

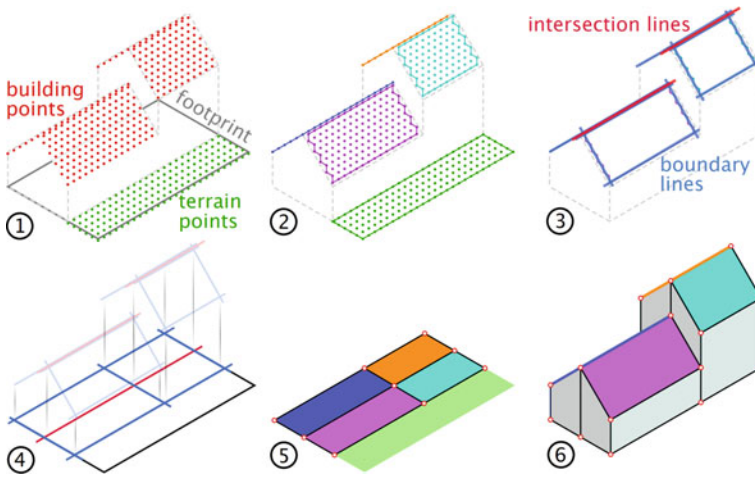


Fig. 6 Overview of GeoFlow pipeline Peters et al. (2022)

Geoflow Peters et al. (2022) is another software that takes as input an airborne point cloud and footprints of buildings to generate 3D building models. As illustrated in Fig. 6, the process involves detecting roof planes using a region growing algorithm and eliminating outliers and points on wall planes. The boundary lines of the roof planes are then identified using the α -shape of each detected plane, and the intersection lines are generated where adjacent planes meet. To reduce the complexity of the roof partition, a graph-cut optimization technique is applied, which minimizes the total length of the edges between faces of different roof planes. This results in a low model complexity, with a minimum number of edges and vertices. The final roof partition is composed of roof parts, which are extruded from the ground level to generate a mesh, producing a LoD2.2 model.

3 Method

This study aims to reduce the computational requirements needed to generate large-scale three-dimensional building models. The proposed approach involves indexing, tiling and parallel-processing of the input data for efficient and cost-effective modeling. As shown in Fig. 7, the method relies primarily on denoising the point cloud for accurate reconstruction, then both the point cloud and the buildings' footprints are indexed using spatial indexing structures. The processing areas are generated based on the buffered tiles. The buffering operation ensures accurate modeling by providing enough points around each building. Using parallel tasks, 3D building models are efficiently generated through independent processing of separate areas, resulting

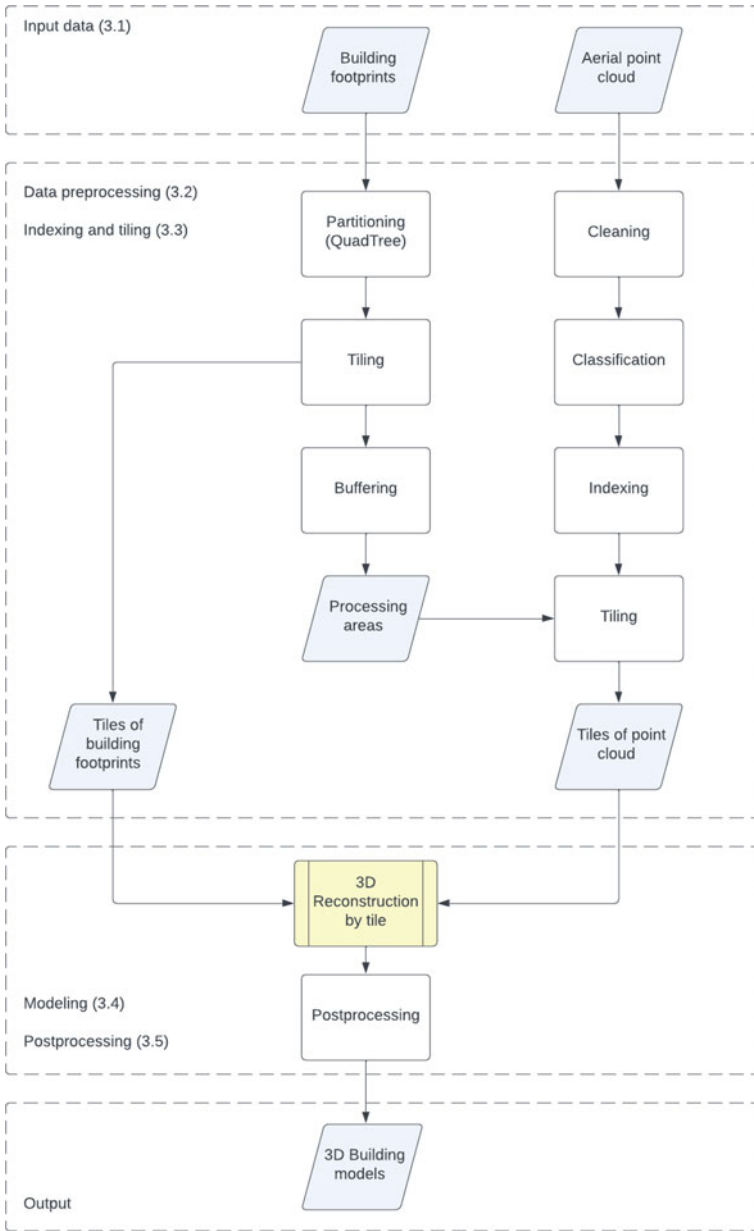


Fig. 7 Our proposed method for efficient and scalable 3D reconstruction of large-scale building models

in tiled final outputs. Additionally, further steps are taken to correct inaccuracies in the final model and to ensure high precision.

In the upcoming sub-sections, we provide a comprehensive overview of the proposed methodology.

3.1 *Materials*

For the reconstruction process, two types of input data are required: three-dimensional aerial LiDAR data and two-dimensional footprints of buildings. The LiDAR data is used to estimate the height of each building and to recreate the detailed geometry of roofs. Conversely, the footprints of buildings are required for instance segmentation of buildings, i.e. extract individual buildings from the point cloud.

For testing purposes, we used AHN-4 data which is a digital-height model covering the whole Netherlands. The dataset was collected using airborne LiDAR. As a result, the height at ground level of every square meter in Netherlands is known to an accuracy of 5 cm. The dataset is available via the AHN viewer: <https://ahn.arcgisonline.nl/ahnviewer>. In addition, the footprints of buildings were retrieved from OpenStreetMap.

3.2 *Data Pre-processing*

The pre-processing step aims to prepare LiDAR data for 3D modeling by: first, filtering the noise that can affect the reconstruction accuracy, and then classifying each point of the dataset into one of these classes: ground points, buildings' points and unclassified points.

This classification method is for unclassified datasets as it proposes an unsupervised method to automatically annotate the point cloud. The ASPRS (American Society for Photogrammetry and Remote Sensing) specifications for classification field should be considered.

1. Filtering the noise using SOR filter

The SOR (Statistical Outlier Removal) filter is used to remove the noisy points within a local neighborhood, resulting in a clean dataset without outliers. It works by computing the mean and standard deviation of the distance between each point and its k -nearest neighbors. Points that are farther away from their neighbors than a specified threshold are considered outliers and removed. For each point in the dataset, the threshold can be represented mathematically as:

$$T = \bar{d} + k \cdot s \quad (1)$$

where \bar{d} is the mean of the distances between a point and its k -nearest neighbors, and s is the standard deviation:

$$\bar{d} = \frac{1}{k} \sum_{i=1}^k d_i \quad (2)$$

$$s = \sqrt{\frac{1}{k-1} \sum_{i=1}^k (d_i - \bar{d})^2} \quad (3)$$

d_i represents the distance between a point \mathbf{p} and one of its k -nearest neighbors \mathbf{p}_i where $i = 0, 1, \dots, k$.

$$d_i = |\mathbf{p} - \mathbf{p}_i| \quad (4)$$

All the points with $d_i > T$ are then removed.

2. Ground points segmentation using CSF algorithm

The CSF (Cloth Simulation Filter) algorithm is used for separating the ground points from non-ground points in LiDAR data. Among many ground filtering algorithms, the CSF algorithm presented the best filtering results and produced the lowest total errors Yilmaz et al. (2018). The filter is based on the Cloth Simulation method, which is an algorithm used in 3D computer graphics to simulate fabric attached to an object. As proposed by Zhang et al. (2016), the first step consists of inverting the LiDAR point cloud, then a rigid fabric is used to cover the inverted surface. By comparing the fabric nodes to the corresponding LiDAR points, these nodes can be used to generate an approximation of the ground surface. Finally, ground points can be extracted from the LiDAR data by comparing the original points to the generated ground surface.

3. Buildings extraction from non-ground points

The second segmentation focuses on extracting buildings from non-ground points based on height and shape criteria. The height of each point in non-ground points is calculated by comparing a point's z -coordinate to the ground elevation at the location of the point:

$$H_{point} = Z_{point} - Z_{ground} \quad (5)$$

Buildings and other objects, such as trees, have different height distribution in elevation data. Buildings are typically taller and have more uniform distribution of heights. Consequently, a height threshold (e.g., 3 m) was used to identify potential building points. However, height alone is not sufficient and shape-based criteria are often used to further recognize the roof parts and walls. As buildings tend to have a more planar shape, the coplanarity test Limberger and Oliveira (2015) can be used to distinguish between buildings and other high objects.

3.3 *Indexing and Tiling*

Quad-tree and kd-tree are both considered as space driven indexing structures, where objects in two-dimensional space are partitioned into rectangular cells Azri et al. (2013). However, since the tiles should have a restricted number of buildings, quad-tree is a better choice as it allows to control the level of detail in each tile. This ensures that the time and resources required for tile reconstruction are consistent, and that the resulting tile files have approximately the same size Peters et al. (2022). The tile boundaries, which are basically the total bounding box of all the buildings in the tile, serve as a processing area for 3D reconstruction, while buffering ensures sufficient surrounding ground points for accurate ground-floor elevation estimation.

Indexing is a crucial step in processing airborne LiDAR data, as datasets can be extensive and computationally demanding. By using a spatial indexing structure, only the required parts of the data can be streamed, thus avoiding the need to parse the entire dataset. This significantly reduces the processing time of spatial queries, which otherwise would require a sequential scan of every record in the point cloud dataset. Unlike the buildings' footprints, the main purpose of point cloud indexing is not the spatial partitioning, but rather reducing the processing time, including the tiling operation. Therefore, two spatial indexing structures were tested and their performances compared: kd-tree and octree.

3.4 *Modeling*

The reconstruction is performed using Geoflow given that it's more adapted to aerial LiDAR datasets compared to other solutions. Considering the top-down view mode of acquisition, walls may be missing or lack sufficient points to accurately represent their geometry, resulting in false and/or less accurate models, especially when using PolyFit or Manhattan-world urban reconstruction pipeline. In contrast, Geoflow eliminates wall points and their geometry is reconstructed by extruding the roof parts from the ground level. This also ensures accurate ground-floor elevation. Additionally, the preference for Geoflow over 3dfier is justified by its ability to provide more accurate and detailed models. The generated model consists of three level of details where the highest is LoD2.2 that represents the detailed geometry of the building's roof (Fig. 8).

As urban areas may have thousands to millions of buildings, iterative reconstruction of all instances can be less-efficient and time consuming. Therefore, we opt for parallel processing by splitting the iterations of the loop into separate computations. This includes the use of multiple processors or cores to perform the modeling process of all tiles simultaneously, among parallel tasks, which significantly speeds up the process.

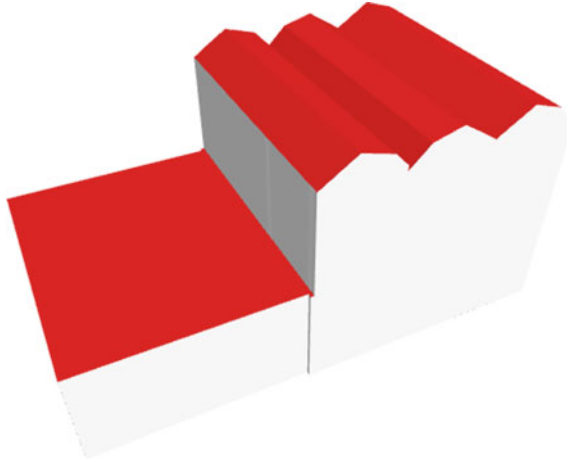


Fig. 8 LoD2.2 building model generated using GeoFlow

3.5 Post-processing

The generated models may contain inaccuracies and imperfections that require refinement. One issue is inaccurate ground-floor elevation, where the bottom of a building has significantly different elevation compared to the surroundings. This is mainly due to the radius neighboring method used by Peters et al. (2022) to identify the ground points around each building and use them to estimate the ground-floor elevation. However, this method may fail for buildings surrounded by others leading to inaccurate results.

Our proposed approach suggests K-nearest neighbors algorithm instead of radius searching to define the ground surface neighborhood, then correct the Z-coordinate of the vertices of each building's ground surface. The kd-tree structure is used to accelerate retrieval of the nearest neighbours. The elevation is then calculated by averaging neighboring point heights and correcting the ground vertices Z-coordinate accordingly.

4 Results and Discussion

In this section, we provide detailed results of our method for efficient and scalable reconstruction of large-scale detailed 3D building models. The study was conducted on an area of 17,236 buildings in Amsterdam, Netherlands. The entire AHN-4 tile contains 1.1 billion points. A sample of 89 millions points covering 4191 buildings was retrieved to compare the performances of the process on both large and small datasets. The workflow underwent rigorous testing on a single machine equipped with 64GB of RAM and an Intel(R) Core(TM) i9-10900X CPU operating at 3.70 GHz.

Table 1 Summary of the processing time

| Step | Sample | AHN-4 tile |
|--------------------------------|----------|------------|
| Footprints indexing and tiling | 00:00:27 | 00:01:54 |
| LiDAR tiling without indexing | 00:02:46 | – |
| LiDAR kd-tree indexing | 00:00:54 | – |
| LiDAR octree indexing | 00:03:22 | 00:59:40 |
| Indexed LiDAR tiling | 00:00:29 | 00:27:10 |
| Iterative reconstruction | 00:09:46 | 03:19:04 |
| Parallel reconstruction | 00:03:41 | 00:49:12 |
| Post-processing | 00:01:05 | 00:03:13 |

Table 2 LiDAR data classification results for different sets of parameters

| H [m] | K-NN | Accuracy | Precision | Recall | F1 Score | IoU |
|-------|------|----------|-----------|--------|----------|--------|
| 2.0 | 5 | 0.8111 | 0.8315 | 0.8110 | 0.8184 | 0.7113 |
| 2.0 | 10 | 0.8790 | 0.8930 | 0.8790 | 0.8826 | 0.7948 |
| 2.0 | 20 | 0.8885 | 0.9035 | 0.8885 | 0.8919 | 0.8084 |
| 2.0 | 30 | 0.8879 | 0.9046 | 0.8879 | 0.8914 | 0.8076 |
| 2.0 | 40 | 0.8865 | 0.9049 | 0.8865 | 0.8901 | 0.8056 |
| 2.5 | 5 | 0.8210 | 0.8408 | 0.8210 | 0.8184 | 0.8278 |
| 2.5 | 10 | 0.8865 | 0.9011 | 0.8865 | 0.8890 | 0.8056 |
| 2.5 | 20 | 0.8937 | 0.9098 | 0.8937 | 0.8971 | 0.8163 |
| 2.5 | 30 | 0.8928 | 0.9110 | 0.8928 | 0.8962 | 0.8150 |
| 2.5 | 40 | 0.8916 | 0.9118 | 0.8916 | 0.8951 | 0.8132 |
| 3.0 | 5 | 0.8310 | 0.8505 | 0.8110 | 0.8310 | 0.7337 |
| 3.0 | 10 | 0.8964 | 0.9123 | 0.8964 | 0.8996 | 0.8203 |
| 3.0 | 20 | 0.9027 | 0.9206 | 0.9027 | 0.9058 | 0.8299 |
| 3.0 | 30 | 0.8999 | 0.9199 | 0.8999 | 0.9031 | 0.8256 |
| 3.0 | 40 | 0.8968 | 0.9186 | 0.8968 | 0.9002 | 0.8210 |

A summary of the processing time required for each step of the process is shown in Table 1.

Before performing the indexing and reconstruction process, the point cloud was firstly classified as described in Sect. 3.2. The classification results for different parameters' values are presented in Table 2. As shown in Fig. 9, the best results were obtained for a height threshold of 3.0 meters and 20 neighboring points for coplanarity test, with a F1 Score of 90.58% and an Intersection-over-Union of 82.99%. The reason for choosing this approach is that it's unsupervised and can be applied to very different datasets. Alternatively, other supervised classification methods can be considered, leading to better results.

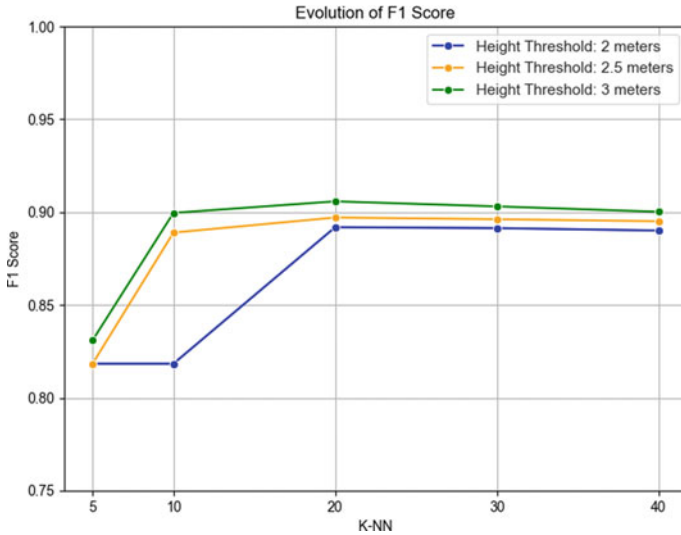


Fig. 9 Variation of F1 Score for different height thresholds and K-NN values

The sample data relatively small size makes the tiling process of LiDAR data possible without indexing. However, reading and processing an entire AHN-4 tile is hardware-extensive and results in memory and processor saturation. Therefore, using a spatial indexing structure was necessary to partition. Although the results on sample data show that building kd-tree was relatively faster than building the octree structure, this doesn't guarantee overall speed since the performances of kd-tree on larger datasets were not evaluated. However, using a spatial indexing structure, whether kd-tree or octree, speeds up the tiling process. Additionally, the parallel processing accelerates the reconstruction for the larger dataset by nearly five times compared to an iterative approach. However, machine performances are a significant factor that affects the computation time as the processor and memory capacities should be considered to prevent any saturation or failure issues.

As illustrated in Fig. 10, using the K-nearest neighbors instead of radius search shows better results for estimating correctly the elevation for buildings that lack sufficient ground point coverage. K-NN's ability to handle irregular point distribution ensures that relevant ground points are considered, even in regions with unevenly distributed points. This makes K-NN more robust and reliable method for estimating building's ground-floor elevation, outperforming the fixed-radius approach.

Overall, the results show that this method presents significant advancements in large-scale detailed 3D building modeling, providing a reliable and efficient solution for handling extensive urban data and producing accurate and detailed building models.

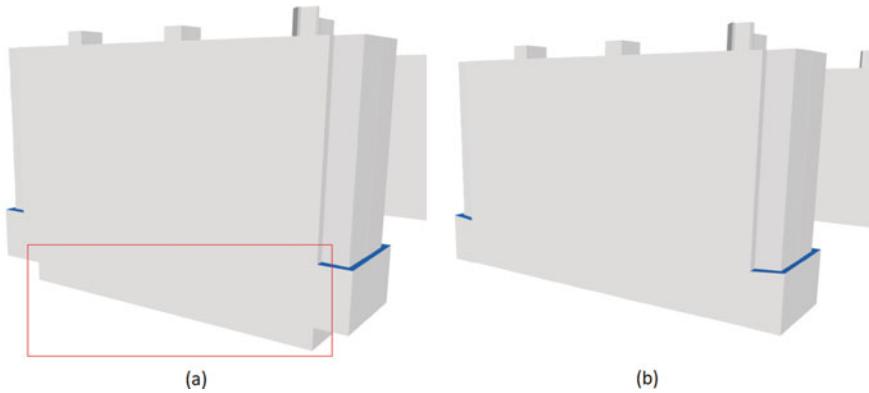


Fig. 10 Ground-floor elevation correction: (a) original model (b) corrected model

5 Conclusion

Our study presents an efficient and scalable approach for generating large-scale detailed 3D building models from aerial LiDAR data. The proposed method involves data indexing and tiling, as well as parallel processing, to optimize the modeling process. By using spatial indexing structures like kd-tree and octree, the tiling process is significantly expedited, overcoming the limitations of handling large datasets. Moreover, parallel processing accelerates the reconstruction by nearly five times compared to an iterative approach, enhancing overall efficiency. The proposed K-NN approach for defining the ground surface neighborhood yields superior results for estimating the ground-floor elevation of buildings lacking sufficient surrounding ground points. K-NN's adaptability to irregular point distributions ensures that relevant ground points are considered, making it a more robust and accurate method than the traditional radius search approach.

Future research could explore the application of supervised classification methods for improved classification results and further optimization strategies to enhance performance on even larger datasets.

Acknowledgements This research was conducted as part of the TrackGen project and supported by Logistics in Wallonia. We sincerely appreciate their financial support.

References

- Azri S, Ujang U, Anton F, Mioc D, Rahman AA (2013) Review of spatial indexing techniques for large Urban data management. International Symposium and Exhibition on Geoinformation, Kuala Lumpur, Malaysia
- Biljecki F, Stoter J, Ledoux H, Zlatanova S, Çöltekin A (2015) Applications of 3D city models: state of the art review. *ISPRS Int J Geo-Inf* 4:2842–2889. <https://doi.org/10.3390/ijgi4042842>

- Biljecki F, Ledoux H, Stoter J (2016) An improved LoD specification for 3D building models. *Computers, Environment and Urban Systems* 59:25–37. <https://doi.org/10.1016/j.compenvurbsys.2016.04.005>
- Huang J, Stoter J, Peters R, Nan L (2022) City3D: large-scale building reconstruction from airborne LiDAR point clouds. *Rem Sens* 14(9):2254. <https://doi.org/10.48550/arXiv.2201.10276>
- Ledoux H, Biljecki F, Dukai B, Kumar K, Peters R, Stoter J, Commandeur T (2021) 3dfier: automatic reconstruction of 3D city models. *J Open Sour Softw* 6:2866. <https://doi.org/10.21105/joss.02866>
- Limberger FA, Oliveira MM (2015) Real-time detection of planar regions in unorganized point clouds. Presented at the May 20. <https://doi.org/10.1016/j.patcog.2014.12.020>
- Li M, Wonka P, Nan L (2016) Manhattan-world Urban reconstruction from point clouds. In: Leibe B, Matas J, Sebe N, Welling M (eds) *Computer vision—ECCV 2016*. ECCV 2016. Lecture Notes in Computer Science, vol 9908. Springer, Cham. https://doi.org/10.1007/978-3-319-46493-0_4
- Nan, L. & Wonka, P.: PolyFit: Polygonal Surface Reconstruction from Point Clouds (2017). <https://doi.org/10.1109/ICCV.2017.258>
- Peters R, Dukai B, Vitalis S, van Liempt J, Stoter J (2022) Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the Netherlands. *Photogram Eng Rem Sens* 88(3):165–170. <https://doi.org/10.14358/PERS.21-00032R2>
- Sayar A, Eken S, Öztürk O (2015) Kd-tree and quad-tree decompositions for declustering of 2D range queries over uncertain space. *Front Inf Technol Electr Eng* 16:98–108. <https://doi.org/10.1631/FITEE.1400165>
- Tomljenovic I, Höfle B, Tiede D, Blaschke T (2015) Building extraction from airborne laser scanning data: an analysis of the state of the art. *Rem Sens* 7:3826–3862. <https://doi.org/10.3390/rs70403826>
- Venugopal V, Kannan S (2013) Accelerating real-time LiDAR data processing using GPUs. In: *IEEE 56th international Midwest symposium on circuits and systems (MWSCAS)*, Columbus, OH, USA, 2013, pp 1168–1171. <https://doi.org/10.1109/MWSCAS.2013.6674861>
- Yilmaz SC, Yilmaz V, Gungor O (2018) Investigating the performances of commercial and non-commercial software for ground filtering of UAV-based point clouds. *Int J Rem Sens* 39:5016–5042. <https://doi.org/10.1080/01431161.2017.1420942>
- Zhang W, Qi J, Wan P, Wang H, Xie D, Wang X, Yan G (2016) An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Rem Sens* 8:501. <https://doi.org/10.3390/rs8060501>

An Overview of the 3D GeoInfo Conference Series

The 3D GeoInfo Journey from 2006 to 2022: A Reflection



Alias Abdul Rahman

Abstract This paper describes research on 3D GIS and the related studies undertaken by the community that were published and presented at the 3D GeoInfo conferences from 2006 to 2022: the first conference was organized in Kuala Lumpur and the 17th conference in Sydney. Several earlier works (published prior to 2006) are also part of the compilation, forming a basis of the 3D GIS research literature and part of the introduction of this paper. Almost all events in the series are described, with the exception of the Wuhan version. The descriptions are categorized into two decades, and the reflections of the 3D GIS research, and development (R-&-D) are highlighted towards the end of the paper.

Keywords 3D spatial data · 3D spatial modelling · 3D data processing · 3D spatial database · 3D visualization · 3D city modelling · Digital twins · 3D geoinformation

1 Introduction

This paper reports the 3D GIS research and development by the community over the two decades from 2006 to 2022, as presented at the 3D GeoInfo conference series. Most of the research was published in the conference proceedings, which were mainly organized by academia (universities) and the International Society for Photogrammetry and Remote Sensing (ISPRS) and documented as books in the Lecture Notes on Geoinformation and Cartography (LNGC). However, not all conferences published with the LNGC but were rather as published as proceedings by university presses. The first event was held in 2006 in Kuala Lumpur, Malaysia. In general, the 3D GIS domain has been investigated in four main aspects: spatial data acquisition, spatial data processing, spatial database, and visualization as to

This article was selected based on the results of a double-blind review of an extended abstract.

A. A. Rahman (✉)

3D GIS Research Lab, Faculty of Built Environment and Surveying, Universiti Teknologi Malaysia, Johor Bahru, Johor 81310, Malaysia
e-mail: alias@utm.my

what was being researched thus far in the 2D domain. Generally, issues related to the four aspects generate and trigger more advanced concepts and techniques of spatial data modelling and visualization. Interoperability and standards related to 3D spatial data also being investigated in parallel by the community—for example, for sharing and disseminating data and information. This paper aims to share how the 3D GeoInfo conferences from 2006 to 2022 addressed, discussed, and highlighted recent developments at particular times, as well as issues related to geospatial domains and disciplines. This paper also describes a few early works on 3D GIS research, such as (Pilouk 1996; Zlatanova 1998 and Abdul-Rahman 2000). In general, 3D geoinformation research has been investigated in academia (including institutes) and by major software vendors. Section 2 describes the first decade of the 3D GeoInfo journey. Section 3 describes the second decade of the journey. Finally, Sect. 4 reflects on the research conducted by the community, as presented in the events, and then concludes the conference journey.

2 The First Decade

In general, we observed that 3D GIS was initially explored and investigated at universities and research institutes for postgraduate studies, especially at the PhD level, for example—the works by Pilouk (1996) and Zlatanova (1998) at the ITC, in the Netherlands. In later years, several software vendors including ESRI, Bentley, and Oracle—began to research the subject. Pilouk (1996) investigated the 3D domain by using a triangular irregular network (TIN) data structure as a way of representing and modelling 2.5D data. This data structure offers object representations using primitives such as nodes, edges, and surfaces of TIN geometry. Additionally, generating several tables of those primitives (with computer coding) made it possible and straight forward to establish the topological relationships of the objects. Figure 1 illustrates the TINs for representing 2.5D data and simple 3D data (boreholes data), as reported by Pilouk (1996). Subsequently, a few researchers made use of the same data structure to for manipulate and model 3D objects—Zlatanova (1998) and Abdul-Rahman (2000) at the ITC in the Netherlands and the University of Glasgow (in Scotland). This data structure offers a unique way of generating 3D objects thus, developing basic 3D spatial information tools and software. Zlatanova (1998) used the Web for displaying 3D objects online with ITC buildings as a case study, (Fig. 2).

In August 2006, the first 3D GeoInfo conference was in Kuala Lumpur, Malaysia. The papers were focused and categorized into the following topics: 3D spatial data acquisition (LiDAR and digital photogrammetry), 3D spatial data modelling and representation, and 3D GIS frameworks; 3D object reconstruction; 3D city modelling; 3D mapping, cadaster and utility; 3D visualization; 3D terrain modelling and digital orthophoto generation; and 3D modelling of historical objects (from a photogrammetry point of view). The two keynotes—addressed 3D geometries in spatial DBMS; and Web 3D service for navigation applications. The first conference dealt with 3D spatial data modelling—how 3D data could be transformed

Fig. 1 2D and 3D TIN for the spatial objects

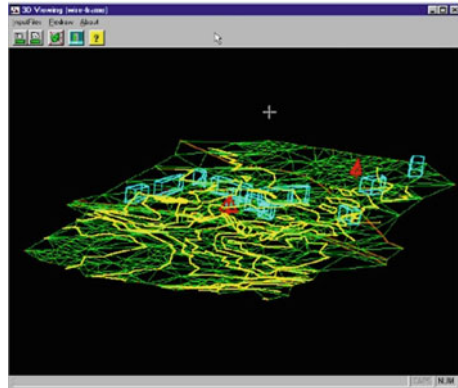
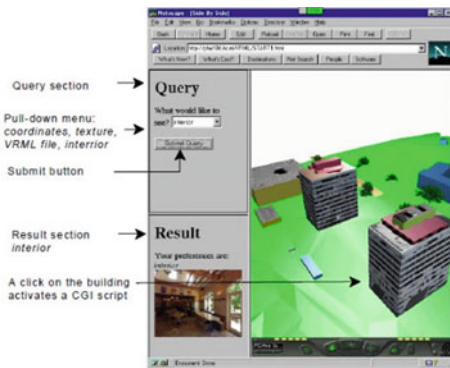


Fig. 2 Visualization of 3D objects via the web



and converted to be recognized as 3D objects with databases and visualization. Approaches for combined or amalgamated 3D data collections like LiDAR and digital photogrammetry were part of the discussions. The data structures for the data as a part of automated photogrammetry data acquisition were considered as relevant research then. All papers can be found in Abdul-Rahman et al. (2006).

In 2007, the second 3D GeoInfo event took place at TU Delft, where papers presented on the topics of 3D spatial data infrastructure (SDI), laser scanning for 3D topography, raster DTM for dike modelling, Web services for data sharing and querying, laser scanning for tree mapping, and 3D GIS applications. Research on 3D city modelling from LiDAR data began. Researchers from TU Delft also discussed the Poincaré²-TEN data structure. The Delaunay triangulation was part of the paper presentation, as well as modelling and topological data management in 3D GIS. TU Delft researchers also presented a paper on mathematical proofs for 2D and 3D representations. In the end, there were five position papers: the requirements and applications of 3D GIS; data collection and 3D reconstruction; modelling 3D geoinformation; spatial data analysis in 3D GIS; and 3D geo-visualization which were deliberated by academic colleagues. The level of research had improved, allowing

for more detailed knowledge on spatial data modelling, such as using 3D data structures for 3D spatial objects with 3D Voronoi diagrams and tetrahedrons. All papers can be found in van Oosterom et al. (2007).

In Seoul 2008, there were three keynotes—a virtual geographic environment for simulating air pollution; Thomas Kolbe's presentation on representing and exchanging 3D city models with CityGML; and the Korean Survey and Mapping Authority's presentation on their research and development program for a 3D GIS project in Korea. The proceedings covered the following topics: operators for modelling 3D objects; multi-layered space-event models for navigation in indoor space; a framework for automatic building generation in CityGML using BIM; 3D routing from terrain surface and street network; B-rep data structure to query 9-intersection topological relationships in 3D GIS; Oracel Spatial 11g query processing; 3D geo database interoperability persistent on CityGML; 3D indoor modelling and visualization with 3D laser scanner; solar radiation over urban texture from LiDAR data and image processing for environmental analysis; constraint-based generation and visualization of 3D city models; and 3D dynamic simulation and visualization for GIS overland flow modelling. The topic of 3D city modelling seems to have received the most attention from the audience. All papers can be found in Lee and Zlatanova (2008).

In Ghent 2009, the proceedings discussed several important aspects of 3D GIS research, such as 3D point cloud data acquisition for large areas like city centers. Presentations covered the topics of Euler operators for building indoor navigation; 3D visualization of glaciers; interactive Web view services; 3D tool for planning by using Google Earth; 3D geological model reconstruction; data validation in 3D Cadaster; 3D topological relationships; 3D city information with sensors for vehicle positioning in urban environment; visibility through visual landmarks in 3D navigation using geo-DBMS; 3D inclusion test on large dataset; and 3D volumetric soft objects for dynamic runoff modelling were presented. The conference also discussed 3D GIS applications. All papers can be found in Neutens and de Maeyer (2009).

In Berlin 2010, TU Berlin academics highlighted the developments in CityGML and 3D city modelling were highlighted, as the university was known as the center of the CityGML standard development. Several interesting papers presented on the detailed standards and schemas for analysis and applications of CityGML-based models. Other aspects of research also part of the proceedings, such as Malaysian 3D SDI; building licensing concept based on standardized 3D geoinformation; automated healing of 3D urban models from HFT Stuttgart; BIM and GIS interoperability within SQL; energy rehabilitation estimation for Berlin city building models using CityGML; automatic landmark detection for 3D urban models; BIM4GEOA; assignment of occluded textures concept within 3D city models stored in CityGML; 3D geoinformation standards in the Netherlands; Delaunay tetrahedralization; and dynamic features in 3D city models as energy system. The proceedings also discussed the subjects—of integrated 3D modelling of multi-utility networks; modelling of complex building interiors; large scale constraint Delaunay triangulation, CityGML and IFC models interoperability, 3D Cadaster for Quebec province; 2D and 3D spatial objects integrated representation for rigorous correct query and manipulation; CityGML and

GeoBIM extension, 3D modelling of geoscientific datasets with discrete Voronoi diagram, framework for full 3D building models; and semantic 3D city modelling and visual exploration. The attendees had the opportunity to informally discuss the various aspects of CityGML research and 3D GIS matters in general. In other words, it was a gathering oriented toward 3D city modelling. All papers can be found in Kolbe et al. (2010).

In Quebec 2012, the modelling of 3D topographic space against indoor navigation was the focus. The requirements for such modelling were the core message of the paper. Other papers covered the topics of–visibility of labels in 3D navigation maps; semantic 3D modelling of multi-utility network in cities for analysis and 3D visualization; generalization of visualization of 3D building models in CityGML; virtual forest generation from volumetric algorithm; camera control concept in 3D geovirtual environments; and representing 3D topography in DBMS with a star-based data structure. A paper from UCL in the UK described the topological pre-culling of faces for rendering the performance of city models in Google Earth as part of the author’s PhD work. A German colleague presented on the problems and benefits of 3D topology on geometries in geomorphology. Researchers from HfT Stuttgart presented on the geometric-semantic consistency validation of CityGML models. A colleague from the University of Stuttgart discussed a topic on advancing DB4Geo. The paper dealt with the database for 3D mass data and the new implementation idea that supported GML data and WebGL 3D interface. Colleagues from the Netherlands presented on indoor localization using Wi-Fi-based fingerprinting and trilateration techniques for LBS applications; integrating scale and space in 3D city models; modelling and application domain extension of CityGML in UML; representation and reconstruction of TIN with vertical walls; 3D geospatial modelling and visualization for marine environment; three step procedure to enrich augmented reality games with CityGML 3D semantic modelling; implementation of a national 3D–in the case of Netherlands; and open building models–towards a platform for crowdsourcing virtual 3D cities. The Montreal event revealed several major developments, including semantic 3D city modelling, 3D generalization, the 3D topography database with a star-based data structure; the validation of 3D building models; and national standards for 3D data in the Netherlands. All papers can be found in Pouliot et al. (2012).

3 The Second Decade

In Istanbul 2013, the community discussed issues related to the validation of CityGML models, updating 3D buildings using LiDAR data, and using neural networks and genetic algorithms for indoor navigation during evacuation. Researchers from Germany and the Netherlands also addressed enhanced conceptions of level of details (LoDs) for 3D city models and CityGML. The discussions at the event also included 3D Web service and databases, 3D spatial data modelling for underground objects, mobile rendering performance between LoD1 and LoD2,

and the 3D visualization strategy for underground objects. Energy demand estimations were examined from the perspective of 3D city models and statistical data. Participants also discussed semantic BIM and GIS for energy efficient buildings as initiatives for national 3D data and services. All papers can be found in Isikdag (2013).

In Dubai 2014, the focus was on 3D modelling and visualization of spatial objects like 3D buildings; city objects and underground water pipelines were manipulated using the tetrahedron network (TEN) data structure for underground objects and CityGML structure for buildings objects. Researchers from TU Berlin and Karlsruhe Institute of Technology in Germany demonstrated the related applications. The organizer presented 3D city models of Dubai city, which was more on the 3D visualization aspect of the city models. Other papers on 3D city models also discussed topics such as the generalization of 3D IFC models, multi-LOD CityGML datasets, change detection in CityGML documents and change detection in cities in general, and indoor route planning using semantic 3D building models with cloud computing. The conference also discussed integrating 3D city models with other data sources like remotely sensed SAR data as for the post-disaster mapping of urban areas. Additionally, Dutch researchers discussed the national coverage of 3D models between research and practices. All papers could be referred to Breunig et al. (2014).

In Kuala Lumpur 2015, a colleague from Oracle Inc. presented a keynote on the point cloud data management benchmark. A group of TU Delft researchers presented on the investigation of finer LoDs for shadow estimation. Other important papers included those by TU Berlin and TU Munich researchers on BIM and 3D city models; the 3D LADM prototype in Interlis. The TU Berlin and TU Munich continued to address CityGML continued to be addressed with a paper on versioning and history management. A mix of authors from TU Berlin and TU Munich also presented 3D traffic noise analysis within CityGML. Malaysian researchers presented on the octree data structure for 3D GIS, mainly for neighborhood searches. Temporal and spatial database applications remained actively investigated by the community. State-of-the-art cartographic enrichment of 3D city models was also part of the proceedings, from a paper jointly authored by German and Malaysian researchers from Universiti Teknologi Malaysia (UTM). All papers can be found in Rahman et al. (2015).

In Athens 2016, CityGML and 3D city modelling dominated the discussions. The research aspects of LoDs within CityGML for 3D building generation, space partitioning, and semantics 3D city modelling were discussed. Presentation also investigated BIM and 3D documentation for heritage structures, 3D modelling of trees from LiDAR point clouds, light laser scanners for UAV, and 3D surveying and modelling for a university campus in Indonesia. The proceedings addressed 3D indoor mobile positioning based on LiDAR data. All papers can be found in Dimopoulou and van Oosterom (2016).

In Melbourne 2017, TU Munich researchers presented several papers on CityGML and 3D city modelling. The community also investigated LADM and Interlis for 3D cadaster. The Singapore Land Authority presented a national-scale project using CityGML and 3D mapping. Other papers addressed WebGIS to support GPR 3D data acquisition as a step for the integration of underground utility networks in 3D

city models, and indoor space location models based on LBS. All papers can be found in Kalantari and Rajabifard (2017).

In Delft 2018, the joint GeoDelft 2018 conference featured three major events—3D GeoInfo, 3D Cadastre, and Smart Data Smart City conferences. This version of 3D GeoInfo 2018 included several papers on important aspects of 3D research, such as IFC and CityGML integration; analysis and applications of 3D city models and other aspects such as the national scale of 3D data acquisition to 3D city models generation using CityGML models (the Singapore case); and the dynamic 3D visualization of floods (in the case of the Netherlands). The community also discussed the knowledge of BIM-GIS integration. Participants noted the interest in further investigation on solar potential analysis with semantic 3D city models and mesh models. All papers can be found in Ohori et al. (2018).

In Singapore 2019, the event featured four papers on—CityGML and the 3D city modelling aspects of research, which still dominated the community. Solar radiation modelling on 3D objects was also presented. Researchers from the Netherlands and the UK jointly presented GeoBIM as part of the EuroSDR project. Contemporary research revisited 3D cadaster. One paper presented on the LADM standard for underground utility objects (in the case of Singapore). The Singapore Land Authority also took the opportunity to showcase its 3D city model project, with several applications planned as part of the Smart Nation initiative. All papers can be found in Stouffs et al. (2019).

London 2020 event focused on BIM/GIS integration, with the proceedings addressing the nation-wide discussion on the nation-wide discussion on generation, storage, updating, and dissemination. Participants investigated and presented on the automatic conversion of CityGML to IFC. Other aspects such as IoT sensors and 3D indoor models with IndoorGML were addressed as well. The OGC working group discussed the 3D representation of underground objects for data definition and integration. The visualization of CityGML objects and ADE continued to be part of the paper presentations in the conference series. Furthermore, geoinformation and VR/AR were also being investigated by the community. All papers can be found in Wong et al. (2020).

The New York 2021 conference was an online event due to the COVID-19 pandemic, jointly organized with the 3D Cadaster Workshop. Here, the selected papers were categorized into several aspects: data creation and management; data integration and information fusion; 3D data processing and visualization; AI and big data; BIM, digital twin, and smart infrastructure; and 3D city modelling. The proceedings included the nation-wide 3D building models quality assessment (of the Netherlands). Solar potential analysis and calculation from 3D building models was described with roof semantics. 3D building generation from drone and laser-scanned data continued to be part of the discussion as well as the use of deep learning with CityJSON and 2D floor plan for automatic 3D building generation. BIM and GIS data interoperability remained a discussion point in the conference. Digital twins of 3D objects were also interesting to note, where voxels and octree-based growing regions were used for the process of object generation. All papers can be found in Truong-Hong et al. (2021).

In Sydney 2022, digital twins and 3D city models were among the focal research areas by the community. HFT Stuttgart researchers presented on CityGML 3.0 for 3D cadaster. The University of Melbourne group presented on the use of CityGML 3.0 for 3D underground land administration. Point clouds and 3D building reconstruction were still being addressed in the proceedings, and BIM and GIS continued to be a part of research in the community. One paper on CityJSON for energy demand calculation was especially interesting, examining the JSON as an alternative (or a complement) of CityGML. The proceedings also described 3D indoor navigation for pedestrian interaction purposes. All papers can be found in Aleksandrov et al. (2022).

4 Summary and Conclusions

The 3D GeoInfo conference series continue to be an important gathering for the community–academia and research institutes, as well as several national mapping agencies, including Ordnance Survey of the UK, Dutch Kadaster of the Netherlands, the German NMA, and the Singapore Land Authority. Major vendors such as ESRI Inc., and Oracle Inc. participated in keynotes and as 3D GIS tutorial convenors in several 3D GeoInfo events. As time has passed, the research domain has evolved from distinctive primitives of 3D data structures, 3D indexing and storage techniques, 3D data modelling, spatial databases, and 3D visualization to more advanced techniques, systems, and services. In reflecting the first decade–3D data acquisition was the focus e.g., how the data could be automatically utilized for the next tasks or at least to minimize data processing. Significant effort has been spent on data transformations from 2.5D to 3D, ranging from algorithm developments to the subsystems of display user interfaces. Databases and 3D visualization were also “heavily” investigated among the researchers, as these two influenced the outcome of the overall system. Furthermore, the development of spatial data standards including CityGML, LADM and CityJSON–generates better systems and services for 3D spatial data either standing alone or online (via the Web). These aspects of standards research were presented in the 3D GeoInfo conferences in Delft, Melbourne, and recently in Sydney (the 17th version of the event, in 2022). In the second decade, more and more studies have been directed toward digital twins of the city and urban models (the “big brother” of the 3D city models). The development of various aspects related to the domain remains active, and it is not an exaggeration to say that the pace looks even “faster” today. Ultimately, the 3D GeoInfo conferences have been an interesting 3D geoinformation conference series for the geospatial science community since 2006. Above all, 3D GeoInfo has been a specialized conference in geoinformation industry for two decades, and it is also being recognized for many years to come.

References

- Abdul-Rahman A (2000) 2D and 3D TIN data structures for 3D GIS development. PhD thesis, University of Glasgow, Scotland, U.K.
- Abdul-Rahman A, Zlatanova S, Coors V (2006) In: Proceedings of 3D Geoinfo conference, LNGC Series. Springer, Heidelberg, Berlin, Germany
- Aleksandrov M, Barton J, Zlatanova S (2022) THE 17th 3D Geoinfo conference: preface archives. In: The international archives of the photogrammetry, remote sensing and spatial information sciences, 17th 3D GeoInfo Conference, Sydney, Australia, vol XLVIII-4/W4-2022, pp 1–3
- Breunig M, Al-Doori M, Butwilowski E, Kuper PV, Benner J, Haefele K-H (2014) In: Proceedings of 9th 3D Geoinfo Conference, Dubai, UAE
- Dimopoulou E, van Oosterom P (2016) Preface. In: The international archives of the photogrammetry, remote sensing and spatial information sciences, 11th 3D Geoinfo conference, Athens, Greece, vol XLII-2/W2, p 1
- Isikdag U (2013) In: ISPRS annals of the photogrammetry, remote sensing and spatial information sciences, 8th 3D Geoinfo Conference, WG II/2 workshop, vol II-2/W1
- Kalantari M, Rajabifard A (2017) In: The international archives of the photogrammetry, remote sensing and spatial information sciences, 3D Geoinfo conference, vol XLII-4/W7
- Kolbe TH, König G, Nagel C (2010) In: Proceedings of 3D Geoinfo conference, LNGC Series, Springer, Heidelberg, Berlin, Germany
- Lee J, Zlatanova S (2008) 3D Geo-information sciences. In: Proceedings of 3D Geoinfo conference, Seoul, Korea, p 215
- Neutens T, de Maeyer P (2009) In: Proceedings of 3D Geoinfo conference, LNGC Series. Springer, Berlin, Germany. <https://doi.org/10.1007/978-3-642-04791-6>
- Ohuri KA, Labetski A, Agugiaro G, Koeva M, Stoter J (2018) Preface. In: ISPRS annals of the photogrammetry, remote sensing and spatial information sciences, 13th 3D GeoInfo conference, Delft, The Netherlands, vol IV-4/W6, p 1
- Pilouk M (1996) Integrated modelling for 3D GIS. PhD thesis, ITC, Enschede, The Netherlands
- Pouliot J, Daniel S, Hubert F, Zamyadi A (2012) In: Proceedings of 3D Geoinfo conference, LNGC Series, Springer, Heidelberg, Berlin, Germany
- Rahman UA, Isikdag, Castro FA (2015) In: The international archives of the photogrammetry, remote sensing and spatial information sciences, 3D Geoinfo conference, vol XL-2/W4
- Stouffs R, Biljecki F, Soon KH, Khoo V (2019) Preface. In: The international archives of the photogrammetry, remote sensing and spatial information sciences, 14th 3D GeoInfo Conference, Singapore, vol XLII-4/W15, p 1
- Truong-Hong L, Jia F, Che E, Emamgholian S, Laefer D, Vo AV (2021) 3D GeoInfo. In: The international archives of the photogrammetry, remote sensing and spatial information sciences, 16th 3D GeoInfo Conference 2021, New York City, USA, vol XLVI-4/W4-2021, pp 1–4
- van Oosterom PMJ, Zlatanova S, Penninga F, Fendel E (2007) In: Proceedings of 3D Geoinfo conference, LNGC Series. Springer, Heidelberg, Berlin, Germany
- Wong K, Ellul C, Morley J, Home R, Kalantari M (2020) 3RD BIM/GIS integration workshop and 15th 3D GEOINFO conference 2020—preface. In: The international archives of the photogrammetry, remote sensing and spatial information sciences, 3rd BIM/GIS integration workshop and 15th 3D GeoInfo Conference, London, UK, volume XLIV-4/W1-2020, p 1
- Zlatanova S (1998) 3D GIS for urban development. PhD thesis, ITC, Enschede, The Netherlands