

Lecture Notes
in Geoinformation and Cartography

LNG&C

Alias Abdul-Rahman *Editor*

Advances in 3D Geoinformation

 Springer

Lecture Notes in Geoinformation and Cartography

Series editors

William Cartwright, Melbourne, Australia

Georg Gartner, Wien, Austria

Liqu Meng, Munich, Germany

Michael P. Peterson, Omaha, 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.

More information about this series at <http://www.springer.com/series/7418>

Alias Abdul-Rahman
Editor

Advances in 3D Geoinformation

 Springer

Editor

Alias Abdul-Rahman
Department of Geoinformation
Universiti Teknologi Malaysia
Johor Bahru
Malaysia

ISSN 1863-2246 ISSN 1863-2351 (electronic)
Lecture Notes in Geoinformation and Cartography
ISBN 978-3-319-25689-4 ISBN 978-3-319-25691-7 (eBook)
DOI 10.1007/978-3-319-25691-7

Library of Congress Control Number: 2016952002

© Springer International Publishing AG 2017

This work is subject to copyright. All rights are reserved 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, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

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

Contents

Realistic Benchmarks for Point Cloud Data Management Systems.	1
Peter van Oosterom, Oscar Martinez-Rubi, Theo Tijssen and Romulo Gonçalves	
Does a Finer Level of Detail of a 3D City Model Bring an Improvement for Estimating Shadows?	31
Filip Biljecki, Hugo Ledoux and Jantien Stoter	
Interactive and View-Dependent See-Through Lenses for Massive 3D Point Clouds	49
Sören Discher, Rico Richter and Jürgen Döllner	
Representation of CityGML Instance Models in BaseX.	63
Sabine Koch and Marc-O. Löwner	
A Spatio-Semantic Query Language for the Integrated Analysis of City Models and Building Information Models	79
S. Daum, A. Borrmann and T.H. Kolbe	
A Methodology for Modelling of 3D Spatial Constraints	95
Daniel Xu, Peter van Oosterom and Sisi Zlatanova	
Reconstructing 3D Building Models with the 2D Cadastre for Semantic Enhancement.	119
Frédéric Pedrinis and Gilles Gesquière	
A 3D LADM Prototype Implementation in INTERLIS	137
Eftychia Kalogianni, Efi Dimopoulou and Peter van Oosterom	
Web-Based Tool for the Sustainable Refurbishment in Historic Districts Based on 3D City Model	159
Iñaki Prieto, Jose Luis Izkara and Rubén Béjar	

Terrestrial Laser Scanners Self-calibration Study: Datum Constraints Analyses for Network Configurations	171
Mohd Azwan Abbas, Halim Setan, Zulkepli Majid, Albert K. Chong, Lau Chong Luh, Khairulnizam M. Idris and Mohd Farid Mohd Ariff	
Managing Versions and History Within Semantic 3D City Models for the Next Generation of CityGML	191
Kanishk Chaturvedi, Carl Stephen Smyth, Gilles Gesquière, Tatjana Kutzner and Thomas H. Kolbe	
Cartographic Enrichment of 3D City Models—State of the Art and Research Perspectives	207
Stefan Peters, Mathias Jahnke, Christian E. Murphy, Liqiu Meng and Alias Abdul-Rahman	
Comparison of 2D & 3D Parameter-Based Models in Urban Fine Dust Distribution Modelling	231
Yahya Ghassoun and M.-O. Löwner	
Investigating Semantic Functionality of 3D Geometry for Land Administration	247
George Floros, Eva Tsiliakou, Dimitrios Kitsakis, Ioannis Pispidikis and Efi Dimopoulou	
3D Complete Traffic Noise Analysis Based on CityGML	265
Lu Lu, Thomas Becker and Marc-Oliver Löwner	
Highly Efficient Computer Oriented Octree Data Structure and Neighbours Search in 3D GIS	285
Noraidah Keling, Izham Mohamad Yusoff, Habibah Lateh and Uznir Ujang	
Framework for on an Open 3D Urban Analysis	305
Marc-O. Löwner and Thomas Becker	
Usability Assessment of a Virtual Globe-Based 4D Archaeological GIS	323
Berdien De Roo, Jean Bourgeois and Philippe De Maeyer	
Temporal and Spatial Database Support for Geothermal Sub-surface Applications	337
M. Jahn, M. Breunig, E. Butwilowski, P.V. Kuper, A. Thomsen, M. Al-Doori and E. Schill	
Automatic Semantic and Geometric Enrichment of CityGML Building Models Using HOG-Based Template Matching	357
Jon Slade, Christopher B. Jones and Paul L. Rosin	

Stochastic Buildings Generation to Assist in the Design of Right to Build Plans 373
 Mickaël Brasebin, Julien Perret and Romain Reuillon

3D Marine Administration System Based on LADM 385
 Aikaterini Athanasiou, Ioannis Pispidikis and Efi Dimopoulou

Assessing the Suitability of Using Google Glass in Designing 3D Geographic Information for Navigation 409
 Kelvin Wong and Claire Ellul

Review and Assessment of Current Cadastral Data Models for 3D Cadastral Applications 423
 Ali Aien, Abbas Rajabifard, Mohsen Kalantari and Ian Williamson

The Hierarchical Three-Dimensional (3D) Dynamic Water Infiltration on Multi-layers of Soil According to Voronoi Sequence Nodes Based on the Three-Dimensional Triangular Irregular Network (3D TIN) 443
 Siti Nurbaidzuri Reli, Izham Mohamad Yusoff, Habibah Lateh and Uznir Ujang

A Data Model for the Interactive Construction and Correction of 3D Building Geometry Based on Planar Half-Spaces 461
 Martin Kada, Andreas Wichmann, Nina Manzke and Yevgeniya Filippovska

The Potential of the 3D Dual Half-Edge (DHE) Data Structure for Integrated 2D-Space and Scale Modelling: A Review 477
 Hairi Karim, Alias Abdul Rahman, Pawel Boguslawski, Martijn Meijers and Peter van Oosterom

Towards Integrating BIM and GIS—An End-to-End Example from Point Cloud to Analysis 495
 Claire Ellul, Gareth Boyes, Charles Thomson and Dietmar Backes

Realistic Benchmarks for Point Cloud Data Management Systems

Peter van Oosterom, Oscar Martinez-Rubi, Theo Tijssen
and Romulo Gonçalves

Abstract Lidar, photogrammetry, and various other survey technologies enable the collection of massive point clouds. Faced with hundreds of billions or trillions of points the traditional solutions for handling point clouds usually under-perform even for classical loading and retrieving operations. To obtain insight in the features affecting performance the authors carried out single-user tests with different storage models on various systems, including Oracle Spatial and Graph, PostgreSQL-PostGIS, MonetDB and LAStools (during the second half of 2014). In the summer of 2015, the tests are further extended with the latest developments of the systems, including the new version of Point Data Abstraction Library (PDAL) with efficient compression. Web services based on point cloud data are becoming popular and they have requirements that most of the available point cloud data management systems can not fulfil. This means that specific custom-made solutions are constructed. We identify the requirements of these web services and propose a realistic benchmark extension, including multi-user and level-of-detail queries. This helps in defining the future lines of work for more generic point cloud data management systems, supporting such increasingly demanded web services.

P. van Oosterom (✉) · T. Tijssen
Faculty of Architecture and the Built Environment, Department OTB,
Section GIS Technology, TU Delft, Delft, The Netherlands
e-mail: P.J.M.vanOosterom@tudelft.nl

T. Tijssen
e-mail: T.P.M.Tijssen@tudelft.nl

O. Martinez-Rubi · R. Gonçalves
Netherlands eScience Center, Amsterdam, The Netherlands
e-mail: O.Rubi@esciencecenter.nl

R. Gonçalves
e-mail: R.Goncalves@esciencecenter.nl

1 Introduction

Traditionally point clouds have been converted to grids, vector objects or other types of data to support further processing in a GIS environment. Today point clouds are also directly used for estimating volumes of complex objects, visibility analysis, roof solar potential analysis, 3D visualizations and other applications. In addition, the usage of point cloud data over the web, usually for visualization purposes, have also increased significantly. For example in archaeology, as shown in Fig. 1, point clouds are crucial for the 3D documentation and analysis of sites (Mapping the Via Appia in 3D 2015; De Kleijn et al. 2015). In addition to managing grids, vectors or TINs, it is more and more demanded that data management solutions are able to handle massive point clouds.

This paper extends the results presented in our previous work (van Oosterom et al. 2015) where we identified the various systems for managing point cloud data on the market, we defined a conceptual benchmark based on user requirements and we investigated the performance of the various point cloud management systems (PCDMS's) via an executable benchmark. Moreover, several methods for the improvement of the various PCDMS's were suggested and tested (Martinez-Rubi et al. 2015). In this paper we present new benchmark results motivated by the latest developments of the Point Data Abstraction Library (PDAL). In addition, we study the usage of some web services based on point cloud data which currently rely on specific solutions. Based on this analysis, we identify ways to improve the existing executable benchmark in order to reflect the needs of such services. This can also be used as a guideline for the new developments required by the PCDMS's in order to have more generic solutions which could also be used for these demanding services.



Fig. 1 Snapshot of a web application based on point clouds for an archaeology application

The outline of this paper is as follows: The different PCDMS's are introduced in Sect. 2. A summary of the executable benchmark defined in our previous work (van Oosterom et al. 2015) together with its results (second half 2014) are presented in Sect. 3, while the results of the new benchmark execution (summer 2015) are detailed in Sect. 4. In Sect. 5 we describe two services based on point cloud data, the 'Actueel Hoogtebestand Nederland' (AHN, height model of The Netherlands) 2D viewer (AHN viewer 2015) and the AHN2 3D web viewer and download tool (AHN2 3D viewer 2015), and we perform an analysis of their usage. Motivated by this study, in Sect. 6 we propose an extension of the executable benchmark to cover the challenges in supporting visualization tools and multi-user interactions. Finally, the conclusions and future work are described in Sect. 7. At the end of the document there is an Appendix section which contains extended information on the previous (second half 2014) benchmark execution.

2 Point Cloud Management Systems

The suitability of Database Management Systems (DBMS) for managing point cloud data is a continuous debate. File-based solutions provide efficient access to data in its original format, however, data isolation, data redundancy, and application dependency on such data format are major drawbacks. Furthermore, file-based solutions have also poor vertical and horizontal scalability (Fiore et al. 2013). We consider and compare both DBMS and file-based solutions for PCDMS's. In the latter points are stored in files in a certain format, and accessed and processed by solution specific software. Within DBMS solutions two storage models can be distinguished:

- Blocks model: nearby points are grouped in blocks which are stored in a database table, one row per block
- Flat table model: points are directly stored in a database table, one row per point, resulting in tables with many rows.

All file-based solutions use a type of blocks model and we have chosen to test LAStools by Rapidlasso (Rapidlasso GmbH 2015) as basis for our used file-based PCDMS. The DBMS with native point cloud support based on the blocks model are Oracle Spatial and Graph and PostgreSQL-PostGIS. Even though in principle any DBMS with spatial functionality can be used as a PCDMS with the flat table model, we tested Oracle Spatial and Graph (Oracle Database Online Documentation 12c Release 1 (12.1) 2014), PostgreSQL-PostGIS (Group, T.P.G.D. 2014) and the column-store MonetDB (Martinez-Rubi et al. 2014). For more information regarding the various PCDMS's and their tuning (block size, compression), we refer the reader to our previous work (van Oosterom et al. 2015).

In addition to using native blocks solutions, it is also possible to use third-party blocking solutions. Point Data Abstraction Library (PDAL 2015) is a software for manipulating point cloud data and it is used, in general, as an abstraction layer on management operations. Thus the same operations are available independently

on which system (DBMS or file-based) actually contains the data. It has recently released its version 1.0.0. It is compatible with a long list of point cloud file formats including LAS and LAZ and it has a set of commands that can be used to create a stand-alone file-based PCDMS, thus offering an alternative to LAStools.

PDAL has database drivers for Oracle Spatial and Graph, PostgreSQL and SQLite, i.e. tools to load and retrieve data to/from a database using the blocks model. Unfortunately the PDAL point cloud format used for Oracle Spatial and Graph is not fully compatible with the native format provided by the SDO_PC package (Oracle Database Online Documentation 12c Release 1 (12.1) 2014). On the other hand, PDAL is compatible with the native format provided by the PostgreSQL-PostGIS point cloud extension (Ramsey 2015). SQLite does not have native point cloud support, thus the functionality is limited to the PDAL features.

Based on the LASzip compression (Rapidlasso GmbH LASzip—free and lossless LiDAR compression 2014) the PDAL team has recently included the laz-perf compression (laz-perf 2015) which can be used in Oracle Spatial and Graph and SQLite. It achieves a compression rate similar to LAZ.

3 Executable Benchmark

The required point cloud functionality and usage profile for governmental institutions, industry and academia was obtained through a set of interviews (Suijker et al. 2014). The profile showed that spatial selections for data analysis or point cloud data extraction for further analysis by external tools were the ones ranked higher. In addition it was also detected that the data preparation was also a concern or challenge for the users. This information was used to define a conceptual benchmark for PCDMS's. An executable benchmark was derived from it. It has several stages with incremental data sets and tested functionality. The data sets vary from a few million points to several hundred billion points and they were subsets of AHN2 (Actueel Hoogtebestand Nederland (AHN) 2015), the second National Height Model of the Netherlands, which in its raw format consists of 640 billion points.

The first stage of the executable benchmark is the *mini-benchmark* which goal is to gain experience with the used systems. In this stage each system loads a small data set with 20 million points and performs seven queries (retrieval operations) based on selecting points that are contained in rectangular, circular or polygonal regions.

The second stage of the benchmark is the *medium-benchmark* and it is used to test the scaling performance of the systems as well as more advanced functionality. In this stage each system loads four different data sets, from 20 million, 210 million, 2.2 billion to 23 billion points and executes 20 queries which include, in addition to the *mini-benchmark* queries, elevation queries, such as points with elevation lower than 10 m in a circular region, queries on large regions or very complex regions and nearest neighbor queries.

The third stage is the *full-benchmark* and its goal is to see the performance of the system in the loading of a massive data set such as the AHN2 and its behavior in the execution of 30 queries which include, in addition to the *medium-benchmark* queries, extra large region queries and aggregated queries like computing the highest point in a province.

The fourth and last stage of the benchmark is the *scale-up-benchmark*, which was only defined and not yet executed, envisioned a test with very massive data sets with 20 trillion points and even more tested functionality. In Appendix A there is more information on the used data sets and the benchmark queries.

Several stages of the benchmark were executed for the various existing PCDMS's using data sets. We tested the blocks model with the native support of Oracle Spatial and Graph and PostgreSQL-PostGIS, in the latter the loading was done using PDAL. We tested flat table models in Oracle Spatial and Graph, PostgreSQL-PostGIS and MonetDB. They were compared with a file-based PCDMS based on LAStools with both LAZ and LAS formats.

All systems run on the same platform, a HP DL380p Gen8 server with 128 GB RAM and 2×8 Intel Xeon processors E5-2690 at 2.9 GHz, RHEL 6 as operative system and different disks directly attached including 400 GB SSD, 5 TB SAS 15 K rpm in RAID 5 configuration (internal), and 2×41 TB SATA 7200 rpm in RAID 5 configuration (in Yotta disk cabinet).

We also tested an implementation of the flat table model with Oracle Spatial and Graph in an Oracle Exadata X4-2 hardware (Oracle Exadata Database Machine X4-2 2015), an Oracle SUN hardware designed for the Oracle database with an advanced architecture including hardware hybrid columnar compression (HCC), massive parallel smart scans/predicate filtering and less data transfer.

3.1 Storage, Preparation and Loading

In Appendix B the reader can find the tables with the loading results of the execution of the *medium-benchmark* and the *full-benchmark*. Compared to flat table systems the blocks model DBMSs are faster and compress the data better during preparation and loading. Flat table systems enable modifications of the table definition or the data values as in any database table which is more complicated in the blocks model. For both, the integration with other types of data is straight and all the key features of DBMSs are present, i.e. data interface through the SQL language, remote access and advanced security.

LAStools prepares data faster than any DBMS since no loading is needed, only resorting and indexing. The storage requirements of the LAZ format are lower than those of the DBMSs, but with its fixed file format the data model loses flexibility as one is restricted to the specified format. For example the standard LAS format allows only one byte for user data. In addition, when dealing with large data sets (stored in many LAS/LAZ files), LAStools needs to be aided by a DBMS to maintain its performance. The DBMS is used to store the extent of the files and it is used as

a pre-filtering stage in the retrieval operations. Storage requirements and speed of loading in Oracle Exadata is comparable to LAsTools. However, the used hardware is different.

3.2 Querying

In Appendix C the reader can find the tables with the retrieval/querying results of the execution of the *medium-benchmark* and the *full-benchmark*. As previously stated, data retrieval was tested by selecting points within rectangles, circular areas, simple and complex polygons, and nearest neighbor and aggregated queries. The DBMS results were obtained by CTAS (create table as select) queries, while the LAsTools results were obtained by storing the selection in an output LAS file.

Blocks model DBMSs perform well on larger areas or complex polygons, independent of the number of points. However, the blocks model adds an overhead which affects simple queries most. The flat table model DBMSs perform well for simple queries on small point clouds; for large point clouds the native indexing methods become inefficient. Alternative flat table models based on space filling curves provided nearly constant response times, independent of number of points (Martinez-Rubi et al. 2015). The file-based solution using LAsTools performs best for simple queries. The queries to LAZ data are slower than to LAS data because of the need of uncompressing the data. In addition, massive point clouds require an external DBMS to maintain good performance.

Data retrieval in Oracle Exadata is comparable to LAsTools but complex queries run significantly better because of massive parallelization. However and as previously stated, the systems run in different hardware.

4 New Benchmark Execution

From our previous benchmark execution we could conclude that when a file-based solution fulfills the user requirements it is effective to use that solution. However, if more flexibility and/or more advanced functionality are required DBMSs offer a good alternative. At the time of the previous benchmark execution (second half 2014), the *full-benchmark* stage (loading 640 billion points and executing 30 queries) could only be executed with decent performance in LAsTools, Oracle Exadata and PostgreSQL-PostGIS (results of the latter were not presented), thus limiting the choice of PCDMS. Moreover, most systems miss two important features. Firstly, though data preparation and loading can be easily parallelized with additional tools, only MonetDB supports parallel processing. For the DBMS for which we applied out-of-the-core parallel queries for data retrieval the performance improved significantly.

The developer teams of Oracle Spatial and Graph, PDAL and MonetDB have been actively improving their systems and these are reaching a more mature and robust state. In order to assess the recent improvements a new execution of the benchmark is required. In the previous execution of the *full-benchmark* stage we compared PCDMS's running in different systems. In this paper we present the results of a new *full-benchmark* execution of LAStools with two file formats, LAS and LAZ, and Oracle Spatial and Graph interfaced with PDAL and the new laz-perf compression, but all running in the same hardware, the HP DL380p Gen8 previously described. Another *full-benchmark* run also considering MonetDB and its the new developments and PostgreSQL-PostGIS with the latests PDAL improvement is planned for the near future.

4.1 Storage, Preparation and Loading

Contrary to the previous benchmark execution we used a cleaned version of the AHN2 data set where duplicate and erroneous points are deleted. The cleaned data set is distributed in 37,588 LAZ files and contains 638 billion points instead of the raw format with 640 billions points.

In the case of LAStools we prepare (resort and index) the data and, as previously explained, we require a DBMS which contains the spatial extent of the files. In the case of Oracle Spatial and Graph we use the external third-party PDAL library to load the data from the LAZ files into Oracle using the blocks model. Note that in both cases the loading is not natively parallel but can be easily done since each file can be loaded independently. In Table 1 we present the results of the loading in the tested systems, in all the cases we used 16 simultaneous processes.

The loading of the data is faster in LAStools. However, the required time for in Oracle has enormously decreased with PDAL. The speed is now 18,854 million points per hour while with the loading method used before it was around 431 million points per second.

In storage terms we can observe the laz-perf compression with PDAL is almost as efficient as the LAZ, thus there is not much penalty anymore in storage terms by loading the data in a DBMS. The estimation of storage terms in Oracle Spatial and Graph without using laz-perf compression is of 20 Tb for the AHN2. Also note

Table 1 *Full-benchmark* loading results for LAStools with both LAS and LAZ and Oracle Spatial and Graph with PDAL. In all cases 16 processes were used

System	LAStools/LAS	LAStools/LAZ	Oracle/PDAL
Total load time (h)	22:54	19:41	33:53
Total size (Tb)	12.18	1.66	2.07
#points	638,609,393,087	638,690,670,516	638,860,225,350

that the number of points after loading differs for the various PCDMS's, this is due to several LAsTools processes crashing for files large than 60 million points (issue reported to developer).

4.2 Querying

We use `lasmerge` and `lasclip` commands to perform the queries in the LAsTools PCDMS's and we use PDAL approach for performing the queries in Oracle/PDAL approach. In this new benchmark, the output was store in LAS files (also in case of Oracle/PDAL). Note that we can not use native SQL/SDO_PC Oracle features because of the compatibility issue between the PDAL format and the native Oracle Spatial and Graph format.

A subset of the *full-benchmark* queries was executed with the tested systems, they consist on selecting the points in rectangular, circular or polygonal regions. Both systems currently have limited functionality and the nearest neighbor and aggregated queries could not be executed.

In Table 2 we present the result of the executed queries in the tested systems. For each query we detail the number of returned points as well as the response time. The queries are executed in sequential order and each query is executed twice, the value in the table is the response time of the second execution (as this value is usually a bit more stable). Both LAsTools and PDAL processes are single-core processes.

From the results of Table 2 we observe that LAsTools with LAS offers the best performance. However, it deals with uncompressed data which is not the case in the other PCDMS's. Hence, the most fair comparison is between LAsTools/LAZ and Oracle/PDAL and in this one LAsTools/LAZ is faster in most of the cases but the response times are in the same order of magnitude. We noticed that the first execution of the queries (not shown in the table) in Oracle/PDAL was, in general, about 30 % slower that the second execution while this difference was almost not noticeable in the LAsTools solutions. Moreover, a closer look at the table reveals some interesting issues:

- The number of returned points differ in Oracle/PDAL and LAsTools. There are two issues that may cause this difference: (i) PDAL uses GEOS library for spatial operations which is different compared to the library used in LAsTools. Concretely, they have different rules for points on the edges. (ii) Some of the input files in LAsTools failed in importing (for being too large) and this may cause that some points are missing.
- Query #24 (long diagonal area) in Oracle/PDAL has less points and requires much more time that LAsTools solutions. The execution of this query is severely damaged by the high amount of affected blocks in this query which produces that the PDAL process performs wrongly. This issue is under investigation.

Table 2 Full benchmark query results of LAStools with both LAS and LAZ and Oracle/PDAL

Query	LAStools/LAS			LAStools/LAZ			Oracle/PDAL		
	#points	Time (s)	#pts/s	#points	Time (s)	#pts/s	#points	Time (s)	#pts/s
1	74850	0.03	2495000	74850	0.11	680455	74818	0.25	299272
2	717959	0.08	8974488	717959	0.42	1709426	717869	0.97	740071
3	34691	0.02	1734550	34691	0.09	385456	34667	0.23	150726
4	563037	0.09	6255967	563037	0.43	1309388	563013	1.16	485356
5	182861	0.15	1219073	182861	0.36	507947	182861	0.57	320809
6	460096	1.40	328640	460096	1.79	257037	387134	1.26	307249
7	45811	0.24	190879	45811	0.68	67369	45813	1.49	30747
8	2365590	3.72	635911	2365590	5.52	428549	2273056	6.86	331349
9	620390	2.02	307124	620390	3.34	185746	620392	4.2	147712
13	896803	280.52	3197	896803	330.96	2710	896802	581.37	1543
14	765961	87.38	8766	765961	133.65	5731	765951	874.35	876
15	3992098	0.38	10505521	3992098	1.60	2495061	3991903	4.29	930514
17	2202833	0.21	10489681	2202833	0.94	2343439	2201079	3.33	660985
21	382284	0.48	796425	382284	9.15	41780	378454	12.40	30520
24	2468151	238.31	10357	2468151	512.66	4814	787912	5515.86	143
27	27453	0.05	549060	27453	0.14	196093	27452	0.36	76256

- The number of points in LAStools solutions for queries #6 and #8 are higher than in Oracle/PDAL. The spatial region of such queries have holes and LAStools clipping methods ignore holes. This issue has already been reported.

This comparison demonstrates that a DBMS-based PCDMS can be almost as efficient as a specific and tailor-made file-based solution and, in theory, with all the benefits of DBMS systems. Obviously, much work is still to be done in the lines of (i) better cross-compatibility between PDAL and Oracle Spatial and Graph in order to actually exploit all the DBMS features and of (ii) a better exploitation of multi-process architectures to solve retrieval operations.

5 Point Cloud Web Services

In this section we describe two web services based on point cloud data and we analyze their usage based on their generated log files.

5.1 *AHN Viewer*

The first service that we analyze is based on ESRI ArcGIS and provides a 2D map of the Netherlands in which the user can add layers with elevation data derived from the various AHN data sets (AHN2 and also some parts of the upcoming AHN3) (AHN viewer 2015). The data is structured in a multi-resolution imaging tiling structure.

We analyze the log file of the year 2012 produced by the server that provides the tiles for the AHN2 data set in order to identify the usage pattern of this service. Each row in the log file contains one connection to the server, i.e. the IP address of the connection, the date and the four coordinates of the bounding box of tile that was requested.

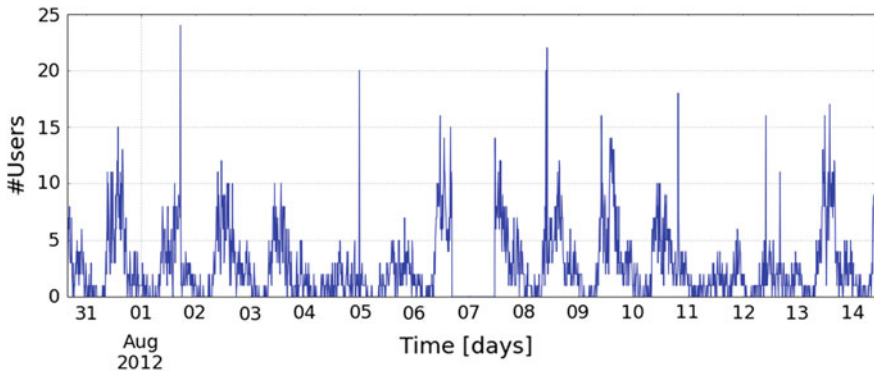
The analysis of the log file was done with Python/pandas (Python 2015). There are 3,788,770 rows/connections in the file from 3,353 different IP's/users. From the coordinates of the bounding box we can compute the area for each requested tile. In Table 3 we present the number of requested tiles for each level of the multi-resolution structure where the level is derived from the area of the requested tile.

We focus in the period of time where the service was most used, this is the first two weeks of August 2012. In Fig. 2 we depict the number of simultaneous users of the service in that period. We can observe that such a service can get up to 25 simultaneous users. The number of simultaneous users is defined as number of different IP's addresses in 10 min intervals.

When analyzing the typical usage pattern of a single user we detected that, in general, a user starts with requesting a tile with the highest area, which is the tile with all Netherlands, and then zooms in and out to smaller areas. A typical user can

Table 3 Number of requests per level which is derived from the area of each requested tile

Level	Area (km ²)	#requests
0	65536.000	7069
1	16384.000	203053
2	4096.000	205947
3	1024.000	214446
4	256.000	239431
5	64.000	286880
6	16.000	326066
7	4.000	652608
8	1.000	509057
9	0.250	483089
10	0.062	366020
11	0.016	210958

**Fig. 2** Number of simultaneous users using the service during August 2012

make up to 35 tile requests per second, thus the response time of each request should be below 28 ms (in order to stay within 1 s response for all requests).

5.2 AHN2 3D Web Viewer and Download Tool

The second service that we analyze is the AHN2 3D web viewer and download tool (AHN2 3D viewer 2015) developed by the author team. Several renderers exploiting WebGL have become available for the point cloud web visualization such as plasio (<http://plas.io/>) or potree (<http://potree.org/>). In our project we extended potree to be able to visualize massive point clouds such as the complete AHN2 data set.

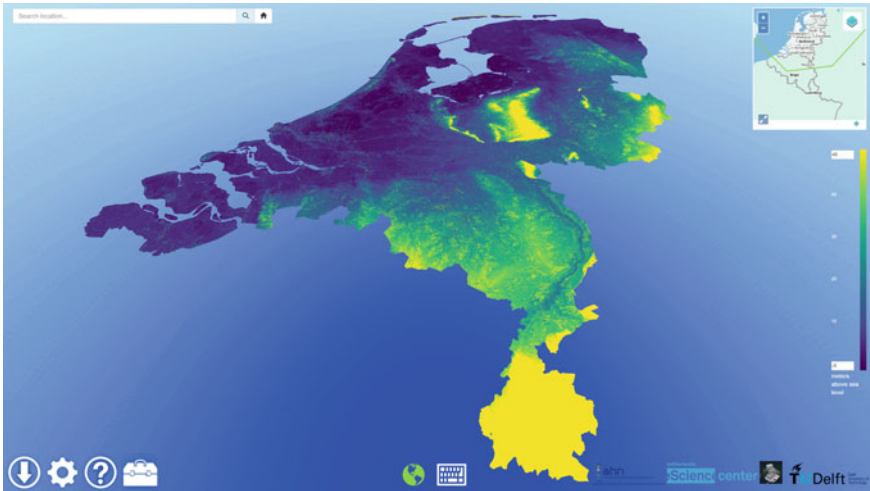


Fig. 3 Snapshot of the AHN2 3D web viewer and download tool

We have developed a publicly available web service with free and open-source tools for the 3D visualization of AHN2. In addition the service also has a multi-resolution download tool, a search bar, a measurement toolkit, a 2D overview map with field of view depiction, a demo mode and the tuning of the visualization parameters. In Fig. 3 we show an snapshot of the web service.

Potree uses a specific data structure, a multi-resolution octree, that requires the data set to be reorganized. This reorganization is very time consuming for large data sets. We have created a free and open-source tool that divides the generation of a massive octree into the generation of smaller octrees which can later be combined. The small octrees generation tasks are independent and can be distributed on different systems. Their combination is possible because the extent of all the nodes of the octrees is known and fixed.

The creation of the whole octree for AHN2 took around 15 days with processing distributed in different machines and processes. In Table 4 we show an overview of the octree structure with the number of LAZ files and points per level, and the ratios of these for consecutive levels. Due to the flat nature (in elevation) of such country-wide point clouds (and more in the Netherlands where the highest point is at 380 m) this octree structure is more similar to a quadtree structure (the ratios are in most cases similar to 4). Note that the total number of points is not 638 billion points, 6.5% of the points are dropped because of the way the tree is created. This can be decreased by tuning the tree creation parameters but the processing time would increase as well. Future work is also expected on trees that contain all the points.

With Python/pandas (Python 2015) we analyze the usage of the AHN2 web viewer and download tool by analyzing a log file with activity information of the server that contains the octree structure around the day of its official release

Table 4 Overview of the AHN2 octree

Level	#files	Files_fact	#points	Points_fact
0	1		34045	
1	4	4,00	134786	3,96
2	14	3,50	541973	4,02
3	41	2,93	2205484	4,07
4	143	3,49	8833283	4,01
5	499	3,49	36081908	4,08
6	1804	3,62	155411383	4,31
7	6767	3,75	668597511	4,30
8	25939	3,83	2834989373	4,24
9	101057	3,90	11355433955	4,01
10	398423	3,94	39911483676	3,51
11	1584598	3,98	112993998398	2,83
12	6671815	4,21	259014500658	2,29
13	29442790	4,41	170207571211	0,66
Total	38233895		597189817644	

Table 5 Number of requests per level

Level	#requests
13	1036
12	635
11	607
10	690
9	904
8	1001
7	993
6	993
5	1120
4	2088
3	2656
2	1510
1	455
0	153

announcement, on the 28th July 2015. The log file contains the requests to the data server, each line contains one connection to the server, i.e. the IP address of the connection, the date, the file that was requested, its size and the number of points in the file. Each LAZ file contains a node of an octree and each node contains points in cubic areas with certain density depending on the node level, from the file name we can derive the level of the node/file and the location.

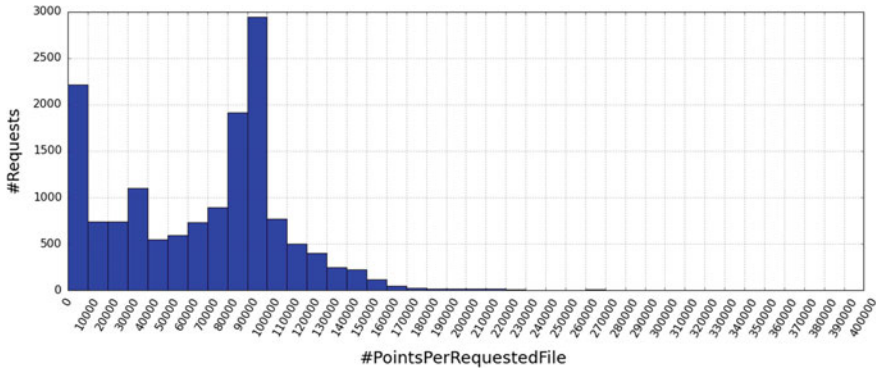


Fig. 4 Histogram for the number of points of the requested files

In the three days which are contained in the log file a total of 54 IP’s/users used the service (with 7 simultaneous users). In Table 5 we show the number of requests for each level, we observe that most of the requested files are of levels 3 and 4. In Fig. 4 we show an histogram for the number of points of the requested files. We see that the majority of files have a number of points of less than 10.000 or between 80.000 and 100.000.

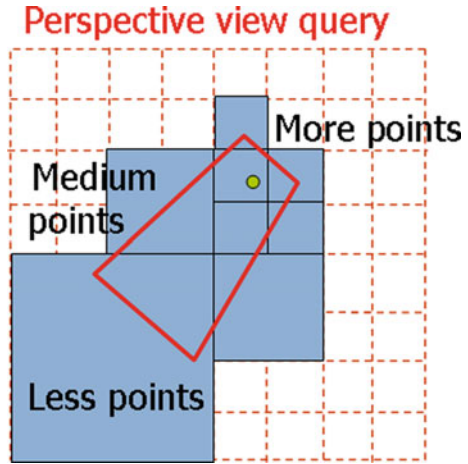
Like on the 2D web viewer, we observed that a typical user starts with requesting level 0 (all NL extent), then zooms in and out to deeper levels. For a smooth visualization experience the user requests up to 18 files per second, i.e. the response time of each request should be below 55 ms.

6 Executable Benchmark Extension

Web services such as the previously described are becoming very popular. Currently for these services the data is reorganized in multi-resolution data structures such as the octree used in the AHN2 3D web viewer and download tool. These services usually have simultaneous users. The typical user requests are to nodes from different level-of-detail’s (LoD’s) of the data structures. From our previous analyses we extracted an indication of their desired per-user performance to provide a smooth visualization experience.

Due to the lack of the required functionality and/or decent performance of the more generic PCDMS’s introduced in Sect. 2 these services currently rely on specific solutions for their data management layers. In order to change this trend new developments are required. In addition, the executable benchmark should be extended to reflect the needs of these services.

Fig. 5 Perspective view query

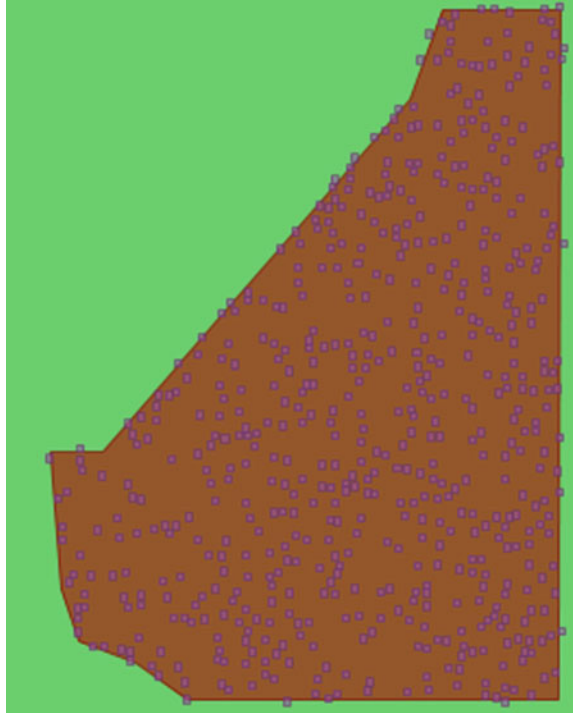


6.1 LoD Queries

We propose to extend the executable benchmark with LoD queries, i.e. queries where we select a representative sub-sample of the points in the queried region. For example get a representative 1 % of the points of the whole point cloud (the Netherlands in our case) or queries such as the illustrated in Fig. 5 where a perspective view query is depicted, i.e. several rectangular regions with different detail/density of points are queried.

However, note that the current web services approaches and their related data structures have a major drawback, they use a discrete number of LoDs and the viewer may notice the difference in the point density between neighboring tiles/nodes at different levels. In (van Oosterom et al. 2015) we present suggestions to solve this issue based on the vario-scale LoD research (Oosterom and Meijers 2014).

As previously stated most of the tested PCDMS's have no native support for multi-resolution/LoD data structures for point cloud data, only Oracle Spatial and Graph (see Usage notes of the SDO_PC_PKG. INIT method in the Oracle SDO_PC_PKG documentation Oracle Database Online Documentation 12c Release 1 (12.1) (2014)). Hence, at the moment in order to test the LoD performance in most of the PCDMS's hybrid solutions or specifically developed solutions have to be developed and used.

Fig. 6 Multi-user queries

6.2 *Multi-user Queries*

The second extension that we propose is to add multi-user queries, i.e. how the system reacts when multiple users are using it. Ideally the multi-user queries should be combined with LoD queries but due to the lack of LoD support on most of the PCDMS's this is currently complicated. Our proposed starting point is to define a set of similar queries and to simulate a pool of several users that simultaneously query the set. For example, in Fig. 6 we show a set of equal-area rectangular queries spread over the extent of the data set with 23 billion points (subset of AHN2) used in the *medium-benchmark* stage of the current executable benchmark. In this test there should be several executions with different number of users and for each execution we determine which is the average query time. In addition to these synthetically generated queries, it is also possible to 're-run' the queries from the log files (at various speeds).

The results of this test could be used to determine for which number of users the performance decreases to an unacceptable level, thus giving an indication of when scaling-out (horizontal scalability) is required. Cloud solutions are perfect candidates for such scaling operations. As part of our research, and together with the Microsoft Azure research team, such solutions are currently being investigated.

7 Conclusions and Future Work

In our previous work we defined a conceptual benchmark from which we derived an executable benchmark that was executed for various PCDMS's. In this paper, and motivated by the recent developments in PDAL, we have re-executed the *full-benchmark* stage with LAStools and Oracle/PDAL to find out that DBMS solutions for PCDMS's can be a real and attractive alternative to specific file-based solutions such as the efficient LAS/LAZ file format with the LAStools. However, several improvements are still required in the Oracle/PDAL approach: exploitation of multi-process architectures for faster data retrieval operations (also true for LAStools) and full compatibility between Oracle Spatial and Graph and PDAL in order to fully benefit from all the features of DBMS systems. In the future we will perform more *full-benchmark* executions with other systems such as MonetDB, which point cloud support has also been recently improved, PostgreSQL or the native Oracle Spatial and Graph point cloud that is also being currently improved.

We described two web services based on point cloud data. The first one is based on ESRI ArcGIS and provides a 2D visualization while the second one, which is developed by the authors team, delivers a novel 3D web visualization tool and a download tool. We analyzed the usage patterns of such services as well as their used data structures and extracted which performance is required by the systems feeding them.

Due to the lack of efficient LoD support in the more generic PCDMS's, usually the web services rely on specific solutions for their data management layer. New developments are required in the PCDMS's in order to be usable for such applications. We proposed an extension of the executable benchmark to take into account the needs of the web services based on point clouds, more concretely we added LoD and multi-user queries. In the future, and as the required functionality becomes available, the extended executable benchmark will be re-executed.

Several activities for point cloud standardization have been recently initiated. Several initiatives, in which the authors team participate, have been recently started with focus on standardization for point cloud data and their usage. The OGC (Open Geospatial Consortium) has created a Point Cloud Domain Working Group (DWG) to address issues on the interoperability when sharing and processing point cloud data. There have been discussions on a possible cooperation with ISO TC211, the ISO technical committee for Geographic information/Geomatics. In parallel the OSGEO PointDown initiative was started with the aim of creating an overview on

the usage of point cloud data over the web in order to provide a generic service definition.

Acknowledgments We thank all the members of the project Massive Point Clouds for eSciences, which is supported in part by the Netherlands eScience Center under project code 027.012.101. Also special thanks for their assistance to Mike Horhammer, Daniel Geringer, Siva Ravada (all Oracle), Markus Schütz (developer of potree), Martin Isenburg (developer of LAsTools), and to Howard Butler, Andrew Bell and the rest of PDAL developers.

Appendix

Appendix A: Executable Benchmark Data Sets and Queries

Data sets

Tables 6 and 7 contains information on the used data sets in the executable benchmark and their usage in the different stages. Figure 7 shows the extent of the used data sets.

Table 6 Data sets name, benchmarks in which they are used, number of files and disk size

Data set name	Benchmark	Points	Files	Format	Size (GB)
20M	Mini/medium	20,165,862	1	LAS	0.4
210M	Medium	210,631,597	17	LAS	4.0
2201M	Medium	2,201,135,689	153	LAS	42.0
23090M	Medium	23,090,482,455	1,492	LAS	440.4
639478M	Full	639,478,217,460	60,185	LAZ	986.7
638860Mc	Full and clean	638,860,225,350	37,588	LAZ	2534.2

Table 7 Data sets area and description

Data set name	Area (km ²)	Description
20M	1.25	TU Delft campus
210M	11.25	Major part of Delft city
2201M	125	City of Delft and surroundings
23090M	2,00	Major part of Zuid-Holland province
639478M	40,000	The Netherlands (full AHN2)
638860Mc	40,000	The Netherlands (full cleaned AHN2)

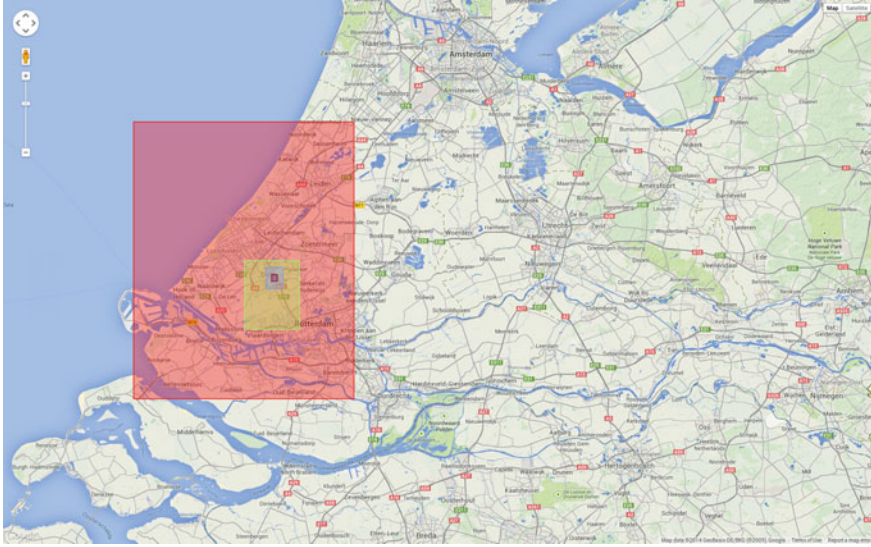


Fig. 7 Approximated projection of the extents of the used datasets in Google Maps: *Purple area* is for 20M dataset, *cyan area* is for 210M dataset, *green area* is for 2201M dataset and *red area* is for 23090M dataset

Note that for the full AHN2 we include two versions of the data set. The first one, *639478M* was used in our previous *full-benchmark* execution while the second one, *638860Mc* is the one used for in the new execution and does not contain erroneous and duplicate points that are found in the first version. Also note the difference in the data sets sizes. This is due to the fact that for the cleaning process that was required to generate the second (cleaned) version of the full AHN2 data set the points in the files needed to be resorted and that affected dramatically the compression performance of LAZ. In the first version the files were separated according to their nature, in object and terrain files, and that improved the compressor performance. However, as part of the cleaning processes, these files were joined and the compression ratio was affected. The compression ratio improves when the data is resorted by LAStools as part of the benchmark execution (see Table 1) but even in that case it is not optimal due to the mixing of point cloud from different nature (total size 1,66 Tb).

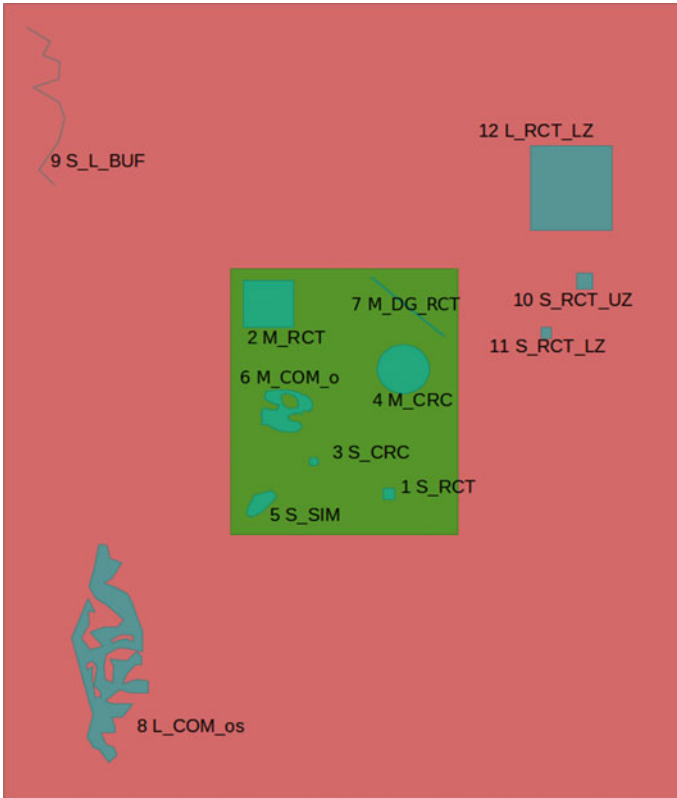


Fig. 8 Queries used in the medium benchmark (up to 210M extent)

Queries

Figures 8 and 9 show the first 20 query geometries that were used in the several benchmark stages. Table 8 describes all of them, their ID, the number of points in the boundary of the query geometry (*Pnts*) and the test data set name in which the query geometry is located.

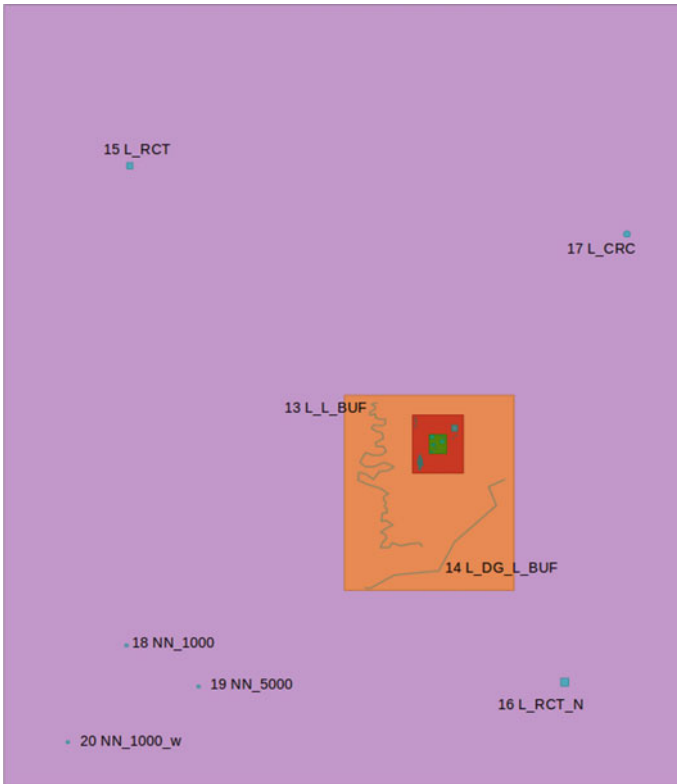


Fig. 9 Queries used in the medium benchmark

Appendix B: Executable Benchmark Loading Results

Table 9 contains the loading details of the *medium-benchmark* execution for various PCDMS's and data sets. The results of LAs tools are when using LAS (instead of LAZ). The PCDMS using the blocks model were using the compression available at that time (second half 2014) and with optimal block sizes previously computed. Note that all the Oracle Exadata approaches (*oe** on the table) run in a different hardware than the other approaches.

Table 8 Description of the different queries

ID	Key	Pnts	Test	Area (km ²)	Description
1	S_RCT	5	20M	0.0027	Small axis-aligned rectangle
2	M_RCT	5	20M	0.0495	Medium axis-aligned rectangle
3	S_CRC	97	20M	0.0013	Small circle, radius 20 m
4	M_CRC	379	20M	0.0415	Medium circle, radius 115 m
5	S_SIM	9	20M	0.0088	Small, simple polygon
6	M_COM_o	792	20M	0.0252	Medium, complex polygon, 1 hole
7	M_DG_RCT	5	20M	0.0027	Medium, narrow, diagonal rectangular area
8	L_COM_os	89	210M	0.1341	Large, complex polygon, 2 holes
9	S_L_BUF	94	210M	0.0213	Small polygon (10 m buffer in line of 11 pts)
10	S_RCT_UZ	5	210M	0.0021	Small axis-aligned rectangle, cut in max. z
11	S_RCT_LZ	5	210M	0.0051	Small axis-aligned rectangle, cut in min. z
12	L_RCT_LZ	5	210M	0.1419	Large axis-aligned rectangle, cut in min. z
13	L_L_BUF	237	2201M	0.0475	Large polygon (1 m buffer in line of 61 pts)
14	L_DG_L_BUF	39	2201M	0.0499	Large polygon (2 m buffer in diag. line of 8 pts)
15	L_RCT	5	23090M	0.2342	Large axis-aligned rectangle
16	L_RCT_N	5	23090M	0.1366	Large axis-aligned rectangle in empty area
17	L_CRC	93	23090M	0.1256	Large circle
18	NN_1000	1	23090M	0.0000	Point for NN query, 1000 nearest points
19	NN_5000	1	23090M	0.0000	Point for NN query, 5000 nearest points
20	NN_1000_w	1	23090M	0.0000	Point in water for NN query, 1000 nearest points
21	L_NAR_RCT	5	639478M	0.0236	Large narrow axis-aligned rectangle, points
22	L_NAR_DG_RCT	5	639478M	0.6399	Large narrow diagonal rectangle, min(Z)

(continued)

Table 8 (continued)

ID	Key	Pnts	Test	Area (km ²)	Description
23	XL_NAR_DG_RCT	5	639478M	42.5573	Very large narrow diagonal rectangle, max(Z)
24	L_NAR_DG_RCT_2	5	639478M	0.1208	Large narrow diagonal rectangle, points
25	PROV_DG_RCT	5	639478M	3022.0427	Provincial size diagonal rectangle, min(Z)
26	MUNI_RCT	5	639478M	236.7744	Municipality size diagonal rectangle, max(Z)
27	STREET_DG_RCT	5	639478M	0.0016	Street size diagonal rectangle, points
28	VAAALS	1565	639478M	23.8972	Municipality Vaals, avg(Z)
29	MONTERLAND	1565	639478M	106.6520	Municipality Montferland, avg(Z)
30	WESTERVELD	1569	639478M	282.7473	Municipality Westerveld, avg(Z)

Table 9 Times and sizes of the data loading procedure for the different PCDMSs and datasets. The names of approaches encode the PCDMS name (o for Oracle, p for PostgreSQL, etc.), flat or blocked model (f and b, respectively), and the dataset name. For example *ob2201M* stands for the dataset 2201M loaded in the Oracle blocks PCDMS

Approach	Time (s)		Load	Close	Size (MB)		Points	Points/s
	Total	Init.			Total	Index		
pf20M	44.07	2.00	13.86	28.21	1558.71	551.83	20,165,862	457,587
pf210M	771.63	2.09	71.44	698.10	16249.24	5762.20	210,631,597	272,970
pf2201M	9897.37	1.15	722.91	9173.31	169776.13	60214.46	2,201,135,689	222,396
pf23090M	95014.05	3.64	8745.90	86264.51	1780963.91	631663.71	23,090,482,455	243,022
of20M	128.43	0.51	123.92	4.00	879.50	453.85	20165862	157018
of210M	275.72	0.32	230.01	45.39	9124.50	4739.94	210,6315,97	763,933
of2201M	1805.68	0.31	1228.81	576.56	95825.00	49533.74	2,201,135,689	1,219,007
of23090M	15825.53	0.14	9226.37	6599.02	997062.50	519621.48	23,090,482,455	1,459,065
mf20M	7.15	1.14	3.70	2.31	475.78	14.09	20,165,862	2,820,400
mf210M	29.15	1.13	16.63	11.39	4888.05	66.95	210,631,597	7,225,784
mf2201M	304.14	4.91	198.76	100.47	50661.30	281.17	2,201,135,689	7,237,245
mf23090M	8490.90	0.93	5466.99	3022.98	529448.70	949.37	23,090,482,455	2,719,439
pb20M	153.41	22.03	130.53	0.85	101.77	0.69	20,165,862	131,451
pb210M	129.14	0.81	121.00	7.33	1009.13	5.18	210,631,597	1,631,033
pb2201M	754.22	0.86	687.49	65.87	10245.61	53.05	2,201,135,689	2,918,427
pb23090M	12263.05	0.87	7450.10	4812.08	106781.48	552.77	23,090,482,455	1,882,931
ob20M	296.22	0.35	228.17	67.70	226.50	0.20	20,165,862	68,077
ob210M	1246.87	0.25	557.70	688.92	2244.50	1.44	210,631,597	168,928
ob2201M	16737.02	0.86	7613.39	9122.77	21220.50	13.25	2,201,135,723	131,513

(continued)

Table 9 (continued)

Approach	Time (s)		Init.	Load	Close	Size (MB)		Points	Points/s
	Total					Total	Index		
ob23090M	192612.07		0.31	96148.06	96463.70	220085.50	165.55	23,090,482,953	119,881
lt20M	9.49		0.03	9.46	0.00	384.65	0.02	20,165,862	2,124,959
lt210M	30.07		0.02	28.68	1.37	4021.37	3.88	210,631,597	7,004,709
lt2201M	218.06		0.02	216.18	1.86	41992.78	9.40	2,201,135,689	10,094,174
lt23090M	2129.30		0.03	2116.64	12.63	440484.48	68.04	23,090,482,455	10,844,166
oenc21090M	508.71					537600.00	0.00	21,090,482,455	41,458,753
oeql21090M	619.14					218504.00	0.00	21,090,482,455	34,064,157
oeqh21090M	928.84					94240.00	0.00	21,090,482,455	22,706,269
oeal21090M	1149.55					93992.00	0.00	21,090,482,455	18,346,729
oeah21090M	1767.53					59096.00	0.00	21,090,482,455	11,932,177

Table 10 *Full-benchmark* loading results for the LAStools and Oracle Exadata PCDMSs

System	LAStools	Oracle exadata
Total load time (h)	22:54	4:39
Total size (Tb)	12,181	2,240
#points	638,609,393,087	639,478,217,460

Table 10 contains the loading details of the *full-benchmark* execution that was done with LAStools and Oracle Exadata PCDMS's. Note that for this execution the *6394784M* data set was used, i.e. the AHN2 version with duplicate and erroneous points. For an in-deep analysis of these results we refer the reader to our previous work (van Oosterom et al. 2015).

Appendix C: Executable Benchmark Querying Results

Table 11 contains the number of returned points and the response times of the first seven queries for the different PCDMS's and data sets. Note that each query was executed twice, the numbers in the table are from the second execution, usually called hot query because of the fact that the PCDMS may be able to reuse cached data either by the PCDMS itself or the file system or the operative system (OS).

Table 12 contains the number of returned points and the response times of the execution of the 30 *full-benchmark* queries for the LAStools and Oracle Exadata PCDMS. Note that for LAStools two columns are given. The first one is when using a DBMS in a pre-filtering step for the queries and the other is without it. For an in-deep analysis of these results we refer the reader to our previous work (van Oosterom et al. 2015).

Table 11 Comparison of number of points returned and response times by the hot queries 1 to 7 in the different approaches

Approach	Number of points							Time (s)						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7
pf20M	74947	718131	34637	562919	182792	387134	45805	0.35	2.25	0.24	1.90	1.18	1.72	0.84
pf210M	74947	718131	34637	562919	182792	387135	45805	0.42	2.50	0.27	2.26	0.91	1.65	1.34
pf2201M	74947	718131	34637	562919	182792	387135	45805	4.92	19.03	2.90	18.28	9.37	17.71	10.16
pf23090M	74947	718131	34637	562919	182792	387134	45805	5.17	18.02	3.32	17.75	9.42	15.46	13.50
of20M	74872	718021	34691	563037	182861	387145	45813	0.24	0.37	0.28	1.85	0.75	1.32	1.32
of210M	74872	718021	34691	563037	182861	387145	45813	0.45	0.58	0.52	1.27	1.12	1.47	1.79
of2201M	74872	718021	34691	563037	182861	387145	45813	1.47	3.87	1.29	4.26	4.38	6.60	5.24
of23090M	74872	718021	34691	563037	182861	387145	45813	1.25	18.20	2.34	22.75	6.99	27.18	635.06
mf20M	74872	718021	34691	563037	182861	387134	45813	0.06	0.13	0.06	0.20	0.96	187.16	38.71
mf210M	74872	718021	34691	563037	182861	387135	45813	0.13	0.26	0.15	0.28	9.95	185.65	38.56
mf2201M	74872	718021	34691	563037	182861	387135	45813	0.64	0.90	0.64	0.77	10.37	186.38	39.17
mf23090M	74872	718021	34691	563037	182861	387134	45813	7.21	16.74	9.70	9.88	17.94	198.51	43.96
pb20M	74947	718131	34697	563108	182930	387142	45821	0.32	2.14	0.20	1.69	0.61	1.72	0.41
pb210M	74947	718131	34697	563108	182930	387142	45821	0.32	2.15	0.20	1.65	0.64	1.62	0.46
pb2201M	74947	718131	34697	563108	182930	387142	45821	0.31	2.19	0.21	1.67	0.67	1.63	0.41
pb23090M	74947	718131	34697	563108	182930	387142	45821	0.32	2.19	0.21	1.68	0.68	1.68	0.44
ob20M	74947	718131	34697	563110	182930	387145	45821	0.41	1.38	0.34	1.21	0.62	1.38	0.53
ob210M	74947	718131	34697	563110	182930	387145	45821	0.38	1.28	0.36	1.22	0.62	1.29	0.54
ob2201M	74947	718131	34697	563110	182930	387145	45821	0.39	1.36	0.36	1.23	0.60	1.33	0.50

(continued)

Table 11 (continued)

Approach	Number of points							Time (s)						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7
ob23090M	74947	718131	34697	563110	182930	387145	45821	0.40	1.30	0.34	1.21	0.60	1.40	0.53
lt20M	74840	717931	34695	563049	182849	460068	45834	0.04	0.12	0.03	0.09	0.66	1.48	0.51
lt210M	74840	717931	34695	563049	182849	460068	45834	0.06	0.14	0.05	0.14	0.67	1.51	0.51
lt2201M	74840	717931	34695	563049	182849	460068	45834	0.06	0.12	0.05	0.14	0.70	1.51	0.51
lt23090M	74840	717931	34695	563049	182849	460068	45834	0.05	0.15	0.05	0.14	0.68	1.50	0.52
oeqh21090M	40368	369352	19105	290456	132307	173927	9559	0.18	0.35	0.59	0.72	0.66	0.67	0.46

Table 12 Full benchmark query results of LAStools and Oracle Exadata. *Notes* (a.) Nearest neighbours queries (#18, #19 and #20) were not executed as functionality was not implemented, and (b.) Oracle Exadata query #25 was also re-run using an MBR instead of a geometry close to an MBR with and improved the time to 353.93 s with 3.6546E+10 selected points

Query	LAStools			Oracle Exadata	
	#points	Time (s)		#points	Time (s)
		DB	No DB		
1	74861	0.07	0.90	74863	0.48
2	718057	0.16	0.87	718070	0.79
3	34700	0.07	0.78	34675	1.22
4	563119	0.16	0.92	563082	1.69
5	182871	0.70	33.24	182875	1.43
6	460140	1.52	32.79	387201	1.29
7	45831	0.55	32.29	45815	1.71
8	2365925	3.72	36.21	2273469	2.86
9	620568	2.34	34.76	620719	1.58
10	2413	0.08	0.88	2434	0.40
11	591	0.05	0.84	591	0.44
12	343168	0.26	1.03	343171	0.60
13	897042	412.29	829.49	897359	23.34
14	765989	102.19	424.91	766029	15.05
15	3992330	0.49	1.39	3992290	2.23
16	0	0.04	0.75	0	0.00
17	2203066	0.32	1.18	2201280	2.51
21	382395	2.28	20.74	382335	0.95
22	12148049	142.09	1115.27	12147802	113.37
23	691422551	313.14	828.51	691422526	330.85
24	2468239	234.40	4261.77	2468367	393.21
25	–	–	–	3.5319×10^{10}	1193.10
26	2124162497	282.89	1124.04	2124162754	25.79
27	27443	0.13	923.59	27459	1.23
28	809097723	1553.87	1885.54	866802585	67.67
29	1509662615	3438.34	5697.79	1509662411	120.02
30	–	–	–	1.3443×10^{10}	3569.54

References

- Actueel Hoogtebestand Nederland (AHN). (2015). <http://www.ahn.nl/>.
 AHN viewer. (2015). <http://ahn.maps.arcgis.com/apps/webappviewer/index.html>.
 AHN2 3D viewer and download tool. (2015). <http://ahn2.pointclouds.nl/>.
 De Kleijn, M., De Hond, R., Martinez-Rubi, O., & Svetachov, P. (2015). A 3d geographic information system for mapping the via appia. Tech. rep. research memorandum (VU-FEWEB) 2015-1, VU University Amsterdam, Amsterdam, The Netherlands.

- Fiore, S., DANca, A., Palazzo, C., Foster, I., Williams, D., & Aloisio, G. (2013). Ophidia: Toward big data analytics for escience. *Procedia Computer Science*, 18, 2376 – 2385. <http://dx.doi.org/10.1016/j.procs.2013.05.409>, <http://www.sciencedirect.com/science/article/pii/S1877050913005528>. 2013 international conference on computational science.
- Group, T.P.G.D. (2014). PostgreSQL 9.3.5 Documentation. Tech. Rep. 9.3.5, The PostgreSQL Global Development Group.
- laz-perf. (2015). <https://github.com/verma/laz-perf.git>.
- Mapping the Via Appia in 3D. (2015). <http://mappingtheviaappia.nl/4dgis/>.
- Martinez-Rubi, O., Kersten, M., Goncalves, R., & Ivanova, M. (2014). A column-store meets the point clouds. FOSS4GEurope.
- Martinez-Rubi, O., van Oosterom, P., Gonçaves, R., Tijssen, T., Ivanova, M., Kersten, M. L., et al. (2015). Benchmarking and improving point cloud data management in monetdb. *SIGSPATIAL Special*, 6(2), 11–18. doi:10.1145/2744700.2744702. <http://doi.acm.org/10.1145/2744700.2744702>.
- Oracle Database Online Documentation 12c Release 1 (12.1): Spatial and Graph Developer's Guide / SDO_PC_PKG Package (Point Clouds). (2014). https://docs.oracle.com/database/121/SPATL/sdo_pc_pkg_ref.htm.
- Oracle Exadata Database Machine X4-2. (2015). <https://www.oracle.com/engineered-systems/exadata/database-machine-x4-2/index.html>.
- PDAL. (2015). <http://www.pdal.io/>.
- Python/pandas. (2015). <http://pandas.pydata.org/>.
- Ramsey, P. (2015). A PostgreSQL extension for storing point cloud (LIDAR) data. <https://github.com/pramsey/pointcloud>.
- Rapidlasso GmbH. (2015). <http://rapidlasso.com/>.
- Rapidlasso GmbH LASzip—free and lossless LiDAR compression. (2014). <http://www.laszip.org/>.
- Suijker, P. M., Alkemade, I., Kodde, M. P., & Nonhebel, A. E. (2014). User requirements massive point clouds for eSciences (WP1). Tech. rep., Delft University of Technology. <http://repository.tudelft.nl/view/ir/uuid%3A351e0d1e-f473-4651-bf15-8f9b29b7b800/>.
- van Oosterom, P., Martinez-Rubi, O., Ivanova, M., Horhammer, M., Geringer, D., Ravada, S., Tijssen, T., Kodde, M., & Gonçaves, R. (2015). Massive point cloud data management: Design, implementation and execution of a point cloud benchmark. *Computers and Graphics*, 49, 92–125. <http://dx.doi.org/10.1016/j.cag.2015.01.007>, <http://www.sciencedirect.com/science/article/pii/S0097849315000084>.
- van Oosterom, P., & Meijers, M. (2014). Vario-scale data structures supporting smooth zoom and progressive transfer of 2d and 3d data. *International Journal of Geographical Information Systems*, 28(3), 455–478.

Does a Finer Level of Detail of a 3D City Model Bring an Improvement for Estimating Shadows?

Filip Biljecki, Hugo Ledoux and Jantien Stoter

Abstract 3D city models are characterised by the level of detail (LOD), which indicates their spatio-semantic complexity. Modelling data in finer LODs results in visually appealing models and opens the door for more applications, but that is at the expense of increased costs of acquisition, and larger storage footprint. In this paper we investigate whether the improvement in the LOD of a 3D building model brings more accurate shadow predictions. The result is that in most cases the improvement is negligible. Hence, the higher cost of acquiring 3D models in finer LODs is not always justified. However, the exact performance is influenced by the architecture of a building. The paper also describes challenges in experiments such as this one. For instance, defining error metrics may not always be simple, and the big picture of errors should be considered, as the impact of errors ultimately depends on the intended use case. For example, an error of a certain magnitude in estimating the shadow may not significantly affect visualisation purposes, but the same error may considerably influence the estimation of the photovoltaic potential.

1 Introduction

Level of detail (LOD) is a fundamental concept in GIS and 3D city modelling: it indicates the data set's resolution, usability, and the degree of abstraction of reality (Biljecki et al. 2014b). The concept is borrowed from computer graphics, where

F. Biljecki (✉) · H. Ledoux · J. Stoter
3D Geoinformation, Delft University of Technology, Delft, The Netherlands
e-mail: f.biljecki@tudelft.nl

H. Ledoux
e-mail: h.ledoux@tudelft.nl

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



Fig. 1 Our research in a nutshell: the estimation of the shadow differs between the different levels of detail, and the accuracy of the prediction seems to increase with each LOD. However, this should be investigated numerically and it is not straightforward as it may appear

it is used to balance the computational complexity and the visualisation quality (fidelity). The latter, i.e. how similar the object looks like to the original one, can be assessed with error metrics such the deviation between the geometry of the original model and the geometry of its simplified counterpart (Luebke et al. 2003).

Such rationale can be applied to geoinformation science, in order to analyse whether it is worth to invest funds and computational resources in data set of a particular LOD. However, in 3D GIS, models are often used beyond visualisation for several purposes which significantly differ from each other—they have different behaviour, different requirements for LODs, and an outcome of different nature. For instance, 3D city models are used to estimate the real estate net internal area in m^2 (Boeters et al. 2015), insolation of buildings in kWh/year (Biljecki et al. 2015a), and noise in dB (Stoter et al. 2008), resulting in different kinds of errors, differently influenced by a particular LOD.

The aim of this paper is to investigate the influence of the resolution of spatial data on the quality of a particular spatial analysis: the estimation of shadows in an urban environment (Fig. 1). This use case is frequent in 3D GIS, and it is used in several application domains, for instance to assess the shadow impact of new buildings to their surroundings (Sect. 2).

For a number of LODs, we compute their errors when used for this purpose. However, conducting such analysis is burdened by difficulties, among others: (1) multi-LOD data sets are seldom available hindering such studies (Biljecki et al. 2015b); (2) available multi-LOD data sets normally contain real-world acquisition errors, inhibiting the comparison between LODs, since it is hard to isolate the error induced by the degree of abstraction from the error caused by the acquisition; (3) the outcome of each spatial analysis may not be unambiguously quantifiable, so the accuracy cannot be easily expressed (Biljecki et al. 2015c); and (4) from the implementation aspect, it is not always easy to automatically run a spatial operation for a large data set and extract results in a format that is suitable for error analysis. We mitigate these problems with an approach supported by a procedural modelling engine to automatically construct CityGML models in multiple LODs (Sect. 3). The results (Sect. 4) are relevant for practitioners because they can aid them to decide whether it is worth to acquire buildings in finer LODs considering that they come with an increased cost of acquisition and storage.

2 Related Work and Background

2.1 Influence of Data Granularity on Spatial Analysis

Studies on the influence of the scale, LOD, and resolution to the quality of a GIS operation are focused on 2D and to raster representations. Hengl (2006) discusses the importance of considering the resolution of a raster, and underlines that in GIS projects the resolution is usually selected without any scientific justification. Usery et al. (2004) determined the resolution effects on watershed modelling by resampling input rasters, and concluded that the resolution has a significant effect on the accuracy of the result. Booij (2005), Chaubey et al. (2005), Ling et al. (2008), and Pogson and Smith (2015) performed similar analyses with similar results.

In 3D GIS such studies are rare. A possible reason is that 3D city models cannot be simply “resampled” to easily obtain additional LODs for analysis.

Brasebin et al. (2012) tested the fitness for use of LOD1 and LOD2 models of two French data sets in the determination of the sky view factor (SVF). The result is that for 75 % of the analysed samples the improvement in accuracy is less than 2 %. Besuievsky et al. (2014) carry out a similar study with SVF focusing on the windows of buildings, with a few variants of high-detailed architectural models, and find a significant difference in the results.

Strzalka et al. (2011) investigate the use of 3D city models for forecasting energy demand, and argue that the suitability of an LOD depends on the configuration of buildings (i.e. for an area with predominantly flat roofs they suggest that an LOD1 suffices). However, experiments with other LODs are not documented.

Kibria et al. (2009) survey the perceptual value of a few LODs in spatial planning. It turns out that in some planning phases a finer LOD is actually undesirable.

2.2 The Role of Shadow in GIS

The estimation of shadows cast by buildings is a common topic in geoinformation science, and it is important for a number of application domains, such as thermal comfort (Hwang et al. 2011; Yezioro and Shaviv 1994) and solar energy (Carneiro and Golay 2009; Strzalka et al. 2012). This use case may be related to the visibility operation which is used for diverse purposes (Bartie et al. 2010; Peters et al. 2015). In this context, the estimation of shadow is a variation of the visibility problem, with the particularity that the sun is practically an infinitely distant point resulting in parallel rays, and that its position is variable.

Herbert and Chen (2015) underline that understanding shadow is crucial in urban planning, for assessing the effect new buildings induce on existing ones. They perform a survey among urban planners on the quality of the visualisation of the shadow based on different visual representations (e.g. level of transparency, 2D vs. 3D), and also include a query about the suitability of the LOD. However, only LOD1 was

given as an option, and the participants were given the opportunity to perceptually assess whether a 3D model of LOD1 is sufficient or not for such analysis.

Estimating shadows is important for determining solar envelopes, the subset of urban space with a certain period of assured access to sunshine (Knowles 2003). These are defined in terms of discrete numbers of hours of sun, but they can also be defined in terms of solar irradiation (Morello and Ratti 2009). Solar envelopes are to a degree enshrined in local and state laws, where residents are protected with the right to solar access (e.g. the façade of houses must receive a certain amount of hours of direct sunlight per day; see Den Haag 2011 and City of Mississauga 2012 for exemplary regulations).

In urban planning, shadows are not analysed only for buildings, for instance, Lange and Hehl-Lange (2005) study shadow casting from a proposed wind turbine, and Kumar et al. (1997) forecast occlusion by terrain.

Accounting for shadows is common when estimating the solar potential of buildings (Mardaljevic and Rylatt 2003; Carneiro and Golay 2009; Tooke et al. 2011; Redweik et al. 2013; Eicker et al. 2015; Nguyen and Pearce 2012; Hofierka and Zlocha 2012). Strzalka et al. (2012) develop a method to determine the shadow projected on a roof surface in order to account for the reduced yield of solar panels when estimating the feasibility of their installation. The method is designed as a visibility problem between small triangular partitions of a surface and the sun at various timestamps. The centroid of each partition is tested for visibility to the sun, and if the sun's ray intersects any of the other surfaces, the partition is marked as shaded at that timestamp.

In a related research, Alam et al. (2013) note that while LOD1 block models are sufficient for shading, higher LODs will inherently bring different results. However, this is not supported empirically, and in this paper we bridge this gap.

Jochem et al. (2009) develop a method for estimating the insolation of roofs from point clouds, taking into account shadows. While they deal with a single-LOD representation, they highlight that roof overhangs and chimneys may play an important factor in the magnitude of the shadow. This is important, because different LODs contain a different degree of detail, and in our research we investigate their claim.

Shadowing plays an important role in the research of Helbich et al. (2013). Their premise is that solar radiation is significantly capitalised in flat prices, and they consider the shadow effect in order to improve the accuracy of a hedonic house price model. They highlight that such simulations should be conducted for different positions of the sun because of the considerable differences in the results.

Finally, shadows are crucial in geo-visualisation to increase the quality of the visual communication (Appleton and Lovett 2003).

2.3 The Role of Shadow in Computer Graphics

Shadows have a longstanding underpinning in computer graphics where they play a significant role, as they enhance the realism of the scene and provide cues of spatial

relations such as depth (Williams 1978; Woo et al. 1990). As a result, many algorithms have been developed to estimate shadows and enhance realism, e.g. recursive ray tracing (Whitted 1980). Furthermore, many other computer graphics algorithms are closely related to this topic and frequently applied, e.g. the determination of shaded portion of the roof when estimating the insolation may be considered as a ray-triangle intersection problem (Möller and Trumbore 1997). We consider such algorithms in the design of our experiment.

3 Methodology

Our methodology for estimating whether finer LODs bring improvements in the estimation of shadows consists of the following stages:

1. Producing multi-LOD data: procedural generation of 3D city models (in CityGML).
2. Shadowing: rendering shadows from the models, and obtaining the shadow in a GIS form for analysis. We consider the shadow a building casts on the ground.
3. Analysis: quantification of shadows, and measuring their error for each LOD.

3.1 Source of Data and Considered LODs

In our approach, as a source of multi-LOD data we use procedurally modelled 3D models stored in CityGML in multiple LODs. The models have been generated by Random3Dcity, an open-source project of Biljecki et al. (2014a). The advantage of procedurally generated data is that they are not burdened by acquisition errors, and that a large number of 3D models can be obtained easily. Synthetic data have been previously diversely used in GIS (see, among others, Li et al. 2000 and Burnicki et al. 2007).

In principle, for our tests we use the traditional CityGML LODs (Gröger and Plümer 2012). However, the engine generates multi-LOD data based on a specification of Biljecki et al. (2016), which refines these LODs, providing us with a larger number of representations to benchmark. We use the following representations. LOD1.1 is a block model obtained with extrusion to a single height, LOD1.2 mandates smaller building parts (e.g. alcoves), and in LOD1.3 each building part has its own height (a building does not have a single height as in 1.1/1.2). LOD2.0 is similar to 1.1, with the addition of a simple roof structure. LOD2.1 adds smaller building parts, and LOD2.2 requires dormers and other roof superstructures of similar size. LOD2.3 demands the explicit modelling of roof overhangs (such models are usually constructed in a combination of terrestrial and airborne techniques). LOD3, the finest LOD, contains openings, roof overhangs, and smaller façade details. Because

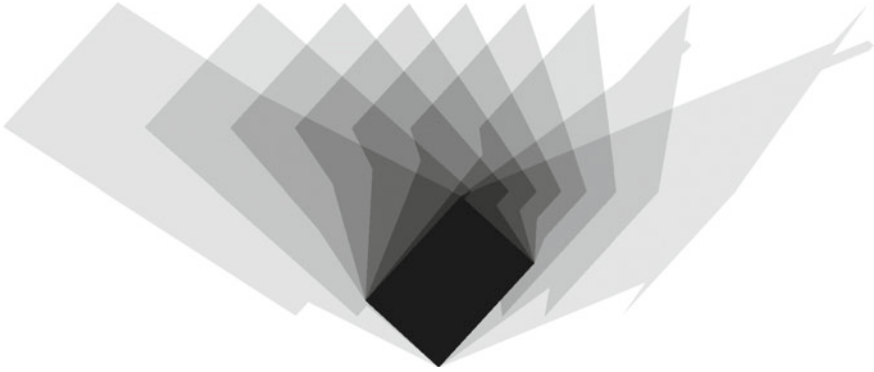


Fig. 2 Orthogonal *top view* composite of 9 shadows during a day (also known as the butterfly shadow diagram; the *outline* of each shadow is drawn at hourly intervals) for an LOD3 model. It suggests the different degree of influence of LOD-related details on the shadow depending on the date, time, and location. The building footprint is shown in *black*

this is the finest representation available, we consider it as ground truth. Figure 1 shows the following LODs: 1.1, 2.0, 2.1, and 3.

3.2 Sun Position and Location on Earth

Figure 2 indicates a substantial difference in the behaviour of this spatial analysis when it comes to the different relative position of the sun, caused by the different time of day and location on Earth. In order to diversify our experiments, we consider two locations: Delft (Netherlands) and Kuala Lumpur (Malaysia), and several timestamps during three characteristic days in 2015: spring equinox (20 March), summer solstice (21 June), winter solstice (22 December), and an arbitrary day: 27 April. This variety results in 81 different positions of the sun spread over daytime.

3.3 Computation of the Shadow

We define the shadow S^{B_i} as the subset of the \mathbb{R}^2 (the ground, a horizontal plane in our case, considered as the shadow receiver) that is occluded by a building B_i . When 3D city models are utilised $S_r^{B_i}$ denotes the shadow forecast with a data set of the representation (LOD) r . We compute the shadow by projecting each polygon of a building model B_i to the plane with a perspective transformation (Blinn 1988), according to the position of the sun. The union of the projected polygons represents the shadow (Fig. 3), however, in the final step we adjust the polygon by removing the footprint of B_i .

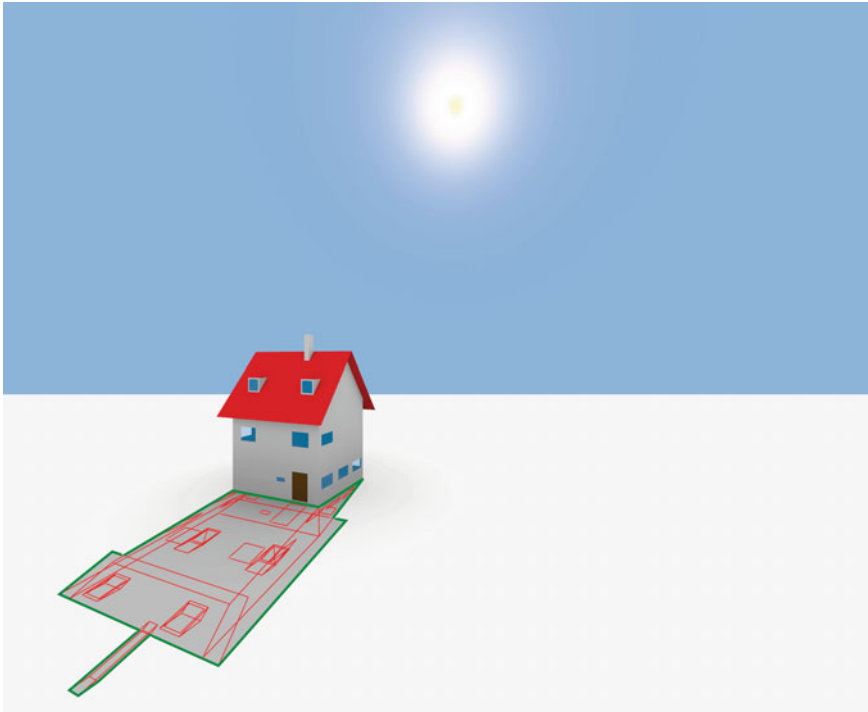


Fig. 3 In our approach and our software implementation, the shadow on the ground is derived as a unionised set (*green*) of projected polygons (in *red*; 51 polygons in this LOD3 case) from the CityGML model, and accounting for the footprint

3.4 Selection of Error Metrics

The shadow cast on the ground is a polygon, thus the first measure that comes to mind to quantify a shadow is its area $a(S^B_i)$, and to compare it to the ground truth:

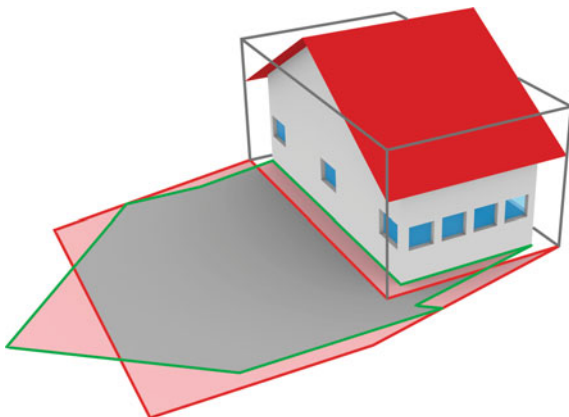
$$\varepsilon_1^r = a(S_r^{B_i}) - a(S^{B_i})$$

However, the deviation of the estimated value is inconclusive, as it appears in Fig. 4. Therefore we introduce a new metric: the area of the symmetric difference (the union without the intersection—see the light red area in the same figure):

$$\varepsilon_2^r = a(S_r^{B_i} \ominus S^{B_i}) = a((S_r^{B_i} \cup S^{B_i}) \setminus (S_r^{B_i} \cap S^{B_i}))$$

In the GIS context this non-negative metric appertains to commission and omission errors, and to false positives and false negatives: the union of the (i) subset that is estimated as shaded but in reality it is not with the (ii) subset of the inverse case.

Fig. 4 A case of two shadows from an LOD1 and an LOD3 (outlined in *red* and *green*, respectively) that have the same area, showing that this error metric can be ambiguous. The area of their symmetric difference (*light red area*) is 28.7 m^2 (33.9 % in relative terms), and the Hausdorff distance in this case is 2.47 m



Because area is only one of the aspects that quantifies the extent of a shape, we compute the similarity between the two shapes, which is a fundamental subject in computer science and GIS (Arkin et al. 1991; Huttenlocher et al. 1993; Samal et al. 2004). There is a variety of methods and metrics to express the correspondence of two shapes in GIS (Ruiz et al. 2011; Goodchild and Hunter 1997), one of the prominent being the Hausdorff distance (Hausdorff 1914). It has been widely used in geoinformation science and 3D city modelling for diverse purposes (Min et al. 2007), for instance, to assess the quality of GIS data (Girres and Touya 2010), to assess the performance of 3D generalisation (Cignoni et al. 1998), to aid map matching (Mustiere and Devogele 2008), to analyse movement trajectories (Liu et al. 2012), and to detect changes between two CityGML models (Pédrinis et al. 2015).

The Hausdorff distance quantifies the mismatch between two geometries by identifying the point on one shape that is the maximum distance from the other shape, therefore we define the third shadow error metric as

$$\varepsilon_3^r = H(S_r^{B_i}, S^{B_i}) = \max(h(S_r^{B_i}, S^{B_i}), h(S^{B_i}, S_r^{B_i}))$$

where $h(A, B)$ is a function that finds the point $a \in A$ that is farthest from any point in B and measures the distance from a to its nearest neighbour in B :

$$h(A, B) = \max_{a \in A} \min_{b \in B} ||a - b||$$

For the three error metrics we compute their root-mean-square error (RMSE). While the Hausdorff distance technically is not an error, it is not uncommon to compute its root-mean-square value (Agarwal et al. 2010; Aspert et al. 2002).

For ε_1 and ε_2 we compute also the relative error (with respect to the true size of the shadow) to put the derived numbers in perspective, which is not possible for ε_3 .

3.5 Implementation

Available implementations do not fully support our methodology. For instance, several GIS tools contain a module to forecast shadows at a specific timestamp, however, such functionality cannot be exploited to our advantage: the shadow cannot be exported as a vector geometry nor most of the tools can be automated. Furthermore, computer graphics software packages usually render a shadow only in raster format. Therefore, we have implemented in Python a software prototype that reads CityGML data, estimates their shadow for a particular location and timestamp, and exports it as a polygon.

The sun positions are taken from PyEphem/XEphem,¹ the implementation of the ephemeris of Bretagnon and Francou (1988). The shadow polygons operations (e.g. union and symmetric difference) are accomplished with Shapely.² The Hausdorff distance has been calculated with PostGIS.³ For validating the correctness of shadows, we have first converted a CityGML model and its calculated polygon shadow to OBJ (with CityGML2OBJS; see Biljecki and Arroyo Ohori 2015), and imported it in Blender,⁴ an open-source 3D computer graphics software. We have rendered the setting for the same date, time, and location, thanks to the Blender add-on Sun Position.⁵ The shadows matched—this is evident in Fig. 3 where the rendered shaded area and the shadow polygon are conflated from independent workflows.

Some of the shadow polygons were found to include long tiny spikes due to floating point errors, which was inhibited with snap rounding (Hobby 1999), and triangulation-based polygon repair with the tool *prepair* (Ledoux et al. 2014).

In addition to calculating the error metrics, the computational cost was recorded (time and number of projected polygons), in order to suggest the load of each LOD.

4 Experiments

We have conducted experiments on 400 different buildings in 8 LODs ($400 \times 8 \times 81 = 259k$ shadows in total). We present the errors in Fig. 5 and Table 1, and discuss them in Sect. 4.1.

¹<https://pypi.python.org/pypi/pyephem/> and <http://www.clearskyinstitute.com/xephem/>.

²<https://pypi.python.org/pypi/Shapely>.

³<http://postgis.net>.

⁴<https://www.blender.org>.

⁵https://wiki.blender.org/index.php/Extensions:2.6/Py/Scripts/3D_interaction/Sun_Position.

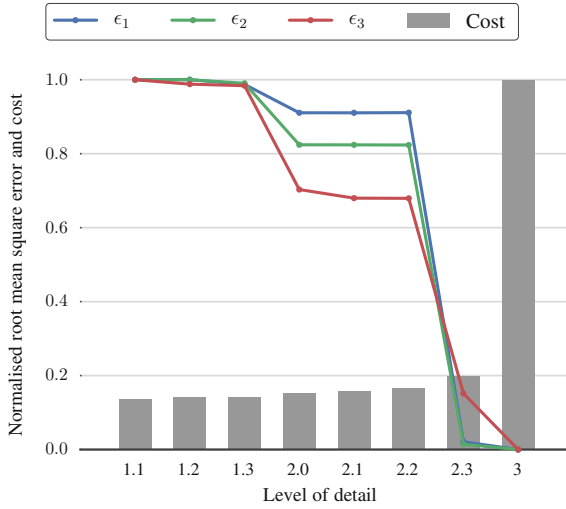


Fig. 5 Errors and computation cost for each LOD. The metrics are normalised according to the least favourable result

Table 1 Numerical results of the experiments. The error metrics are expressed as RMS values

LOD	ϵ_1		ϵ_2		ϵ_3
	(m ²)	(%)	(m ²)	(%)	(m)
1.1	27.6	16.2	40.3	30.1	2.5
1.2	27.6	16.2	40.3	30.1	2.4
1.3	27.2	16.0	39.9	29.9	2.4
2.0	25.1	13.1	33.3	20.7	1.8
2.1	25.1	13.1	33.3	20.7	1.6
2.2	25.1	13.1	33.2	20.6	1.6
2.3	0.5	0.7	0.5	0.7	0.4

4.1 Findings and Discussion

The main findings of the experiments, as shown in Fig. 5 and Table 1, suggest that the relative errors between most LODs are small, and the improvements of each LOD are not significant. Furthermore, we point out other findings:

- The improvement of LOD2 over LOD1 is almost negligible if considering the shadow as a whole (only a 3% reduction in the area error).
- Overall, a finer LOD does bring a more accurate result. However, that is not always the case for each building. The improvement depends on the configuration of the analysed area. As an example, Fig. 7 shows the distribution of ϵ_2, ϵ_3 errors for LOD 2.2. It shows that for many buildings the error is negligible (e.g. in that LOD

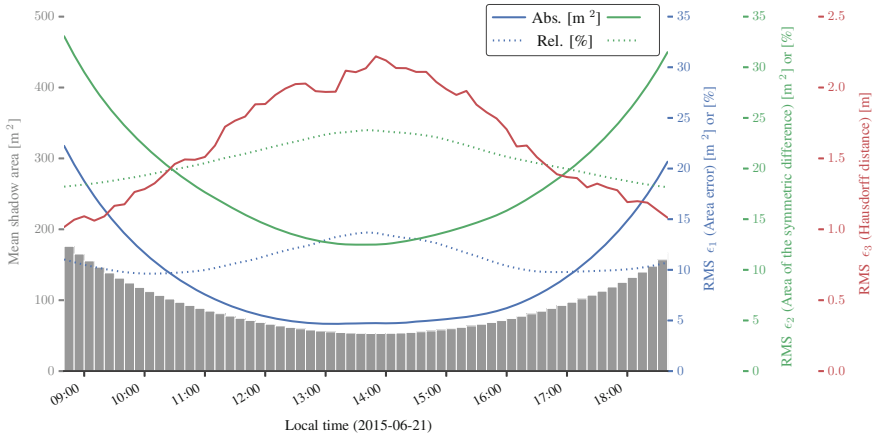


Fig. 6 This combined plot shows the variable behaviour of the three error metrics with respect to the position of the sun, and thus—the size of the shadow in the ground. The values refer to LOD2.2

for 19% buildings there is no error ϵ_2 ; for LOD1.1 that value is at 10%). A more detailed inspection revealed that this applies to buildings with flat roofs and no roof superstructures. If such buildings dominate in an area to be analysed, the acquisition of finer LODs is probably not beneficial.

- Modelling dormers (LOD2.2 and 2.3) and other roof details has a negligible influence on the quality of the prediction. This can be explained by the fact that dormers and chimneys are not present in all buildings, and they make a difference only during a limited time of the day (see the example in Fig. 2).
- Different data (types of buildings) and different settings (day, time, location) result in a different behaviour and magnitude of the error, indicating that related experiments should be diverse. Figure 6 shows the variation of the magnitude of errors as a function of time during one day.
- Figure 6 also shows that while in absolute terms the ϵ_1 and ϵ_2 errors increase with the actual size of the shadow (close to sunrise and sunset), their relative counterparts decrease. Furthermore, the behaviour of ϵ_3 is different.
- LOD3 contains openings, which have no influence on the shadows (unless in the special case of the sun rays passing through two windows, but this triviality was not taken into account). The improvement over LOD2 is mostly caused by overhangs and other smaller details, which on the other hand are probably not appreciated by the use cases that require shadow estimation as an input. Furthermore, an LOD3 model entails a substantial cost of acquisition and processing, which also has to be taken into account.

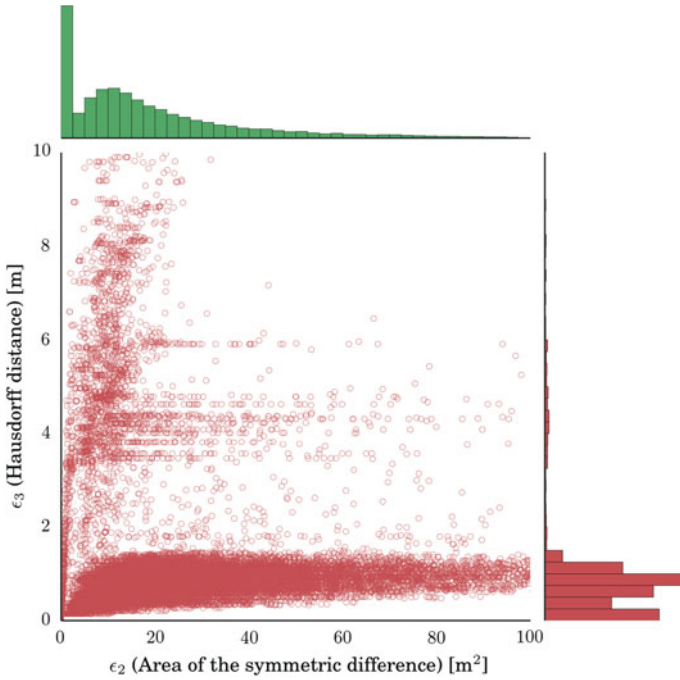


Fig. 7 Scatter plot showing the relation between the error metrics ϵ_2, ϵ_3 and their distribution (for LOD2.2). The histogram on the *right* shows the distribution of ϵ_3 , while the one on *top* shows the distribution of ϵ_2 . The latter shows that for a substantial share of samples the error ϵ_2 is insignificant

4.2 Evaluation of Error Metrics

The results of a seemingly simple analysis of determining the accuracy of a shadow estimation show that errors can be approached from different perspectives. Figure 7 shows the relation between our second and third error metrics, and the distribution for each. We have computed the correlation between the errors to investigate their relation:

$$r_{|\epsilon_1, \epsilon_2} = 0.967 \quad r_{|\epsilon_1, \epsilon_3} = 0.099 \quad r_{\epsilon_2, \epsilon_3} = 0.085$$

An interesting outcome is that there is a low degree of correlation between the area error metrics and the Hausdorff distance. As visible in the scatter plot, there are several cases in which the magnitude of the area error metrics (ϵ_1 and ϵ_2) are small ($<0.1 \text{ m}^2$), with a significant value of ϵ_3 . This is most evident in the results of LOD2.3 where the first two error metrics are low, and the Hausdorff distance is not negligible. Manual inspection revealed that these deviations are caused by small LOD3-only details such as chimneys, which render small shadow area differences, but since they protrude, ϵ_3 is of noticeable value (e.g. see the chimney in Fig. 3).

We are ambivalent on the use of the Hausdorff distance for this purpose. Besides the advantage that ϵ_3 revealed some discrepancies that were not detected by the first two, it helped to put the area errors in perspective, i.e. some large area errors were in fact caused by practically insignificant deviations (e.g. long and narrow strips of shadows). The disadvantage is that the Hausdorff distance is not a stable error metric (see Fig. 6), and it is sensitive to computational and geometric errors, e.g. caused by floating point errors and slivers.

For the first two error metrics we have computed also a relative counterpart, as the relation to the total size of the shadow. This helped to understand the true magnitude of the error (e.g. that the RMS of ϵ_2 for LOD1.2 is 16 % of the shadow size).

5 Conclusions

In this paper we have presented a study on the influence of the LOD of a 3D building model on the quality on a 3D GIS spatial analysis. We benchmark the accuracy of 8 LODs for estimating the shadow, and conclude that investing in a data set with a fine LOD is not always a good idea. For instance, for 50 % of buildings modelled in the coarse LOD1.1, the error (ϵ_1) is within 10.2 % of the size of the shadow (percentile rank of a score), which practically does not have much influence for some use cases. For areas with a higher share of buildings with a flat roof this fact would be even more substantiated.

Therefore, we refute the universally accepted assumption that finer LODs inherently bring more accurate results in spatial analyses, and we argue that such analyses should be conducted to understand the impact of LOD on a specific use case.

Shadows do not have a unique metric, a fact that is also valid for many other spatial analyses. We use three error metrics: area error (ϵ_1), area of the symmetric difference (ϵ_2), and the Hausdorff distance (ϵ_3), which show different observations. In our study, we determine the influence of the resolution of the models on raw shadows as standalone concepts. While we find that the LOD has a variable influence, these smaller improvements may not always benefit a use case. Actually, it may damage it: while the improvement is negligible, the acquisition and processing costs could be substantially higher. This depends on the *weight* a shadow has as an input in a use case. For instance, in geo-visualisation a more accurate shadow probably does not make any difference, while in some other such as predicting the yield of photovoltaic panels it might be tangible.

For future work we plan to investigate the significance and influence of the LOD on the outcome of a use case that uses estimated shadows as input. For instance, the error in the prediction of the duration a wall is shaded during the day, or the error in the estimation of the loss of the solar potential in kWh/year due to shadows.

Furthermore, we plan to investigate the benchmarking of continuous LODs (Arroyo Ohori et al. 2015).

Acknowledgments The detailed observations raised during the peer-review are gratefully acknowledged. We are thankful to the developers of the open-source software that was used in the realisation of this work, and to Ken Arroyo Ohori for suggestions which have accelerated the development of the method. This research is supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by the Ministry of Economic Affairs (project code: 11300).

References

- Agarwal, P. K., Har-Peled, S., Sharir, M., & Wang, Y. (2010). Hausdorff distance under translation for points and balls. *ACM Transactions on Algorithms*, 6(4), 1–26. Aug.
- Alam, N., Coors, V., & Zlatanova, S. (2013). Detecting shadow for direct radiation using CityGML models for photovoltaic potentiality analysis. In C. Ellul, S. Zlatanova, M. Rumor, & R. Laurini (Eds.), *Urban and regional data management* (pp. 191–196). London, UK: CRC Press.
- Appleton, K., & Lovett, A. (2003). GIS-based visualisation of rural landscapes: defining ‘sufficient’ realism for environmental decision-making. *Landscape and Urban Planning*, 65(3), 117–131.
- Arkin, E. M., Chew, L. P., Huttenlocher, D. P., Kedem, K., & Mitchell, J. S. B. (1991). An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3), 209–216. Mar.
- Arroyo Ohori, K., Ledoux, H., Biljecki, F., & Stoter, J. (2015). Modeling a 3D city model and its levels of detail as a true 4D model. *ISPRS International Journal of Geo-Information*, 4(3), 1055–1075.
- Aspert, N., Santa-Cruz, D., & Ebrahimi, T. (2002). MESH: measuring errors between surfaces using the Hausdorff distance. In *IEEE international conference on multimedia and expo (ICME)* (pp. 705–708). IEEE.
- Bartie, P., Reitsma, F., Kingham, S., & Mills, S. (2010). Advancing visibility modelling algorithms for urban environments. *Computers, Environment and Urban Systems*, 34(6), 518–531. Nov.
- Besuievsky, G., Barroso, S., Beckers, B., & Patow, G. (2014). A configurable LoD for procedural urban models intended for daylight simulation. In G. Besuievsky & V. Tourre (Eds.), *Proceedings of the eurographics workshop on urban data modelling and visualisation* (pp. 19–24). Strasbourg, France: The Eurographics Association. Apr.
- Biljecki, F., & Arroyo Ohori, K. (2015). Automatic semantic-preserving conversion between OBJ and CityGML. In *Eurographics workshop on urban data modelling and visualisation 2015* (pp. 25–30). Delft, Netherlands.
- Biljecki, F., Ledoux, H., & Stoter, J. (2014a). Error propagation in the computation of volumes in 3D city models with the Monte Carlo method. *ISPRS annals photogrammetry, remote sensing and spatial information sciences*, II(2), 31–39.
- Biljecki, F., Ledoux, H., Stoter, J., & Zhao, J. (2014b). Formalisation of the level of detail in 3D city modelling. *Computers, Environment and Urban Systems*, 48, 1–15. Nov.
- Biljecki, F., Heuvelink, G. B. M., Ledoux, H., & Stoter, J. (2015a). Propagation of positional error in 3D GIS: estimation of the solar irradiation of building roofs. *International Journal of Geographical Information Science*, 29(12), 2269–2294. Dec.
- Biljecki, F., Ledoux, H., & Stoter, J. (2015b). Improving the consistency of multi-LOD CityGML datasets by removing redundancy. In M. Breunig, A.-D. Mulhim, E. Butwilowski, P. V. Kuper, J. Benner, & K.-H. Häfele (Eds.), *3D geoinformation science* (pp. 1–17). Springer.
- Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., & Çöltekin, A. (2015c). Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4), 2842–2889. Dec. <http://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.

- Blinn, J. (1988). Me and my (fake) shadow. *IEEE Computer Graphics and Applications*, 8(1), 82–86. Jan.
- Boeters, R., Arroyo Ohori, K., F., Biljecki, & S. Zlatanova. (2015). Automatically enhancing CityGML LOD2 models with a corresponding indoor geometry. *International Journal of Geographical Information Science*, 29(12), 2248–2268.
- Booij, M. J. (2005). Impact of climate change on river flooding assessed with different spatial model resolutions. *Journal of Hydrology*, 303(1–4), 176–198. Mar.
- Brasebin, M., Perret, J., Mustière, S., & Weber, C. (2012). Measuring the impact of 3D data geometric modeling on spatial analysis: illustration with Skyview factor. In T. Leduc, G. Moreau, & R. Billen (Eds.), *Usage, usability, and utility of 3D city models—European COST action TU0801* (pp. (02001)1–16), Nantes, France, Oct. 2012. EDP Sciences.
- Bretagnon, P., & Francou, G. (1988). Planetary theories in rectangular and spherical variables. VSOP 87 solutions. *Astronomy and Astrophysics*, 202, 309–315. Aug.
- Burnicki, A. C., Brown, D. G., & Goovaerts, P. (2007). Simulating error propagation in land-cover change analysis: The implications of temporal dependence. *Computers, Environment and Urban Systems*, 31(3), 282–302. May.
- Carneiro, C., & Golay, F. (2009). Solar radiation over the urban texture: LIDAR data and image processing techniques for environmental analysis at city scale. In J. Lee & S. Zlatanova (Eds.), *3D Geo-Information Sciences* (pp. 319–340). Heidelberg: Springer.
- Chaubey, I., Cotter, A. S., Costello, T. A., & Soerens, T. S. (2005). Effect of DEM data resolution on SWAT output uncertainty. *Hydrological Processes*, 19(3), 621–628. Feb.
- Cignoni, P., Rocchini, C., & Scopigno, R. (1998). Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2), 167–174. June.
- City of Mississauga. Standards for shadow studies, Feb. 2012. http://www6.mississauga.ca/online/maps/planbldg/UrbanDesign/ShadowStudiesFinal_Feb2012.pdf.
- Den Haag (2011). Voorstel van het college inzake beleid dakopbouwen. *RIS*, 180461.
- Eicker, U., Monien, D., Duminil, E., & Nouvel, R. (2015). Energy performance assessment in urban planning competitions. *Applied Energy*, 155, 323–333. Oct.
- Girres, J.-F., & Touya, G. (2010). Quality assessment of the French openstreetmap dataset. *Transactions in GIS*, 14(4), 435–459.
- Goodchild, M. F., & Hunter, G. J. (1997). A simple positional accuracy measure for linear features. *International Journal of Geographical Information Science*, 11(3), 299–306. Apr.
- Gröger, G., & Plümer, L. (2012). CityGML—interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71, 12–33. July.
- Hausdorff, F. (1914). *Grundzüge der Mengenlehre*. Leipzig, Germany: Verlag von Veit and Comp.
- Helbich, M., Jochem, A., Mücke, W., & Höfle, B. (2013). Boosting the predictive accuracy of urban hedonic house price models through airborne laser scanning. *Computers, Environment and Urban Systems*, 39(C), 81–92.
- Hengl, T. (2006). Finding the right pixel size. *Computers and Geosciences*, 32(9), 1283–1298. Nov.
- Herbert, G., & Chen, X. (2015). A comparison of usefulness of 2D and 3D representations of urban planning. *Cartography and Geographic Information Science*, 42(1), 22–32.
- Hobby, J. D. (1999). Practical segment intersection with finite precision output. *Computational Geometry*, 13(4), 199–214. Oct.
- Hofierka, J., & Zlocha, M. (2012). A new 3-D solar radiation model for 3-D city models. *Transactions in GIS*, 16(5), 681–690. Oct.
- Huttenlocher, D. P., Klanderman, G. A., & Rucklidge, W. J. (1993). Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9), 850–863. Sept.
- Hwang, R.-L., Lin, T.-P., & Matzarakis, A. (2011). Seasonal effects of urban street shading on long-term outdoor thermal comfort. *Building and Environment*, 46(4), 863–870. Apr.
- Jochem, A., Höfle, B., Rutzinger, M., & Pfeifer, N. (2009). Automatic roof plane detection and analysis in airborne lidar point clouds for solar potential assessment. *Sensors*, 9(7), 5241–5262. July.

- Kibria, M. S., Zlatanova, S., Itard, L., & Dorst, M. (2009). GeoVEs as tools to communicate in urban projects: requirements for functionality and visualization. In *3D geo-information sciences* (pp. 379–395). Heidelberg: Springer.
- Knowles, R. L. (2003). The solar envelope: Its meaning for energy and buildings. *Energy and Buildings*, 35(1), 15–25. Jan.
- Kumar, L., Skidmore, A. K., & Knowles, E. (1997). Modelling topographic variation in solar radiation in a GIS environment. *International Journal of Geographical Information Science*, 11(5), 475–497. July.
- Lange, E., & Hehl-Lange, S. (2005). Combining a participatory planning approach with a virtual landscape model for the siting of wind turbines. *Journal of Environmental Planning and Management*, 48(6), 833–852. Nov.
- Ledoux, H., Arroyo Otori, K., & M. Meijers. (2014). A triangulation-based approach to automatically repair GIS polygons. *Computers and Geosciences*, 66, 121–131.
- Li, Y., Brimicombe, A. J., & Ralphs, M. P. (2000). Spatial data quality and sensitivity analysis in GIS and environmental modelling: the case of coastal oil spills. *Computers, Environment and Urban Systems*, 24(2), 95–108. Mar.
- Ling, Y., Ehlers, M., Usery, E. L., & Madden, M. (2008). Effects of spatial resolution ratio in image fusion. *International Journal of Remote Sensing*, 29(7), 2157–2167. Apr.
- Liu, L., Qiao, S., Zhang, Y., & Hu, J. (2012). An efficient outlying trajectories mining approach based on relative distance. *International Journal of Geographical Information Science*, 26(10), 1789–1810. Oct.
- Luebke, D., Reddy, M., Cohen, J. D., Varshney, A., Watson, B., & Huebner, R. (2003). *Level of detail for 3D graphics*. Morgan Kaufmann Pub.
- Mardaljevic, J., & Rylatt, M. (2003). Irradiation mapping of complex urban environments: an image-based approach. *Energy and Buildings*, 35(1), 27–35. Jan.
- Min, D., Zhilin, L., & Xiaoyong, C. (2007). Extended Hausdorff distance for spatial objects in GIS. *International Journal of Geographical Information Science*, 21(4), 459–475. Apr.
- Möller, T., & Trumbore, B. (1997). Fast, minimum storage ray-triangle intersection. *Journal of Graphics Tools*, 2(1), 21–28.
- Morello, E., & Ratti, C. (2009). Sunscapes: ‘Solar envelopes’ and the analysis of urban DEMs. *Computers, Environment and Urban Systems*, 33(1), 26–34.
- Mustiere, S., & Devogele, T. (2008). Matching networks with different levels of detail. *GeoInformatica*, 12(4), 435–453.
- Nguyen, H. T., & Pearce, J. M. (2012). Incorporating shading losses in solar photovoltaic potential assessment at the municipal scale. *Solar Energy*, 86(5), 1245–1260. May.
- Pédrinis, F., Morel, M., & Gesquière, G. (2015). Change detection of cities. In *3D geoinformation science* (pp. 123–139). Springer.
- Peters, R., Ledoux, H., & Biljecki, F. (2015). Visibility analysis in a point cloud based on the medial axis transform. In *Eurographics workshop on urban data modelling and visualisation 2015* (pp. 7–12). Delft, Netherlands.
- Pogson, M., & Smith, P. (2015). Effect of spatial data resolution on uncertainty. *Environmental Modelling and Software*, 63, 87–96. Jan.
- Redweik, P., Catita, C., Brito, M., & Brito, M. (2013). Solar energy potential on roofs and facades in an urban landscape. *Solar Energy*, 97, 332–341. Nov.
- Ruiz, J. J., Ariza, F. J., Ureña, M. A., & Blázquez, E. B. (2011). Digital map conflation: a review of the process and a proposal for classification. *International Journal of Geographical Information Science*, 25(9), 1439–1466. Sept.
- Samal, A., Seth, S., & Cueto, K. (2004). A feature-based approach to conflation of geospatial sources. *International Journal of Geographical Information Science*, 18(5), 459–489. July.
- Stoter, J., de Kluijver, H., & Kurakula, V. (2008). 3D noise mapping in urban areas. *International Journal of Geographical Information Science*, 22(8), 907–924. Aug.
- Strzalka, A., Bogdahn, J., Coors, V., & Eicker, U. (2011). 3D city modeling for urban scale heating energy demand forecasting. *HVAC&R Research*, 17(4), 526–539.

- Strzalka, A., Alam, N., Duminil, E., Coors, V., & Eicker, U. (2012). Large scale integration of photovoltaics in cities. *Applied Energy*, 93, 413–421. May.
- Tooke, T. R., Coops, N. C., Voogt, J. A., & Meitner, M. J. (2011). Tree structure influences on rooftop-received solar radiation. *Landscape and Urban Planning*, 102(2), 73–81. Aug.
- Usery, E. L., Finn, M. P., Scheidt, D. J., Ruhl, S., Beard, T., & Bearden, M. (2004). Geospatial data resampling and resolution effects on watershed modeling: A case study using the agricultural non-point source pollution model. *Journal of Geographical Systems*, 6(3), 289–306. Oct.
- Whitted, T. (1980). An improved illumination model for shaded display. *Communications of the ACM*, 23(6), 343–349. June.
- Williams, L. (1978). Casting curved shadows on curved surfaces. *ACM SIGGRAPH Computer Graphics*, 12(3), 270–274. Aug.
- Woo, A., Poulin, P., & Fournier, A. (1990). A survey of shadow algorithms. *IEEE Computer Graphics and Applications*, 10(6), 13–32.
- Yezioro, A., & Shaviv, E. (1994). Shading: A design tool for analyzing mutual shading between buildings. *Solar Energy*, 52(1), 27–37. Jan.

Interactive and View-Dependent See-Through Lenses for Massive 3D Point Clouds

Sören Discher, Rico Richter and Jürgen Döllner

Abstract 3D point clouds are a digital representation of our world and used in a variety of applications. They are captured with LiDAR or derived by image-matching approaches to get surface information of objects, e.g., indoor scenes, buildings, infrastructures, cities, and landscapes. We present novel interaction and visualization techniques for heterogeneous, time variant, and semantically rich 3D point clouds. Interactive and view-dependent see-through lenses are introduced as exploration tools to enhance recognition of objects, semantics, and temporal changes within 3D point cloud depictions. We also develop filtering and highlighting techniques that are used to dissolve occlusion to give context-specific insights. All techniques can be combined with an out-of-core real-time rendering system for massive 3D point clouds. We have evaluated the presented approach with 3D point clouds from different application domains. The results show the usability and how different visualization and exploration tasks can be improved for a variety of domain-specific applications.

Keywords 3D point clouds · LIDAR · Visualization · Point-based rendering

1 Introduction

Applications for environmental monitoring (Koenig et al. 2015), urban planning and development (Tomljenovic et al. 2015), as well as disaster and risk management (Canli et al. 2015) require precise and up-to-date surface information for objects, sites, and landscapes. 3D point clouds are digital discrete surface repre-

S. Discher (✉) · R. Richter · J. Döllner
Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
e-mail: soeren.discher@hpi.de

R. Richter
e-mail: rico.richter@hpi.de

J. Döllner
e-mail: juergen.doellner@hpi.de

representations that fulfill these requirements. They can be created automatically and efficiently by means of in situ and remote sensing technology (e.g., laser scanning or photogrammetric approaches) with high density (e.g., 400 points per m^2) and high capturing frequency (e.g., once a month). Due to the massive amount of data managing, processing and visualization of 3D point clouds poses challenges for hard- and software systems (Richter und Döllner 2014; Leberl et al. 2010). Traditionally, geoinformation applications and systems reduce the density and precision, or extract generalized, mesh-based 3D models in a time-consuming process (Lafarge and Mallet 2012). Hence, they do not use the full potential of the data. Out-of-core and external memory algorithms on the other hand are designed to cope with massive amounts of data (Nebiker et al. 2010). As an example, arbitrarily large 3D point clouds can be explored in real-time by combining specialized spatial data structures with Level-of-Detail concepts (Richter et al. 2015; Goswami et al. 2013). Most of these approaches render all points in a uniform way that gives an equal amount of significance to every point. However, points within a 3D point cloud typically differ in terms of relevance. The relevance of a point depends on the kind of information it carries (e.g., timestamp, surface category, color, position) as well as the current use case and visualization task. In this paper, we focus on the following use cases:

- Different Acquisition Types.** Surface information for a site could be available from different surveys such as an airborne, mobile mapping, terrestrial, outdoor and indoor data acquisition. On a broader scale, 3D point clouds of spatial environments might consist of overground as well as subterranean structures (e.g., mine shafts, sewers). By allowing users to see through occluding structures (i.e., by masking out corresponding points) we can facilitate the in-depth exploration of such 3D point clouds (Fig. 1).

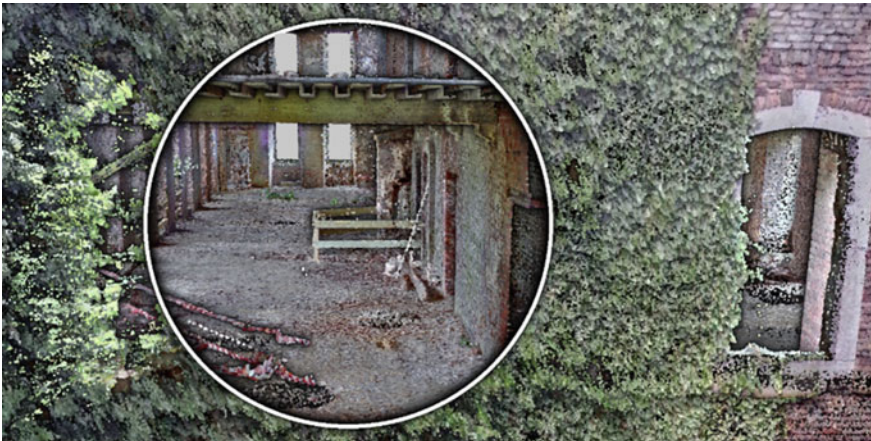


Fig. 1 Example of a massive 3D point cloud consisting of indoor and outdoor scans. It is explored with a see-through lens to inspect the occluded interior of the building in the context of the overall scan



Fig. 2 Multi-temporal 3D point cloud: building points that are not present in the old scan (*upper left*) but captured in the new scan (*lower right*) are highlighted with a red color scheme and by tracing their boundaries

- **Multi-temporal Data.** Many applications require frequent scans and simultaneous use of 3D point clouds taken at different points in time. To assist users in exploring differences and structural changes of the captured site within such multi-temporal 3D point clouds (e.g., constructed, demolished, or modified buildings), points indicating such changes should be highlighted (Fig. 2).
- **Classification-Dependent Rendering.** Typically, points represent different surface categories (e.g., ground, building, vegetation, water, city furniture interior, façade). By applying different point-based rendering techniques and color schemes to each point, taking into account characteristics of its surface category (e.g., fuzziness of vegetation, smooth ground surfaces or planar building roofs), different objects within a 3D point cloud depiction and relations between these objects can be distinguished more efficiently (Richter et al. 2015; Gao et al. 2014). However, relevant objects might still be not visible due to occlusion by less relevant objects (e.g., buildings below vegetation). To facilitate the identification of full and partly occluded objects, users need to see through occluding structures (Fig. 3).

To filter points within a 3D point cloud based on their relevance to the given use case, we define *see-through lenses* for massive 3D point clouds:

- A see-through lens defines a space within a 3D point cloud, in which points of higher relevance are emphasized by masking out less relevant points completely or in parts.
- The area of a see-through lens can be defined *interactively* by the user or automatically with respect to the current view position (e.g., center of the screen), i.e., *view-dependent*.

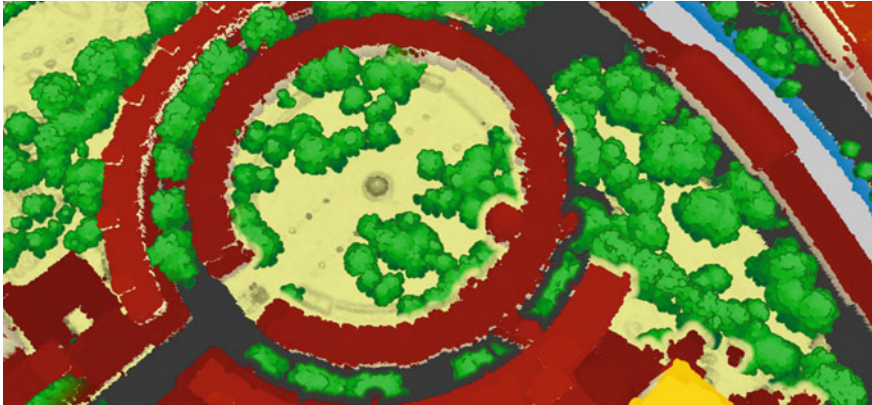


Fig. 3 3D point cloud with semantics: vegetation points are masked out in the *lower right* part to show hidden building points

In this paper we present several interactive and view-dependent see-through lenses for massive 3D point clouds and discuss their specific advantages and disadvantages in different scenarios (Sect. 3). Building upon an existing out-of-core rendering approach for the classification-dependent visualization of massive 3D point clouds, we show how see-through lenses can be efficiently combined with existing point-based rendering approaches as a post processing step (Sect. 4). As we demonstrate on a set of real-world datasets, our approach allows to select and configure see-through lenses at runtime while providing interactive frame rates for arbitrarily large 3D point clouds (Sect. 5). We conclude by giving an outlook on future research directions (Sect. 6).

2 Related Work

An overview of point-based rendering techniques is given by Gross and Pfister (2007). While several photo—(Yu and Turk 2013; Preiner et al. 2012) and non-photorealistic (Goesele et al. 2010; Xu et al. 2004) rendering techniques for 3D point clouds exist, most approaches apply a uniform style regardless of each points significance for the given use case. Some approaches prioritize and highlight points based on precomputed surface categories by combining different rendering styles to emphasize surface category characteristics. Gao et al. (2014) focus on combining point and polygon based rendering techniques to achieve a solid, hole-free visualization of certain surface categories (i.e., buildings, ground). However, this approach does not address the occlusion of structures and surfaces. Richter et al. (2015) show how out-of-core rendering algorithms for 3D point clouds can benefit from per-point surface category information and provide an even broader set of

point-based rendering styles. The proposed rendering system allows to integrate see-through lenses using image compositing.

Interactive occlusion management techniques for virtual 3D environments have been widely discussed for years. However, the focus is on mesh-based 3D models instead of point-based rendering. Elmqvist and Tsigas (2008) differentiate three primary purposes for these techniques: (1) the discovery of yet unknown, occluded objects, (2) the exploration of occluded objects whose position has been known beforehand, and (3) the exploration of an occluded object's spatial relationship to its environment. Since all these purposes are important when exploring 3D point clouds, we introduce techniques addressing all of them. Elmqvist and Tsigas (2008) identify five different categories of occlusion management techniques: Tour Planner based approaches (Andújar et al. 2004; Burtnyk et al. 2002) calculate special camera paths through an environment, ensuring that every significant object is visible at least once while following these paths. Some approaches use multiple views of an environment in parallel. The different views are either rendered into separate viewports (Hasan et al. 2014; Brosz et al. 2011) or combined into a single one by applying multi-perspective projections (Pasewaldt et al. 2014; Brosz et al. 2007). Volumetric probes stay with a single view for an environment. Instead they deform objects (e.g., by manipulating their proportions) within a certain area that is often defined by users (Wang et al. 2005; Sonnet et al. 2004).

The see-through lenses for massive 3D point clouds proposed in this paper belong to the category of *virtual x-rays*. These techniques leave object proportions intact and usually operate in screen space, either masking out occluders completely or making them (semi-)transparent. They can be either active, i.e., requiring user input to define the areas they should be applied to (Trapp et al. 2008; Ropinski et al. 2005), or passive, i.e., finding those areas automatically based on available information about the objects within an environment (Vaaraniemi et al. 2013; Sigg et al. 2012). We decided for visual x-rays, as this allows users to freely navigate the 3D point cloud (which wouldn't be the case for tour planner based approaches) without having to manipulate or deform the data (which would be the case for volumetric probes). We also decided against combining multiple views as this would interfere with existing out-of-core rendering techniques for massive 3D point clouds.

3 Overview

We present several see-through lenses to filter and highlight structures within 3D point cloud depictions based on their significance to the given use case. All lenses operate in screen space and are realized as post processing effects, thus facilitating their integration into existing point-based rendering approaches.

3.1 Data Characteristics

Additional per-point attributes can be calculated in a preprocessing step. Common attributes are:

- **Color.** Color and color-infrared information allows for a more realistic visualization of 3D point clouds. It can be extracted from image data that ideally have been captured simultaneously with the 3D point cloud.
- **Surface normal.** Surface normals describe the orientation and structure of a surface at each point's position (Mitra and Nguyen 2003). They are typically used to align and orient the rendering geometry (e.g., point splats).
- **Topological information.** This information can be derived by analyzing the local point proximity. Topologic attributes are for example the height of a point within its local neighborhood, surface smoothness, and local point density. Adapting the colorization accordingly puts emphasis on these attributes, which can be used to accentuate detailed and geometrically complex object structures (e.g., roof elements).
- **Surface category.** A basic categorization of points into indoor, outdoor, over-ground, subterranean, airborne, or terrestrial can be made directly during the capturing process. A more specific categorization into surface categories such as building, ground, and vegetation can be derived with point cloud classification approaches that are based on analyzing topological attributes or additional geodata capturing the same surface area. (Richter et al. 2013).
- **Temporal information.** Temporal attributes describe the date of data collection.
- **Change information.** Change information describes on a per-point basis the degree of change related to a given reference geometry. That reference geometry might be another 3D point cloud captured at a different point in time or a 3D model describing the same area (e.g., building model, cadastre data, 3D city model).

These per-point attributes can be used to customize the appearance of points individually at runtime (Richter et al. 2015). Hence, points that are *related* to each other (e.g., representing the same surface category or being created at the same point in time) can be identified more easily. If points are occluding each other, these attributes can also serve as a means to assess and highlight the significance of each point for a given use case.

3.2 Priority Levels

Depending on the application, use-case, and kind of information that needs to be explored, the significance of a point may vary. When exploring a densely forested area for example, points representing vegetation are often less interesting than

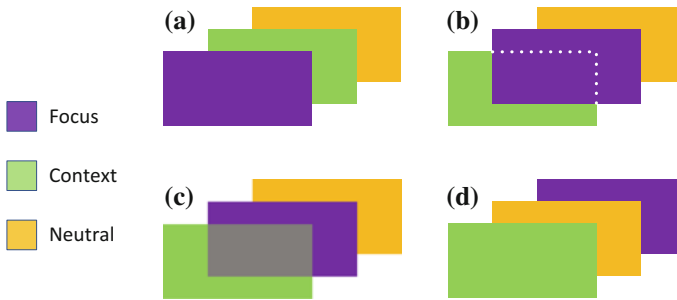


Fig. 4 Illustration of compositions for different priority levels in occlusion situations. **a** Context information is, **b** masked out in favor of **c**, blended with focus information. **d** Neutral information is neither highlighted nor filtered

occluded points belonging to subjacent objects (e.g., buildings). We describe the significance of each point with the following priority levels:

- **Focus.** Essential information of interest and exploration aim (e.g., interior objects occluded by walls, subterranean structures or buildings that have been demolished between consecutive scans). The more points carrying such information are occluded, the less information can be gathered, i.e., the less effective the exploration becomes (Fig. 4).
- **Context.** Information that increases the overall realism of the visualization without being the main focus of the exploration (e.g., vegetation in densely forested areas). Points carrying such information can be safely masked out in favor of focus information (Fig. 4).
- **Neutral.** Depending on the use case, we might want to focus on specific occlusion scenarios, such as solely on buildings that are occluded by vegetation. This requires to define all other information as neutral, i.e., points representing such information are treated as solid geometry that is neither masked out or highlighted (Fig. 4).

These priority levels are used to define the composition of the data for the rendering and visualization of different occlusion scenarios (Fig. 4): Occlusions with the same priority level or including points carrying neutral information are solved by displaying the nearest point to the view position (i.e., similar to the default behavior in 3D rendering). Points carrying context information on the other hand (so-called context points) are masked if focus points are occluded.

3.3 *Interactive and View-Dependent See-Through Lenses*

Simply masking out all context points in occlusion scenarios limits the correct estimation of depth differences and does not provide information about the object shape

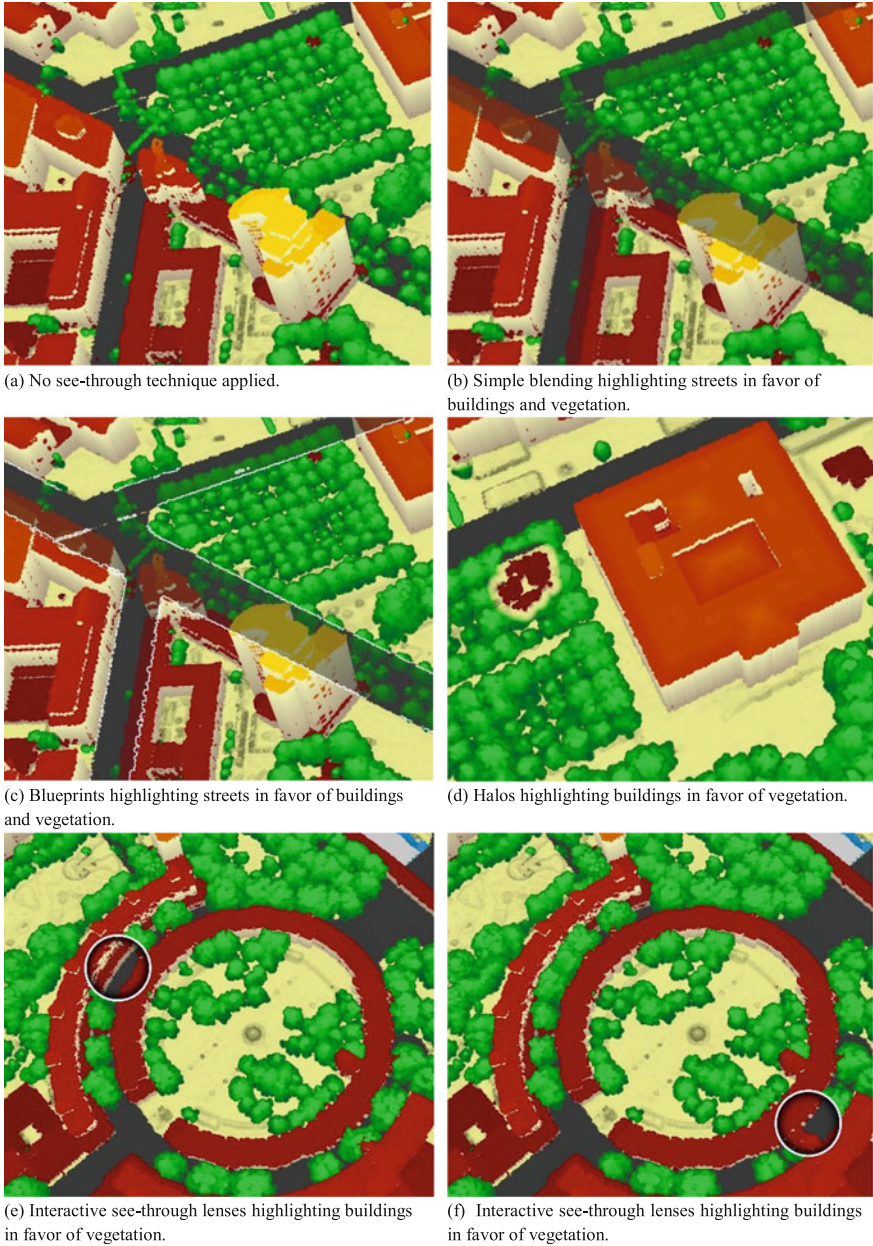


Fig. 5 Examples of different see-through lenses applied to massive 3D point clouds

and boundaries within a 3D point cloud depiction (Fig. 5a). Blending context points and focus points by a certain factor addresses these limitations, however, areas with blended structures might be difficult to recognize during the exploration (Fig. 5b).

3.3.1 Blueprints

Blueprints are a traditional form of technical drawings and known for their characteristic style which originating from the historical contact print process (Nienhaus and Döllner 2004). Construction elements are visualized by tracing outlines using different line widths and a color contrasting favorably with the background. Thus, the focus of the viewer is directed towards the most significant construction elements. We adapt this concept to highlight areas where focus and context points have been merged by tracing the boundaries of those areas with a configurable line width and color (Fig. 5c). This approach is especially effective to highlight changes within multi-temporal 3D point clouds (Fig. 2).

3.3.2 Halos

A stronger emphasis on focus points can be obtained by masking out additional context points in a defined local proximity (Fig. 5d). As a result, groups of neighboring focus points are surrounded with a halo effect, similarly to the real-world phenomenon and the technique used by artists throughout history to emphasize certain individuals. Techniques that highlight objects by removing occluders are known as *cut-away-views* (Vaaraniemi et al. 2013; Sigg et al. 2012). So far, they have not been applied to point-based rendering. The typical use case to apply this technique is the exploration of complex structures that are completely occluded by their surroundings, such as subterranean structures (Fig. 1).

3.3.3 Interactive See-Through Lenses

All techniques that have been introduced are applied automatically to the data where focus information is occluded by context information. Naturally, the number, position, and extent of these areas varies depending on the view position. As opposed to that automated, view dependent approach, interactive see-through lenses (also commonly known as ‘magic lenses’) can be moved freely across the screen. Within an interactive see-through lens, context points are masked out completely, whereas in the surrounding no blending is applied at all (Fig. 5f, e). This is required to focus on occlusions within certain areas whose position is known beforehand (e.g., to explore and show a former state for a certain area). However, if those areas are unknown, interactive see-through lenses are inefficient as the entire 3D point cloud has to be traversed manually to identify all areas.

4 Compatibility to Existing Point-Based Rendering Approaches

The see-through lenses proposed in this paper can be integrated into existing rendering systems for 3D point clouds. We demonstrate this by extending the rendering approach for massive 3D point clouds with surface category information introduced by Richter et al. (2015) (Fig. 6).

The approach is based on a layered, multi-resolution kd-tree to efficiently select relevant points of different surface categories for a given view position and rendering budget. To render selected points multi-pass rendering in combination with G-Buffers (Saito and Takahashi 1990), i.e., specialized frame buffer objects (FBO), is used. Multiple 3D textures store information for color, depth, or normal values. For each surface category a separate rendering pass is applied, allowing to combine different rendering techniques. Originally, Richter et al. (2015) utilize a separate G-Buffer for each surface category and apply a compositing pass to combine these G-Buffers into a final image. To apply see-through lenses, the compositing logic is extended. See-through lenses are implemented via programmable fragment shaders and can be activated and configured at runtime.

To guarantee interactive frame rates even for massive 3D point clouds with a variety of thematic attributes and surface categories, the original approach was modified to have only one G-Buffer for each priority level (instead of one for each surface category). Thus, the compositing pass always combines three G-Buffers, independently from the overall number of surface categories within a 3D point

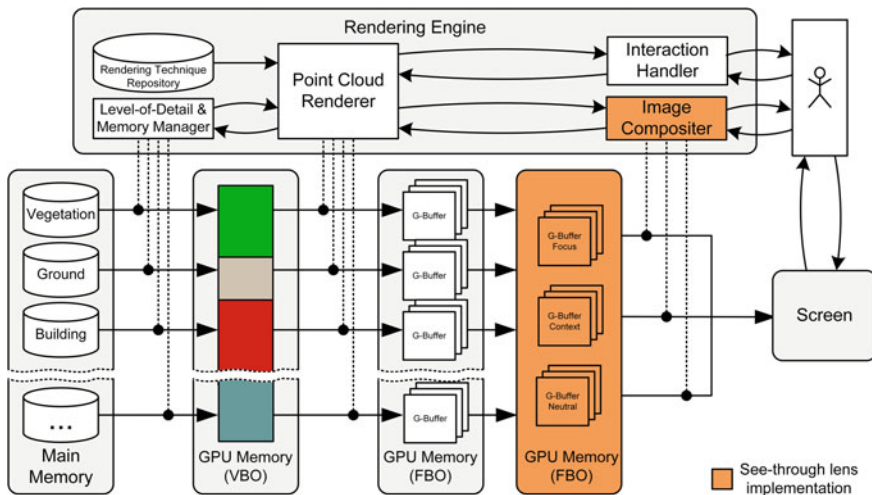


Fig. 6 Schematic overview of the classification-dependent point-based rendering system by Richter et al. (2015) and our modifications. Categorized by surface categories, points are transferred to GPU memory and rendered into separate G-Buffers that are merged based on the respective priority levels before being composed to synthesize the final image

cloud depiction. To enable halos with a defined proximity (e.g., 2 meters), an additional halo mask has to be prepared for surface categories belonging to the focus that determines the area in which context information needs to be masked out. However, creating that mask for each frame has a notable impact on the overall performance (Sect. 5).

5 Results and Applications

We have implemented the proposed rendering approach using C++, OpenGL, GLSL, and OpenSceneGraph. Performance evaluation was conducted based on three real-world data sets of different size and point density (Table 1). The test system consisted of an Intel Xeon CPU with 3.20 GHz, 12 GB main memory, and a NVIDIA GeForce GTX 770 with 2 GB device memory.

Each data set was rendered from a zoomed out and a close up perspective which affected the number of points rendered for a given frame. As shown in Table 2, applying simple blending, blueprints, and interactive see-through lenses has minimal effect on the overall performance. Since all techniques are implemented as screen-based post-processing effects, this stays true independent from the overall number of rendered points. Halos require a halo mask that is created by rendering corresponding surface categories twice, which effectively halves the average frame

Table 1 Characteristics of the datasets used to evaluate the performance of the presented rendering approach for view-dependent and interactive see-through lenses

	Dataset 1	Dataset 2	Dataset 3
Point density (pts/m ²)	10	28	100
Number points (Billion)	5	7.1	80
Data size (GB)	112	159	1788

Table 2 Rendering performance in frames per second (fps) for different see-through lens techniques and varying numbers of rendered points. Each dataset is evaluated for a close and a far perspective. Points are rendered as so-called Point Splats (see Richter et al. 2015)

	Dataset 1		Dataset 2		Dataset 3	
	Far	Close	Far	Close	Far	Close
#Rendered points in million	2.32	0.50	3.42	0.85	4.85	1.04
No technique applied	51.84	214.32	32.27	138.67	23.01	108.63
Simple blending	49.65	210.84	30.39	135.84	21.73	105.96
Blueprints	48.18	208.31	29.84	134.76	21.09	104.85
Interactive lens	49.32	209.68	30.12	135.20	21.46	105.21
Halos	27.68	104.31	23.97	67.31	16.92	52.57

rate if halos are activated. However, interactive frame rates are still achieved for several million points. Since the applied out-of-core rendering approach (Richter et al. 2015) limits the number of rendered points per frame, the proposed see-through lenses can be applied for arbitrarily large 3D point clouds.

6 Conclusions and Future Work

We discussed how the exploration of heterogeneous, time-variant, and semantically rich 3D point clouds can be facilitated by taking into account the relevance of a point for different use cases. The relevance of a point can be assessed based on a point's spatial position or any additional per-point attribute such as surface category, timestamp, or the topology of the local point proximity. We introduce the concept of see-through lenses to interactive point-based rendering and show how such techniques can be integrated into existing rendering systems for massive 3D point clouds as additional post processing effects. Thus, we can improve the visual recognition of relevant, occluded objects within a 3D point cloud depiction by masking out less relevant occluders. The proposed see-through lenses can be configured and selected at runtime offering many degrees of freedom for graphics and interaction design. Therefore, our approach is highly adaptive and applicable to a variety of use cases and visualization tasks from different domains. For future work, we plan to improve the overall performance of the halo-based technique (e.g., by defining halos in screen space, thus, avoiding the costly preparation of halo masks). In addition, we also want to investigate how to efficiently apply alternative techniques to highlight occluded objects (e.g., multiple views or multi-perspective projections) to point-based rendering.

Acknowledgments This work was funded by the Federal Ministry of Education and Research (BMBF), Germany within the InnoProfile Transfer research group “4DnD-Vis” (www.4dndvis.de) and the Research School on ‘Service-Oriented Systems Engineering’ of the Hasso Plattner Institute. We would like to thank virtualcitySYSTEMS and the Faculty of Architecture at the Cologne University of Applied Sciences for providing datasets.

References

- Andújar, C., Vázquez, P., & Fairén, M. (2004). Way-Finder: Guided tours through complex walkthrough models. *Computer Graphics Forum*, 23(3), 499–508.
- Brosz, J., Carpendale, S., & Nacenta, M. (2011). The undistort lens. *Computer Graphics Forum*, 30(3), 881–890.
- Brosz, J., Samavati, F., Carpendale, S., & Sousa, M. (2007). Single camera flexible projection. In *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering* (pp. 33–42).
- Burtnyk, N., Khan, A., Fitzmaurice, G., Balakrishnan, R., & Kurtenbach, G. (2002). StyleCam: Interactive stylized 3D navigation using integrated spatial and temporal controls. In

- Proceedings of the 15th annual ACM symposium on User interface software and technology* (pp. 101–110).
- Canli, E., Thiebets, B., Höfle, B., & Glade, T. (2015). Permanent 3D laser scanning system for alpine hillslope instabilities. In *6th international conference on debris-flow hazards mitigation: Mechanics, prediction and assessment* (pp. 1–1).
- Elmqvist, N., & Tsigas, P. (2008). A Taxonomy of 3D occlusion management for visualization. *IEEE Transactions on Visualization and Computer Graphics*, *14*(5), 1095–1109.
- Gao, Z., Nocera, L., Wang, M., & Neumann, U. (2014). Visualizing aerial LiDAR cities with hierarchical hybrid point-polygon structures. In *Proceedings of the 2014 graphics interface conference* (pp. 137–144).
- Goesele, M. et al. (2010). Ambient point clouds for view interpolation. *ACM Transactions on Graphics*, *29*(4), 95:1–95:6.
- Goswami, P., Erol, F., Mukhi, R., Pajarola, R., & Gobbetti, E. (2013). An efficient multiresolution framework for high quality interactive rendering of massive point clouds using multi-way kd-trees. *The Visual Computer*, *29*(1), 69–83.
- Gross, M., & Pfister, H. (2007). *Point-based graphics*. Morgan Kaufmann Publishers Inc.
- Hasan, M., Samavati, F., & Jacob, C. (2014). Multilevel focus + context visualization using balanced multiresolution. In *2014 international conference on cyberworlds* (pp. 145–152).
- Koenig, K. et al. (2015). Comparative classification analysis of post-harvest growth detection from terrestrial LiDAR point clouds in precision agriculture. *ISPRS Journal of Photogrammetry and Remote Sensing*, *104*(0), 112–125.
- Lafarge, F., & Mallet, C. (2012). Creating large-scale city models from 3d-point clouds: A robust approach with hybrid representation. *International Journal of Computer Vision*, *99*(1), 69–85.
- Leberl, F. et al. (2010). Point clouds: Lidar versus 3D vision. *Photogrammetric Engineering and Remote Sensing*, *76*(10), 1123–1134.
- Mitra, N., & Nguyen, A. (2003). Estimating surface normals in noisy point cloud data. In *19th annual symposium on computational geometry* (pp. 322–328).
- Nebiker, S., Bleisch, S., & Christen, M. (2010). Rich point clouds in virtual globes—A new paradigm in city modeling? *Computers, Environment and Urban Systems*, *34*(6), 508–517.
- Nienhaus, M., & Döllner, J. (2004). Blueprint Rendering and ‘Sketchy Drawings’. *GPU Gems, II*, 235–252.
- Pasewaldt, S., Semmo, A., Trapp, M., & Döllner, J. (2014). Multi-perspective 3D panoramas. *International Journal of Geographical Information Science*, *28*(10), 2030–2051.
- Preiner, R., Jeschke, S., & Wimmer, M. (2012). Auto splats: Dynamic point cloud visualization on the GPU. In *Proceedings of eurographics symposium on parallel graphics and visualization* (pp. 139–148).
- Richter, R., Behrens, M., & Döllner, J. (2013). Object class segmentation of massive 3D point clouds of urban areas using point cloud topology. *International Journal of Remote Sensing*, *34* (23), 8408–8424.
- Richter, R., & Döllner, J. (2014). Concepts and techniques for integration, analysis and visualization of massive 3D point clouds. *Computers, Environment and Urban Systems*, *45*, 114–124.
- Richter, R., Discher, S., & Döllner, J. (2015). Out-of-core visualization of classified 3D point clouds. In *3d geoinformation science: The selected papers of the 3D geoinfo 2014* (pp. 227–242).
- Ropinski, T., Hinrichs, K., & Steinicke, F. (2005). A solution for the focus and context problem in geo-virtual environments. In *Proceedings of the 4th ISPRS workshop on dynamic and multi-dimensional GIS* (pp. 144–149).
- Saito, T., & Takahashi, T. (1990). Comprehensible Rendering of 3-D Shapes. *SIGGRAPH Computer Graphics*, *24*(4), 197–206.
- Sigg, S., Fuchs, R., Carnecky, R., & Peikert, R. (2012). Intelligent cutaway illustrations. *IEEE Pacific Visualization Symposium, 2012*, 185–192.

- Sonnet, H., Carpendale, S., & Strothotte, T. (2004). Integrating expanding annotations with a 3D explosion probe. In *Proceedings of the working conference on advanced visual interfaces* (pp. 63–70).
- Tomljenovic, I., Höfle, B., Tiede, D., & Blaschke, T. (2015). Building extraction from airborne laser scanning data: An analysis of the state of the art. *Remote Sensing*, 7(4), 3826–3862.
- Trapp, M., Glander, T., Buchholz, H., & Döllner, J. (2008). 3D generalization lenses for interactive focus + context visualization of virtual city models. In *12th international conference on information visualisation* (pp. 356–361).
- Vaaranemi, M., Freidank, M., & Westermann, R. (2013). Enhancing the visibility of labels in 3D navigation maps. In *Progress and new trends in 3D geoinformation sciences* (pp. 23–40).
- Wang, L., Zhao, Y., Mueller, K., & Kaufman, A. (2005). The magic volume lens: An interactive focus + context technique for volume rendering. *IEEE Visualization, 2005*, 367–374.
- Xu, H., Nguyen, M., Yuan, X., & Chen, B. (2004). Interactive silhouette rendering for point-based models. In *Eurographics symposium on point-based graphics* (pp. 13–18).
- Yu, J., & Turk, G. (2013). Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Transactions on Graphics*, 32(1), 5:1–5:12.

Representation of CityGML Instance Models in BaseX

Sabine Koch and Marc-O. Löwner

Abstract The Open Geospatial Consortium standard CityGML is an application schema of GML 3.1.1 for the representation, storage and exchange of semantic-rich virtual 3D city models. Here we assess the feasibility of storing, querying and updating CityGML models in the native XML database system BaseX. The features and performance of BaseX are compared with the implementation of the 3DCity-Database which stores CityGML models in a relational database system. The results show that BaseX is a fast, flexible and intuitive tool to store and query even large CityGML documents. Its main advantage is the schema-oblivious storage mechanism that allows schema changes without changes to the database layout and the fast import and export of CityGML models. Using the 3DCityDatabase to manage CityGML data on the other hand is a better choice when spatial analysis and integration with third party software are demanded.

Keywords CityGML · 3D city models · Native XML databases · BaseX

1 Introduction

3D city models are increasingly used in urban planning for analysis, participation, management and visualization purposes (Ross 2010; Löwner et al. 2013). The OGC standard for virtual 3D city models CityGML, an application schema of GML, was designed to improve the storage and exchange of semantic-rich 3D city models by providing a standardized data model.

S. Koch (✉)

Technische Universität Berlin, Berlin, Germany
e-mail: Sabine.koch@posteo.de; sabine.koch87@gmail.com

M.-O. Löwner

Department Methods of Geoinformation Science,
Technische Universität Berlin, Berlin, Germany
e-mail: m.loewner@tu-berlin.de

This paper examines the approach to manage CityGML data by storing, querying and updating them in the native XML database system BaseX. The features and performance of BaseX are compared with the implementation of the 3DCityDatabase, which offers a relational database schema to store CityGML data as well as an Importer-Exporter to load CityGML data into the database and export it back to the CityGML format.

The OGC Standard 12-019 CityGML 2.0 is an “XML-based format for the storage and exchange of virtual 3D city models” (Gröger et al. 2012). It allows for the modeling of cities and their corresponding city objects (e.g. buildings, bridges) in five levels of detail (LODs) (Gröger et al. 2012). Applications can combine the core module with several extension modules as needed or define application domain extensions (ADEs) for feature types or properties not covered by the standard. The modular approach and the possibility to represent features in different LODs make CityGML a very flexible standard that is adjustable to many use cases (Ross 2010; Löwner et al. 2013). Features in CityGML are modeled using a profile of the GML 3.1.1 geometry model, which represents 3D objects using boundary surfaces. In CityGML this model is extended by scene graph concepts. The standard incorporates topology in the model by allowing geometry objects to be shared by features or geometries. These relationships are represented in the data by XLinks containing a reference to the shared geometry. Through the appearance module, textures can be applied to buildings.

CityGML models can be stored in the 3DCityDatabase that was developed to facilitate the management of complex CityGML models (Stadler 2008). The 3DCityDatabase offers a relational database schema to store CityGML data in Oracle or PostgreSQL, two popular relational database management systems with spatial extensions. It includes an Importer-Exporter to load CityGML data into the database and export it back to the CityGML format. An object-relational mapping is necessary to transfer CityGML features to a relational database. As a result the data model represented in the 3DCityDatabase is less complex and less flexible as the data model described in the CityGML Encoding Standard. Additionally, a review of the standard and future schema changes directly lead to changes of the relational schema as well as the mapping algorithm of the 3DCityDatabase.

Databases that follow the approach to map the schema of an XML document to a previously designed database schema are called XML-enabled database (Bourret 2005). The more complex, hierarchical, and irregular the XML data are, the more difficult is the mapping to a relational schema and the storage in an XML-enabled database (cf. Pavlovic-Lažetic 2006; Harms 2008; Zhang et al. 2008). Due to the object-relational impedance mismatch, this mapping process is also often connected with information loss (cf. Stadler 2008; Zhang et al. 2008; Kudrass and Conrad 2002).

The object-relational mapping and its drawbacks can be avoided by using native XML database systems to manage complex XML data. Native XML database systems (NXDs) are schema-oblivious, i.e. the XML document is mapped to a fixed database schema that is directly based on the general XML storage model (Bourret 2005).

According to (Staken 2001), NXDs have the following characteristics:

- They define a storage model for an XML document that is based on the general document structure, not on the actual data in the document.
- They include at least elements, attributes, PCDATA and the document order.
- The fundamental storage unit is the XML document.
- The actual physical storage model is not strictly defined; it can for instance be a relational database or a file system.

In summary, NXDs can store document-oriented, hierarchic and semi-structured data more efficiently than relational database systems. They offer a flexible storage model, different search and query capabilities (query languages like XPath and XQuery, support of full text searches) and changes in the data schema do not pose a problem (Staken 2001).

Although CityGML documents can be defined as data-oriented, rather than document-oriented, CityGML instances can have a highly hierarchical, nested and complex structure (Stadler 2008), which makes them a suitable candidate for the storage in an NXD.

This paper examines an alternative approach to manage CityGML data by storing, querying, and updating them in the native XML database system BaseX. The features and performance of BaseX are compared with the implementation of the 3DCityDatabase in PostgreSQL with PostGIS, a spatial-enabled OpenSource relational database system.

2 Existing Research on the Representation of Spatial Data in NXDs

A number of studies evaluated the use of NXDs and XML query languages for storing and querying XML-based spatial data.

Zhang et al. (2008) stressed the advantages of a native XML storage for GML data with their hierarchic and nested structure. They designed the GML storage model G2NXDBM, using the Ipedo native XML database by Ipedo Inc. USA.

Córcoles and González (2009) compared a structure-mapping approach (LegoDB, XML-enabled) with two different model-mapping approaches (monetDB/XQuery and XParent, native storage approach) for the storage of GML data. They concluded that the structure mapping approach of LegoDB provides the best performance.

Amirian and Alesheikh (2008) developed a geospatial web service, using native database technologies, namely Software AGs Tamino 4.4, and evaluated the performance for storing and retrieving GML data in comparison to Microsoft Server 2000 as XML-enabled database system. The evaluation was performed with three GML datasets containing 1000, 10000 and 100000 features respectively. The authors concluded that the NXD performs better for large amounts of data, as there are no join operations necessary that could decrease the performance of the system. A drawback is the larger disk space occupied by the NXD.

Almendros-Jiménez et al. (2010) adapted the XPath query language to query semantic spatial relations in GML documents. They implemented a system that stores GML data in a relational database management using PostGIS. The main characteristic of their query approach is following the GML schema in the query definition and not the syntactic structure of the GML instance document. The queries are then translated into SQL to be processed by PostGIS.

Concerning the storage, updating, and querying of GML data in BaseX no scientific research exists so far, but several studies on the performance of the system compared with relational database systems or other NXDs have been conducted.

In a comparison involving the two relational database systems MySQL and Oracle and the two NXDs Sedna and BaseX, Risse and Leunissen (2010) came to the conclusion that Oracle performs best for all kinds of datasets and queries. The NXDs were more flexible but are not considered suitable for handling large datasets (>1 million records). BaseX performed better than Sedna in terms of database size, but query performance was in three of five queries poorer than Sedna when the largest dataset was queried.

Freire et al. (2012) assessed the performance of several NXDs for storing and querying electronic health records. The evaluated NXDs are eXist, BaseX, Sedna and Berkeley DB XML. Amongst the four NXDs, BaseX is the one with the best performance, both in terms of database size and query time. However, all NXDs perform poorly compared to MySQL with its relational storage approach.

Although there exists some research on the storage of CityGML in NoSQL databases (e.g. Mao et al. 2014 used MongoDB to store CityGML 3D city models), the storage of CityGML data in a native XML database has not yet been scientifically assessed. Current research on the topic of storage and querying general GML data in native XML databases on the other hand exists and indicates that NXDs might be a possible alternative with many advantages for the storage of CityGML documents, but with possible drawbacks concerning the performance compared to a relational database management system.

3 Working Environment and Method

3.1 *BaseX*

Currently available NXD implementations are Tamino XML Server, Berkeley DB XML, the free and Open Source software applications monetdb/XQuery, Sedna and eXistdb, and BaseX, an open source product originally developed at the University of Konstanz. The development and maintenance of monetdb/XQuery and Sedna stopped several years ago, therefore these products were not considered further for this research. Tamino XML server and Berkeley XML DB were excluded as well due to them being non-open source products with a comparably low amount of documentation and community support. Therefore, the Open Source databases eXistdb and BaseX were considered for testing the suitability of NXDs for

managing CityGML data. BaseX was chosen over eXistdb considering its more extensive set of features, the intuitive GUI and its good performance in several studies (cf. Freire et al. 2012; Grün 2010).

The open source NXD BaseX is a “light-weight, high-performance and scalable XML Database engine and XPath/XQuery 3.1 Processor” (BaseX Team 2014). BaseX is capable of dealing with large XML instances (Grün 2010), and provides an intuitive and interactive GUI.

Besides XML, BaseX also accepts XHTML, HTML, TEXT, JSON, JSONML and CSV files as input data.

According to (Bourret 2010) NXDs can store XML data as documents or using a node-based approach. BaseX uses a fixed table structure to store XML data and therefore follows the node-based storage approach.

The basic encoding of XML nodes in BaseX is as *pre-dist-size* tuples:

- The *pre* value is defined by the position of the node in the document when traversing the XML document in pre-order (starting at the root then traversing the child nodes from left to right).
- The *dist* property represents the relative distance (number of nodes) to the parent node.
- The *size* property contains the number of descendant nodes.

This encoding is used to map the XML document to a table representation. Attributes are inlined, i.e. they are treated as nodes and stored in the same table as the other nodes.

The performance of BaseX is compared to the 3DCityDatabase version 3.0, implemented in PostgreSQL with PostGIS, a spatial-enabled OpenSource relational database system.

3.2 Test Data

To generally test BaseX’ capabilities to manage CityGML models, two datasets of different LODs served as test data. For the performance evaluation the 3D city model of Berlin, the largest CityGML model obtainable so far, was used as input dataset. Several datasets with different sizes in terms of disk space, number of features, and number of separate files were generated from the original Berlin city model. Table 1 lists the characteristics of the input data.

Table 1 Input datasets

Name	Potsdam	House	Berlin_1%	Berlin_10%	Berlin_100%
File size	1.3 MB	20.4 MB	137.5 MB	1353.9 MB	13901 MB
Number of files	1	2	5	50	500
CityObjectMembers	97	1	5440	54365	543652
LODs	LOD 1 (simple solids)	LOD 4 (includes interior)	LOD 2	LOD 2	LOD 2

The performances of both systems were compared in terms of import, export, query, and update time as well as database size. The import, query, and update tests were executed on database instances with and without indexes. For BaseX the tests and report generation was automated using BaseX' Python API. For PostgreSQL the 3DCityDB Importer/Exporter was used as well as psycopg2, a PostgreSQL adapter for Python. The tests were iterated ten times, except for the import tests of the Berlin_10% and Berlin_100% datasets in PostgreSQL, which were reduced to five times (Berlin_10%) and one time (Berlin_100%). The evaluated queries were:

- Non-transactional queries:
 - select one feature by gml:id
 - select all buildings
 - select a set of features by address
 - count all building features
- Transactional queries:
 - Insert one cityObjectMember
 - Insert many elements as child node to a selection of cityObjectMembers
 - Update the value of a set of elements.

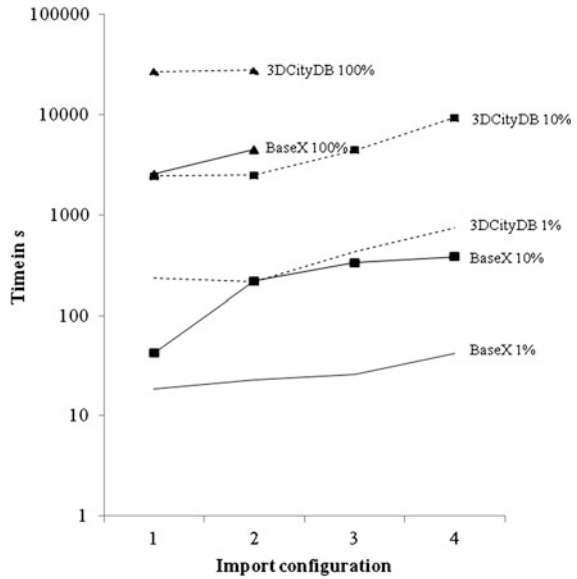
4 Results

4.1 Import/Export

Importing single or multiple CityGML files into BaseX was unproblematic and modifications of the input data were not necessary. Creating a database instance in BaseX, importing data and adding indexes can be done in one step. The resulting database consists of collections, where each collection represents an imported XML document and can be seen as the equivalent to a table in a relational database system (Bourret 2005). BaseX exports the data to the same file structure as before the import. No changes in file content could be observed when comparing the exported data to the original input files. However in the case of the dataset “House” the resulting file was about 1 MB (5 %) smaller than the original file. This difference resulted probably from a different serialization of non-printable characters. To export a CityModel object with only a selection of cityObjectMembers from BaseX, it is necessary to first create an empty CityModel element and insert the selected objects.

Before importing data into the 3DCityDatabase, an empty PostGIS-enabled database has to be created and the tables for the 3DCityDatabase have to be added. A selection can be specified for the import. Additionally, indexes can be created and the XML can be validated during the import. Export of selections is possible based on the criteria gml:id, gml:name, bounding box, or FeatureType. The exported data is merged into one file. Figure 1 display the import times for each of the three input

Fig. 1 Import times for different dataset sizes (1 %, 10 %, 100 % of the original input dataset)
 (Configurations: 1—BaseX: no indexes, internal parser, 3DCityDB: no indexes, no appearances or textures; 2—BaseX: text and attribute index, internal parser, 3DCityDB: indexes, no appearances or textures; 3—BaseX: full text index, internal parser, 3DCityDB: indexes, no appearances or textures; 4—BaseX: all indexes, java parser, 3DCityDB: indexes, appearances and links to textures)



datasets with and without indexes. In all cases more time is needed using the 3DCityDatabase Importer in comparison to the import process in BaseX.

4.2 Storage

BaseX offers a GUI with different visualization options besides the textual representation of the data, such as a tree map visualization (Johnson and Shneidermann 1991), a presentation of the XML hierarchy as a tree, or a scatterplot diagram. In the 3DCityDatabase the data is represented in 60 separate tables.

The original file size of the CityGML input data was compared with the database size after adding the data with different import options. The results show that the full text index has the highest influence on the database size in BaseX compared to the other indexes. The database size for all three datasets and the different input configurations for BaseX and the 3DCityDatabase as well as the size of the input data are displayed in Fig. 2.

The appearance module of CityGML defines classes for including images as textures in a CityGML 3D city model. Image data can be stored in BaseX by importing them as raw, i.e. binary data. The image data is stored to a separate folder; the storage size of the original files and the files in the database stays the same. The raw files are not displayed in any of the visualization options offered by the GUI, but are listed in the Database Properties window. The import of textures into the 3DCityDatabase can be specified during the import. It is also possible to

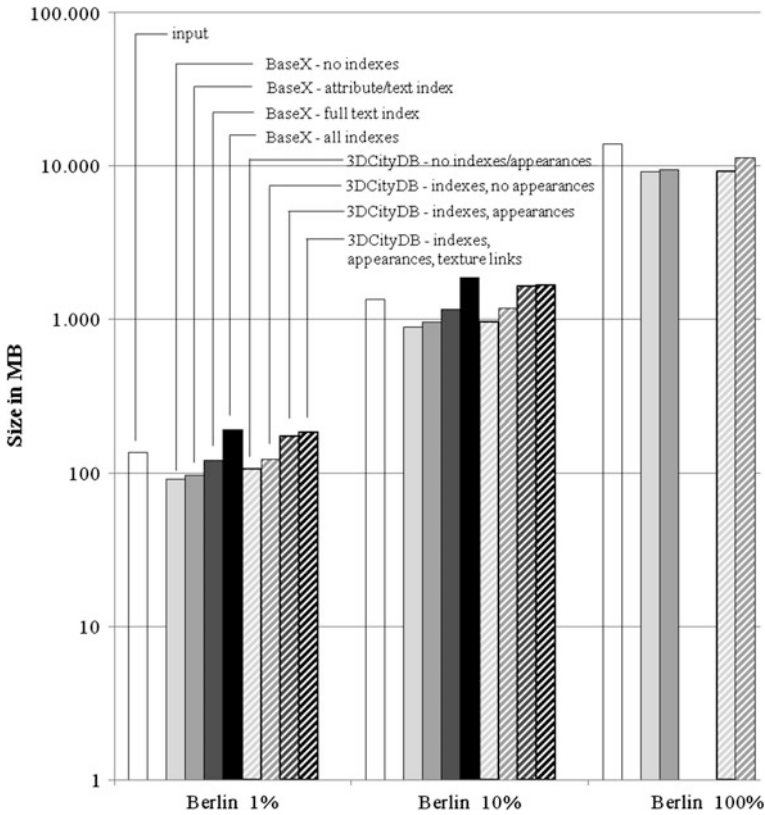


Fig. 2 Databases size for different database configurations (see Fig. 1)

add georeferenced raster data using the PostGIS raster data type. Storing textures along with the GML data is therefore technically possible in BaseX but easier to manage using the 3DCityDatabase.

4.3 Querying

BaseX supports the XML Path Language (XPath) 3.0 and the XML Query Language (XQuery) 3.1 as well as the XQuery Update Facility 1.0 and the XQuery Full Text Recommendation.

XPath uses path expressions to navigate through the XML tree structure. A path expression is formed by one or more steps, each step contains information on the direction of the expression and a node test that specifies the type and name of the nodes that the expression should return. Additionally, predicates can be used for filtering the results. The XML Query Language XQuery is a superset of XPath.

Besides path expressions, it also supports element constructors, FLWOR expressions, list expressions, conditional expressions, quantified expressions, and datatype expressions.

In addition to the set of operations and functionalities described in the W3C Recommendation “XPath and XQuery Functions and Operators 3.1”, BaseX contains several modules that add further functionalities to XPath, partly using libraries from the EXPath and EXQuery Projects. Queries on CityGML can be performed on the whole range of semantic information stored in the features of the model. However, the use of spatial queries is restricted. Spatial queries test relationships between features (intersect/contain/touch/etc.) or perform operations on one feature (calculate area/length/buffer calculation). BaseX provides a Geo Module with spatial functions using the ExPath geo module which is based on the JTS Topology Suite. However, this module only supports operations on geometries conforming to the OGC Simple Features (SF) data model and cannot be applied to the more complex CityGML data. BaseX therefore offers no out-of-the-box spatial query mechanism for CityGML data. Still, simple spatial queries can be executed on the `gml:boundedBy` attribute that is included in several feature types in CityGML 1.0 documents. The `gml:boundedBy` element contains an `gml:Envelope` element, which holds the two opposite points of the bounding box as child elements (`gml:upperCorner/gml:lowerCorner`). Based on this, bounding box selections as well as simple queries on topological relationships can be implemented.

BaseX supports the following index structures:

- Name index—indexes tags and attribute names. Internally, the names are replaced by ids, which can reduce the database size significantly (Grün 2010).
- Path index/Path summary—the successor to DataGuides that were introduced by (Goldman and Widom 1997), serving as structural summaries to speed up queries in semi-structured databases; the path summary is constructed with all unique paths in a document.
- Resource index—indexes the *pre*-values of all nodes in the database.
- Text index—improves queries on the strings stored in text nodes in the database
- Attribute index—indexes all attribute values
- Full text index—is based on the XQuery Full Text language extension, uses text tokens as keys together with a compressed trie structure

The indexes significantly impact the database size (see Fig. 2) and query times (see Table 2).

Table 2 Querying 1 feature by id in BaseX with different indexes—time in s

	Berlin_1%	Berlin_10%	Berlin_100%
No indexes	0.7442	7.4200	307.2323
Attribute/value index	0.0162	0.0172	0.0317
Full text index	0.7563	7.3808	–

4.4 Updates

In the case of NXDs the storage structure often relies on a numbering scheme (e.g. the *pre-size-dist* structure for BaseX, where the *pre* value equals the row number of the node). This can make updates very cost-intensive (cf. Bächle and Härder 2008) as every structural update not only affects the inserted/deleted node, but also values of following and preceding nodes. BaseX implements the XQuery Update Facility, which was released as a W3C Recommendation in 2011 and describes expressions to change XML data persistently.

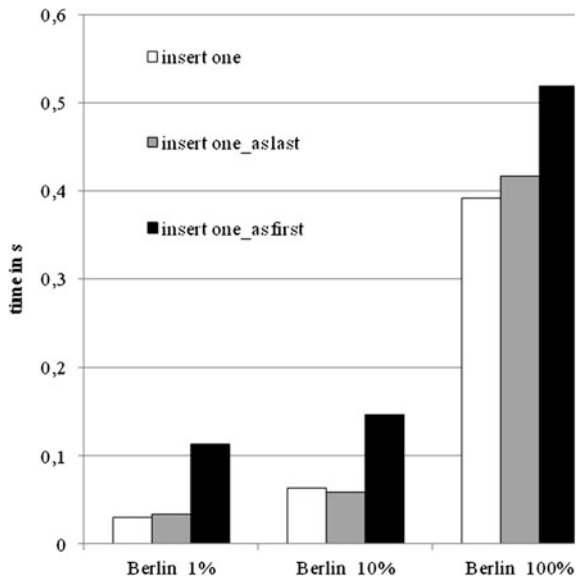
All update statements of a transaction are stored in a Pending Update List (PUL). The PUL collects all update statements of a query, including database updates, optimizes them and then applies them in a defined order. To make updates more efficient, the data is stored on fixed-sized pages.

The impact of inserting a feature without specifying the position, as first feature and as last is displayed in Fig. 3.

4.5 Transaction Management, Version Control and Web Integration

Transactions, e.g. every command or query in BaseX, are controlled by the Transaction Monitor when running BaseX in client-server mode. This permits only one updating transaction per database at a time. As of yet, BaseX does not provide a

Fig. 3 Times to insert one feature at different positions into the database



version or revision control system. The only way to keep track of changes in the database is therefore by creating backups and explicitly storing several versions of a database at once. BaseX offers the possibility to access NXDs via a REST interface and to create XQuery-based web services and applications using the RESTXQ API (Retter and Grün 2015). Kramis et al. (2009) implemented a web service using REST and a native XML data storage. Using REST or RESTXQ, a database storing CityGML models could be queried online.

5 Discussion and Outlook

For the storage of CityGML data in a relational database system, the 3DCityDatabase was developed, a product that provides the necessary scripts to create the relational database schema as well as an Import-Export tool to facilitate addition and retrieval of data to the database. Despite its convenient and easy use, the product has some drawbacks: The schema used in the 3DCityDatabase is a simplified version of the OGC CityGML Encoding Standard and is therefore lacking some of the depth and flexibility of the standard. Changes in the schema of the standard lead to changes in the whole application logic as well as the database. Addressing these drawbacks is the advantage of BaseX: it pursues a schema-oblivious storage mechanism, meaning that data is stored based on the tree structure of the XML data, not on the semantic structure of the data. Schema changes therefore do not pose a problem and a simplification of the standard to fit the data into the database is not necessary. These benefits were the overall incentive for the current research on the advantages and disadvantages of storing CityGML models in BaseX.

The results show that the performance of BaseX can compete with the 3DCityDatabase in many aspects. Still, the NXD is lacking certain important functionalities, which restricts its usability for managing CityGML models.

Due to the reduced number of imports of the Berlin_10% and Berlin_100% datasets into the 3DCityDatabase, the validity of the performance data is limited. The additional time needed to set up an empty 3DCityDatabase was not included in the performance evaluation.

Creating a database and importing the data are necessarily the first steps when CityGML models have to be stored in a database system. Relevant issues in this step are the speed of the import process, the complexity of setting up and importing data, the possibility to restrict the imported data by semantic or spatial criteria and no information loss from file to database.

In the 3DCityDatabase and PostgreSQL in general, an empty database has to be set up, before data can be added. With the scripts provided by the 3DCityDatabase, this is fairly easy, but it is still an extra work step. In BaseX on the other hand, database creation and import of data can be done in one step, but directly importing a subset of a file is not possible. The 3DCityDatabase Importer offers a range of import settings like adding indexes (spatial/normal), importing appearances and/or

textures and validating the XML data during the import. BaseX provides only the possibility to add indexes, though spatial indexes are not supported. The other possible import criteria (chop whitespaces, use java/internal parser, strip namespaces) are not relevant or useful for the work with CityGML data.

Data export is mainly important when data should be exchanged with other systems. The data exported from database has to be a valid CityGML document, no information should be lost from the database to the exported file, and the definition of meaningful export criteria should be possible. Both systems fulfill these criteria, but in BaseX exporting a selection requires several steps. However, the export from BaseX is much faster, as no conversion from relations to objects is necessary.

In contrast to prior expectations, the import of the Berlin datasets took much longer with the 3DCityDatabase Importer than with BaseX. BaseX parses the files and stores the data in the database according to the underlying XML data structure, no conversion between data types is necessary. PostgreSQL has to convert the object data to the relational schema and its data types. It also processes the data, for example by resolving XLinks and converting the GML geometry to PostGIS geometry. This process causes the longer import times but adds a layer of information and usability to the data that BaseX cannot offer.

The GUI of BaseX is very user friendly. Exploring the data—at least for small to medium sized databases—is simplified by the different visualization options, especially the tree map and tree visualization. The 3DCityDatabase on the other hand offers better support for raster data.

The query approaches of the evaluated database systems are entirely different. Where XPath return nodes and their respective child nodes, SQL return fields from records in a table. Creating a subset of a CityGML model is easy to achieve with XPath and XQuery in BaseX, but in the 3DCityDatabase the Exporter tool has to be used. With the Exporter the user is restricted to a small set of selection criteria, which may nonetheless be sufficient for most applications. In PostgreSQL, when information from several tables is required in the result, many join operations are necessary, which leads to bulky queries and long query times.

Queries via the Exporter take much longer than queries in BaseX. This difference can be explained with the object-relational mapping performed during the export.

In BaseX, elements can be queried without having to know the exact schema of the model, using for example the descendant-axis of XPath. The restriction to queries along the XML hierarchy in BaseX is on the other hand a drawback, as CityGML models are not pure trees. Elements are connected not only by relationships defined by the XPath-axes, but also via XLinks. Querying along XLinks is not supported in XQuery (cf. Behrend et al. 2006), and can only be achieved indirectly.

BaseX supports the XQuery Update Facility, it is therefore possible to add, delete, and change the CityGML features stored in the database. To add a new cityObjectMember to the 3DCityDatabase, it can be imported with the Importer. Deletion of features and updating feature information is not possible via the Importer and instead has to be done directly in the database system.

According to (Ross 2010) one of the main uses of 3D city models is spatial analysis in urban planning scenarios. PostGIS offers a variety of spatial analysis functions. BaseX on the other hand is not a spatial database. Though it has a basic set of spatial functionalities for GML Simple Features data, these cannot be applied to CityGML models. However, it was possible to create a bounding box functions that yields results very similar to the bounding box function of the 3DCityDatabase Exporter. This restriction of BaseX is probably the bottleneck for its use to store and manage CityGML models. If the use case contains spatial analysis tasks that exceed operations that can be based on the bounding box of features, it will not be an option to choose BaseX, unless it can be combined with other tools to import and process the GML geometry objects.

6 Conclusion

Relational database management systems have dominated the field of data storage for the last decades. Lately, NoSQL databases have gained importance and play an important role especially for storing large amounts of irregular data. Native XML databases emerged around the year 2000, but were soon considered to be outperformed by relational database systems in most applications, which led to a decrease in interest in and development of these systems.

Today with improved storage and query methods NXDs might again be a competitor to relational database systems when it comes to storing XML data with complex and changing schemas. Therefore the performance of BaseX for storing CityGML data was tested.

The evaluation was performed in two steps: the first step comprised a general assessment of the usage of BaseX for storing, querying and updating CityGML models. It is well suited to store CityGML datasets as well as perform semantic queries on them. Although the import, storing and export of CityGML models can be done in an intuitive way, also a range of limitations of BaseX became visible in this evaluation step. Most importantly, BaseX is not a spatial database. Though it supports basic spatial queries on features complying with the GML SF profile, queries and operations on the more complex 3D city models are not directly possible. Integration with GIS software does not exist so far, integration with other tools to process GML data would have to be done by the user.

On the other hand, the 3DCityDatabase with PostGIS supports the storage of georeferenced raster data and offers many possibilities for spatial analysis. However, setting up a 3DCityDatabase and importing data is a lot more time-intensive than setting up a database in BaseX and adding CityGML data.

The second step in the evaluation comprised the quantitative performance comparison of the two implementations in terms of speed of different operations and database size. The results are quite clear: BaseX outperforms the relational database regarding the import of data and export of databases and features well as the database size for different input data set. The main reason for this lies in the data

conversion from XML to the relational schema that is time-intensive in the 3DCityDatabase, but not necessary in BaseX.

One of BaseX' main advantages is that schema changes do not pose a problem. This is particularly relevant considering the current development of CityGML 3.0 which will probably lead to extensive changes compared to version 2.0 (Löwner et al. 2015). Next to extensions, general concepts like a new Level of Detail concept are under review (Löwner et al. 2013; Benner et al. 2013), which will lead to fundamental schema adaption. For the 3DCityDatabase, this schema change can lead to extensive changes in the relational schema and the application itself, while at the same time having to ensure the compatibility with models in older versions of the standard. Due to the schema-oblivious storage method of BaseX, changes in the data schema do not lead to changes in the database schema. The same holds true for storing and querying data from ADEs to the database, which is possible in BaseX without any schema adaptations.

In conclusion BaseX can be a valuable alternative in the following cases:

- quick assessment of and overview on datasets
- emphasis on (data) visualization
- fast creation of subsets based on semantic criteria or bounding boxes
- frequent schema changes
- adding data from ADEs

Since March 2013, the Berlin City model, an instance of CityGML 1.0, is available as open data and can be downloaded online. This makes CityGML 3D city models available to a broader public, to which BaseX can serve as a user-friendly and intuitive alternative to the storage in PostgreSQL or Oracle.

References

- Almendros-Jiménez, J., Becerra-Terón, A., & García-García, F. (2010). Development of a query language for GML based on XPath. In *Proceedings of the 6th international workshop on automated specification and verification of web system, 2010*, pp. 51–64.
- Amirian, P., & Alesheikh, A. A. (2008). Publishing geospatial data through geospatial web service and XML database system. *American Journal of Applied Sciences*, 5(10), 1358–1368.
- Bächle, S., & Härder, T. (2008). Realizing fine-granular and scalable transaction isolation in native XML databases. In *Proceedings of the SYRCODIS 2008 colloquium on databases and information systems, Saint-Petersburg, Russia, May 29–30, 2008*.
- BaseX Team. (2014). BaseX documentation version 7.9. Publication date 2014-06-27.
- Behrend, E., Fritzen, O., & May, W. (2006). Querying along XLinks in XPath/XQuery: Situation, applications, perspectives. *Presentation at QLQP- Query Languages and Query Processing Munchen*, 31(3), 2006.
- Benner, J., Geiger, A., Gröger, G., Häfele, K. H. & Löwner, M.-O. (2013). Enhanced LoD concepts for virtual 3D city models. In *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences* (Vol. II-2/W1, pp. 51–61).
- Bourret, R. (2005). Going native: Making the case for XML databases. Retrieved Jan 17, 2015, from <http://www.xml.com/pub/a/2005/03/30/native.html>.

- Bourret, R. (2010). XML database products—native XML database. Retrieved April 10, 2015, from <http://www.rpbouret.com/xml/ProdsNative.htm>.
- Córcoles, J. E., & González, P. (2009). GML as database: Present and future. In *Handbook of research on geoinformatics*. Chapter 1.
- Freire, S. M., Sundvall, E., Karlsson, D., & Lambrix, P. (2012). Performance of XML databases for epidemiological queries in archetype-based EHRs. In *Scandinavian conference on health informatics 2012, October 2–3, Linköping, Sweden*.
- Goldman, R., & Widom, J. (1997). DataGuides: enabling query formulation and optimization in semistructured databases.
- Gröger, G., Kolbe, T. H., Nagel, C., Häfele, K.-H. (2012). OGC city geography markup language (CityGML) encoding standard, version 2.0, OGC Doc No. 12-019, Open Geospatial Consortium.
- Grün, C. (2010). Storing and querying large XML instances. Dissertation an der Universität Konstanz.
- Harms, J. (2008). Räumliche datenbanken und GML. Bachelorarbeit an der technischen universität Wien. Ausgeführt am Institut für Rechnergestützte Automation Forschungsgruppe Industrial Software. Wien, February 2008.
- Johnson, B., & Shneiderman, B. (1991). Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2nd conference on visualization '91*, pp. 284–291.
- Kramis, M., Gabathuler, C., Fabrikant, S. O., & Waldvogel, M. (2009). An XML-based infrastructure to enhance collaborative geographic visual analysis. *Cartography and Geographic Information Science*, 36(3), 281–293.
- Kudrass, T., & Conrad, M. (2002). XML-based data management and multimedia engineering. In *EDBT 2002 workshops lecture notes in computer science* (Vol. 2490, pp. 210–227).
- Löwner, M.-O., Benner, J., & Gröger, G. (2015). Aktuelle trends in der entwicklung von CityGML3.0. Recent trends in the development of CityGML 3.0 In Seyfert, E. et al. (Eds.), *Geoinformationen öffnen das Tor zur Welt*, 34. Wissenschaftlich-Technische Jahrestagung der DGPF. 26.-28.03.2014, DGPF-Tagungsband 23, Hamburg.
- Löwner, M.-O., Benner, J., Gröger, G. & Häfele, K. H. (2013). New concepts for structuring 3D city models—an extended level of detail concept for CityGML buildings. In MURGANTE, B. et al. (Eds.), *13th international conference on computational science and its applications, Ho Chi Minh City, Vietnam, June 24–27, 2013, Part III, LNCS* (Vol. 7973, pp. 466–480). Berlin: Springer.
- Löwner, M.-O., Casper, E., Becker, T., Benner, J., Gröger, G., Gruber, U., Häfele, K.-H., Kaden, R., & Schlüter, S. (2013). CityGML 2.0—ein internationaler standard für 3D-stadtmodelle, Teil 2: CityGML in der Praxis. CityGML 2.0—an international standard for 3D city models, part 2: CityGML in practice. *Zeitschrift für Geodäsie, Geoinformation und Landmanagement*, 2, 131–143.
- Mao, B., Harrie, L., Cao, J., Wu, Z., & Shen, J. (2014). NoSQL based 3D city model management system. In *The international archives of the photogrammetry, remote sensing and spatial information sciences, volume XL-4, 2014. ISPRS technical commission IV symposium, 14–16 May 2014, Suzhou, China*.
- Nagel, C., Stadler, A. (2008). Die oracle-schnittstelle des Berliner 3D-stadtmodells. In *Entwicklerforum Geoinformationstechnik*.
- Pavlovic-Lažetic, G. (2006). Native XML databases vs. relational databases in dealing with XML documents. *Kragujevac Journal of Mathematics*, 30, 181–199.
- Retter, A., & Grün, C. (2015). RESTXQ 1.0: RESTful annotations for XQuery. An EXQuery project. Unofficial draft 23 February 2015. Retrieved April 10, 2015, from <http://exquery.github.io/exquery/exquery-restxq-specification/restxq-1.0-specification.html>.

- Risse, J. E. & Leunissen, J. A. M. (2010). A comparison of database systems for XML-type data. In *Silico biology* (Vol. 10, pp. 193–205). IOS Press.
- Ross, L. (2010). Virtual 3DCMs in ULM. Dissertation. TU Berlin.
- Staken, K. (2001). Introduction to native XML databases. Retrieved Oct 31, 2001, from <http://www.xml.com/pub/a/2001/10/31/nativexmlb.html>. [Last updated:31/10/2001].
- Zhang, S., Gan, J., Xu, J., & Lv, G. (2008). Study on NXD based GML storage model. In *The international archives of the photogrammetry, remote sensing and spatial information science* (Vol. 37, Part B4). Beijing.

A Spatio-Semantic Query Language for the Integrated Analysis of City Models and Building Information Models

S. Daum, A. Borrmann and T.H. Kolbe

Abstract Distinct semantic-geometric data models are applied in architecture/engineering/construction and geospatial domains. However, to make decisions within complex urban planning and engineering tasks these two domains and their data models must be combined. Currently, the necessary joint information is created by converting data between the two domains. Because the employed modelling differs conceptually, there is a high risk of information loss during these conversions. To overcome this issue, we present a spatio-semantic query language that allows analysis of IFC building information models and geospatial CityGML models in an integrated context. Rather than converting between IFC and CityGML, a holistic information space is realized by an intermediate layer that abstracts from the two schemas of spatio-semantic modelling.

1 Introduction

The examination of data from diverging scales is necessary for various planning tasks. The scales to be considered range from the detailed construction information of a single building up to the spatial information of an entire city or region. A typical urban planning task is the construction of a new subway line, which involves a large-scale alignment design of the track and detailed designs of new train stations, escape shafts and traffic nodes (Borrmann et al. 2015). Currently, experts use different information sources for these scales, in particular building information modelling (BIM)

S. Daum (✉) · A. Borrmann
Chair of Computational Modeling and Simulation, Technische Universität München,
Munich, Germany
e-mail: simon.daum@tum.de

A. Borrmann
e-mail: andre.borrmann@tum.de

T.H. Kolbe
Chair of Geoinformatics, Technische Universität München, Munich, Germany
e-mail: thomas.kolbe@tum.de

systems and geographic information systems (GIS). For both domains, standardised data schemas for semantic and geometrical modelling exist. However, an integrated analysis of BIM and GIS data with respect to spatial and semantic criteria has not been possible to date.

More and more city models are being represented according to CityGML (Gröger and Plümer 2014). This well-known data model is employed to represent partial and whole cities at five levels of detail (LOD) (Kolbe 2009). In the most detailed level, LOD4, the interior of buildings and other types of construction are also represented. CityGML is designed to represent an as-built situation as captured by a survey. For this reason, the modelling principle of CityGML is based on representing the viewable surfaces of a building, e.g. expressed by classes such as *WallSurface* and *RoofSurface*.

In the architecture, engineering and construction (AEC) domain, the standardised data model industry foundation classes (IFC) is employed to realize open data exchange between BIM tools. In contrast to CityGML, the IFC schema focuses on the modelling of building elements and their relationships. The combination of semantic objects, explicitly stored relationships and geometry representations based on solid modelling meets the requirements of planning, constructing and operating buildings. Thus, an IFC model can be used as a single holistic data pool supporting all phases of a building's life cycle. Conceptually, the IFC model provides the foundation for BIM which promotes a new way of working in the construction domain, that is fully based on high-level digital models. Currently, BIM is gaining wide acceptance in the building industry and is successfully employed in a large number of construction projects worldwide (Eastman et al. 2011).

During the planning of new buildings and infrastructure facilities, many decisions require integrated analysis of information regarding the new (planned) construction in combination with information about already existing structures. The required holistic data pool can be used to process queries that comprise semantic, spatial and temporal aspects. Thus, analyses regarding property situations, construction scheduling and geometrical conflicts can be performed. A number of researchers have developed methods to support such an integrated analysis by converting IFC data sets to CityGML. These approaches are discussed in more detail in Sect. 2. Because of the varying modelling methods and their distinct semantics, there is a high probability that information is lost during this transformation.

This paper presents the QL4BIM query language and its corresponding runtime environment, the QL4BIM system. The QL4BIM language supports integrated analysis of both IFC and CityGML models. Moreover, because no conversion between these two schemas is required, the loss of information caused by data transformations is avoided. The intermediate layer of QL4BIM is the integral part of the query language that abstracts from the analysed data schemas such as IFC and CityGML. Entities from the external schemas are encapsulated by the intermediate layer. Thereby, the query runtime environment can process these items. Although no conversion between the different external schemas is performed, comprehensive queries that operate simultaneously on instances of both schemas can be processed. As demonstrated by IFC and CityGML, QL4BIM is conceptually capa-

ble of analysing schemas that include semantic, spatial and temporal data by using generic operators. The benefit of the proposed approach is that CityGML and IFC can be temporarily brought together in one comprehensive (spatial) context. Therefore, only one query language is required and identical operators can be employed on both data sources. For example, integrated analysis beyond the extent of a single schema enables the detection of spatial conflicts between models of different scales.

The paper is organized as follows: Related work that addresses the combined processing and conversion of BIM and GIS data and existing query languages for BIM are discussed in Sect. 2. In Sect. 3, an overview of QL4BIM is given. Including the extension for geospatial processing, the components of the query runtime environment are discussed in Sect. 4. In Sect. 5, the QL4BIM data model is presented. This model acts as an intermediate layer that realizes the necessary abstraction of the IFC and the CityGML schema. Section 6 discusses the handling of CityGML geometry. In Sect. 7, a case study is shown. Here a combined IFC/CityGML data pool is analysed by two exemplary QL4BIM queries. Conclusions are presented in Sect. 8.

2 Related Work

The integration of approaches from geoinformatics and construction informatics has recently been addressed by a growing number of publications. In addition, the joint processing of data from these two domains is a topical subject. A general mapping between IFC and CityGML schemas for LOD1 to LOD4 has been developed by Isikdag and Zlatanova (2009). In Hijazi et al. (2011) the extension of city models with IFC data concerning utility networks is discussed. Topological analyses and spatial reasoning on both models were employed in Khan et al. (2014) to deduce indoor routing graphs. Their contribution also presents an approach to convert IFC models into CityGML LOD4 with a focus on navigable space. To integrate IFC data into a GIS context, the GeoBIM extension for CityGML can be used. Here an application domain extensions (ADE) is employed to equip the CityGML schema with 17 classes from the BIM domain (Laat and van Berlo 2011). In El-Mekawy et al. (2011) a bi-directional mapping between IFC and CityGML was established using a combined data model. This requires a conversion to the combined model and to the respective data model. Conversely, the QL4BIM data model encapsulates items from IFC and CityGML without a bi-directional conversion operation. An automatic conversion of IFC data to CityGML LOD3 was also discussed in Donkers (2013). The spatial operators of QL4BIM are heavily influenced by spatial GIS analysis. Here, metric and topological analysis functionality which is common in GIS was adapted to the three-dimensional data of building information models (Daum and Borrmann 2014; Borrmann and Rank 2009).

In addition to QL4BIM, a number of query languages have been developed for the BIM domain. The partial model query language (PMQL) is an XML-based language offering *Select*, *Update* and *Delete* operators (Adachi 2003). It has been developed in the context of the EXPRESS modelling languages which is used to define the IFC

schema. However, PMQL does not provide high-level operators for geometric analysis. The building information model query language (BIMQL) is the query interface of the IFC model server (Mazairac and Beetz 2013). The language is inspired by SPARQL which is a query language developed for the data model of the semantic web (Prud'Hommeaux et al. 2008). Although BIMQL offers an interface to extend the language with capabilities for spatial analysis, this is not yet available. A visual BIM query languages for craftsman was introduced in Wülfing et al. (2014). Finally, BERA is a language developed specifically to support code compliance checking (Lee 2011).

3 Overview of QL4BIM

QL4BIM is an imperative and procedural query language for analysing and processing building information models. QL4BIM is designed to be employed by domain experts and offers a carefully selected vocabulary to formulate queries at a high level of abstraction. Only a few patterns must be understood to use the language. Low-level data handling operations such as instantiations and cast operations are hidden from the end user. Queries can be stated in textual and visual notation. In this contribution, the textual notation ^tQL4BIM is used to describe the underlying methodology. For a description of the visual notation ^vQL4BIM, the reader is referred to Daum and Borrmann (2015).

A complete QL4BIM query includes a number of statements executed successively. Each statement represents an assignment that binds the result of an operation to a variable. An operation is the combination of an operator and the passed parameters. Constants such as strings, numbers, floats as well as assigned variables can be used as parameters. In addition, user defined predicates that evaluate the attributes of entities can be handed over to operators. The described pattern is reflected by the textual grammar of QL4BIM that is depicted in extracts in Fig. 1.

Listing 1 shows an example query stated in ^tQL4BIM that selects pairs of specific walls and windows, which are of the topological relationship *Touch*. The query

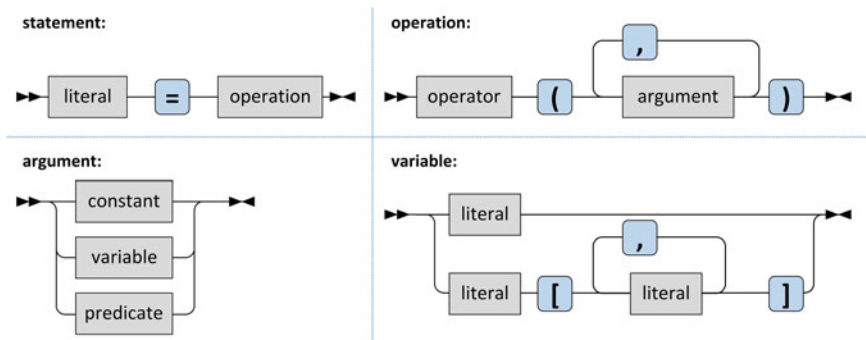


Fig. 1 Section of the QL4BIM textual grammar (syntax diagram)

begins by loading an IFC instance file using the *GetModel* operator. In Line 2, all walls are selected on the basis of their type affinity. The result is then filtered by applying a predicate to each entity. Thus, only walls with an attribute titled *Description* holding the value “Wall-002” are retained. To access an attribute of an entity, the point operator is employed in the predicate definition (Line 3). After retrieving all windows by applying a second *TypeFilter* (Line 4), the set of filtered walls and all windows are analysed topologically by applying the operator *Touch*. The result is assigned to a relational variable that comprises pairs of wall and window objects that satisfy a *Touch* relation.

```

1 model = GetModel("C:\Institute.ifc")
2 allWalls = TypeFilter(model, "IfcWall")
3 someWalls = AttributeFilter(allWalls.Description = "Wall-002")
4 allWindows = TypeFilter(model, "IfcWindow")
5 rel[wall, window] = Touch(someWalls, allWindows)

```

Listing 1 QL4BIM query extracting specific walls and their touching windows

This demonstrates the main pattern of QL4BIM, i.e. repeated variable assignment combined with the application of an operator. In this way, subsequent operations process variables assigned by former calls. Despite its simplicity, this pattern enables flexible and detailed analysis of building models by chaining manageable operations. The complexity of the information retrieval is hidden in the interior processing of each operator. Thus, the domain expert can concentrate on the semantic of each operator rather than define the low-level data handling. The functionality of the language is provided by four categories of operators:

1. Semantic operators for type and attribute extractions
2. Relational operators to analyse links between entities
3. Spatial operators to evaluate topological, directional and metric predicates
4. Temporal operators to examine construction schedules

Figure 2 demonstrates possibilities to select building elements in an IFC model instance using these four operator categories. For example, walls can be selected by evaluating their type affinity (1). To extract tuples of walls and their filling windows, the relations stored in the model are analysed (2). It is also possible to identify tuples of walls and their associated windows by a spatial examination. Here, the occurrence of *Touch* relationships between the geometric representations is used as a selection predicate (3). Finally, the selection of building elements can be based on their position in a construction schedule, e.g. the planned installation start time (4).

4 QL4BIM’s Runtime Environment: The QL4BIM System

The QL4BIM system operates using its own internal data model, which acts as an intermediate layer for abstraction and integrated analysis. This concept offers high flexibility with respect to the processed data. Instances of the data model of QL4BIM

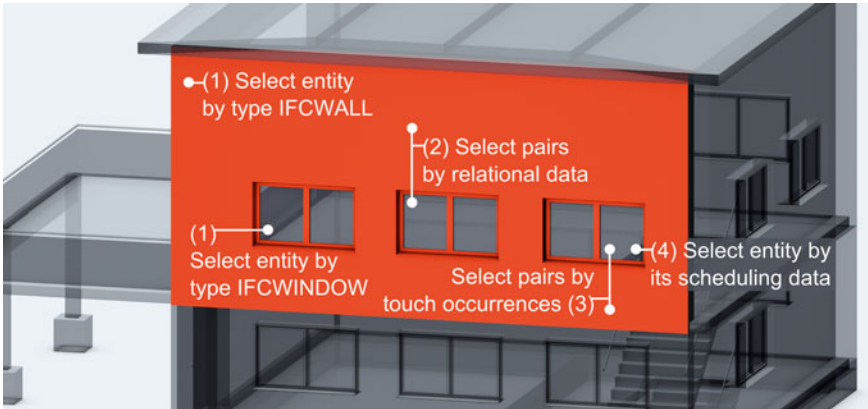


Fig. 2 Applications of the four categories of QL4BIM operators

wrap entities of a genuine data model whereas IFC and CityGML are considered until now. In this contribution, types from QL4BIM will be addressed as *internal* whereas types from genuine data models are referred to as *external*. The QL4BIM data pool, which is the initial data source that can be analysed by queries, is provided in the internal schema. Furthermore, all intermediate variables are instances of this internal schema. To establish the data pool and to create an appropriate infrastructure for query execution, the runtime makes use of several parsers, a query interpreter and a query backend.

Figure 3 shows the components of the runtime environment and their interconnections. Items drawn in white are parts of the original system focused on IFC processing. The components newly developed to handle CityGML are shaded in blue. The included parsers are used for the syntactic analysis of query input, for

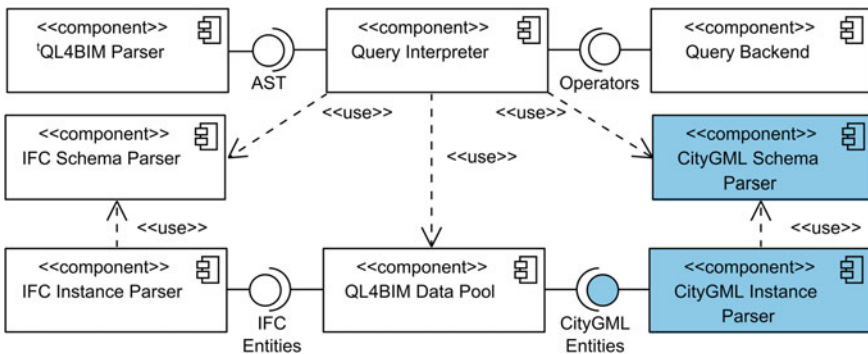


Fig. 3 Components of the QL4BIM system (UML component diagram)

importing the underlying pool data and for examining the external schema in use for the given case.

In the following, the interaction among components is discussed without the explicit consideration of the examined external data schema. As the internal data model of QL4BIM is highly generic, the principle is identical for handling IFC, CityGML or any other schema.

First, the instance parser interprets the referenced file and transfers the external entities to an internal representation. Rather than specific classes for external types, the runtime environment uses a late binding approach for this in-memory representation. As a result, entities can be set up dynamically by means of generic parts. Fine-grained sub-elements such as object identifiers, lists of symbolic references and values of simple types are represented by these parts. The late binding approach provides a great flexibility regarding external schemas. Thus, all IFC versions are directly supported ad-hoc, and alternative data models such as CityGML can be integrated easily by an appropriate instance and schema parser.

The schema parser supports the query execution when external types must be compared against native QL4BIM types. In addition, meta-data such as the attribute names of entities can be read directly from the schema without computationally expensive reflection mechanisms.

If a query has passed the syntactic analysis, the QL4BIM parser translates it to an abstract syntax tree (AST) (Parr 2010). This intermediate representation of a query is consumed by the query interpreter. The AST is a streamlined structure deduced from the textual query input without lexical helpers. The tree structure can be processed in several ways. For example, the QL4BIM system performs validations on the AST that verify the order of variables. Additionally, static type checks are executed and parameter handovers are examined. Finally, the main purpose of the AST is its use for the query interpretation.

Figure 4 shows a QL4BIM query and its AST-based representation. The S-nodes correspond to the four sub-query statements. If a complete query execution is invoked by the user, the interpreter traverses the whole AST to collect results and to call the

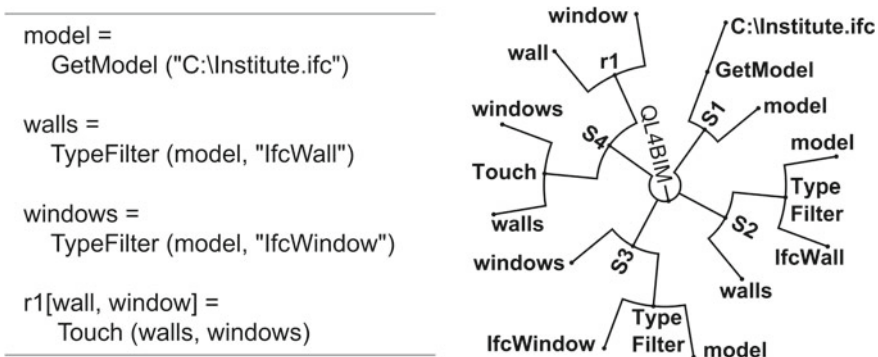


Fig. 4 A QL4BIM query and its AST-based representation

appropriate functions in the query backend. Finally, the last variable in the query includes the final output of the query.

As a matter of principle, the execution of a declarative query language such as SQL processes a complex query in a single step. In contrast, the QL4BIM system supports the execution of all statements and an incremental interpretation. The imperative style of the language combined with the enforced representation of interim results enables this incremental interpretation. Rather than processing all statements in the AST, the runtime environment interprets statements sequentially, according to user request. In this step-wise execution, the content of the current variable is visualized in the QL4BIM user interface. This approach is similar to a debugging session in an integrated development environment for a standard programming language. The approach supports domain experts in formulating complex queries, because the effect of each single statement can be tested.

5 The QL4BIM Data Model

The central structure of the QL4BIM data model is the *Entity* which wraps an object supplied by an external data model. Instance files are imported into the data pool using the *GetModel* operator. After execution, the objects included in these files are present in the pool as a set of entities and are referenced by a variable. By executing QL4BIM queries, the initial sets are processed and intermediate results are bound to new variables. The processing includes the extraction of a single entity and the creation of variables that refer to specific sets and relations of entities.

To incorporate the original data, the *Entity* type is designed as a dynamic container for external data. Fine-grained elements below the level of a complete object are denoted as *Parts* in the QL4BIM system. The *Part* type accommodates primitive data such as strings, enumerations, integers and floats. Thus, an external object and its attributes can be represented in the runtime environment. Figure 5 shows a class

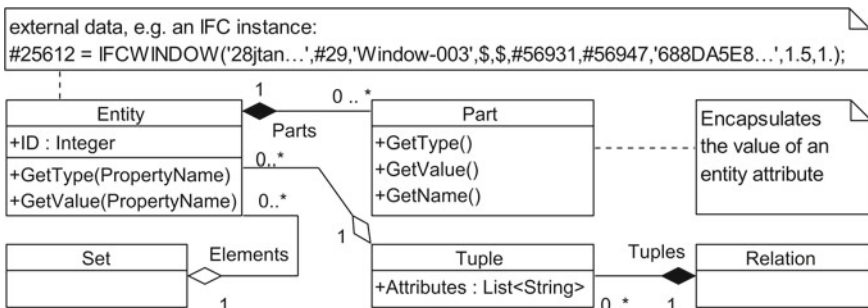


Fig. 5 Central classes of the internal QL4BIM data model (UML class diagram)

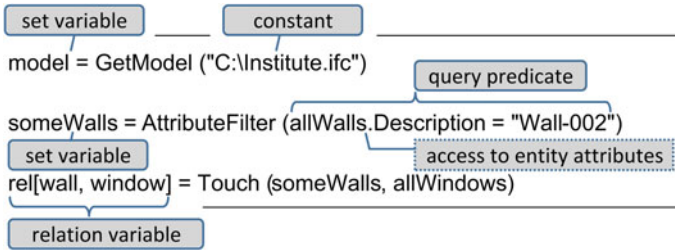


Fig. 6 Implicit typing of *Set* and *Relation* variables and query predicates

diagram that depicts the *Entity* class and its association to the *Part* class. Furthermore, the composition of the internal *Set* and *Relation* classes are shown.

The query language is designed without explicit type declarations. Therefore, no *Entity*, *Set* and *Relation* keywords are used in QL4BIM queries (see Listing 1). A variable is defined by using a new literal in an assignment. The actual type of the variable is inferred by the operator used in this assignment. Figure 6 demonstrates this by annotations for the QL4BIM query in Listing 1.

Besides defined variables that bind to instances, sets or relations, the passing of constants as parameters is also depicted. The sub-elements of entities are processed to evaluate the *AttributeFilter*. Therefore, the point operator, which can be used in passed query predicates, is provided. In the example, each entity of the *allWalls* set is examined for an attribute named *Description*. If this attribute is present, its type is checked. Here, the attribute must be typed as a string as it is tested for equality against a string constant. After query execution, the *someWalls* variable will only include entities with a string typed attribute *Description* holding the value “Wall-002”.

The processing of schemas of examined external data models is required to evaluate attribute names and types. Thus, the integration of any external data model in the QL4BIM system includes the introduction of an instance parser and a schema parser. To support CityGML, this is a parser for GML instance files and a parser for the respective XSD schema. The use of IFC data results in a parser for Part21 files and an EXPRESS schema parser.

Within the QL4BIM data model, the selection of entities is based on an unambiguous object identifier (OID). If an OID is available for any object in the original data, it is reused for this purpose. Otherwise, artificial OIDs are created by the QL4BIM runtime environment. The internal data model of QL4BIM is designed to encapsulate data from external models in a way that supports an efficient data processing. Therefore, internal entities and their corresponding attributes are created and stored in a data structure for random access. In addition, interrelations between entities represent further essential information. In the internal data model, these relations are expressed through foreign OIDs as the content of the respective *Part* of an *Entity* (Fig. 7). The data model of QL4BIM is designed to support this typed graph

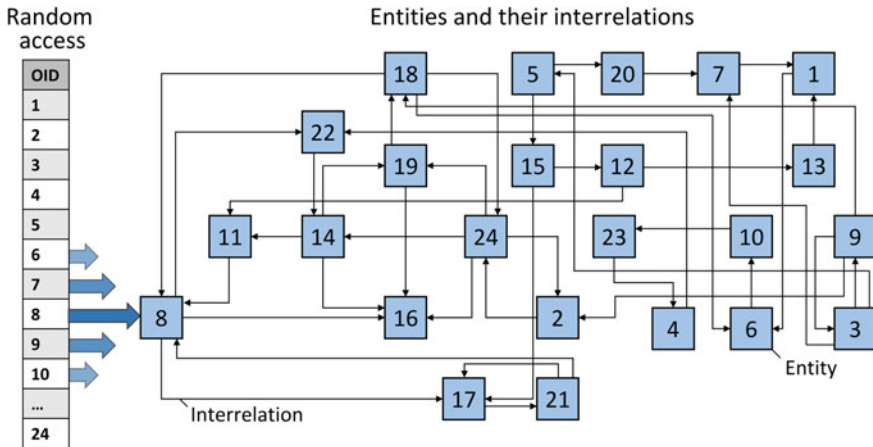


Fig. 7 Data structure for random access to entities and the resolving of interrelations

structure. A hypothesis of this research is that representing entities and particularly their relations is required to analyse any data model in geospatial and BIM domains.

To realize the efficient selection of entities by their OIDs, the QL4BIM runtime environment introduces a hash table-based storage model. As a result, the selection of an entity and resolving an interrelation is executed on average in constant time, independent from the number of entities in the current input. Thus, the duration of query execution is reduced if large data sets are handled. This is crucial because a query language is generally applied if manual processing is not feasible.

The layout of a CityGML instance differs from the relational equipped linear structure of the QL4BIM runtime environment. As CityGML is encoded as XML, data is arranged in a hierarchical, tree-based fashion which is illustrated in Listing 2. In this encoding, an object is described by an XML element. Direct child elements represent the attributes of the object. If an attribute is modelled by a simple type, the actual value is encoded as the enclosed text of the element. Accordingly, the *bldg:Building* instance in Listing 2 holds a *gml:Name* attribute with the value "Example Building LOD4". If an attribute references another object it is encoded as a nested XML element. This is the case for the building with the *gml:boundedBy* attribute which points to a *bldg:GroundSurface* instance. Additionally, *XLink* references can be used to link to objects at other places in the document.

If CityGML instances are parsed and imported into the QL4BIM data model, the nested data structure must be linearised so that CityGML objects can be represented in the hash table-based data pool. The rule for this transformation is that every object is transferred to a QL4BIM *Entity* and its attributes are represented as *Part* instances. If an attribute of an objects is a simple type, it can be stored directly in a *Part*. If another object is referenced in an attribute, a stand-alone *Entity* is instantiated and inserted into the indexed data pool. The linkage between the two entities is expressed by the OID value stored in the attribute of the parent. In contrast to a conversion

between CityGML and IFC, the embedding of the external objects in the QL4BIM data model is bijective. This is evident as the imported data can be re-encoded to its native format.

```

1 <bldg:Building gml:id="GML_7..." >
2   <gml:name>Example Building LOD4 </gml:name>
3   <bldg:measuredHeight uom="#m">5.0</bldg:measuredHeight >
4   <bldg:storeysAboveGround>1</bldg:storeysAboveGround>
5   <bldg:storeyHeightsAboveGround uom="#m">3.0</bldg:storey...>
6   <bldg:boundedBy>
7     <bldg:GroundSurface>
8       <gml:name>Ground Slab</gml:name>
9       <bldg:lod4MultiSurface>
10        <gml:MultiSurface> ...
11        </gml:MultiSurface>
12      </bldg:lod4MultiSurface>
13    </bldg:GroundSurface>
14  </bldg:boundedBy>

```

Listing 2 Tree-based encoding of a CityGML instance

As CityGML is encoded in XML, namespaces are frequently used in the data model. To achieve complete representation of the CityGML data, namespace values are added as meta data to QL4BIM entities and their parts. Furthermore, XML elements can be equipped with attributes. Although, attributes are not heavily used in CityGML, their representation is useful. For example they provide information about the unit used for stated quantities. Thus, XML attributes are upgraded to regular attributes of entities and can be consistently analysed in queries. The main parts of the IFC and CityGML models can be combined in the internal QL4BIM data model without special handling of the respective source. This results in identical queries in many cases. Listing 3 shows the extraction of IFC and CityGML walls whose *Name* attribute ends with “South”.

```

1 ifcModel = GetModel ("C:\Institute.ifc")
2 ifcWalls = TypeFilter (ifcModel, "IfcWall")
3 someIfcWalls = AttributeFilter (ifcWalls.Name = "*South")
4 gmlModel = GetModel ("C:\Building_LOD4.gml")
5 gmlWalls = TypeFilter (gmlModel, "WallSurface")
6 someGmlWalls = AttributeFilter (gmlWalls.name = "*South")

```

Listing 3 Identical queries for IFC and CityGML data

6 Handling of Surface-Oriented CityGML Geometry

The spatial operators of QL4BIM examine the geometry representation of entities in the data pool. In the IFC case, the used geometry representation is based on solids that have a well-defined interior, boundary and exterior (Daum and Borrmann 2014). As CityGML modelling is designed to represent geometry captured by survey, laser scanning or photogrammetry, the use of surface-oriented, non-closed representations is common. Figure 8 shows non-closed geometry in an exemplary CityGML data

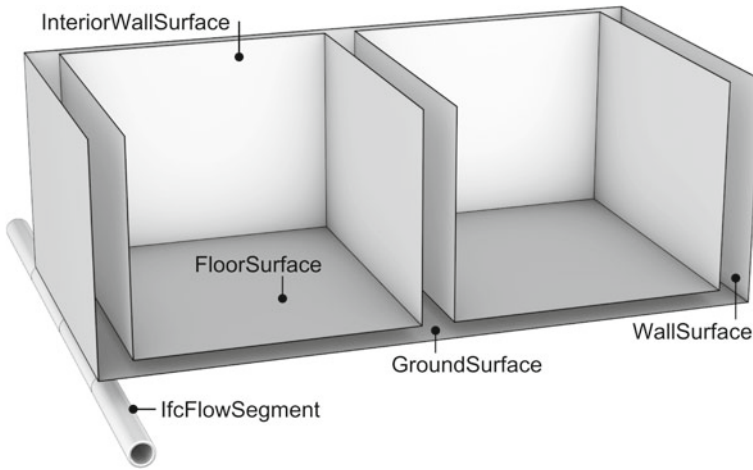


Fig. 8 Non-closed geometry in an exemplary CityGML data set in combination with IFC solids

set in combination with IFC flow segments which are located below the CityGML entities.

A spatial analysis is also possible if the surface-oriented geometry of CityGML is used together with IFC solids. However, because of the lacking interior/exterior classifications within the surface representation, e.g. directional constellations can remain undetected. Here, the wall surfaces can not be classified as laying above the flow segments because of the missing covers of walls. To cope with this, the spatial analysis should be based on an extended set of geometry representations and include the *GroundSurface* instance.

7 Case Study

The following case study demonstrates the concept of an integrated analysis of data from the domains of construction informatics and geoinformatics. The study examines the interactions of subterrestrial constructions that are part of a planned subway track with existing buildings on the surface. Here, the buildings above ground are present as inventory data expressed in CityGML. The planned subway track and its supporting buildings are modelled for construction purpose and are available as an IFC data set. In Fig. 9, the CityGML entities are shaded in grey and IFC entities are shaded in blue.

The following QL4BIM query in Listing 4 begins by loading the city model into the runtime environment. Then, a selection of buildings is executed on the basis of their type affinity. The same is done with the IFC model from which all spaces are retrieved. The final *Above* operator tests which buildings are located above spaces

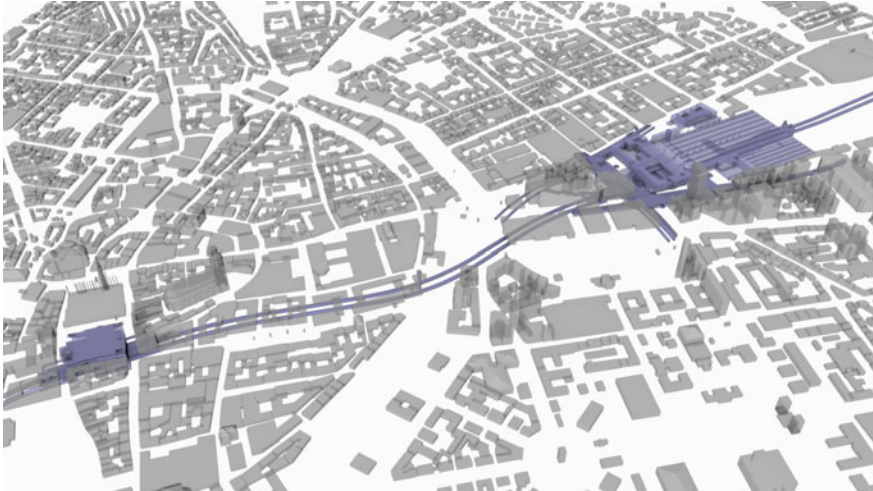


Fig. 9 CityGML data set of Munich (LOD1) combined with a planned subway track encoded as IFC

instances. Thus, the relational $rel[space, building]$ variable represents pairs of *IfcSpace* and CityGML *Building* objects.

```

1 gmlModel = GetModel ("C:\Building_LOD1.gml")
2 gmlBuildings = TypeFilter (gmlModel, "building")
3 ifcModel = GetModel ("C:\supportinBuildings.ifc")
4 ifcSpaces = TypeFilter (ifcModel, "IfcSpaces")
5 rel[space, building] = Above(ifcSpaces, gmlbuildings)

```

Listing 4 QL4BIM query spatially analysing IFC spaces and CityGML buildings

```

1 gmlModel = GetModel ("C:\Building_LOD4.gml")
2 buildings = TypeFilter (gmlModel, "building")
3 highBuildings = AttributeFilter (buildings.StoreysAboveGround > 3)
4 walls = RelationResolver (highBuildings.WallSurface)
5 ifcModel = GetModel ("C:\supportinBuildings.ifc")
6 flowSegments = TypeFilter (ifcModel, "IfcFlowSegment")
7 rel[flowSegment, wall] = NearerThan (flowSegments, walls, 3)

```

Listing 5 QL4BIM query analysing IFC flow segments and walls in CityGML buildings

Listing 5 shows a second example of combined IFC/CityGML processing, demonstrating a more detailed analysis. From the loaded CityGML data, only buildings that are higher than three storeys above ground, are considered. In the next step, walls are selected from these buildings. Then, the set of CityGML walls is examined spatially with a set of flow segments originating from the IFC data set. Finally, a relation is established that contains pairs of one *IfcFlowSegment* and one CityGML *WallSurface* whereby the two entities are located closer than 3 m.

8 Conclusion

This paper presents an extended version of the QL4BIM query language that supports integrated processing of data sets from the AEC and the geospatial domains. Rather than converting between the data models of IFC and CityGML, an intermediate layer as part of the QL4BIM system is introduced. As a result, it is possible to abstract from different modelling approaches. The original data remains valid and integrated data processing is realized. In essence, a new level of integrated semantic and spatial analysis across domain borders of BIM and city models is achieved. Thus, the proposed approach has the potential to ease complex urban and structural planning substantially. Future work will concentrate on the spatial processing of surface-oriented geometry of CityGML in combination with the solid representations of IFC. Furthermore, the spatial referencing in the QL4BIM System will be discussed.

References

- Adachi, Y. (2003). Overview of partial model query language. In *ISPE CE* (pp. 549–555)
- Borrmann A, Kolbe TH, Donaubaue A, Steuer H, Jubierre JR, Flurl M (2015) Multi-scale geometric-semantic modeling of shield tunnels for GIS and BIM applications. *Computer-Aided Civil and Infrastructure Engineering* 30(4):263–281
- Borrmann A, Rank E (2009) Topological analysis of 3D building models using a spatial query language. *Advanced Engineering Informatics* 23(4):370–385
- Daum, S., & Borrmann, A. (2015). Simplifying the analysis of building information models using tQL4BIM and vQL4BIM. In *EG-ICE 2015*
- Daum S, Borrmann A (2014) Processing of topological BIM queries using boundary representation based methods. *Advanced Engineering Informatics* 28(4):272–286
- de Laat, R., & van Berlo, L. (2011). Integration of BIM and GIS: The development of the CityGML GeoBIM extension. In *Advances in 3D Geo-Information Sciences* (pp. 211–225). Springer
- Donkers, S. (2013). *Automatic generation of CityGML LoD3 building models from IFC models*. Ph.D. thesis, TU Delft, Delft University of Technology
- Eastman, C., Teicholz, P., Sacks, R., & Liston, K. (2011). *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors* (Vol. 2)
- El-Mekawy, M., Östman, A., & Shahzad, K. (2011). Towards interoperating cityGML and IFC building models: A unified model based approach. In *Advances in 3D Geo-Information Sciences* (pp. 73–93). Springer
- Gröger, G., & Plümer, L. (2014). The interoperable building model of the European Union. In A. Abdul Rahman, P. Boguslawski, F. Anton, M. N. Said and K. M. Omar (Eds.), *Geoinformation for Informed Decisions*, Lecture Notes in Geoinformation and Cartography (pp. 1–17). Springer International Publishing
- Hijazi, I., Ehlers, M., Zlatanova, S., Becker, T., & van Berlo, L. (2011). Initial investigations for modeling interior Utilities within 3D Geo Context: Transforming IFC-interior utility to CityGML/UtilityNetworkADE. In *Advances in 3D Geo-information sciences* (pp. 95–113). Springer
- Isikdag, U., & Zlatanova, S. (2009). Towards defining a framework for automatic generation of buildings in CityGML using building Information Models. In *3D Geo-Information Sciences* (pp. 79–96). Springer
- Khan, A. A., Donaubaue A., & Kolbe, T. H. (2014). A multi-step transformation process for automatically generating indoor routing graphs from existing semantic 3D building models. In *Proceedings of the 9th 3D GeoInfo Conference*

- Kolbe, T. H. (2009). Representing and exchanging 3D city models with CityGML. In *3D geo-information sciences* (pp. 15–31). Springer
- Lee, J. K. (2011). Building environment rule and analysis (BERA) language and its application for evaluating building circulation and spatial program
- Mazairac W, Beetz J (2013) BIMQL—An open query language for building information models. *Advanced Engineering Informatics* 27(4):444–456
- Parr T (2010) Language implementation patterns: Create your own domain-specific and general programming languages. *The pragmatic programmers*. Pragmatic Bookshelf, Raleigh, N.C
- Prud'Hommeaux, E., & Seaborne, A., et al. (2008). SPARQL query language for RDF. *W3C Recommendation*, 15
- Wülfing, A., Windisch, R., & Scherer, R. J. (2014). A visual BIM query language. *eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2014* (Vol. 157)

A Methodology for Modelling of 3D Spatial Constraints

Daniel Xu, Peter van Oosterom and Sisi Zlatanova

Abstract In this work we demonstrate a new methodology to conceptualise and implement geo-constraints in 3D, which has not been widely explored yet. This is done in four stages from natural language to implementation, in which geometric primitives and Object Constraint Language (OCL) play a crucial role to formulate the constraints. A database including various 3D topographic objects (e.g. buildings, trees, roads, grass, water-bodies and terrains) from CityGML (no constraints yet) is used as a case study to apply the developed methodology. In this research, a first attempt to formulate 3D geo-constraints in OCL is made. Unified Modelling Language (UML) class diagram has been extended with graphical symbols for indicating constraints between classes (in addition to the additional compartment within a class for a class constraint). These constraint expressions can be tested and translated to other models/implementations when the OCL standard is extended with spatial types and operations. During this research, new types of constraints are defined as follows: general-level constraints (applicable to all object sub-classes), parameterised constraints (containing numeric values, e.g. maximum distance), constraints allowing exceptional instances (to resolve cases that have not been defined) and constraints relating to multi-scale representations (to check the consistency between two levels of detail which model the same object). Additionally common sense rules to detect conflicting constraints are specified as well.

D. Xu

Accenture Technology Solutions, Amsterdam, The Netherlands
e-mail: xusifeng@gmail.com

P. van Oosterom (✉)

Faculty of Architecture and the Built Environment, Section GIS Technology, Department OTB,
TU Delft, Delft, The Netherlands
e-mail: p.j.m.vanoosterom@tudelft.nl

S. Zlatanova

Faculty of Architecture and Built Environment, Department of Urbanism,
TU Delft, Delft, The Netherlands
e-mail: S.Zlatanova@tudelft.nl

© Springer International Publishing AG 2017

A. Abdul-Rahman (ed.), *Advances in 3D Geoinformation*,
Lecture Notes in Geoinformation and Cartography,
DOI 10.1007/978-3-319-25691-7_6

1 Introduction

Nowadays the field of geo-information is undergoing major changes, and the transition from 2D to 3D is having a major influence on these changes. A significant amount of 3D datasets are stored in spatially enabled databases. Experts are aware that new quality control mechanisms need to be built into the database systems in order to secure and guarantee data integrity.

Constraints set up rules that must never be violated and thus are used to maintain the data integrity. Many research project have investigated implementation of spatial constraint but still at a theoretical level. Few implementations at specific GIS application domain have been carried out but most of them are in 2D or in 2.5D (Louwsma 2004). But 3D city/urban models are becoming increasingly popular and the need for 3D data models is still growing. 2D models have already created such a big margin for errors, let alone adding one more dimension. An extension of current constraint theory from 2D to 3D would be a great help to stabilise the data.

A generic methodology in modelling 3D constraints is developed in this research. At first, constraints are designed and expressed using natural language. Then objects in the sentences are abstracted by geometric primitives, and their interrelationship by topological relationships. By doing so, spatial constraints become more specific and clearer to the others. Following the well-defined spatial types and operations as proposed in the ISO19107 standard and using various tools, an attempt is made to formalise these constraints using OCL. Finally, the constraints are translated to executable code, e.g. Procedural Language/Structured Query Language (PL/SQL), and with a small modification realised in the database by trigger mechanisms.

A database including various 3D topographic objects and their attributes is selected as a case study to test this approach. Objects such as buildings, trees, sensors (stationary and mobile), land use (e.g. grass fields, water bodies, roads) and terrain are modeled and stored in the Oracle Spatial (11g) database (Geomatics 2010). This database plays a key role in Campus Spatial Data Infrastructure to store, retrieve and exchange the information of in-campus objects for climate study, e.g. simulation and analysis.

This paper is organised as follows: Sect. 1 is the introduction; Sect. 2 reviews the related work presented in papers; Sect. 3 presents details of modelling 3D geo-constraints in the four stages; Sect. 4 shows how this methodology can be applied to a 3D topographic model and pseudo 3D OCL expressions; Sect. 5 tells new insights of constraint nature and classification methods/classes; Sect. 6 gives conclusion and recommendations for future work.

2 Related Work

In order to define the constraints, the classification of typical constraints has been studied. The main types of traditional (explicit) constraints include domain constraints, key and relationship structural constraints, and general semantic integrity

constraints. These classical types have been then extended by (Cockcroft 1997) with new spatial classifications: topological, semantic and user-defined constraints. This classification has been again elaborated by (Louwsma et al. 2006) with different types of relationship: direction, topology, metric, temporal, quantity, thematic, topological and mixed, and was also proven by (van Oosterom 2006).

A commonly adopted method of modelling geo-constraints is OCL, as suggested by research projects (Casanova et al. 2000; van Oosterom 2006; Duboisset et al. 2005; Louwsma et al. 2006; Pinet et al. 2007), ISO standard (ISO 2008) and data specifications of INSPIRE (INSPIRE 2010a, b). OCL is a formal language used to describe the constraints applying to objects, and is part of UML which is a preferred concept modelling schema. An advantage is that it can be translated by available tools, e.g. Dresden OCL2 toolkit, into executable code such as Java or SQL. And with small modifications database triggers can be formulated from SQL code.

Beside OCL, there are also other ways to express constraints, such as the constraint decision table by (Wachowicz et al. 2008), meta data repository by (Cockcroft 2004), ontology by (Mäs et al. 2005), GML/XML by (Reeves et al. 2006). Nevertheless, unlike the rich results of theoretical study, literature gives much less examples of implementing constraints in the GIS application domain. Some of them are found in a landscape design system (Louwsma et al. 2006), field data capture system (Wachowicz et al. 2008), an agricultural information system (Pinet et al. 2004), a cadastral data maintenance system (TOP10NL) (van Oosterom 2006), a land administration system (Hespanha et al. 2008), and a traffic flow control system (Reeves et al. 2006). These examples only deal with 2D constraints.

A fast growth of 3D GIS models can be observed since the emergence of CityGML. It is an information model to represent 3D urban objects, which also considers multi-scale modelling. A single object is able to have four levels of details, each of which corresponds to a certain geometry. Although CityGML does not define constraints, the classes and relationships of objects in a city, including their spatial (geometric/topological), semantic and appearance characteristics, are modelled therein (Kolbe et al. 2009).

As the spatial model evolves from 2D to 3D, more and more database vendors support 3D data storage in their products. An example is Oracle. Since Spatial 11g, 3D object data types are supported, such as 3D (multi) points, (multi) lines, (multi) polygons and (multi) solids. 3D data geometry can be validated according to GML 3.1.1 and ISO 19107 specifications, e.g. geometry validation via `SDO_Geom.Validate_Geometry_With_Context` (Oracle 2010b). With the hierarchy of single objects and composite objects, it enables storing multiple Levels of Detail (LoDs) from CityGML, which can model regions, cities/sites and architectural interiors. 3D queries concerning 3D visibility, volumetric analysis and spatial/semantic attributes are available in 11g (Ravada 2008) as well.

Despite the fact that 3D GIS is rapidly advancing and databases are able to implement geometric constraints, so far little research has been carried out on 3D constraints at a database level. Detailed research about implementing 3D constraints in databases is still highly demanding.

3 A Methodology for Geo-constraints Modelling

A methodology of modelling 3D geo-constraints is proposed here and can be used as a generic approach for all spatial-related constraints specifications in four stages:

1. Natural Language
2. Geometric/Topological Abstractions
3. UML/OCL Formulations
4. Model Driven Architecture (MDA)
 - a. Database PL/SQL Code
 - b. Exchange Format XML
 - c. Graphic User Interface ArcGIS

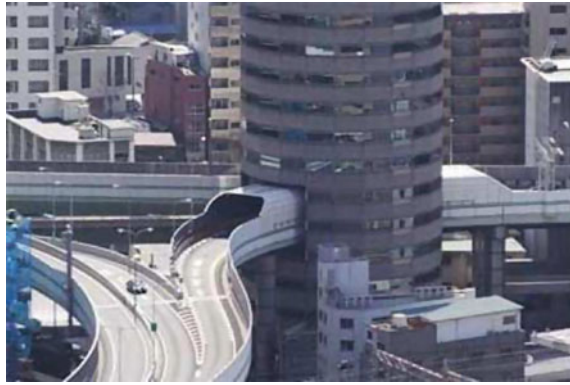
Natural language is a simple way to specify a constraint statement relating to spatial objects. But it is subjective to the individuals and therefore a more objective specification is necessary. A logical next step is making drawings of the objects (mostly the ‘nouns’ in a sentence) in order to illustrate the shape of the objects. After that, the objects interactions (mostly the ‘verbs’) can be explained better by formal descriptions of topological relationships, e.g. Egenhofer 9 intersection matrices (9IM) (Egenhofer 1989). Constraint statements thus become more specific and clearer to the others and not subject to multiple verbal interpretations. In order to let machines understand the constraints and automate the model translation, a further specification should be made considering MDA. UML/OCL as a modelling aid/tool therefore is the choice at this stage. Under the support of various tools/software, the constraints implementation in the database (e.g. PL/SQL code), data exchange (e.g. XML schema), graphic user interface (e.g. ArcGIS) or any other domains can be automated. Here we focus on the constraints implementation in the database. With a small modification the generated code can be used in database triggers, which finally realises the implementation of constraints check. The detailed analysis of expressing and translating constraints will be described in the section below.

3.1 Constraints in Natural Language

Defining geographic constraints by natural language is a challenging task. Natural language is not formalised as machine language but subjective to individuals’ interpretations of real-world phenomena. Not every geographic rule can be expressed as clearly and precisely as non-geographic rules, e.g. ‘a book from the library can only be lent to one person at a time’, or ‘the grade of a supervisor must be higher than his supervisees’ grade’.

An example of geo-constraint is ‘a road cannot cross a building’. The designer would mean to avoid a clash between a car and a building. But the statement might be confusing if one is confronted with the data model which exactly depicts a real world scenario as in Fig. 1. A person may start wondering if this constraint is satisfied

Fig. 1 An innovative architectural design where a building appears to be crossed by a road



by the model or not. Therefore it is difficult to avoid confusion when formulating constraints merely by natural language.

3.2 *Geometric and Topological Abstractions*

In the spatial model, the real world objects can be described by clearly-defined geometric primitives (e.g. solid, surface, line, point). Then the exact shapes of an object become clear, and the geometric primitives can also be specified by the components: interior, boundary and exterior. Based upon these primitives, the spatial relationships between objects can be formulated by 9IM and its extensions, such as calculus-based method from (Clementini et al. 1993) and 3D topological relationships from (Zlatanova 2000).

The example ‘a road cannot cross a building’ given in Sect. 3.1 can be specified using this abstraction. Here, three things in the text need to be clearly defined, ‘what is a building?’, ‘what does cross mean?’, and ‘what is a road?’.

- A building: is a structure consisting of a set of surfaces that divide the whole space into buildings interior and exterior;
- A road: is a path that allows vehicles to travel along it at a certain speed;

Therefore we can model the building using *solid* geometry and the road *surface* geometry. And the term ‘cross’ can be replaced by 9IM ‘intersect’.

The situation the constraint intends to forbid can be rephrased as:
 ‘A surface must not intersect a solid’.

By using 9IM, the real world scenario in Fig. 1, which would be modelled as the one in Fig. 3, can be distinguished from the forbidden situation as illustrated in Fig. 2.

Fig. 2 An illustration of ‘a surface intersects a solid’

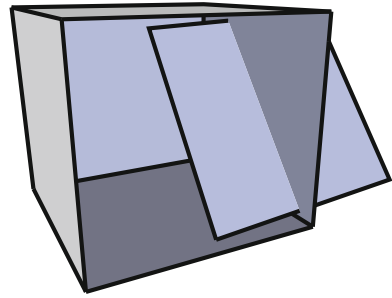
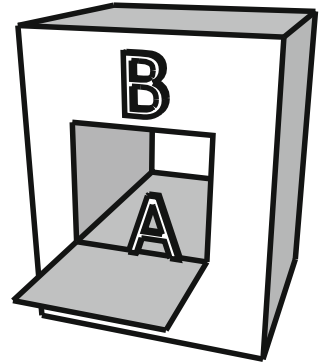


Fig. 3 Geometric model for Fig. 1. A is a road surface ‘through’ the hole in solid building B. In this case the constraint ‘road A cannot cross building B’ is not violated



3.3 UML/OCL Formulations

The UML diagram together with OCL notation has been found to be a suitable tool to express the designed constraints. An advantage of UML/OCL, comparing it to text-only alternatives, is that it unveils the relationships between objects visibly in diagrams. The OCL notations give formal (well defined and unambiguous) meanings that can be understood by both humans and machines. Another advantage is that it allows the designer to represent constraints on both geographic and alphanumeric data. And these two types of object can be used in one OCL constraint. For example, to indicate that a topological relation between geographical objects depends on the value of alphanumeric attributes.

UML is a common modelling tool and has a standard way to describe object classes, attributes and their interrelationships. Various tools support automatic generation of, e.g. SQL script, from UML class diagrams so that an implementation of the model can be created in the database by a few clicks. If OCL constraints can also be integrated to this code generation, the constraints can be integrated to database model and function through the whole lifespan of the database.

A challenge of specifying a relationship constraint is that currently there is no a rigorous mechanism in UML2.2 to indicate a constraint. When two object classes are related in the standard way (i.e. by association, specialisation, composition or aggregation), the constraint can be attached to the association. But when two object classes are not explicitly connected with an association link, the method to mention constraints about their relationships could lead to discussion.

For example, in general a road class and a building class may not have an explicit association. However, when some spatial constraint such as minimum distance between them is to be specified, a constraint association between the road class and the building class needs to be considered. Inspired by (Louwsma 2004), the normal association plus colour ‘red’ is proposed in this research as a new type of association link in UML class diagram (see examples in Figs. 6, 8 and 9 in Sect. 4).

A consequence of ‘borrowing’ association is the consideration of multiplicity. Because the constraint association will deal with class level rather than instance level, multiplicity which applies to instance level will not be considered.

Another difficulty which is somewhat related to the lack of constraint association in UML is that OCL itself does not support constraint expressions that have multiple classes involved. In normal OCL formula, if somebody wants to express a constraint related to a different class than the context class, he has to use the name of association end role to navigate from one class to the other. For example, to specify the no-intersect rule between a building instance and a road instance, the class ‘Road’ must be available in the context ‘Building’. In other words, the class ‘Road’ should have a property that is of ‘Building’ type, or ‘Building’ class should have a property that is of ‘Road’ type. But in this example of no-intersect between a road and a building, neither ‘Building’ nor ‘Road’ has a property to type (typify) each other. So an expression can be:

```
context Building
inv BldRoadNoIntersect:
  Road.allInstances()->forall(r | intersect(r.geometry, geometry)
= false)
```

Here, an extension of multiple classes in OCL is necessary to enable a ‘navigation’ from the context class Building (‘self’) to another class Road without an explicit association link. This extension can also be accompanied with our proposed constraint association in UML through which multiple class navigation is possible. More example expressions using multiple classes can be found in the pseudo OCL expressions in Sect. 4.

A workaround can be to set the context in a super-class which is shared by all the classes in the diagram. But this requires an ultimate super-class which can be navigated to by all classes, which is not always desirable. Therefore, it is better to extend OCL with multiple class expression.

3.4 Code Generation—Focus on Database PL/SQL Code

The Model Driven Architecture principle, being supported by Object Management Group (OMG), provides a framework to define how models in one domain-specific language (e.g. UML, OCL) can be translated to models in the other languages (Kleppe et al. 2003). In this paper, we focus on the implementation of constraints in an object-relational database. Some software/tools that are found useful to translate UML/OCL formulations into PL/SQL code are studied in this research. Their strengths and weaknesses in terms of OCL translation are given below.

- Enterprise Architect: EA is a UML drawing software used by ISO TC211 and INSPIRE for defining and visualising their data models and standards (Truyen 2005). A UML model can be automatically translated into Data Definition Language (DDL) and SQL script and thereby creates a corresponding database schema. Its OCL exporting function was used in the research project presented in (Hespanha et al. 2008). But except the referential key (Primary Key/Foreign Key) constraints and check constraints, the standard EA templates do not support the translation of other constraints such as OCL constraints to DDL or other implementation model. In our research, the standard syntax checking of EA appears to be weak and unreliable. Statements which are obviously incorrect in syntax can still be validated successfully. And spatial data types and operations are not supported.
- Dresden OCL2SQL: It's a part of Dresden OCL2 toolkit which provides a set of tools to parse and evaluate OCL constraints on various models, i.e. UML. Basically, it can transform a constraint to a SQL query that finds out all records that violate this rule this SQL code is integrated into triggers. The input for code generator includes a UML diagram of the appointed model and the associated OCL notations. Given a UML model together with correctly parsed constraints, within several clicks an executable SQL script can be generated.

For spatial constraints, the 3D geometries standardised in ISO19107 and 9IM topological names are not yet included in the OCL library. Articles from Pinet et al. (2007) and Werder (2009) show the possible extensions for 2D objects and 2D topological relationships. To achieve automatic SQL generation the extensions toward 3D, spatial types and operators can be added in a similar manner.

PL/SQL is Oracle's procedural extension language for SQL and its relational database. It combines the data manipulating power of SQL with the processing power of procedural language (Oracle 2010a). PL/SQL enables a trigger mechanism which is used to monitor the database and take a certain action when a condition occurs (Elmasri and Navathe 2003).

When a user modifies (create/insert/update/delete) certain datasets and then tries to commit the modification to the database, the trigger will be fired. Once it detects that a constraint is not satisfied in this commitment, it will give an error message to the front-ends and reject the transaction. By this means, a trigger is able to response to the data modification at run-time and guard the database integrity.

Given the trigger mechanism, if the OCL expressions are translated into SQL scripts, the spatial constraints check can be carried out by the spatial functions (e.g. `distance()`, `buffer()`, `intersect()`) supported by the database. In this sense, the power of data maintenance and spatial functions from database can be combined to have 3D geo-constraints integrated in database seamlessly.

However, the existing 3D functions in Oracle Spatial are relatively new and not extended. Many spatial and topological constraint checks can not be immediately implemented yet. The most useful function in Oracle Spatial database to calculate 3D topological relationships is `SDO_AnyInteract`. It is able to detect if two 3D objects are ‘disjoint’ or not. But it does not tell more details about what is happening in the ‘non-disjoint’ part. For example, two geometries which have ‘touch’ and ‘intersect’ do not make any difference to it. To be able to distinguish them in 3D, a new function named ‘`3D_SurfaceRelate`’ is developed in this research.

4 Examples of Constraints on 3D Objects

The methodology (1. natural language, 2. geometry/topology, 3. UML/OCL, 4. implementation) can be applied to many 3D topographic models, such as city models. CityGML and more specifically 3DCityDB (Kolbe et al. 2009) was selected as a 3D topographic model in this project. Similar to other research project, such as (Emgård and Zlatanova 2008) which extends the standard CityGML model, we also modified the model to fit the needs of this research. A new class ‘Sensors’ was added to the model and represented with point geometry.

As was mentioned in related work (Sect. 2), not many OCL formulations found in the existing work deal with 3D geo-constraints, especially in the application domain. The necessary constraints regarding to city objects in 3D space are discovered and will be described in the following sections in natural language first. Then figures will be given to illustrate the geometric/topological abstraction of some of them. And a first attempt to formalise these constraints in UML/OCL—(*pseudo 3D Geo-OCL*)—will also be explained. As was mentioned in Sect. 3.3, the constraint association links use colour ‘red’ and the name of constraints is attached to the class diagram.

The well-defined 3D geometric primitives from ISO19107: 2003 standard (ISO 2008)—`GM_Point`, `GM_Curve`, `GM_Surface`, `GM_Solid` and the aggregational and compositional types of them—are used as spatial types in UML class diagrams.

Spatial operators from ISO19107 such as `distance()` and `intersect()`, as well as Oracle functions `inside()` and `validateGeometry()` are used in these formulations.

4.1 Generic Constraints

Some constraints are applicable to all classes. Instead of repeating them everywhere and to keep the UML diagram clean and brief, it is better to specify them on the

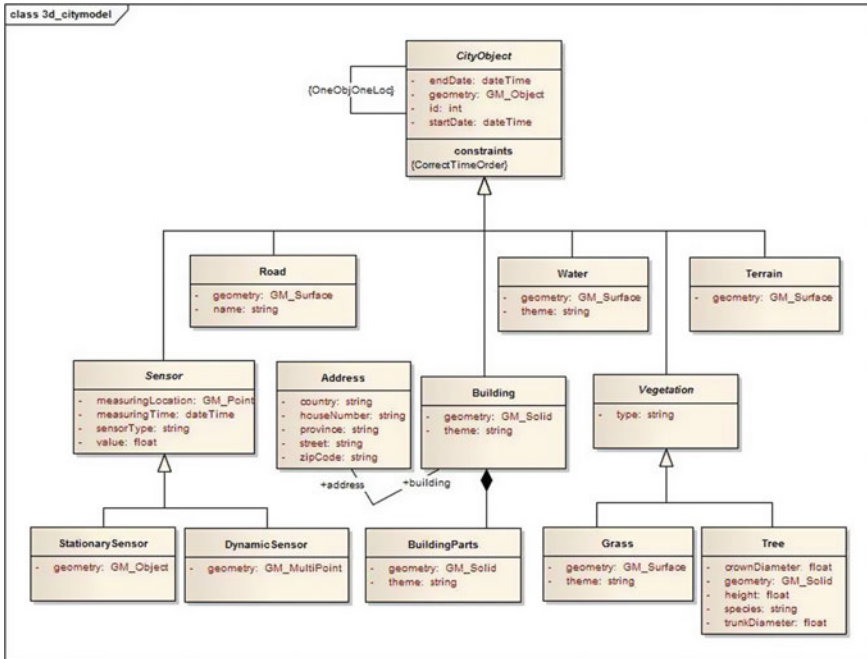


Fig. 4 UML model of constraints related to general city modelling (based on CityGML with extension on *sensor* class)

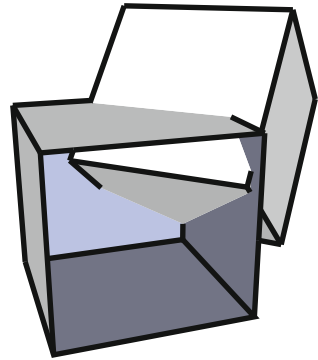
abstract/super-classes. With the inheritance, they can apply to all sub-classes (see *OneObjOneLoc* and *CorrectTimeOrder* in Fig. 4).

1. *OneObjOneLoc*: Given a local neighbourhood, there is only limited space to place an object. It is important to make sure the inserted/updated object do not clash into its neighbours (i.e. other buildings, trees) or stand on the wrong place (e.g. on the road, water).

A geometric/topological abstraction of this constraint can be seen in Fig. 5 where two solid objects intersect. This situation must be forbidden. Notice that this constraint applies to two or more instances under the same class. To distinguish this between-instances constraint from the constraint for single instance, e.g. alphanumeric constraint applying to attribute, a reflexive relation is used.

2. *CorrectTimeOrder*: The starting time and end time are general attributes for every object class. They can be used to review the changes of a certain area in time, e.g. land-use type for urbanisation monitor. It is important to keep these two time factors in a correct order—the starting time of an object must not occur after its end time.

Fig. 5 Two solid objects intersect. A situation to be restricted by constraint OneObjOneLoc



```

context CityObject
inv OneObjOneLoc:
    CityObject.allInstances()->forall(obj1, obj2 | obj1 <> obj2
        implies intersect(obj1.geometry, obj2.geometry) = false)
inv CorrectTimeOrder:
    startDate < endDate
    
```

4.2 Building Constraints

Building constraints—ValidSolid, AddressMatchesRoadName, MinDistTreeBld, CorrectTimeOrderIn-Composition, are attached to the UML class diagram in Fig. 6.

The extended symbol in UML class diagram—constraint association—is presented in the figure also.

3. ValidSolid: A building is treated as a composition of different parts. Each part can be modelled independently. This constraint checks if geometries (solids) of all building parts that belong to one building together form an (Oracle) valid solid. Oracle function *validateGeometry()* is used here which checks if the simple solids composing the whole object are topologically correct (adjacent).

An example of the geometric abstraction of this constraint is shown in Fig. 7. The three parts of a complex house must be adjacent and altogether form a valid solid.

4. MinDistTreeBld: This constraint checks if between one tree and one building the distance is larger than a minimum threshold. This is a parameterised constraint in which ‘minDistBld2Tree’ is an open parameter that can vary depending on different factors (see also Sect. 5.4).

5. AddressMatchesRoadName: The address of a building usually is related to its surrounding road. If a building in the model has a name that cannot be found in any name of roads in its local neighbourhood, this building must have been placed

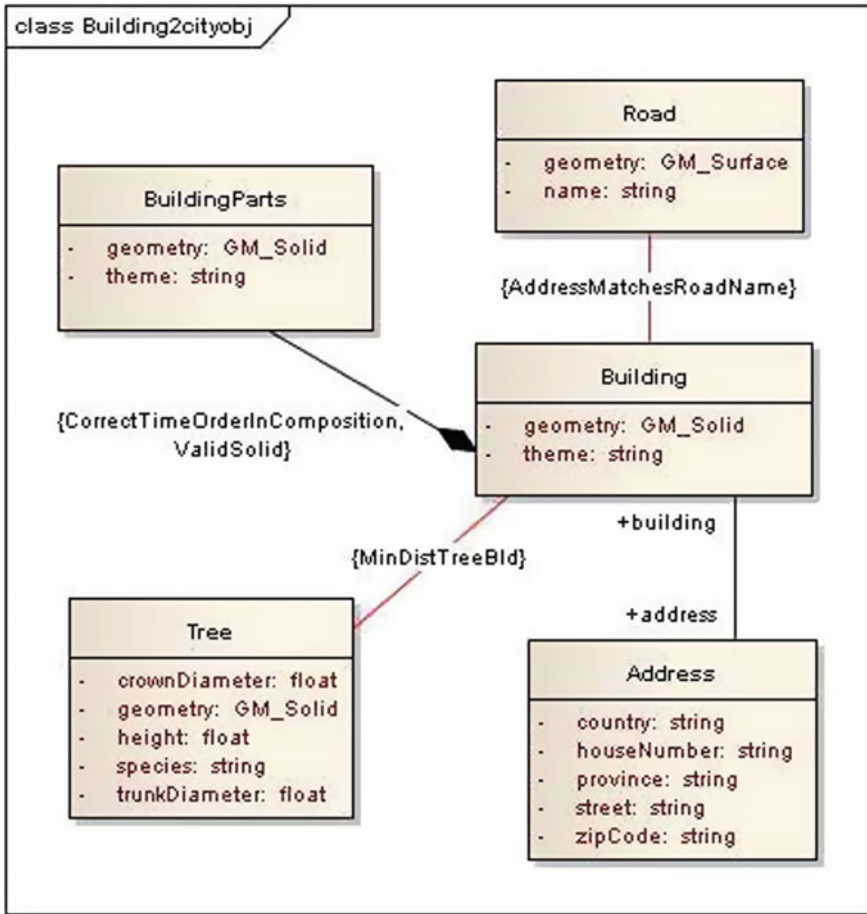


Fig. 6 UML model of constraints relevant to building object class

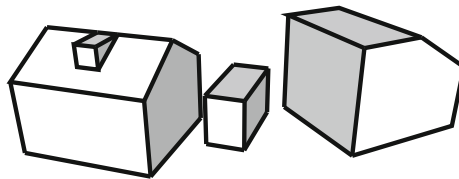


Fig. 7 The three parts of a house must altogether form a valid (e.g. Oracle definition) solid

in a wrong location. As a result, a relational constraint is specified to make sure a building is put in the right local area by comparing its address and the road name. This expression is also an example of using multiple classes in OCL.

6. *CorrectTimeOrderInComposition*: Since a building can be assembled by different parts and each part has its own starting time and end time, there is a temporal constraint taking place in the composition relation. The starting time of a part must be later than its assembly. And the end time of a part must occur before the end time of its assembly.

```

context Building
inv ValidSolid:
    buildingParts->validateGeometry(geometry) = true
inv MinDistTreeBld:
    Tree.allInstances()->forall(t | distance(t.geometry, geometry) > minDistBld2Tree)
inv AddressMatchesRoadName:
    Road.allInstances()->one(r | distance(r.geometry,
    geometry) < maxDistBld2Road and address.street = r.name)
inv CorrectTimeOrderInComposition:
    buildingParts->forall(b | b.startDate >= startDate) and
    buildingParts->forall(b | b.endDate <= endDate)

```

4.3 Tree Constraints

Tree constraints—*MinDist2Tree*, *MinDist2Road*, *Aquatic_Close2Water*, *LimitedCrown/Height/TrunkSizes*—are attached to the UML class diagram in Fig. 8.

7. *MinDist2Tree/Road*: ensures minimum distances between two tree instances and between a tree instance and a road instance. These two are also parameterised constraints. Each distance rule can have a different min distance threshold (i.e. the min distance of tree-tree ‘*minDistTree2Tree*’ would be different from tree-road ‘*minDistTree2Road*’). And the expression of tree-road constraint uses multiple classes.

8. *AquaticClose2Water*: Depending on the species, a tree object has a particular area to be planted in. Some aquatic plant can only grow close to or in the water. It is necessary to constrain the close distance of this species to water in the model. The expression below checks if an aquatic plant is close enough (comparing to a user-defined threshold) to water. This expression is parameterised and uses multiple classes.

9. *LimitedCrown/Height/TrunkSizes*: Every species of tree has a different limited size in terms of crown diameter, height and trunk diameter. It is unusual to see some bush plant growing into the size of an oak tree. Thereby, a rule to avoid unrealistic size of a tree is given. The species—*maxCrownvalue/Height/Trunkvalue* are constants for each species. But these parameters can differ from one species to another. Multiple-class expression is also used here.

```

context Tree
inv MinDist2Tree:
    Tree.allInstances()->forall(t1, t2 | t1 <> t2 implies

```

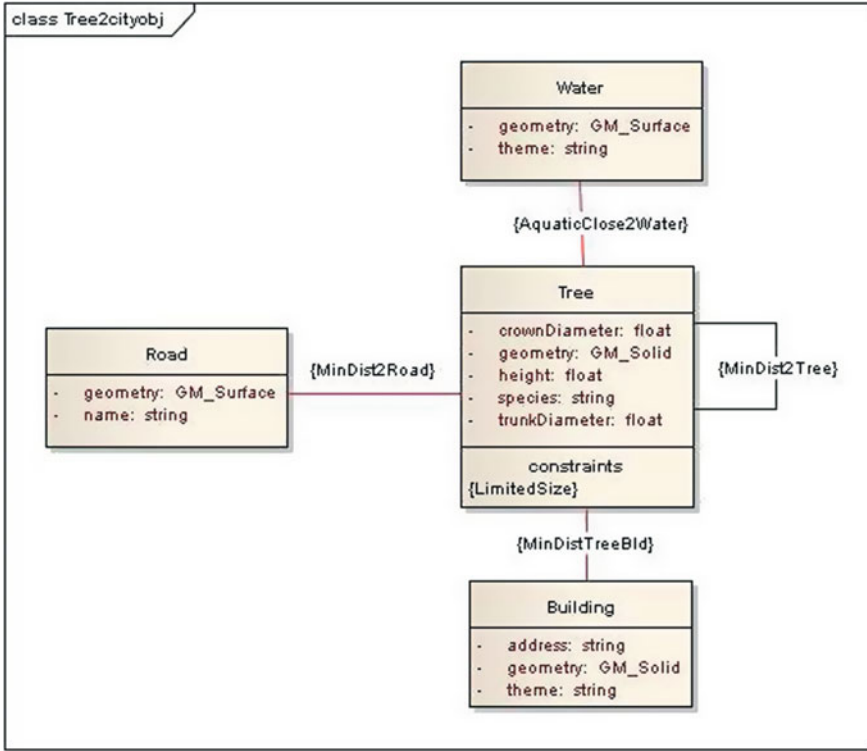


Fig. 8 UML model related to constraints in tree object class

```

distance(t1.geometry, t2.geometry) > minDistTree2Tree)
inv MinDist2Road:
    Road.allInstances()->forall(r | distance(r.geometry, geometry)
    > minDistTree2Road)
inv AquaticClose2Water:
    Water.allInstances()->exists(w | species = 'Aquatic' and
    distance(w.geometry, geometry) <= maxDistAquatic2Water)
inv LimitedCrownSize:
    crownDiameter < speciesMaxCrownValue
inv LimitedHeight:
    height < speciesMaxHeight
inv LimitedTrunkSize:
    trunkDiameter < speciesMaxTrunkValue
    
```

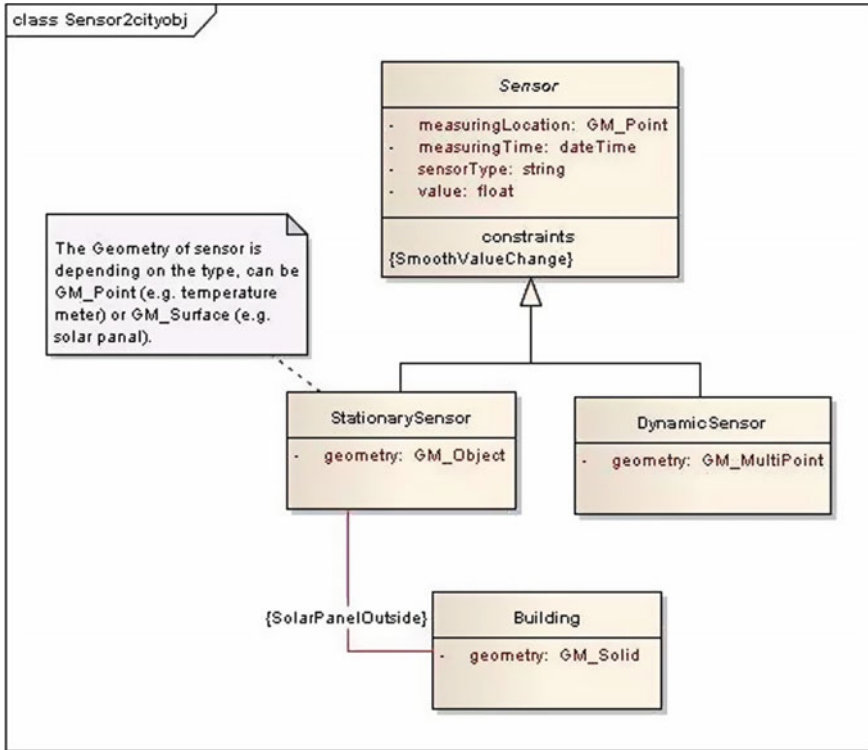


Fig. 9 UML model related to constraints in sensor object class

4.4 Sensor Constraints

The inheritance from generalisation is helpful in optimising the property constraints for sensors (static and mobile types). The necessity of having measurements of location and time for all sensors and smooth value changes is specified in the abstract type *Sensor* so that they can be automatically inherited by dynamic sensors and stationary sensors (see Fig. 9).

10. *SmoothValueChange*: Sensing data, which includes all different kinds of climate observations, is sensitive to erroneous value. One possible error in general is: two observations in a space of time have too big a difference. This is unusual for climate parameter(s), e.g. temperature, and thus needs to be marked as outliers.

The expression ensures that between two sensor instances from the same type the value change is within a threshold. The threshold parameter can be derived from change per time interval and vary with sensor type. Since this rule applies to every sensor object, it is added to the abstract class.

11. *SolarPanelOutside*: There are solar sensors (panel) equipped on some weather station and building roof. Usually the solar panels are installed on the roof,

or on the outside (wall) of building. Just shifting the panel object into building by 1 m can make its position become inside the building. This shift may not be obvious in terms of metric distance, but it definitely breaks the principle of using solar panel, which requires a direct reception of sunshine.

```
context Sensor
inv SmoothValueChange:
  sensorType = 'UserdefinedType' implies Sensor.allInstances()
->forall(s1, s2 | s1 <> s2 implies abs((s1.value - s2.value)
  / (s1.measuringTime - s2.measuringTime)) <= userdefinedMaxChangeRate)
inv SolarPanelOutside:
  Building.allInstances()->forall(b | sensorType = 'Solar Panel'
  implies inside(b.geometry, geometry) = false)
```

As was discussed in Sect. 3.4, the current OCL standard does not support 3D spatial types and operations. Therefore, fully automatic PL/SQL code generation is not possible; the constraints need to be implemented in the database manually.

5 New Insights of Constraint Nature

During the research and the constraint design toward the case study, some general aspects regarding the nature and classification of constraint were discovered. Two ways of stating the same constraint, i.e. forced and restricted, were introduced from early literature by (Louwsma 2004). And in this research, a general rule at this point is specified to keep the statements brief and constraints easier to implement. New classes (types) of constraints are found, that is, constraints with/without exceptions, abstract/specific constraints, constraints with/without parameters, common sense constraints to detect conflicting constraints, and constraints related to consistency of multi-scale representations.

5.1 *Forced and Restricted Specifications*

There are basically two ways to formulate a constraint in natural language, forced (positive: must be/always have to) and restricted (negative, cannot/must not) (Louwsma 2004). Some constraints stated in one way would appear to be much more clear and efficient than stated in the other way. For example, the constraint ‘grass colour always has to be green’ has the same meaning as ‘grass colour cannot have the colour black, white, gray, blue, red, yellow, purple’. The former is a forced way and is much briefer than the latter—the restricted constraint. As you may notice, what matters is the number of defined situations/values. ‘Grass colour should be green’ only defines one situation, whilst the other way defines many, potentially unlimited situations.

In the implementation stage, the difference does not only stay in length of text but also the performance of constraints. For example, constraint `OneObjOneLoc`

in Sect. 4.1 employs a restricting statement. According to that statement, an error will be given once the database detects relationship ‘intersect’ (9IM). The forced way in OCL-like manner can be:

```
inv OneObjOneLoc:
  CityObject.allInstances()->forall(obj1, obj2 | obj1 <> obj2
  implies disjoint(obj1.geometry, obj2.geometry) = true or
  touch(obj1.geometry, obj2.geometry) = true)
```

which states the relationships can be ‘disjoint’ or ‘touch’ (and even more relationships that are allowed). If we would implement this statement, then at least two different topological relationships (instead of one—‘intersect’) have to be calculated before the database can tell if the constraint is satisfied.

In order to reduce the complexity of constraint specifications and implementations, a constraint design principle is that:

‘All constraints should be compared using both statements. The statement that requires fewer values should be used’.

Of course, which way is more efficient may not be universal for all cases, but depends on the specific constraints and implementation.

5.2 Constraints With/Without Exceptions

According to (Servigne et al. 2000) and (Werder 2009), exceptions make a certain constraint more usable and realistic. In city environment, some conditions do not have to be strictly forbidden for the whole city. They may in general be unusual or irregular but in certain cases can nevertheless happen. These situations can be coped with by using exceptions.

Take the constraint expression `MinDistTreeBld` (from Fig. 8 in Sect. 4.2) as an example. Let us assume the ‘`minDistBld2Tree`’ is two meters. In most cases this statement is satisfied. But in some specific instances, the tree just leans on the wall or stands closer to a building than the allowed distance. It is good to reveal the common distance relation between a building and a tree and keep the distance, but it is difficult to give a clear-cut rule which describes all situations that have to be rejected. Instead of using ‘must be’ or ‘cannot have’ and the exact value, this example distance condition can be stated in natural language as:

‘a tree and a building have at least two meters in between’ (should constraint in class level) *or* ‘instance pair tree-building is an exception’ (exception in instance level) = true

So the principle can be summarised as:

An instance can be accepted (i.e. is true) if the ‘should’ constraint is satisfied or this instance is marked as an exception. In OCL-like format it can be expressed as:

```

Exception
  obj1 : OclAny
  obj2 : OclAny
context Building
inv minDistBld2Tree:
  Tree.allInstances()->forall(t | distance(t.geometry, geometry)
> minDistBld2Tree
  or
  Exception.allInstances()-> one(e | e.obj1.isKindOf(
Building) and e.obj2.isKindOf(Tree) and e.obj1.id = id and
e.obj2.id = t.id))

```

5.3 *Abstract/Specific Constraints*

Some rules, such as `OneObjOneLoc` in Sect. 4.1, as well as `SmoothValue Change` in Sect. 4.4, are applicable to many sub-classes. Some other rules, such as `SolarPanelOutside` from Sect. 4.4, are only applicable to specific object sub-classes. The former type of rules can be generalised into a higher level and inherited by subclasses. This distinction can lead to a new classification of constraints in a generic context—abstract/specific constraints.

5.4 *Constraints With/Without Parameters*

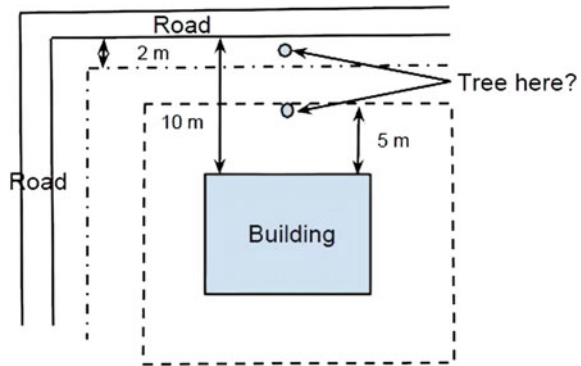
Parameterised constraints which function upon exact values of objects and their properties can be distinguished from non-parameterised constraints. Examples as such are `MinDist2Tree`, `MinDist2Road` in Sect. 4.3. It is difficult to assign an exact value to the class and reject all the instances which do not satisfy it, whilst the non-parameterised rules can use one statement (such as `ValidSolid` from Sect. 4.2) and reject all violating instances.

5.5 *Common Sense Rules*

When the constraints are many, it may happen that some rules contradict, which means no solution can be found to satisfy them all. In practice, the result is that no instance of a certain class can be stored in the database (some tables will always be empty). An example is made up here: three parameterised rules regarding distances amongst tree, building and road are proposed as (see also illustration in Fig. 10):

- `Tree-Building`: tree must be placed within a distance of five meters from building.

Fig. 10 A possible scenario of inserting a tree when tree-building-road distance constraints contradict



- Building-Road: there must always be at least 10 meters distance between a building and a road.
- Tree-Road: tree must be at a distance of less than two meters from a road.

As you can see, if buildings and roads are correctly stored, then it becomes impossible to have tree objects in the database. Because no solution can be found to satisfy both tree-building and tree-road constraints. Plainly, this outcome contradicts to the intention of having tree class in the database for it is created to allow storage of tree objects.

Explicit statement of this type of intention (let's call it common sense rules), which reflects human intuition or common sense, is subtle (easy to neglect) and yet important for constraints reasoning. They can be considered as rules of rules and may not directly apply to a specific object/attribute/relationship in the model. Therefore the common sense rules should be distinguished from the rest.

5.6 Multi-scale Consistency Rules

When a real world object or phenomenon is modelled with multiple scales, e.g. CityGML LoDs, it is necessary to have constraints that keep the geometry consistency between two different scales. For example, the same building represented in a finer scale and a coarser scale under the same reference system must not have coordinates that are too far apart (e.g. 100 m) in space. Neither should the volumes of a structure be too different in different scales. This type of constraints deals with relationships between the same objects in different scales, instead of relationships between different objects in the same scale. Therefore they should be distinguished from the same-scale-dealing constraints.

5.7 Indoor Consistency Rules

The best example of 3D constraints can be seen in 3D indoor modelling. Almost all 3D CityGML LOD4, IFC and IndoorGML maintain volumetric objects for rooms (CityGML), construction elements (IFC) and/or spaces (IFC, IndoorGML). The very common constraint is then room/spaces should not ‘intersect’ or must ‘touch’. In addition, some of the objects are nested in each other. For example a furniture is inside a room, or a window is inside a wall. This can generate a large number of topological relationships. Interesting example is IndoorGML. It is a relatively new standard providing a framework for deriving a network for navigation. The framework is based on duality graph and requires complete subdivision of space (topographic, security, sensor, etc.). Intersections of and gaps between spaces (navigable or non-navigable) are not allowed. The only possible relationship is ‘touch’. Despite the obvious need for constraints in 3D indoor modelling, to ensure good quality data, these constraints are not yet formalized and not yet part of today’s standards (IFC, IndoorGML). A similar approach as presented in this paper should be conducted to obtain the constraints. This is part of future work.

6 Conclusion

This paper demonstrates a developed methodology to model and implement 3D geo-constraints, following four stages: *natural language*, *geometric/topological abstractions*, *UML/OCL formulations*, *PL/SQL code*. This methodology also addresses the third dimension which has not been discussed much in present papers (mostly 2D), and can be applied to a broad context of geo-constraints modelling in both 2D and 3D.

A database in application domain that stores 3D topographic objects commonly seen in the urban environment was used to apply the methodology. Some constraints regarding the objects in 3D space have been formulated in this work. Because currently automatic model translation from OCL to SQL is not available, so PL/SQL code was written by hand. The challenges of automatic translation lie on the support of spatial types and operations in OCL standard (see also Sect. 4), multiple class expression of OCL (see Sect. 3.3) and sufficiency of spatial functions in the database (see Sect. 3.4).

New observations about constraint nature have also been discussed (in Sect. 5). Two ways of specifying a constraint can have sharp difference in terms of statement briefness and performance. New classes and classification methods have been found and proposed for a better understanding of constraints.

During this research, standards have been studied and used. The experience gained in using them has led to the conclusion that some improvements could be made by the owning organisations.

- OCL: It would be highly productive to extend current OCL standards with 3D spatial types/operations and topological relationships (see Sect. 3.4). The extension with multiple class expression is also highly demanded (see Sect. 3.3) which can also be accompanied with the extension of the constraint association in UML.
- CityGML: It would be advisable to have geo-constraints incorporated into the UML model, such as multi-scale consistency constraints (see also Sect. 5.6), so that city modelling could be more concise and well-defined (see also Sect. 3.4). Also indoor constraints should be developed (as in CityGML LoD4 and the new IndoorGML) which covers both Indoor-Indoor consistency and Indoor-Outdoor consistency. Examples for Indoor-Indoor are that rooms, corridors and other building interiors should not overlap but connect and partition whole building. Examples for Indoor-Outdoor are that indoor elements should be inside the outdoor model (envelope). This at all stages in time, both old, existing and future (planned) buildings.
- UML: It would be recommended to create a symbol—constraint association—to connect two classes that do not have normal types of association, and also allow using multiple class in OCL (see Sect. 3.3). At the finishing of this paper, research from (Egenhofer 2011) and [citemas2008reasoning was found which add multiplicity to the relational constraints between two classes with e.g. for-all, for-some. Therefore, the multiplicity (at instance level) of the involved classes can be added to the proposed constraint association. In order to avoid its inconsistency with the OCL expression, it would be better if this multiplicity is generated by the UML editor, e.g. EA or Eclipse.

Suggestions regarding to future work for researchers who wish to extend the work carried out in this research are:

1. The pseudo 3D Geo-OCL expressions from Sect. 4 need to be tested in conjunction with the UML diagrams.
2. It would be useful to extend EA, or Dresden OCL2SQL or other OCL code generation tools to enable automatic model translation from OCL (esp. spatial constraints) to SQL (see also Sect. 3.4). This of course can also be provided by software developers.
3. Further study can be conducted into detecting contradicting (spatial and non-spatial) constraints, as explained in Sect. 5.5. It can be done by using OCL verification tools, e.g. HOL-OCL, UMLtoCSP (Chimiak-Opoka et al. 2010). Or through First Order Logic reasoners that are adapted/extended to understand geographic predicates and types.
4. Corresponding functions in database need to be developed, esp. 3D and solids related, to implement 3D spatial predicates from extended OCL. This extension could be a common effort dedicated by the manufacturer, developers and researchers, as was presented in Sect. 3.4.
5. Test the performance of the trigger that uses 3D geometric operators. Geometric operations are often time consuming and it is often impossible to use trigger in

each update. Furthermore, when the amount of geometric objects in one update is great, how the performance of trigger will be is an interesting issue to study.

References

- Casanova, M., Wallet, T., & D'Hondt, M. (2000). Ensuring quality of geographic data with UML and OCL. In A. Evans, S. Kent and B. Selic (Eds.), *UML'00 Proceedings of the 3rd International Conference on The Unified Modeling Language: Advancing the Standard* (pp. 225–239). Heidelberg, Berlin: Springer.
- Chimiak-Opoka, J., Demuth, B., Awenius, A., Chiorean, D., Gabel, S., Hamann, L., et al. (2010). OCL tools report based on the IDE4OCL feature model. In *Proceedings of the Workshop on OCL and Textual Modelling (OCL 2010)* (Vol. 36, p. 18).
- Clementini, E., Di Felice, P., & van Oosterom, P. (1993). A small set of formal topological relationships suitable for end-user interaction. In *The 3rd International Symposium on Large Spatial Databases*.
- Cockcroft, S. (1997). A taxonomy of spatial data integrity constraints. *GeoInformatica*, 1(4), 327–343.
- Cockcroft, S. (2004). The design and implementation of a repository for the management of spatial data integrity constraints. *GeoInformatica* 8(1):49–69.
- Duboisset, M., Pinet, F., Kang, M.-A., & Schneider, M. (2005). Precise modeling and verification of topological integrity constraints in spatial databases: From an expressive power study to code generation principles. In L. Delcambre, et al. (Eds.), *Conceptual Modeling—ER 2005* (pp. 465–482). Klagenfurt, Austria: Springer.
- Egenhofer, M. (2011). Reasoning with complements. In *Advances in Conceptual Modeling. Recent Developments and New Directions* (pp. 261–270).
- Egenhofer, M. J. (1989). A formal definition of binary topological relationships. LNCS 367:457–472.
- Elmasri, R., & Navathe, S. B. (2003). *Fundamental of database systems* (4 ed.). Addison Wesley.
- Emgård, L., & Zlatanova, S. (2008). Implementation alternatives for an integrated 3D information model. In *Advances in 3D Geoinformation Systems, LNGC* (pp. 313–329).
- Geomatics, M. (2010). Measure the climate, model the city—acquiring and storing temporal and spatial data for climate research. Technical report. GM2100 Synthesis Project, 112 pp.
- Hespanha, J., van Bennekom-Minnema, J., van Oosterom, P., & Lemmen, C. (2008). The model driven architecture approach applied to the land administration domain model version 1.1, with focus on constraints specified in the object constraint language. In *FIG Working Week* (p. 19).
- INSPIRE. (2010a, April 26). D2.8.i.5 INSPIRE data specification on addresses guidelines.
- INSPIRE. (2010b, April 26). D2.8.i.6 INSPIRE data specification on cadastral parcels guidelines.
- ISO. (2008, September 17). 19107:2003. In *Geographic information—Spatial schema*. ISO.
- Kleppe, A., Warmer, J., & Bast, W. (2003). *MDA Explained: The model driven architecture: Practice and promise*. Addison-Wesley Professional.
- Kolbe, P. D. T. H., König, G., Nagel, C., & Stadler, A. (2009). *3D-Geo-Database for CityGML*. Institute for Geodesy and Geoinformation Science, Technische Universität Berlin, 2.0.1 edition.
- Louwsma, J. (2004). Constraints in geo-information models—applied to geo-VR in landscape architecture. Master's thesis, TU Delft. GIRS-2004-17, 104 pp.
- Louwsma J, Zlatanova S, van Lammeren R, van Oosterom P (2006) Specifying and implementing constraints in giswith examples from a geo-virtual reality system. *GeoInformatica* 10:531–550.
- Mäs, S., Wang, F., & Reinhardt, W. (2005). Using ontologies for integrity constraint definition. In *Proceedings of the 4th International Symposium on Spatial Data Quality'05*, Beijing, China.
- Oracle. (2010a). *Oracle Database, PL/SQL Language Reference, 11g Release 2 (11.2)*. E17126-04 edition, 756 pp.

- Oracle. (2010b). *Oracle Spatial Developer's Guide 11g Release 2 (11.2)*, 916 pp.
- Pinet, F., Duboisset, M., & Soulignac, V. (2007). Using UML and OCL to maintain the consistency of spatial data in environmental information systems. *Environmental Modelling & Software*, 22.
- Pinet, F., Kang, M.-A., & Vigier, F. (2004). Spatial constraint modelling with a GIS extension of UML and OCL. In U. K. Wiil (Ed.), *Metainformatics* (pp. 160–178). Salzburg, Austria: Springer.
- Ravada, S. (2008). *Oracle spatial 11g technical overview*. Presentation. Retrieved from, December 2010.
- Reeves, T., Cornford, D., Konecny, M., Ellis, J. (2006). Modeling geometric rules in object based models: An XML/GML approach. *Progress in Spatial Data Handling* 3:133–148.
- Servigne, S., Ubeda, T., Puricelli, A., Laurini, R. (2000). A methodology for spatial consistency improvement of geographic databases. *GeoInformatica* 4(1):7–34.
- Truyen, F. (2005). Implementing model driven architecture using enterprise architect—MDA in practice. White paper practice, Sparx Systems. Version 1.1.
- van Oosterom, P. (2006). Constraints in spatial data models, in a dynamic context, Chapter 7. In *Dynamic and Mobile GIS: Investigating Changes in Space and Time* (pp. 104–137). CRC Press.
- Wachowicz, S., Wang, F., and Reinhardt, W. (2008). Extending geographic data modeling by adopting constraint decision table to specify spatial integrity constraints, Chapter 2. In *The European Information Society: Leading the Way with Geo-information. LNGC* (pp. 435–454). Heidelberg, Berlin: Springer.
- Werder, S. (2009). Formalization of spatial constraints. In *12th AGILE International Conference on Geographic Information Science*. Germany: Leibniz Universität Hannover.
- Zlatanova, S. (2000). On 3D topological relationships. In *DEXA '00 Proceedings of the 11th International Workshop on Database and Expert Systems Applications* (pp. 913–919). London, UK: Greenwich.

Reconstructing 3D Building Models with the 2D Cadastre for Semantic Enhancement

Frédéric Pedrinis and Gilles Gesquière

Abstract Virtual city models are increasingly used in urban land management processes, which involve the use of different sources of spatial information. This heterogeneous data is, however, often complementary and it may be necessary to give the possibility to join information provided by different sources. This paper presents a method to enhance 3D buildings by using usual 2D vectorial polygon database. These polygons may represent districts, building footprints, or any segmentation of the urban area that adds information to the city model. The enhancement consists in using this polygon database to split the 3D buildings into a set of city objects where each element possesses a 3D geometry and the semantic information of the polygon it is linked to. In this paper, for an illustration purpose, we will present how to create this link between 3D buildings and the cadastre map, in order to create a set of semantically rich 3D building models.

Keywords 3D virtual city · Cadastre · CityGML · Semantic information

1 Introduction

Many cities possess a virtual 3D model representing their territory. They have at their disposal various methods to build their virtual double, depending on the available data and the desired model. Therefore, international standards are currently being developed to manage these 3D city models, like the CityGML standard¹ created in 2008 by the Open Geospatial Consortium² (Gröger et al. 2008;

¹<http://www.citygml.org/>.

²<http://www.opengeospatial.org/>.

F. Pedrinis (✉) · G. Gesquière
LIRIS, University of Lyon, Lyon, France
e-mail: frederic.pedrinis@liris.cnrs.fr

G. Gesquière
e-mail: gilles.gesquiere@liris.cnrs.fr

Kolbe 2009). It is a format that allows the complete description of a building, including 5 different Levels of Details (LoD) and semantic information, but these are rarely present in a newly generated 3D model.

3D urban model construction processes mainly use imagery and/or LiDAR scans to create the 3D geometries of the city (Musialski et al. 2012). They focus on the elaboration of correct 3D geometries in order to propose a visually convincing representation of the city. The semantic richness of buildings defined in the 3D geometries is not really discussed, and these objects can't be directly connected to urban databases. However, other representations of the city can contain the information needed to complete the database. 2D polygonal databases, like for instance the cadastre map, can be used to generate this enhancement of a 3D city model. Such a 2D polygonal database describes an urban area and provides semantic information for each of its elements represented by a polygon. An example would be a partition of the city into polygonal zones according to some characteristics like the buildings date of construction or the average age of their inhabitants. More generally, the method could be used with every 2D database providing a partition relevant to city objects.

In this paper, we choose to focus on the use of cadastre maps to enhance the 3D buildings. In addition to a precise notion of buildings, the cadastre provides semantic information and can grant access to non-georeferenced databases. For example, we can access to the date of construction of a building, its owner's name, its number of housings, or to collect the result of a thermal simulation attached to cadastral buildings. This is why we want to develop a method to link a given 3D building model with the corresponding cadastral polygons. This link is used to create a new model containing, for each building initially defined in the cadastre, a 3D geometry computed from the initial 3D model. The union of these buildings would thus have a 3D model similar to the initial one, and a possible connection with all databases accessible from the cadastre. Figure 1 illustrates this possibility. The cadastral ID provides an access to semantic information. Thanks to our enhancement process described in this paper, the link between the cadastre and the initial 3D model of buildings transfers this accessibility to the resulting model.

Two different ways to generate the link between the cadastre, or any other polygons database, and 3D models can be identified. The first one is to directly involve the 2D polygons in the construction process of 3D geometries, as in Durupt and Taillander (2006), Vallet et al. (2011), and Aguiaro (2014), who used cadastral data to design a robust 3D reconstruction of buildings. The second way consists in a post-process intervention on the geometries in order to associate them to the needed semantic information. In this case, we don't need a process to build a 3D model from raw data, as we can just use an existing one. Ross et al. (2009) propose to add information to 3D city models to create complex 3D Land Information Systems. Among the concerned data, they choose to automatically add attribute information extracted from the cadastre map to the 3D geometries of buildings. The authors do not really describe how this is processed. However, such additions should require a certain overlap between the 3D geometry and the cadastre to be relevant, which is not the case in our datasets. For each object initially called "building" in the 3D model (Fig. 2, top-left), multiple cadastral objects, which also represent "buildings"

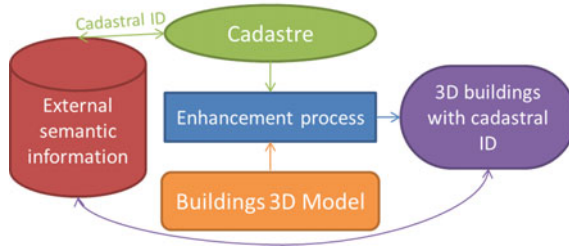


Fig. 1 The link created between 3D buildings and the cadastre generates a new 3D model composed of precise buildings that can access to external databases with their cadastral ID

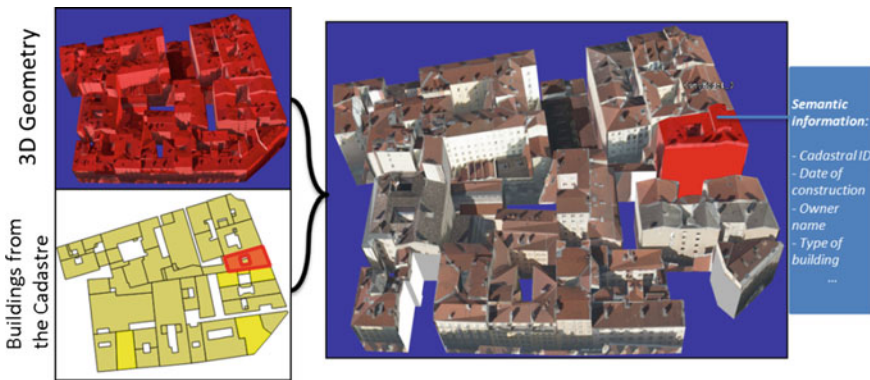


Fig. 2 A single building from a 3D city model (*top-left*) and the cadastral buildings of the same area (*bottom-left*) are linked; a split 3D model according to the cadastre is generated by our method (*right*). A building is picked (highlighted in red), and some other ones are hidden (yellow polygons on the *bottom-left*). Each of these buildings has a set of semantic information collected thanks to the cadastre

according to their definition (Fig. 2, bottom-left), can match. We have to decide which cadastral object may be linked to the 3D geometry, and which should be ignored. The proposed solution is to split the geometry to provide a new 3D model for each cadastral footprint (Fig. 2, right). The geometry and the texture coordinates are extracted from the initial 3D buildings.

Our tests are made on the data of the city of Lyon (France), which are available on the Lyon Data website.³ We implemented our method on the software 3D-Use developed by our team.⁴

In this paper, we will first present an overview of our method (Sect. 2). Then, we will present how we create the link between the two datasets with 2D processes (Sect. 3), before involving the 3D to get a better result (Sect. 4). The paper

³<http://data.grandlyon.com/>.

⁴<http://liris.cnrs.fr/vcity/wiki/doku.php>.

continues with the generation process of the new 3D model, which contains the semantic information stored in the cadastre (Sect. 5). We will then propose some tests and discussions (Sect. 6) before concluding and presenting our future works (Sect. 7).

2 Overview of Our Method

To illustrate our method, we will use two data sets. This first one is a CityGML file containing 3D models of buildings in LoD2 (Fig. 3, left). The second one is the French Cadastre which is a 2D vectorial polygons database (Fig. 3, right).

Since the buildings' definition is conveyed by the cadastral polygons, we want to modify these polygons so that they completely cover the surface represented by the footprints of the input 3D models of buildings. Since buildings are modelled in LoD2, covering their footprints means being able to entirely assign their geometry to cadastral polygons and to regenerate 3D geometries, as we will see in Sect. 5. However, if we look closely at the pictures of Fig. 3, we can already identify some inconsistencies between the footprints, such as lacks of courtyards or some differences for small pieces of buildings. This reflects that the two databases do not properly overlap. We illustrate this observation on the left picture of Fig. 4, which is a focus on the rectangular selection of the right picture of Fig. 3.

These differences may have several explanations that are listed below. The first one is linked to the precision that is not the same for the two datasets. A second set of difference is due to the acquisition date. Third, these acquisition methods are simply not comparable. For example, in CityGML LoD2 models of buildings a footprint is defined by the roof shape. As we can see on the right picture of Fig. 4, there are common cases of buildings where such footprints are irrelevant, when



Fig. 3 CityGML buildings are on the *left*, cadastral buildings on the *right*. The *rectangular* selection on the cadastre corresponds to a zoom used in Fig. 4

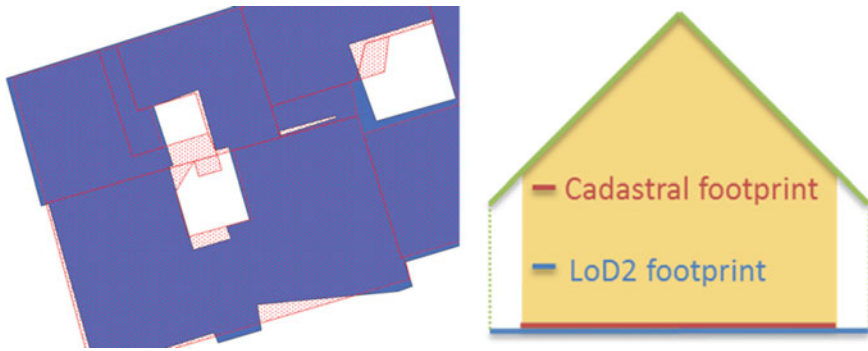


Fig. 4 Inconsistencies between footprints of 3D buildings (*blue*) and cadastral polygons (*red*)

there is an offset between the wall and the roof border, and the cadastre logically does not use them. The footprints are wider than the footprint generated from the real walls that the cadastre probably use. Even if the cadastre footprint logically represents the more realistic representation, our objective here is to split the LoD2 3D geometry and link it with cadastral areas.

The exact process of generating 3D geometries from a footprint and a 3D model will be presented in Sect. 5. If we use this process on the raw data of the cadastre, some wall and roof polygons of the 3D model will not be assigned to the newly generated buildings. More generally, the 3D geometry will be unpredictable, and probably visually unrealistic. We would only obtain a cut of the 3D geometry with an unrelated mask defined by the cadastre, and a part of this geometry would be lost. Figure 5 illustrates this problem: the cut process without any preparation of the footprints is presented on the left, while the initial model is on the right. The lost part of the 3D geometry is illustrated in green, wall and roof polygons that are not entirely on the area defined by the cadastre cannot be naïvely assigned to a building in the final 3D model.

In our dataset based on Lyon CityGML files, we have calculated that the proportion of 3D models, in terms of superficies, that are not covered by the cadastre is about 8 % of the total building superficies. If we cut only with the raw cadastre, we would thus lose a significant part of the 3D model, especially as these losses mainly concern the external envelope of the buildings, which is the most visible part mapped with texture.

Another problem that may appear with a raw cut is the gap between the visible separations of two buildings on the 3D model and the corresponding split defined in the cadastre. A case is illustrated on Fig. 6. A cut is generated with the unchanged footprints of the cadastre (left). The cut generated by our method, which adapt the footprints according to the 3D geometry, is presented on the right and is visually more satisfying.

This means that at least one of the model has to be modified. Since we don't necessary have the data to construct a new 3D textured model, we focus on

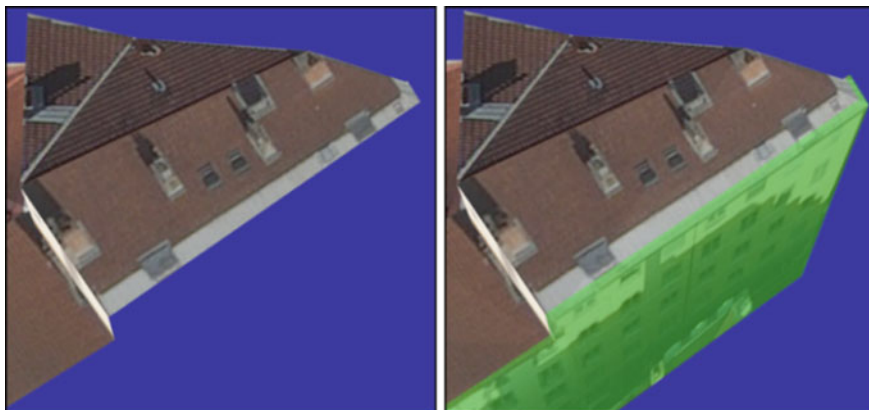


Fig. 5 Cut of the 3D model according to the cadastre without correction will lead to have missing parts (*left*). The lost part of roof and wall polygons are represented in *green* on the initial model (*right*)

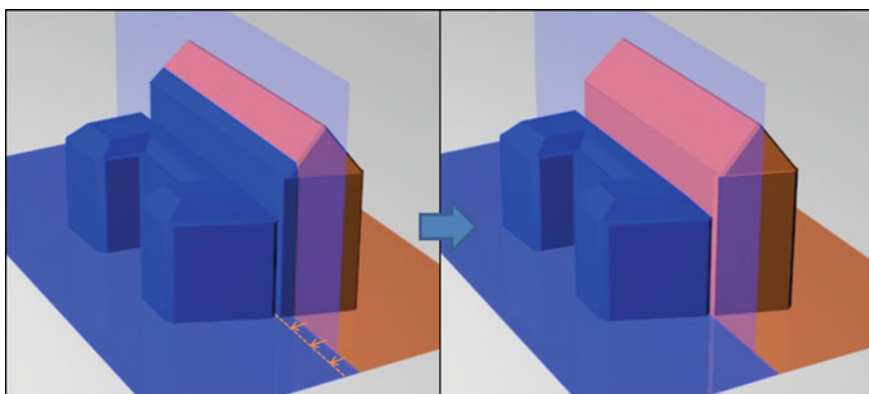


Fig. 6 *Left* image corresponds to a rough cut with the unmodified cadastral footprints. The *right* one is generated with the corrected footprint that takes into account the 3D geometry of the buildings models

adapting the cadastral footprints to the initial 3D geometry, and to keep a link between these new polygons and the original cadastre.

These problems have been identified in Pedrinis et al. (2015) and a first method has been proposed to solve this pairing between the cadastre and 3D buildings in order to provide a better detection of changes between two 3D models available at two different dates. However, this first tentative is not enough to solve all the problems presented in this section.

A global overview of our method is proposed in Fig. 7; it also introduces the notations used in the following of this paper. G is a set of 3D geometries that are

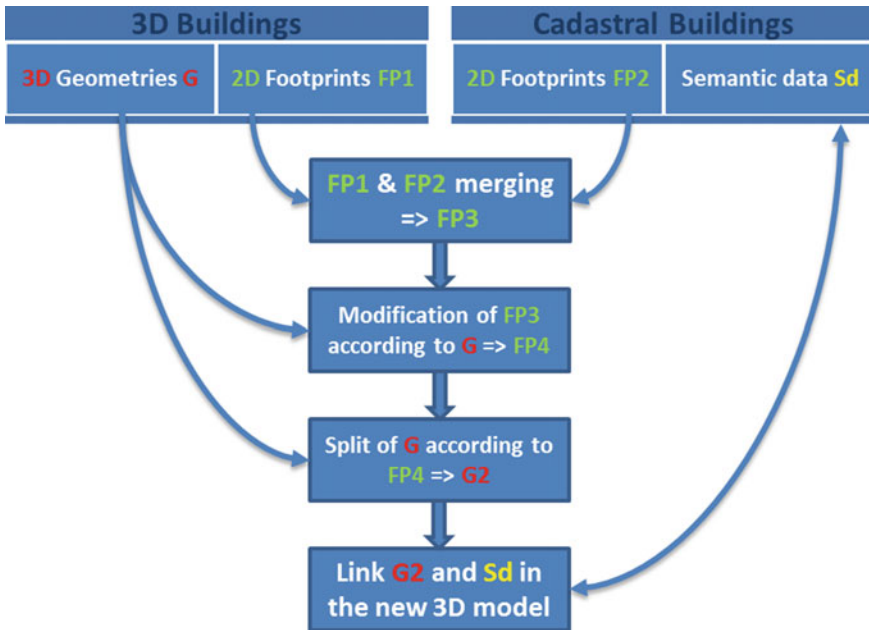


Fig. 7 Presentation of the main steps of our method. The sets of 2D polygons are in *green*, while the 3D geometries are in *red*

used to generate building’s footprints by projecting the roof polygons on the ground (*FP1*). On the other side, the set of polygons *FP2* represents the footprints according to the cadastre. Each footprint may have semantic information *Sd*. The two sets of footprints are merged to generate a new one, *FP3*, defining the cadastral footprints from *FP2* that are expanded to cover the entire surface of *FP1*. This step corrects the error shown in Fig. 5. For that presented in Fig. 6, *G* is used to modify *FP3* by adapting footprints to the related 3D geometry. This new set of footprints is called *FP4*. It is then possible to generate a new 3D model *G2* by cutting the 3D polygons of *G* according to polygons of *FP4*. A natural link is now available and *G2* can access to the semantic data *Sd* initially proposed by the cadastre.

3 Merging Footprints from the 3D and Cadastral Models

The previous section has pointed out the need to adapt cadastral polygons to the footprints generated from the 3D models of buildings, in order to be able to use them for the generation of the 3D geometries of new buildings. We thus have two sets of polygons *FP1* and *FP2* and we want to generate a new set of polygons *FP3* that will contain footprints of *FP2* modified to cover the entire surface of polygons from *FP1*. For the example represented on the right picture of Fig. 4, this means

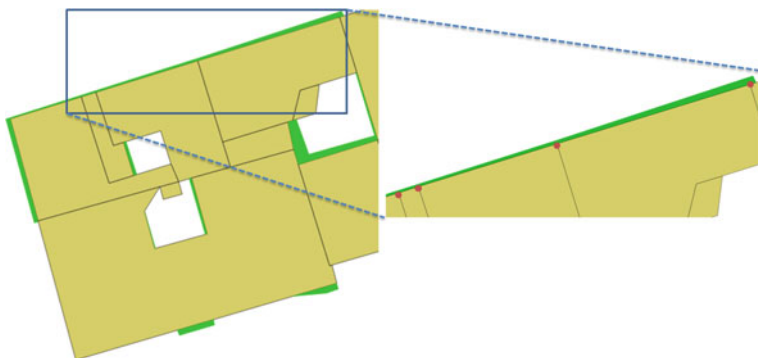


Fig. 8 Difference between $FP1$ and $FP2$ (beige polygons). The resulting polygons $Diff$ are represented in green. Red points that are on the boundary of two cadastral footprints of $FP2$ are then detected

that the red polygons will entirely cover the blue one. To achieve this, the difference between $FP1$ and $FP2$ is computed to produce a set of polygons $Diff$. These areas correspond to the surface of $FP1$ that was not initially covered by $FP2$ (presented in green in Fig. 8).

These polygons $Diff$ represent the area that needs to be filled by the expansion of the polygons of $FP2$. However, this expansion has to be controlled: the result has to be visually convincing considering that it will define the future division for 3D models of buildings. Each $Diff$ polygon is split according to its neighbouring polygons from $FP2$, in order to assign the best possible shape to each of them.

Points on $Diff$ that are on the boundary of two cadastral footprints are detected (red points of Fig. 8). These points are used to split $Diff$ into distinct polygons, which will be assigned to the concerned cadastral polygons. The resulting footprints need to be visually credible.

The more convincing line passing through P that extends the border line must be chosen to split the $Diff$ in two parts. To reach this goal, P is projected on other edges; the best projection P' according to parameters of distance or angles. The angles between $[PP']$ and the two neighbouring edges of P should be close and the $[PP']$ segment should be as short as possible. In our implementation, we chose the projection with the shortest segment, except if the two angles are too different. In the example of Fig. 9, $P2$ and $P3$ are eliminated by P' and $P1$ due to their smaller distance to P . $P1$ is rejected because the angles formed with the neighbouring edges of P are too different. P' is thus chosen and the line defined by P and P' will be used to split $Diff$ into two resulting polygons.

This process applied on the selected area of Fig. 8 provides the result presented in Fig. 10. The last step of this process consists in assigning each sub-polygon (blue ones in Fig. 10) to the nearest cadastral footprint (beige polygons) to expand it. The chosen corresponding footprint is the one with which the polygon shares the longest border.

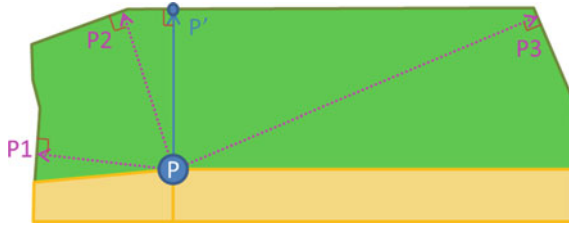


Fig. 9 Orthogonal projections of point P from polygon $Diff$ (green) on its other edges gives points $P1$, $P2$, P' and $P3$. Due to angle and distance criteria, P' is chosen. Beige polygons are cadastral footprints that were used to detect P

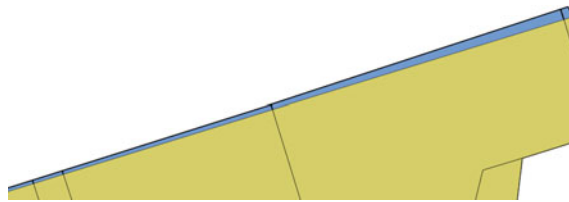


Fig. 10 $Diff$ is split into blue polygons, which are assigned separately to their nearest cadastral polygon (beige)



Fig. 11 On the left, cadastral polygons are extended to cover the footprint $FP1$ extracted from the 3D model. On the right, blue polygons are intersected with $FP1$ to delete the non-needed part (not present in the 3D model) and obtaining $FP3$. The orange rectangle corresponds to a zoomed area used in Fig. 12

The result on the entire area is shown on Fig. 11 (left). A cadastral footprints covering $FP1$ is now available, but they do not perfectly fit: some parts of these footprints are not present in $FP1$, and are not required since they do not correspond to the 3D geometry G . The intersection between this set of footprints and $FP1$ is then computed (Fig. 11, right). This last result is called $FP3$ according to Fig. 7 notations. It represents the exact footprint of G split in a set of polygons corresponding to cadastral objects.

4 Correction of Generated Footprints by Using 3D Geometry

This set of polygons $FP3$ allows to generate a 3D model, with the process explained in Sect. 5, since these footprints now cover the entire surface of FPI . However, as we can see in Fig. 12 (left), which is a focus on the orange selected area of Fig. 11, the result would not be visually satisfactory. Initially, the footprints generated from CityGML buildings only take into account the (x, y) coordinates of roof polygons (Fig. 12, middle). This means that 3D discontinuities of the roof, as seen on the Fig. 12 (left), do not interfere in the choice of splitting edge between two new footprints.

In order to prepare better 3D models, we have to use this information to propose a new version of footprints that will consider the 3D shape of the future generated buildings. The footprints of $FP3$ is split according to the roof discontinuities: if a wall of the input 3D model crosses a footprint of $FP3$, it must be split along the 2D line representing this wall (Fig. 12, right). The split results in a set of polygons for each element of $FP3$. These polygons are represented on Fig. 13. Each of them keeps a link to its original footprint of $FP3$. Among these polygons, those that are characteristics of visual inconsistencies are the small or elongated ones: they represent footprints spilling over their neighbours. They are represented in light green in Fig. 13 and we will call them Sp . We need to determine which of these polygons must be reallocated to other footprints in order to correct the visual inconsistencies.

The polygons are added to Sp if they fall into one of these 3 categories: the very small polygons, the small and non-compact polygons, and the elongated ones. Two parameters are proposed to detect the concerned polygons: the area and the Gravelius Compactness Coefficient (Kg) of a polygon. The second parameter Kg for a polygon P is defined by $Kg(P) = Perimeter(P) / (2 * \sqrt{\pi * Area(P)})$. This coefficient is initially used to analyse watersheds (Pareta and Pareta 2011). Kg is equal to

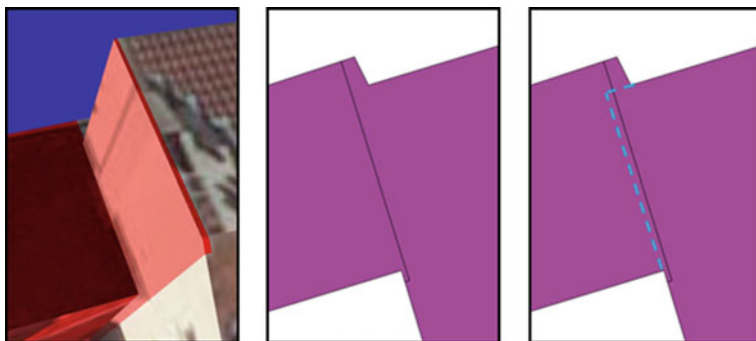


Fig. 12 On the *left* is shown the corresponding 3D cut of the footprints (*middle*), which are taken from the selected area (*orange rectangle*) of Fig. 11. The expected cut corresponding to the wall of the 3D model is represented on *blue* (*right*)

Fig. 13 Split polygons according to 3D discontinuities. *Light green* ones have been selected for their elongated shape or small area



1 if P is a circle and is inversely proportional to its compactness (Fig. 14). This parameter is used to detect elongated polygons.

Each of the polygons Sp is then compared to the neighbouring footprints of $FP3$. The polygon is reallocated to the footprint with which it shares the longest

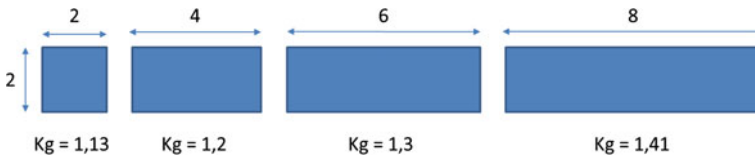


Fig. 14 Some values of the Gravelius Compactness Coefficient (Kg)

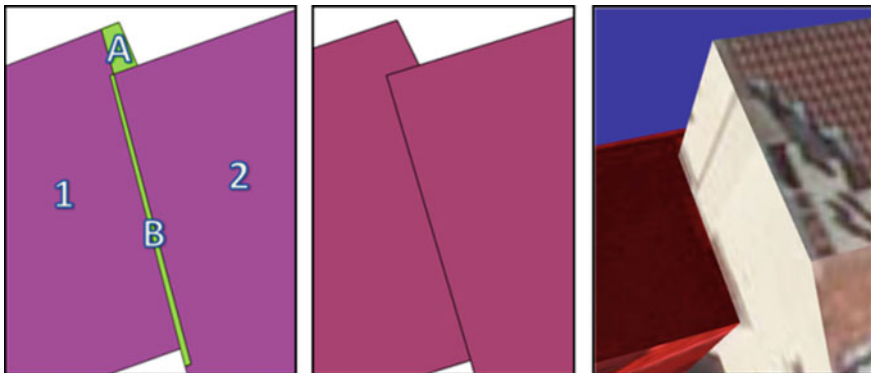


Fig. 15 Elongated polygons A and B are detected (*left*) and are respectively merged to footprints 1 and 2 (*middle*) in order to propose two distinct footprints that will give a visually satisfying 3D geometry (*right*)

Fig. 16 Final version of footprints *FP4* that will be used to generate 3D models of buildings



continuous border in the 3D model. If no borders are found, the polygon is left to its original footprint.

For the illustration presented in Fig. 12, we detect on the left of Fig. 15 the elongated polygons *A* and *B* (green polygons). We test them with the two cadastral footprints *1* and *2* (purple polygons). *A* and *1* (resp. *B* and *2*) are considered as the better neighbours. We can see on the middle of Fig. 15 the two resulting footprints and, on the right, the resulting 3D models.

The final version of the buildings footprints *FP4* is presented in Fig. 16. This set of polygons can be used now to generate the 3D model.

5 Creation of the New 3D City Model

Each footprint stored in *FP4* represents a building. We want to create a new 3D city model containing all these buildings, so we have to generate 3D geometries *G2* for each of them. Since we want to generate a 3D model for each element of *FP4*, we split the 3D polygons of *G* accordingly to the footprints of *FP4*, before assigning the resulting polygons to the concerned footprints in the new CityGML buildings. Textures coordinates are then computed for each building by using those stored in the input 3D model. However, existing walls are not enough if we want an independent and complete building in the final CityGML file: we have to generate hidden walls to close each building. Otherwise, we would have holes in the geometry when a created building would be displayed alone. We thus detect edges of *FP4* that represent border between two footprints (green edges in Fig. 17), and we create walls or extend existing walls at their position. For the example of Fig. 15, we can see a pre-existing wall on the border between footprints *1* and *2*. This wall has to be extended to the ground to generate two closed geometries. These generated walls do not have textures because they were initially hidden, so they are

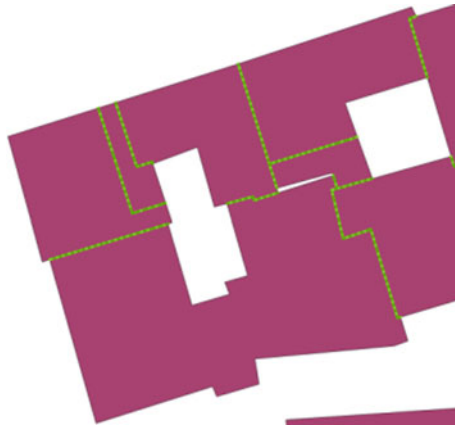


Fig. 17 Detection of missing walls for generated buildings



Fig. 18 Semantic information extracted from a footprint and linked to a 3D building

displayed with a default colour (here white) and can be viewed by hiding their neighbours (Fig. 2, right).

Since polygons of *FP4* are originally extracted from cadastral footprints, the semantic related information can be assigned to the corresponding generated 3D model (Fig. 18). For instance, here, data describing the construction and modification dates, the surface and the perimeter are available.

6 Performance and Discussion

We have tested our method on two districts of the city of Lyon (France). The result on the first district is presented Fig. 19. Table 1 gives some performances data generated with a computer on Windows 7 with an I7-4790K @ 4.00 GHz CPU.

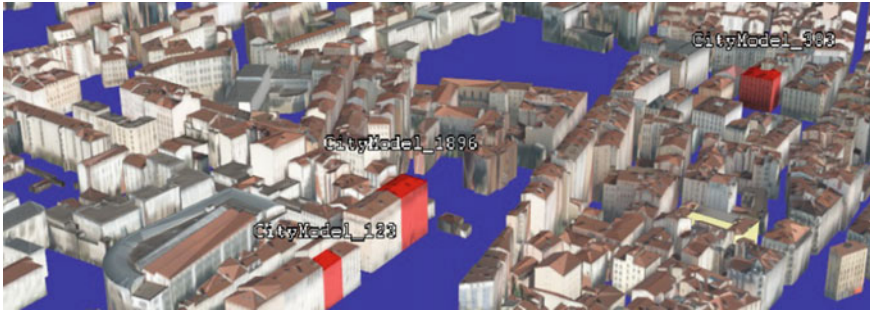


Fig. 19 Enhanced 3D model describing the first district of the city of Lyon. A multi-selection of generated buildings is highlighted in *red*

Table 1 Performance of enhancing CityGML for two districts of the city of Lyon. Each building of the output CityGML file corresponds to a cadastral footprint

District (superficy)	Input CityGML file		Enhancement process	Output CityGML file	
	Number of buildings	File size (Mb)	Time	Nb of buildings	File size (Mb)
Lyon 01 (1.5 km ²)	409	110	1197 s (20 min)	2640	132
Lyon 04 (2.9 km ²)	999	118	1461 s (24 min)	4339	148

We can see on Table 1 that the number of buildings logically increases after the process; however it does not mean that all buildings from the initial CityGML file or those from the cadastre have been kept. It is possible that some footprints did not find any equivalent in the other database. This information can be exported by our software 3D-Use since it may be useful in a validation process of the cadastre for example: these differences can be representative of an out of date cadastre. In our case, this may signify that our process should then require a human intervention to verify the corresponding buildings, on the field or with another model of the city. After discussing with urban operators who manually realize a similar geometry cut, we do think it is anyway necessary to keep a human validation step in our global enhancement process. Some cases lead to disagreements even between different human operators, so changes should be possible on the created model by following the process depicted in Fig. 20. *FP4* are used to generate *G2* but an expert has the possibility to detect inconsistencies in the 3D model cut, eventually guided by computed indicators based on 3D analysis processes. Concerned footprints are then exported to allow modify manual correction by the expert before reinserting them in the set of footprints used to generate a new version of the buildings 3D model. This process can be repeated several time and stopped by the expert or when the possible inconsistencies become under a given threshold.

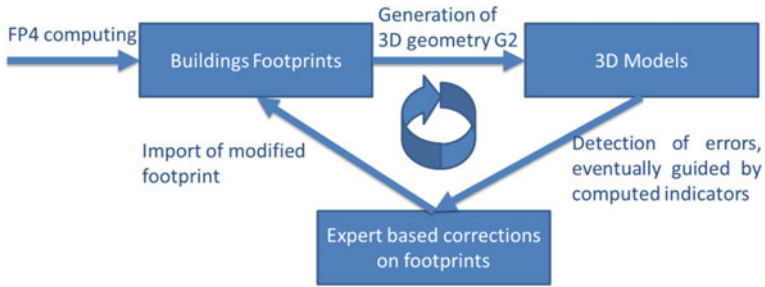


Fig. 20 Semi-automatic validation and correction of the computed footprints

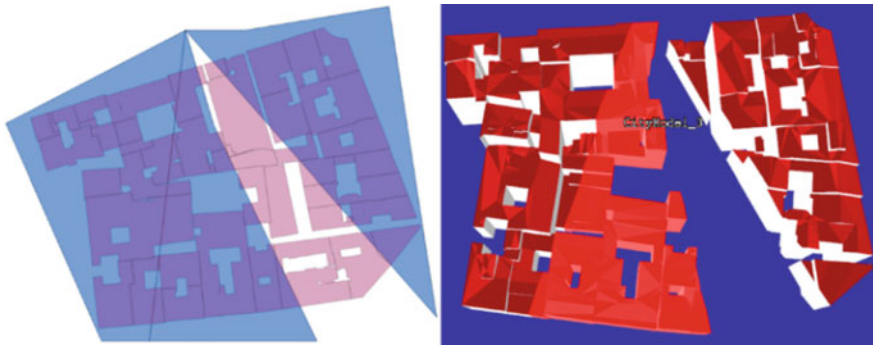


Fig. 21 Cut of a 3D building (on the right) with some polygons (blue ones, on the left)

Decomposing the method in two parts also gives the possibility to use an extra set of polygons in order to cut 3D geometry. We illustrate this possibility in Fig. 21: a set of random polygons (in blue) are used to cut the initial building. In this way, the second part of our method could be used in several other applications like a tiling process for example, or for any method that requires to group buildings accordingly to 2D shapes, which could eventually contain semantic information.

7 Conclusion and Future Works

Nowadays, a large amount of 3D city models are available. The accuracy of reconstruction processes of these models has been increased but the linked semantic is often poor and it may be difficult for a user to link the needed information with these 3D models. On the other side, a large diversity of semantic information linked to 2D models like the cadastre is available. In this paper we propose a method to link these two kinds of data in order to enhance dedicated data linked to 3D buildings.

Since these two models have no reason to be correctly fitted, some processes are required to pair their representation and to generate a merging model. We have proposed corrections on the 2D polygons to make them split the entire 3D geometry and to get convincing new 3D models of buildings. The two datasets are thus closely related. The 3D geometry is involved in the description of the footprints based on the 2D database, while each of them is then used to cut the 3D model and generate a new 3D building. This link between two datasets can also be used to compare two models representing the city at two different dates (inducing a real enhancement of the method proposed in Pedrinis et al. (2015)). It also allows to evaluate one of the model by a second which is deemed true.

The presented processes are not necessarily sufficient to generate a city model wholly convincing. They may need to be followed by a semi-automatic procedure involving an urban expert to produce a better 3D model of the city.

The presented method has been presented to experts. The feedbacks are very positive. Our approach could provide a good alternative to their actual time consuming methods.

Nevertheless, some works remain to be done, in particular to propose a semi-automatic loop for a guided correction by experts in our software. For example, even if the process that automatically corrects error such as 3D inconsistencies in the resulting model can be improved, it should be more relevant to begin by just proposing a detection process of these errors to help the expert correction task. For example, we may propose an evaluation of our results by computing matching indicators between a cadastral footprint and the corresponding 3D geometry.

The method presented in this paper was developed with LoD2 buildings according to the CityGML definition, it may also be necessary to extend it to more precise models.

Acknowledgments This work was performed within the BQI program of Université Lyon 1. This work was also supported by the LABEX IMU (ANR-10-LABX-0088) of Université de Lyon, within the program “Investissements d’Avenir” (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR). Data are provided by “Lyon Métropole”. The authors would like to thanks the DINSI team of Lyon Métropole for their valued feedbacks during this collaboration.

References

- Aguiaro, G. (2014). From sub-optimal datasets to a CityGML-compliant 3D city model: Experiences from Trento, Italy. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-4, 7–13.
- Durupt, M., & Taillandier, F. (2006). Automatic building reconstruction from a digital elevation model and cadastral data: An operational approach. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(Part 3), 142–147.

- Gröger, G., Kolbe, T. H., Czerwinski, A., & Nagel, C. (2008). OpenGIS city geography markup language (CityGML) encoding standard, Version 1.0.0, International OGC Standard, Open Geospatial Consortium, Doc. No. 08-007r1.
- Kolbe, T. H. (2009). Representing and exchanging 3D city models with cityGML. In J. Lee & S. Zlatanova (Eds.), *3D geoinformation sciences*. Berlin: Springer.
- Musialski, P., Wonka, P., Aliaga, D. G., Wimmer, M., van Gool, L., & Purgathofer, W. (2012). A survey of urban reconstruction. Eurographics 2012—State of the Art Reports (pp. 1–28).
- Pareta, K., & Pareta, U. (2011). Quantitative morphometric analysis of a watershed of Yamuna basin, India using ASTER (DEM) data and GIS. *International Journal of Geomatics and Geosciences*, 2(1), 248–269.
- Pedrinis, F., Morel, M., & Gesquière, G. (2015). Change detection of cities. In *3D geoinformation science* (pp. 123–139). Springer International Publishing.
- Ross, L., Bolling, J., Döllner, J., & Kleinschmit, B. (2009). *Enhancing 3d city models with heterogeneous spatial information: Towards 3d land information systems*. In *advances in GIScience* (pp. 113–133). Berlin: Springer.
- Vallet, B., Pierrot-Deseilligny, M., Boldo, D., & Brédif, M. (2011). Building footprint database improvement for 3D reconstruction: A split and merge approach and its evaluation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(5), 732–742.

A 3D LADM Prototype Implementation in INTERLIS

Eftychia Kalogianni, Efi Dimopoulou and Peter van Oosterom

Abstract The massive developments and uses of high-rise buildings indicate that the demand for use of space above and below the ground surface is rapidly increasing in recent years. The same applies to Greece, where the existing cadastral model does not cover the 3D needs and does not conform to international standards. In this paper, a model is proposed, considered as an effort for overcoming these shortcomings, based on international standards, including the representation of a wide range of different types of spatial units, organized in levels according to the LA_Level structure of ISO19152 LADM. It is a proposal for a comprehensive multipurpose LAS supporting 2D and 3D cadastral registration in Greece. A prototype system was developed to exploit the strengths and limitations of the proposed conceptual model, as well as to investigate the efficiency of technological tools. Experience from the prototype will be used to further improve the conceptual model. The steps that were followed were: the description of the prototype in UML diagrams, the implementation via INTERLIS, a Swiss standard modeling language for geodata exchange, the selection of the most appropriate technical model/format to implement and visualize the result in 3D environment and finally the conversion and/or creation of sample data into the model. In this paper it is explored how INTERLIS can be used in actual implementation of land administration system based on LADM. During the development of the prototype many design decision have been taken and these are then analyzed, together with technical problems and challenges for future work.

E. Kalogianni (✉) · E. Dimopoulou
School of Rural and Surveying Engineering, National Technical University
of Athens, Athens, Greece
e-mail: efkaloyan@gmail.com

E. Dimopoulou
e-mail: efi@survey.ntua.gr

E. Kalogianni · P. van Oosterom
Faculty of Architecture and the Built Environment, Delft University
of Technology, Delft, The Netherlands
e-mail: P.J.M.vanOosterom@tudelft.nl

Keywords Land Administration Domain Model (LADM) · INTERLIS · 3D cadastre · Constraints · Technical model

1 Introduction

Competition for space resulting from population growth, limitations of land supply for infrastructure developments in urban areas together with the complicated use of space led to the development of multi-level and vertical constructions. All those developments not only increase the complexity of urban environments, but also affect land property ownership interests attached to the underlying land (Aien and Rajabfard 2014). The complexity of the property ownership interests cannot be sufficiently addressed in all situations by the existing 2D cadastral systems. For that reason, 3D Cadastre has been introduced as a solution to this problem and FIG decided to establish a 3D Cadastre Working Group to make further process (van Oosterom 2013).

The key for successful management of uses and ownerships of property objects and spaces above and below the ground and their corresponding uses and ownerships rights is the availability of a reliable information system with a multipurpose character. Such information system is expected to be able to handle information related to 2D and 3D geometries and their corresponding attributes that reflect complexities of rights, restrictions and responsibilities of 3D objects (Budisusanto et al. 2013).

A Multipurpose 3D Cadastre can be defined as an integrated land information system containing legal (e.g. tenure and ownership), planning (e.g. land use zoning), revenue (e.g. land value, assessment and premium) and physical (e.g. cadastral) information (Choon and Seng 2013). Under this concept, the need to organize all the information related to land registries in one model and enable transparent geo-information exchange is growing.

The development of multipurpose land administration systems (MLAS) is a complex task. Taking this situation into account, there is a need to standardize the process of land management, introducing models for the effective management of the properties and the property rights attached to them with a multipurpose character.

Land Administration Domain Model (ISO 19152 2012), 3D Cadastre Data Model (3DCDM), SDTM and ePlan (2010) are some modeling approaches concerning how 3D MLAS can be based and organized accordingly. Those cadastral models manage and maintain 3D land and property ownership interests (including 3D RRRs) and represent legal objects. However, one of the most important elements of a cadastral model is the ownership boundary. Since ownership boundaries are often represented by building's physical structures, such as walls and floors, cadastral data model should relate—or even integrate—both legal and physical objects of urban features (Aien 2013).

Towards this integration, the derivation of a technical model from the conceptual is key challenge. Nowadays, many tools have been developed in order to facilitate and automate this conversion. Enterprise Architect, a UML modeling tool from Sparx Systems, and INTERLIS, a SWISS standard that allows co-operation between geographic information systems, are two of the most significant tools in this domain (COGIS 2006).

Based on the points mentioned above, this paper deals with the development of a prototype of 3D multipurpose land administration system that is capable to facilitate the registration of multi-level and multi-use property spaces using LADM principles and comply with national law. The methodology is implemented for Greek cases and in particular, for a comprehensive land administration system based on LADM for 2D and 3D cadastral registration system. The model takes into account the existing spatial and administrative registration systems in Greece, and is partly based on new developments inspired by the LADM standard and its implementation in other countries.

The main focus of this paper is to describe how a technical model, a database schema (SQL/DDDL) or a data exchange format (GML, XML, etc.) can derive from the conceptual model described in UML class diagrams and discuss how reliable this conversion can be by using existing tools.

The remainder of this paper is organized as follows. Section 2 elaborates the basic modeling principles, the current standardization options on the land administration domain and the interaction between multipurpose land information systems and the current standardization developments.

Section 3 provides the steps on the development of the prototype, elaborates the materials and methods used in each step and discusses the result. Finally, Sect. 4 summarizes the paper concluding the major findings and proposes future development of the prototype.

2 A 3D Multipurpose LAS

2.1 3D LAS Standardization Options

Significant progress has been made in advancing the concept of 3D cadastres and related technologies to facilitate their realization. These advances have been gaining momentum: over the last few years, the endorsement of different standards has occurred, and jurisdictions that currently develop prototype systems and undertake pilot trials to test conceptual boundaries and appropriateness significantly increased.

After prior research and prototype developments, a new era has arrived with the implementations and pilot programs of the first 3D cadastral systems in operation. Therefore, the use of existing standards and shared-terminology and concepts are key challenges towards this direction (van Oosterom 2012).

Standardization is a well-known subject since the establishment of LAS; even since paper-based systems existed. Over the last years, a lot of research has been

conducted in the domain of standardization regarding land administration practices. Many standards all over the world have been developed dealing with the modeling of the whole system, the data types, the metadata, the geometry types, the representation, etc. However an international standard for land administration systems didn't exist.

Land Administration Domain Model, ISO19152, was introduced as a model to create standardized information services at an international context, where land administration domain semantics have to be shared between regions or countries, in order to enable necessary translations. It is a simple model with flexible and extensible elements based on the pattern “*people—land*” relationships.

The LADM was approved as an official International ISO Standard on November 1st (2012). It covers basic information-related components of land administration (including those over water and land and elements above and below the surface of the earth). The model provides a conceptual schema with three basic packages (Lemmen 2012):

1. Parties, which means people and organizations that perform transactions,
2. Basic administrative units, including rights, restrictions and responsibilities,
3. Spatial units, mostly parcels and the legal space of buildings and utility, including the sub package “*surveying and spatial representation (topology and geometry) and spatial sources*”.

LADM covers land registration and cadastre in a broad sense by describing spatial and administrative components, source documentation and the ability to link with external registrations, such as physical object registration (buildings, utilities, etc.). The model also includes agreements on data about administrative and spatial units, land rights in a broad sense and source documents (e.g. titles, deeds or survey documentation). The rights may include real and personal rights as well as customary and informal rights and the restrictions and responsibilities can be similarly represented to document the relationships between people and land.

Recent works suggest that the utilization of LADM international standard for cadastral domain is significant as mentioned by several researchers: Lemmen (2012), van Oosterom et al. (2011), Pouliot (2011), etc. Many countries such as Poland, Republic of Korea, Malaysia, Indonesia, Croatia, etc. have proposed country profiles based on LADM. A key challenge is to include the third dimension in such models and develop prototypes that can support it.

With the maturation and accessibility of 3D technologies, there is little doubt that we are now moving to embrace a 3D digital environment for land administration business practices, including an object-oriented approach to managing information about land and property RRRs (Rajabifard 2014).

There are many formats for the storage and visualization of the spatial data, some focusing on the description of geometry (e.g. InfraGML, X3D Scarponcini 2013) and some others also including the representation of semantic and thematic properties and aggregations (Kolbe 2009) (e.g. CityGML). Exploring opportunities for leveraging 3D technologies, such as the use of Building Information Models

(BIM), Industry Foundation Class format (IFC) (IAI 2008) and CityGML facilitates greater collaboration around the function of 3D Cadastres.

In this direction, apart from international standards, national standards have also been developed. The most representative standard in this category is INTERLIS (Germann et al. 2015). INTERLIS is a standard for the modeling and integration of geodata allowing co-operation between information systems, especially geographic information systems, see Fig. 1. It is a Conceptual Schema Language (CSL) and a neutral transfer format and also defines a system neutral XML based data exchange format. Using its tools (e.g. compiler, checker) it is possible to quality check INTERLIS data and is compatible with the most relevant international standards (UML, XML, XML Schema, GML).

INTERLIS is an Object Relational modeling language, a very precise, standardized language on the conceptual level to describe data models. Both humans and computers can read it and it has built in data types for GIS; e.g. the geometry types. Transfer formats are derived from data models by transfer rules and there is strict separation of transfer and modeling aspects (model driven approach).

The main benefits from INTERLIS are that it:

- Supports freedom of methods through system neutral approach;
- CSL is easily understandable by IT and domain experts;
- Data can be directly processed and checked by computers;
- Allows for automated data quality control (checker, check service);
- Allows the automation of many cadaster related processes;

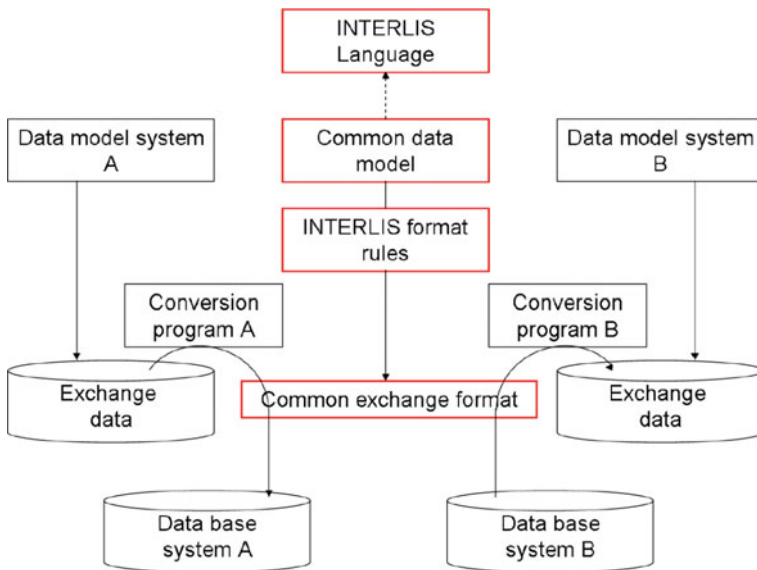


Fig. 1 INTERLIS (COGIS 2006)

- Has built-in geometric data types (point, polyline, polygon), making it especially suitable for models in the geo-information domain.
- Provides reference manuals, translated to many languages.

Furthermore, it has relation with other standards. INTERLIS uses UML as graphic representation of its data models (.ili files). Moreover, GML is supported by INTERLIS through additional transfer rules (eCH—0118 2011). In particular, INTERLIS allows INTERLIS (.ili) and XMI files to be imported and SVG, JPEG, INTERLIS, WMS and XML Schema files to be exported from INTERLIS tool chain (Germann 2015).

The last years, there is a growing interest in the implementation of conceptual models into technical. By describing a conceptual model with a tool based on CSL and then derive the technical model results in (directly) implementable model descriptions based on the conceptual models. Therefore, the description of LADM with INTERLIS enables the exchange of LADM data between IT systems, which can be used to initialize databases or transfer LADM data via XML.

2.2 A 3D Multipurpose Land Administration System for Greece

The implementation of the tools mentioned above is done using a 3D multipurpose LAS proposed for Greece based on LADM, in order to derive the technical model from the conceptual. The party and administrative package of this model together with an overview of the spatial part, described in UML is presented in Appendix II.

Greece presents several deficiencies in adapting with international requirements and best practices. The Hellenic Cadastre (HC) is still an ongoing project relying on its data model, not conforming to international standardization criteria. This model presents heterogeneity concerning geospatial data and cannot act as reference base for National Spatial Data Infrastructure.

In this paper, the proposed model is considered as an effort for overcoming previous shortcomings, introducing a model based on international standards, covering a broader perspective than the one of the National Cadastre and Mapping Agency SA (NCMA SA), including objects and interests that are not registered to the existing model. More precisely, networks (both the legal and their physical part), planning zones, marine parcels and 3D parcels are not included nor provided by the existing model.

Greek planning law comprises a wide range of instruments which extent from strategic plans at national level to regulatory town plans and zones at local level. Therefore, planning zones are very important as they define activities, policies, land uses and restrictions for the entire country and it was considered necessary to be included in this model. On the other hand, marine parcels are not recorded in the

HC today, despite the fact that sea covers a big part of the Greek territory. Because the model is future-proof it was considered necessary to include marine parcels as a separate level.

Conforming to a standard like LADM would reduce time for new data production, the number of people and resources, while land transactions would be safer and of higher-quality and data update would be assured. Technically, the structure of LADM supports 2D and 3D cadastral registration of legal items and the link with their physical part. The following paragraphs analyze the need for a multipurpose land information system in Greece, organizing in groups the wide range of different spatial units and drawing the attention on the third dimension.

The land question and RRRs in Greece present great diversity and specificities, as it largely depends on localized historical, economic, social, political and cultural factors. At this model, an attempt is made to cover all Greek land administration related information, which is maintained by different organizations today. Despite its small surface area, Greece is endowed with a particular rich and diversified natural environment; with unique geomorphology and intense contrasts. All those diversities conclude to complex scenery, with different characteristics, which should be registered and managed into a coherent and unified system. Moreover, the rights, restrictions and responsibilities attached to that scenery, the activities developed, the responsible parties, as well as the difficulty to represent the outline of it need to be described explicitly.

Therefore, the 3D MLAS should contain information about administrative records, tenure, value and sale and purchases records, base maps, cadastral and survey boundaries, categories of land use, streets addresses, census utilities, all rights, all parties etc.

A new data model could facilitate the provision of data to internal and external users in a more flexible format for the community's needs. That means improving the structure of property rights, restrictions and responsibilities, as well as all relative stakeholders, in a direction of harmonizing with international land administration systems and standardization processes in this field. In order to improve insight in the spatial component of rights established on 2D parcels, the current system needs to be extended and introduce the registration of 3D situations. In order to make the model comprehensive and future proof, all spatial units can be supported in such a model, even if not supported today by the NCMA S.A. or another organization.

But why registering the third dimension?

The implementation of a 3D cadastral model in Greece is specifically required for the registration of SPROs (but also for other types of spatial units with RRRs attached as mines, marine parcels, utility networks and even "normal" parcels, 3D will be more important in the future, see Appendix II). The SPROs are currently the only registrations with 3D tags in the existing model. They include very common cases in several Greek islands where land parcels and buildings are partially or totally overlapping to each other. A 3D challenge demonstrates in Fig. 2, where anogia and katogia (typical SPROs) are represented in Santorini Island with steep slope, where most houses are dug in the volcanic soil led to overlapping properties.



Fig. 2 Special property right objects in Santorini Island (Dimopoulou et al. 2006)

Recent work (Tsiliakou and Dimopoulou 2011) (Fig. 3) has shown that those characteristic cases need the description of the third dimension and the visualization of the physical objects on 3D environment is of a great importance.

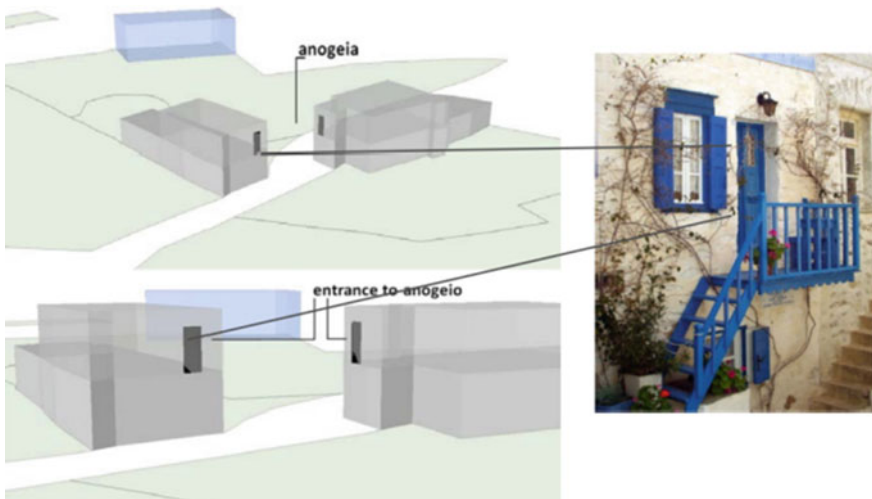


Fig. 3 Anogia in Syros Island (Tsiliakou and Dimopoulou 2011)

3 From Conceptual to Technical Model

3.1 Prototype Development

Towards the implementation of the prototype, the creation of the conceptual model is followed by the transformation of the logical data model into a physical database. Through this conversion, many implementation and design decisions have been taken, from the conceptual understanding of possible errors or duplicates at the proposed model, to the selection of the most appropriate DBMS.

The paper also introduces the implementation of LADM with INTERLIS following the methodological steps discussed in (Germann et al. 2015), drawing particular attention to the formulation of constraints and the 3D, as until now this wasn't included. The main purpose is to investigate the process of deriving a database schema or an exchange format from the conceptual model (UML diagrams), using technological tools, which support the initial attributes, constraints and aggregations.

Figure 4 presents the methodological concept followed during the development of the prototype. As a first step the 3D multipurpose LAS was described in UML diagrams using Enterprise Architect software. The UML model was exported in XMI (XML metadata format) in order to be imported in INTERLIS. However, due to a compatibility problem, this conversion was not successful.

Therefore, the conceptual model was (manually) described in INTERLIS language. The INTERLIS model was imported into INTERLIS compiler for quality check. Real data concerning both the legal and physical aspects of the object has

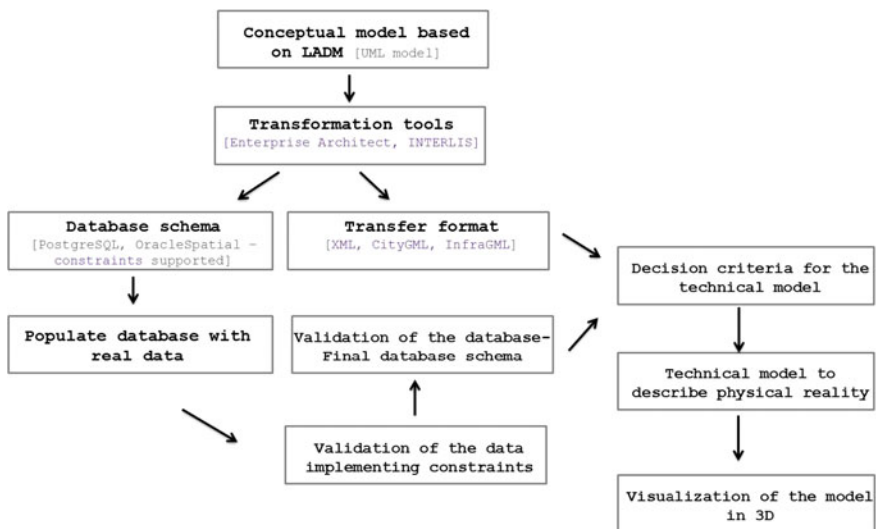


Fig. 4 Prototype development: methodological steps

gathered from the responsible authorities and populate the database. The data refers to the majority of the levels created in the spatial part of the proposed model mostly concerning condominiums, 2D rural parcels, archeological spaces, mines, polygons created from planning zones and data for utility networks.

The next step is to visualize the result with physical objects. There is a plethora of candidates to describe the physical reality of the data; CityGML, IFC, LandXML, InfraGML, etc. For that reason four decision criteria were set in order to choose the appropriate one, as presented below:

- Is it used? Do we have real data available?
- Will it be maintained in the future? Is there a Consortium/Institution responsible for maintaining this format?
- The data we want can be described within that model?
- Is there a conversion to XML available or possible? As XML-based format has become the default for many tools, it has also been employed as the base language for communication protocols, which means that it is compatible with most of the existing formats.

Based on those criteria, CityGML and IFC are considered to be the two most appropriate formats to describe the physical reality of this model. Both formats fulfill the requirements set by the criteria described above and also support semantics. However, the two standards have different concepts, i.e. they represented the building structure from two distinct views: the constructor (IFC) and the user (CityGML) view. Which model is more appropriate to be used for the physical description of this model is further to be studied in detail.

In recent years, various 3D visualization solutions have been developed, some of them web-based. Shojaei et al. (2014) at their study summarize the common 3D web-based solutions and compare them and their conclusions are helpful towards the selection of the best candidate format.

Another aspect that is of great importance and requires formal description is the constraints of the model. Constraints are often initially described in natural language; however practice has shown that this results in ambiguities. For the prototype, constraints have been defined from the first step of a prototype in order to reduce complexity.

According to Paasch et al. (2013) constraints can be categorized as: constraints derived from the properties of objects (thematic, temporal, spatial and mixed constraints), constraints derived from spatial relationships between objects (thematic, spatial, temporal and mixed) as well as constraints grouped according to their dimensional aspects (constraints that concern the 2D ground plane and/or the 3D objects).

In this paper, the constraints are defined on the UML model using the Object Constraint Language (OCL), which enables to express the constraints at a conceptual level in a formal way. EA supports OCL syntax on the model and also supports the ability to validate the OCL statement against the model itself. This means that it enables the verification beyond just the syntax that the OCL statement

is expressed correctly in terms of actual model elements, and that the kind of validation syntax that it uses corresponds to the actual data types defined for these elements: numbers, strings, collections, etc.

Database implementation offers better management of constraints. The types of constraints that can be found in a database are unique constraints, referential, primary key and check constraints. Those are useful but not powerful enough to support all the spatial constraints. For that reason, the use of triggers is proposed as an alternative solution for general constraint implementation in DBMS.

For the proposed model there are many types of constraints that can be found, i.e. primary key must be unique, end date of unfinished construction must be equal to start date of a building, if there is a co-ownership in a parcel, the sum of the RRR share should always be 1, boundary of 2D parcel must be closed, if we refer to the marine parcel the surface relation type should only take the values below or mixed etc.

3.2 *The Prototype—Case Study in Greece*

As mentioned before, the methodological steps are implemented using as basis a proposed model for Greece based on LADM. Different attributes are added to the GR_SpatialUnit class such as *HasTopoMap* (already a class of HC model, showing whether there is a topographic map attached together with the ownership declaration or not), *InsideMap* (whether the property is inside the city plan or not), etc.

According to ISO 19152 (2012), LA_Level and therefore, GR_Level is a collection of spatial units with a geometric or thematic coherence. This concept is important for organizing the spatial units. In this way, in relation to the principle of “*legal independence*” (Kaufmann and Steudler 1998) different groups of coherent spatial units can be created. For the proposed model, this structure allows for the flexible introduction of spatial data from different sources and accuracies, including utility networks, buildings and other 3D spatial units, such as mining claims, or construction works, etc.

The various types of spatial units are organized in levels using the class GR_Level. For this class there is an attribute type that describes level type of the spatial unit, which will include: archeological space, land parcels, marine parcels, panning zones, mines and SRPOs. The code list for these attributes can refer to GR_LevelContentType.

For Greece, the following levels are proposed: level 1 for archeological, level 2 for 2D parcel, level 3 for 3D parcel, level 4 for mines, level 5 for SPROs, level 6 for planning zones and level 7 for marine parcel. In the involved classes a constraint has been added to make this more explicit. For instance, GR_Mine has a constraint GR_Level.name = “level 4”.

Taking for example the level referring to SRPOs. Special Real Property Objects considered an individual entity also in the existing model of the HC. This is due to the fact that they are properties built above or below other properties, usually found

in Greek islands. Customary law applies in most Aegean islands creating complex RRRs, mixed up in multiple layers below or above the surface. Legal relations on those RRRs are presented through characteristic cases and examples of the complex 3D reality.

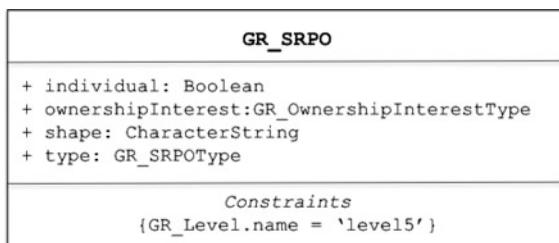
As SRPOs are already separate entities at the HC data model it is considered that a new level for them should be created (Fig. 5). RRRs are connected and affect each other, as well as connect parties and property units. Separate ground floor (e.g. katoi) and upper floor (e.g. anoi) residences have been traditionally under a system of horizontal property, evidently not complying with the Roman accession rule. The owner of the ground floor also owned the land parcel, while the owner of the upper floor owned the roof (and the air), having no land share. Under this special system of co-ownership, each floor's rights, even without land share, are separate, transferable and registrable. It is clear that there is no way to explicitly describe those relationships in two dimensions. The integration of these legally defined spaces to a 3D cadastral system should leave no doubt about their 3D registration.

What makes the development of this model unique is that it supports a wide range of spatial units, each of them having different requirements. The model also includes the content of various code lists, which are an important aspect of standardization and unique for each country. Code lists are used to describe more open and flexible enumeration values and are useful for expressing a long and potentially extensible, list of potential values.

Because of the special characteristics and complex structure of the proposed model, it was a challenge to use it as case study to test the prototype methodology. At the beginning, the model was implemented in Enterprise Architect software, described by UML. EA apart from the modeling part also supports generation and reverse engineering of source code; the automated conversion from the UML class diagrams to the technical model in Data Definition Language (DDL) and then creation of the SQL statements and generation of the database schema.

Although this process is automated, some limitations appear. The most important is that most of the constraints set at the UML diagram are lost through this conversion and because the number of inter-relationships between the classes of the proposed model is many, it is difficult to directly translate them in SQL. Moreover, the code lists are also a key challenge, as the enumeration list derived from the automated conversion cannot be extended.

Fig. 5 GR_SRPO class of the proposed model; constraint added (Kalogianni 2015)



Therefore, the result needs manual interpretation. At that point the Swiss standard INTERLIS was considered a challenging solution to get computer processable model description and transfer LADM classes via XML (Germann et al. 2015).

INTERLIS was used as intermediate to derive the technical model from the conceptual. As a first step, the XMI (XML metadata format) was exported from EA describing the proposed model and imported in INTERLIS UMLeditor tool. It was expected that the model would be described in INTERLIS language and also a representation in diagrams would be available.

However, the XMI versions of EA and INTERLIS are not (yet) compatible, as INTERLIS uses Rational Rose XMI and EA exports to XMI 1.1. and XMI 2.1. Figure 6 depicts the technical problem presented during the development of the prototype.

Therefore, 3D MLAS model was translated manually into INTERLIS language and imported into INTERLIS compiler to quality check the INTERLIS model. Then the database schema from INTERLIS is derived. Below, is presented the GR_BAUnit class described in INTERLIS language.

More INTERLIS described classes are presented in the Appendix I.

```
CLASS GR_BAUnit EXTENDS LADM.Administrative.LA_BAUnit =
name (EXTENDED): CharacterString;
type (EXTENDED): MANDATORY LADM.Administrative.LA_BAUnitType;
KAEK : CharacterString;
horizontalPropertyID : int;
verticalPropertyID : int;
extArchiveID : Oid;
END GR_BAUnit;
```

The next step was to import sample data into the model and test its efficiency with real data. It is interesting to see whether some constraints are being violated, how much time a spatial query needs to be interpreted, etc.

The most important steps during the INTERLIS implementation were the formulation of code lists (avoiding the creation of enumeration types, which are fixed values and not extensible), the expression of constraints (as presents at the code fragments below), the aggregations between the classes and the support of 3D geometry for the representation of spatial units. Additionally, the description of multiple ISO standards that LADM uses as basis, such as ISO19107, ISO19111, ISO19156, and so on, were already described in INTERLIS and ready to use, from the Swiss Land Management.

In particular, there is a stack of INTERLIS models described from the Swiss Land Management: the ISO191xx base and also the datatypes that are used by LADM are specified (*LADM_Base*); then, the 3D MLAS model (*LADM_GR*) was described and each layer uses the model layer illustrated at the following figure, starting from the base to the top. This step facilitates the progress as most of the data types needed for the description were already defined (Fig. 7).

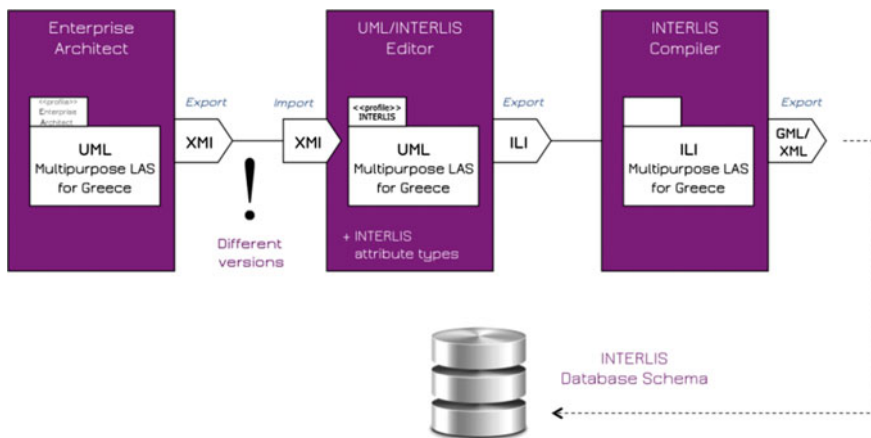
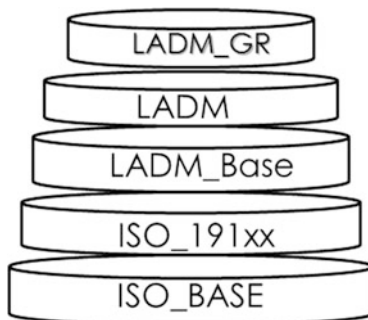


Fig. 6 Technical steps and problems through the development of the prototype

Fig. 7 Stack of INTERLIS models used to describe LADM country profile for Greece



A sample INTERLIS code fragment from each of the model layers is displayed below. In particular, a part of ISO19107 and a part of LADM_Base in INTERLIS are presenting below. Additionally, an example of an external class (*ExtLevelOfAdministrativeDivision*) of the proposed model described both in UML and INTERLIS is presented.

The relatively recent integration of Greek municipalities under the new (Kalikratis Plan) organizational schema does not remove the need for a separate management of the (former) sub-municipalities and thereby a more flexible land administration is provided. Because of the multiple levels of administrative division that have been created, the list of values is dynamic and there is great likelihood that additional values will be added, code lists are selected instead of enumeration types.

Please note the *IMPORTS* statement to include a model layer below.

```

!!-----
!!
!! ISO 19107 Geographic Information - Spatial Schema
....

DOMAIN

....

STRUCTURE GM_Point =
  geometry: Coord3D;
END GM_Point;

STRUCTURE GM_MultiCurve =
  geometry: LIST (1..*) OF Curve3DListValue;
END GM_MultiCurve;

STRUCTURE GM_MultiSurface =
  geometry: LIST (1..*) OF Surface3DListValue;
END GM_MultiSurface;

END ISO19107.

```

External::ExtLevelOfAdministrativeDivision + AFMrepresentative: Int + levelOfAdministrativeDivisionID: Oid + name: CharacterString + type: GR_LevelOfAdministrativeDivisionType
--

```

TOPIC External [Abstract] =
STRUCTURE ExtLevelOfAdministrativeDivision
  AFMrepresentative : Int;
  levelOfAdministrativeDivisionID : Oid;
  name : CharacterString;
  type : GR_LevelOfAdministrativeDivisionType;
END ExtLevelOfAdministrativeDivision;

```

«codeList» GR ExtLevelOfAdministrativeDivisionType + AD01 - Municipality + AD02 - Region + AD03 - Decentralized Administration + AD04 - Nation + AD05 - European level + AD06 - International level
--

```

GR_LevelOfAdministrativeDivisionType =
[Municipality, Region, Decentralized
Administration, Nations European level,
International level]

```

```

!!-----
!!
!! ISO 19152 LADM modelled with INTERLIS 2
...

IMPORTS UNQUALIFIED ISO_Base;
IMPORTS UNQUALIFIED ISO19115;

DOMAIN

STRUCTURE VersionedObject =
  beginLifespanVersion: MANDATORY DateTime;
  endLifespanVersion: DateTime;
  quality: LIST {0..*} OF DQ_Element;
  source: LIST {0..*} OF CI_ResponsibleParty;
!! MANDATORY CONSTRAINT
!! (endLifespanVersion(n-1) = startLifespanVersion(n));
END VersionedObject;

STRUCTURE Fraction =
  denominator: MANDATORY Integer;
  numerator: MANDATORY Integer;
!! Functions()
!! equals(Fraction): Boolean;
!! real(): Real;
!! invariant
!! denominator > 0;
!! numerator > 0;
!! numerator <= denominator;
END Fraction;

...

END LADM_Base.

```



Fig. 8 UML and INTERLIS description of the proposed model

Figure 8 depicts two classes of the Greek 3D MLAS and a code list described both in UML and INTERLIS. To make a code list class semantically more meaningful, an attribute indicating the parent of the code list, in case that it exists, was added, other-wise it was considered as top-level code list. This means that if a code list has been introduced for the proposed model and has not been previously defined at the LADM description in INTERLIS, it is considered as a top level code list. Also, the code list is versioned, which means that it could be used to update the description of a specific code list value. On the other hand, if the code list has already been defined, the parent to which it is associated it is also defined.

An example with *LA_PartyRoleType* is mentioned below.

```

CLASS LA_PartyRoleType EXTENDS VersionedObject =
  partyRoleTypeCode_ID: MANDATORY Oid;
  parentCode_ID: Oid referring to
  LA_PartyRoleType.partyRoleTypeCode_ID;
  description: CharacterString;
  !! Possible code list values: surveyor, notary, other
END LA_PartyRoleType;
  
```

Of course this raises questions; for instance, can anyone who models create a top-level code-list? And if yes, then how communication between the different modelers can be achieved? However, such questions are out of the scope of this paper.

INTERLIS closed enumeration types can be extended, for instance, in most of the code lists described in UML there is the value “other” that can be further extended. However, this results in a new enumeration type; and the classes that use this type also need to be remodeled.

3.3 Problems and Strengths of the Prototype

The prototype is still on an ongoing process. So far, the conversion from the conceptual UML model to a physical database has been achieved. More is yet to come, as the database will be populated with real data that have been gathered in order to test the efficiency of the database, implement the necessary changes and finalize the model. The criteria for choosing the best candidate for the description of the physical part of the objects need further exploitation, as the technological developments are rapid.

Until now, some technical modeling difficulties have been addressed; e.g. the compatibility problem between UML and INTERLIS while some modeling obstacles need to be further faced.

4 Conclusions and Discussion

This paper describes the prototype development based on LADM, INTERLIS and a model proposed for Greece. Many technical design and implementation decisions have been elaborated on during the conversion of the conceptual model to database schema.

The conceptual model is a proposal covering a broader perspective than the one of the HC, including 2D and 3D objects and interests that are not registered to the HC. The various types of spatial units are organized in levels using the class GR_Level. The levels of spatial units include: archeological space, 2D and 3D land parcels, marine parcels, panning zones, mines and SRPOs.

By applying INTERLIS to the proposed model, we get directly implementable data models, which speed up the implementation of LADM model. INTERLIS was chosen among other tools as it allows formal description of constraints. For the proposed model there are many types of constraints that can be found, i.e. end date of unfinished construction must be equal to start date of a building, if there is a co-ownership in a parcel the sum of the RRR share should always be 1, (Sect. 3.2.). Also, automated quality control of the data is offered using INTERLIS compiler tool.

One of the most difficult steps towards the prototype was the formulation of the code lists, as they should be extended and semantically rich. They were given a unique identifier (e.g. PR07), which will facilitate the exchange of information. Additionally, an extra attribute was added to support hierarchy in code lists; it needs further identification as a next step.

Finally, INTERLIS supports 3D point/line and polygons. Although there are not special 3D-volumetric types at the moment (Germann et al. 2015), it is still considered appropriate for the development of this prototype as it provides the basic 3D geometry types which can be used as basis for the definition of more complex types. Therefore, the next step is to further improve the 3D volume types supported by introducing polyhedron and/or solid types.

Further research aims to investigate the best candidate for the description of the physical objects and the implementation in 3D environment in order to investigate similarities and differences, as well as patterns between the physical and the legal reality.

Appendix I—INTERLIS Implementation

```
!!-----
!!
!! LADM_GR modeled with INTERLIS 2

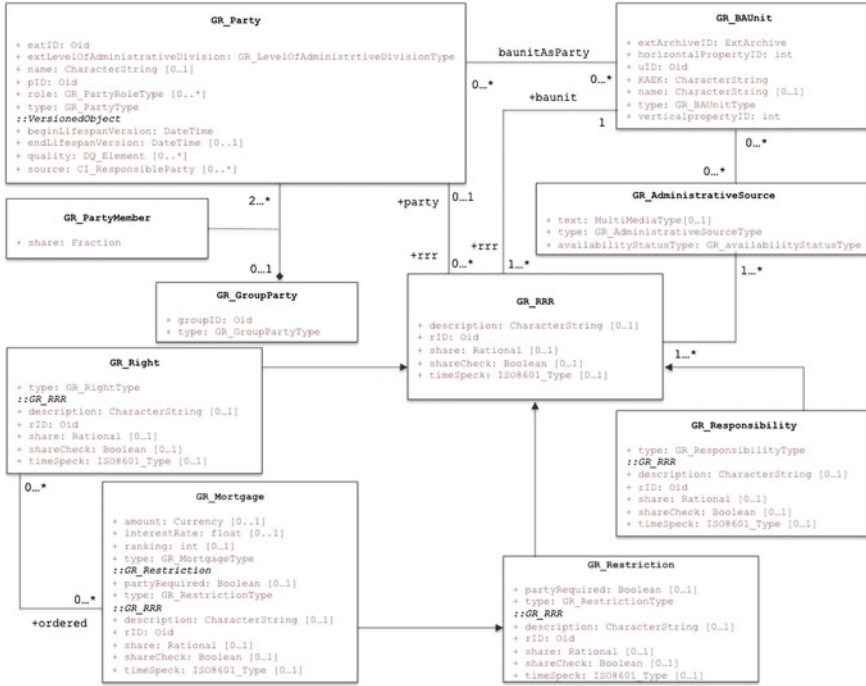
....
TOPIC Party(ABSTRACT) =

DEPENDS ON Spatial_Unit, Administrative, Surveying and
Representation;

DOMAIN

GR_PartyType = (group,natural_person,nonnatural_person,
basic_administrative_unit,Greek_public_state,
foreign_state, European_Union,unknown);

GR_PartyRoleType = (lawyer,bank,notary,citizen,
institution, tax_office,insurance_organization,
church,surveyor,metropolis,parish,court,
courtof_appeal,high_court,state_council,
legislative_authority,expropriation_committee,
ministry,local_authority,urban_planning_authority,
general_secretary_of_the_region,to_be_filled);
```

References

Aien, A., Kalantari, M., Rajabifard, A., Williamson, I., & Wallace, J. (2013). Towards integration of 3D legal and physical objects in cadastral data models. *Land Use Policy*, 35, 140–154.

Aien, A., & Rajabifard, A. (2014). Integrating legal and physical dimensions of urban environments. *ISPRS International Journal of Geo-information*. ISSN 2220-9964.

Budisusanto, Y., Aditya, T., & Muryamto, R. (2013). LADM implementation prototype for 3D cadastre information system of multi-level apartment in Indonesia. In *5th land administration domain model workshop, Kuala Lumpur, Malaysia*.

Choon, T., & Seng, L. (2013). Towards Malaysian multipurpose 3D cadastre based on land administration domain model (LADM)—an empirical study. In *Proceedings of the LADM workshop, Kuala Lumpur, Malaysia*.

COGIS. (2006). INTERLIS 2.3 reference manual. Coordination, geo-information and services (COGIS), a division of the Swiss Federal Office of Topography.

Dimopoloulou, E., Gavanas, I. & Zentelis, P. (2006). 3D registration in the Hellenic Cadastre. In TS 14-3D and 4D cadastres, shaping the change. XXII FIG Congress. Munich, Germany.

eCH—0118. (2011). GML Kodierungsregeln für INTERLIS.

ePlan. (2010). ePlan model version 1.0 ePlan (p. 61): Intergovernmental Committee on Surveying and Mapping (ICSM).

Germann, M. (2015). Standardized data modeling and its benefits. In *WCS-CE—the world cadastre summit, congress and exhibition. Istanbul, Turkey*.

- Germann, M., Kaufmann, J., Steudler, D., Lemmen, C., van Oosterom, & de Zeeuw. (2015). The LADM based on INTERLIS. In FIG working week 2015, Sofia, Bulgaria.
- IAI. (2008). International alliance for interpretability. Retrieved from <http://www.iai-international.org>.
- ISO. (2012). Geographic information—land administration domain model (LADM). ISO 19152:2012(E). International Organization for Standardization (ISO), Geneva, Switzerland.
- Kalogianni, E. (2015). Design of a 3D multipurpose land administrative system for Greece in the context of LADM. Athens, National Technical University of Athens (NTUA, Master Thesis).
- Kaufmann, J., & Steudler, D. (1998). Cadastre 2014—A vision for a future cadastral system. F.I. G., Commission 7, Working Group 7.1. In *XXI international congress, Brighton, England*.
- Kolbe, T. H. (2009). Representing and exchanging 3D City Models with CityGML. In *Proceedings of the 3rd international workshop on 3D geo-information, Seoul, Korea*.
- Lemmen, C. H. J. (2012). A domain model for land administration. Delft, Technical University Delft (TUD), University of Twente Faculty of Geo-Information and Earth Observation (ITC), 2012. ITC Dissertation 210, ISBN: 978-90-77029-31-2.
- Paasch, J., van Oosterom, P. J. M., Lemmen, C., & Paulsson, J. (2013). Specialization of the LADM—modeling of non-formal RRR. In *5th international workshop on 3D cadastres, Kuala Lumpur, Malaysia*.
- Pouliot, J., Marc, V., & Abbas, B. (2011). Spatial representation of condominium/co-ownership: Comparison of Quebec and French cadastral system, based on LADM specifications. In *2nd international workshop on 3D cadastres*.
- Rajabifard, A. (2014). 3D cadastres and beyond. In *4th international workshop on 3D cadastres, Dubai, United Emirates*.
- Scarponcini, P. (2013). InfraGML proposal. OGC Land and Infrastructure DWG/SWG.
- Shojaei, D., Rajabifard, A., Kalantari, M., Bishop, I., & Aien, A. (2014). Design and development of a web-based 3D cadastral visualization prototype. *International Journal of Digital Earth*.
- Tsiliakou, E., & Dimopoulou, E. (2011). Adjusting the 2D hellenic cadastre to the 3D complex world—possibilities and constraints. In *Proceedings of the 3rd international workshop on 3D cadastres, Delft, The Netherlands*.
- van Oosterom, P., Fendel, E., Stoter, J., & Streilein, A. (2011). In *Proceedings 2nd international workshop on 3D cadastres, Delft, The Netherlands*.
- van Oosterom, P. J. M., Guo, R., Li, L., Ying, S., & Angsüsser, S. (2012). In *Proceedings 3rd international workshop on 3D cadastres: developments and practices, Shenzhen, China*.
- van Oosterom, P. J. M. (2013). Research and development in 3D Cadastres. *Computers Environment and Urban Systems*, 40.

Web-Based Tool for the Sustainable Refurbishment in Historic Districts Based on 3D City Model

Iñaki Prieto, Jose Luis Izkara and Rubén Béjar

Abstract The objective of this article is to describe a web-based tool that aims the management and conservation of urban heritage by means of the usage of a decision making system. The decision making system will be feed by a 3D city model of a historic district, enabling the storage and presentation of data at city and building scales. 3D city model contains both geometric and semantic data into a single data model. Access to stored city objects in the 3D city model is made through standard web services defined by the Open Geospatial Consortium (OGC). The presentation of the information is performed through an user friendly interface based on the interactive representation of the 3D city model using Web3D technologies (HTML5 and WebGL).

1 Introduction

One of the main problems in historic cities, in addition to their physical deterioration, is the loss of urban habitability and quality. One of the key aims of any historic city conservation and management strategy should be done improving the quality of life of its inhabitants.

The 40 % of energy consumption in Europe is produced by the building sector, which is also responsible for 36 % CO₂ emissions (OECD/IEA 2013). This situation is closely related to the emissions caused by existing buildings, as 50 million buildings across Europe are 50 years old or more, with the majority of these

I. Prieto (✉) · J.L. Izkara
Construction Unit, Tecnalia, Derio, Spain
e-mail: inaki.prieto@tecnalia.com

J.L. Izkara
e-mail: joseluis.izkara@tecnalia.com

R. Béjar
Computer Science and Systems Engineering Department,
Universidad de Zaragoza, Zaragoza, Spain
e-mail: rbejar@unizar.es

buildings found in European cities (Department United Nations. Population of Economic 2010). Historic urban districts also represent an element that is of essential importance to European culture and heritage, meaning that improving energy efficiency will help protect this legacy for future generations.

The creation of innovative methods and management tools that make up a systemic, holistic and participatory approach with regard to the urban scale can become a key element in this process. Analysis and processing of the information through the use of new technologies is potentially of great help in prioritizing and decision-making within management and refurbishment processes (Delgado et al. 2012). This project focus on the strategies and tools required to confront the conservation challenges that face from a multi-scale perspective.

Nevertheless, the majority of current development regarding energy efficiency focuses on new building work, ignoring the specific problems of historic buildings. New solutions also tend to focus on individual buildings without taking into account the urban dimension, where the connections between buildings and other urban infrastructures create an altogether different context (Bakar et al. 2015).

Historic cities, like any other urban ecosystem, create a great deal of diverse information: different scales, different uses, different applications and formats. In the near future, this information will increase exponentially, making its modelling, storing handling a crucial and strategic aspect in management and decision-making (Ferretti 2014).

2 3D City Models for the Management of Historic Cities

The management and conservation of urban heritage require an approach that considers each of the historic buildings and other city elements as forming part of an environment that should be conserved, brought up-to-date and showcased. This focus requires the integration of Geographic Information Systems (GIS) and Building Information Models (BIM), while at the same time bearing in mind the particular nature of urban heritage (Döllner and Hagedorn 2007).

The solution, based on 3D digital models, has grown in importance over recent years as it offers durable support which is easily brought up-to-date, allowing information storage and visualization on an urban scale. The visualization of 3D and semantic information of historic cities and town centers is a more natural way to represent the spatial properties of urban elements, of great use to those taking part in the management, conservation, use and enjoyment of such towns and cities. The general aim is to combine geometric information and the characteristics relating to buildings, urban environments and archaeological sites into a single integrated data model. Egusquiza et al. (2014) identifies the CityGML as the data model that allows 3D geo-referenced and semantic information associated with geometry to be stored in a single data model (Gröger et al. 2012).

One way to tap into the real potential of the model (its use with international standards, its interoperability with other data models and other analysis, management

and decision-making tools etc.) is through the development of a service ecosystem to make urban planning and management easier, by creating new cloud-based applications (Chen 2011). This ecosystem takes into account and employs various standards to help define and to take full advantage of a common data model. These services include sustainable refurbishment which assists in administration and maintenance of urban intervention, aimed at improving sustainability and energy saving in towns and cities. Services can also be designed to ensure the efficient management of energy resources, for the administration and optimization of urban mobility, tourist, cultural and service information which facilitate e-government and the participation of the general public, improving interaction and communication between government bodies and local people, among others (Gröger and Plümer 2012).

However, the semantization of 3D city model is still a challenge. After generating and adding the necessary semantic information into the data model, a continuous maintenance is needed in order to have an up to date data model. As Navratil said 'the system should be kept up-to-date and not be created new for each decision' (Navratil et al. 2013). In the maintenance geometry and also the semantic data needs to be updated. Without this process it will be an useless model in the near future. An investigation in the obsolescence factors and obsolescence-prevention strategies has been carried out and the implemented strategies in order to overcome these were described in Morton et al. (2012). As explained by Chen, maintenance related task usually consists in having accumulated historical data from different sources and updating the database with this data (Chen et al. 2013). Another approach consists in maintaining the data model automatically using reconstruction by components (Steinhage et al. 2013). In this work they used information derived from different sensors and map data from a geographic information system together with semantic and component-based approach to model and reconstruct complex buildings. They use the derived geometry and semantic in a spatial information system that allows making queries for the maintenance and update of the data model.

There are different alternatives of the usage of 3D city models in end user applications using Web 3D technologies. In urban planning it enables the city administrators to present urban plans to the public to engage with stakeholders and facilitates also the public participation. Dambruch and Kramer has used HTML5 and X3DOM along with Liferay to present a public participation in urban planning with 3D web tool (Dambruch and Krämer 2014). Another alternative is the usage of 3D city models for analyse the information within the data model, for later visualization by mean of different web services such as W3DS, WFS and WVS (Soave et al. 2013). An approach of configurable instances of 3D city models allowing the simplification of 3D models in different application has been presented by Klein et al. (2014).

The web tool developed in this project and described in this short paper is focused on the identification and prioritization of urban interventions for sustainable refurbishment of historic district.

3 Overall System Architecture

A multi layer architecture has been designed to present 3D and semantic information of 3D city models on multiple platforms using open standards. The architecture must meet the following requirements: easy reuse of the 3D city model, ensure the interoperability with other tools or services to integrate new functionality and abstracting the details/complexity of the model to end users of the system.

The architecture is split in three layers (see Fig. 1):

- Infrastructure layer. The data layer contains the 3D city model stored in PostgreSQL with PostGIS extension. In addition low level services using web services defined by OGC are deployed in this layer, such as WFS or W3DS. Also contains the intermediate API, which allows an easy access to the data in order to develop applications based on the 3D city model in an easier and faster way.
- Application layer. The Application layer contains the services which implement the functionality of the web tool. These services are mainly building state identification, the generation of vulnerability maps and the generation of prioritization maps.

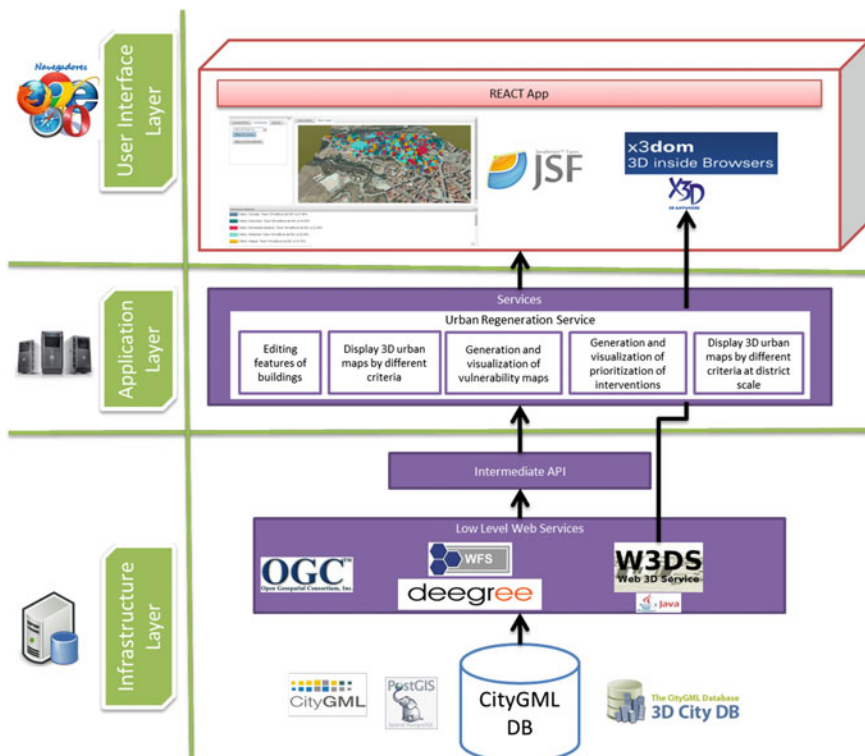


Fig. 1 Application architecture

- User Interface layer. Through the presentation layer the user can access to the application layer services as a result X3D files are visualized by means of X3DOM framework.

PostgreSQL relational database has been chosen augmented by PostGIS extensions which add support for geographic referenced 3D objects. Data Model storage is based on the 3D City Database. 3DCityDB is used to represent and store the same information modelled by CityGML. This database implementation is completed with an Importer/Exporter tool in order to process, store and retrieve efficiently and quickly CityGML dataset.

Since one of the objectives is to improve interoperability, instead of creating new web services interfaces, standard Web service interfaces defined by the OGC such as the W3DS (Web 3D Service) and the WFS (Web Feature Service) have been used. W3DS allows retrieving geometric information and WFS the semantics of the data model.

Deegree3 have been selected to deploy the WFS because it can be easily integrated with PostGIS. This service allows getting and modifying the semantic information of the CityGML model (Vretanos 2005). In that way, it is possible to separate the access to the geometry (W3DS) and semantics (WFS). Communication with the service (requests and responses) is done via XML files according to the XML schema specifications of the OGC.

The W3DS (Web 3D Service) (Schilling and Kolbe 2010) allows to retrieve the geometry and appearance of three-dimensional models for visualization. It is still in draft phase as a previous step to final approval by the OGC. There is a reference implementation developed in OSM3D-Germany project (Goetz and Zipf 2012) which implements almost all features of the standard specification. However, the software has not been open sourced, so we have developed our own implementation.

The user interface is developed using following technologies:

- JavaServer Faces to ease the creation and manipulation of the interface objects
- Primefaces to enrich the JSF components and to improve the aesthetics of the web interface
- X3DOM is used as the 3D engine, allowing to visualize the 3D city model by means of X3D files.

4 Application for Sustainable Refurbishment of Historic Districts

Segovia has been selected as the case study for the validation of the implementation of the data model. Segovia is located near Valladolid and the Spanish capital, Madrid, with around 55,000 inhabitants. The old town of Segovia and its aqueduct is a historical town with an excellently preserved civil engineering work from the Roman period ca. 50 AD. which has been declared World Heritage Site by

UNESCO in 1985. The city of Segovia is used as the pilot city for the application; however, any city can be used in that application.

The main purpose of this application is to develop a management tool in which functionality such as identification, prioritization and maintenance activities at urban scale will be incorporated. It is intended to provide support and easy understanding of complex heritage environment for technicians and managers of historic districts in the decision making process. The awareness will facilitate the implementation of shared solutions and consensus within collaborative environments. For this, an application that helps in identifying and making decisions to prioritize consolidation and rehabilitation interventions in historic centers from urban areas has been developed. It also aims to develop a management program that allows the administration to manage and maintain the interventions in order to improve the sustainability and energy savings of the historic center.

The main objective of the application is to identify the most vulnerable districts, and therefore, the priority districts within the district; allowing focus the decision making in those districts. The application is based on a methodology that provides a vulnerability map using basic data, those data regarding the vulnerability are weighted with the information regarding the density of those buildings obtaining an prioritization or opportunity map.

The application interface is split in three sections. In the left side, there are three tabs with building properties, vulnerability and prioritization map generation and building search. In the right side, there is the 3D engine with city and building scale. Once a building is selected in the city scale, if a Building Modelling Information is available in IFC format, it is presented in building scale (see Fig. 2). As a future work, if an IFC model is available, it should be converted to CityGML LoD4 in order to represent different LoD's within the same CityGML model. Finally, in the bottom side, the additional textual information is presented.

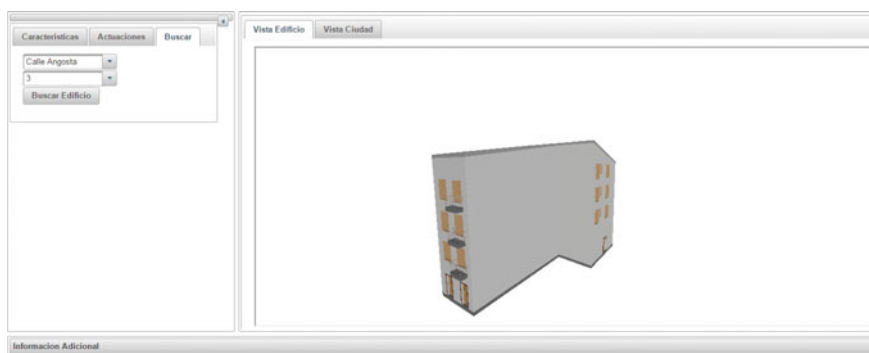


Fig. 2 Visualization at building scale

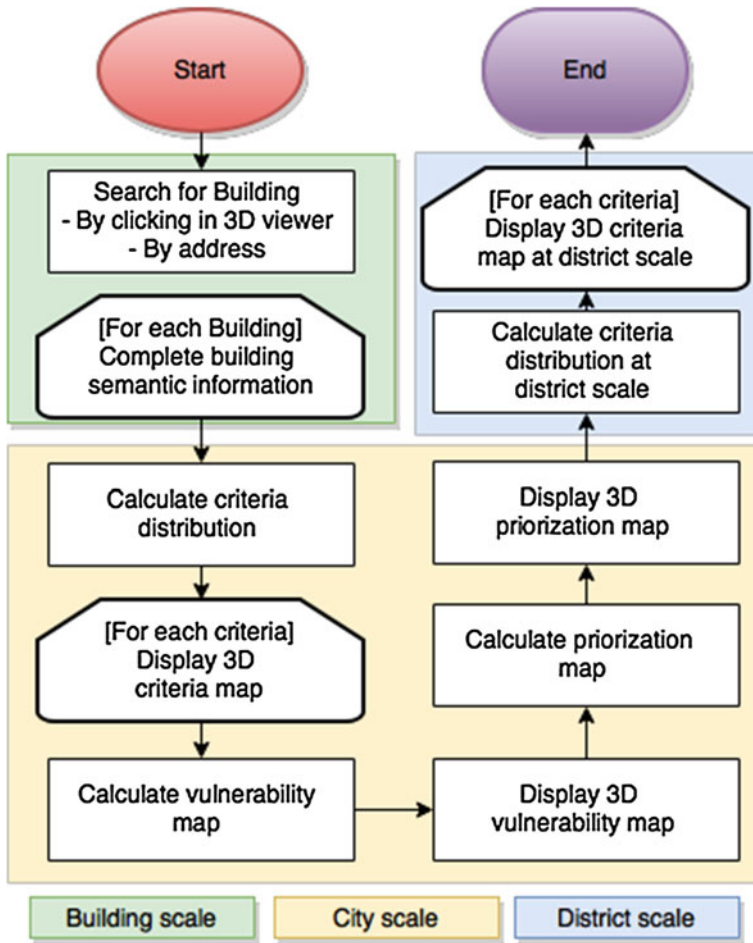


Fig. 3 Application workflow

The application workflow is intended to be as follows (see Fig. 3). Three scales are differentiated in the workflow, in green, the building scale; in yellow, the city scale and in blue, the district scale.

Firstly, if some semantic information is still incomplete; a specific building can be selected clicking directly in the 3D engine at city scale or using building search module, where a building can be selected giving its address. Once a building is selected, it is painted in black color in the 3D viewer and building properties tab CityGML core properties and ADE properties in terms of historical and energy can be fulfilled.

Then, the display of 3D urban maps by different criteria should be performed (see Fig. 4). This feature allows the visualization of a color map in 3D according to



Fig. 4 Visualization at city scale

the criteria chosen at building scale. The criteria that can be selected are: protection degree, usage, year of construction, refurbished or not, energy performance, building condition and number of incidents. The number of buildings that meet each criteria value and the percentage that represents is also presented in the additional information section.

The next step is the identification of vulnerable districts. This feature allows the visualization of 3D color map at district scale according to the vulnerability value of the district. For each district, a vulnerability value (between 0 and 10) is obtained. Districts are gradually painted depending in the vulnerability value, from green (0 vulnerability) to red (10 vulnerability). In the additional information section the parameters used in the vulnerability calculation and the color associated with value of vulnerability are presented. Once the vulnerability map is shown, if a building is selected, the entire district in which the building is presented in black. In the same way, in the additional information section the vulnerability calculation results are presented.

With the vulnerability map, the most vulnerable districts can be identified. However, it can be reasonable to prioritize districts where the impact is higher. It sounds reasonable that it is better to intervene in an district with 1000 inhabitants rather than an district with 50. Even if the second one is more vulnerable than the first one, the impact in the first is much higher. With that idea in mind, the next functionality is the prioritization map generation.

Prioritization maps allows the visualization of 3D color map at district scale according to the prioritization value of the district (see Fig. 5). For each district, a prioritization value (between 0 and 10) is obtained, being 10 high priority and 0 without priority. Districts are gradually painted depending in the prioritization value, from green (0 prioritization) to red (10 prioritization). In the same way as

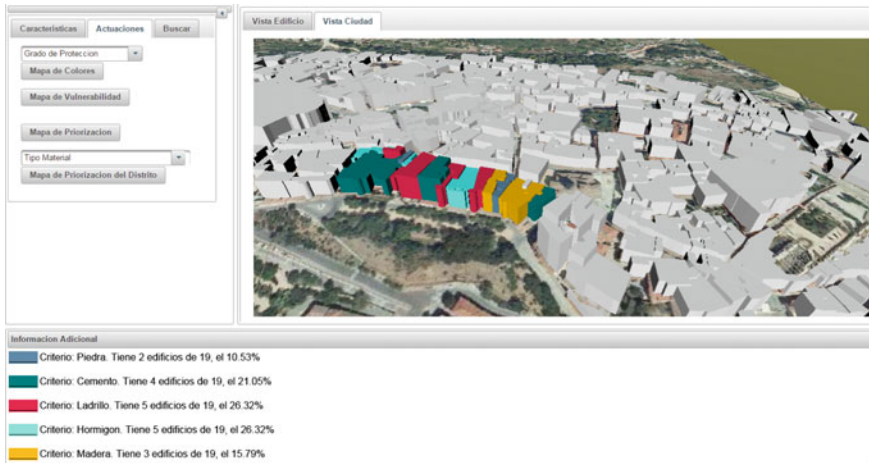


Fig. 5 Visualization at district scale

with vulnerability maps, once the prioritization is calculated and a building selected, prioritization calculation results are presented in the additional information section.

In this points, the most vulnerable and prioritized district should be identified. Once this is done, a visualization of 3D urban maps by different criteria at district scale (the selected one) can be performed. This functionality gives an overview of the different criteria in the district to be intervened.

The user interface is developed using following technologies: JavaServer Faces to ease the creation and manipulation of the interface objects. Primefaces to enrich the JSF components and to improve the aesthetics of the web interface. X3DOM is used as the 3D engine, allowing to visualize the 3D city model by means of X3D files.

The web tool is available in the following URL:
<http://3dcity.tecnalia.com/ReactApp/faces/ReactApplication.xhtml>.

5 Conclusions and Future Work

Decision-making to prioritize the action to be taken and to improve the sustainability of historic districts and their subsequent management should be based on integrated approaches with intuitive user-friendly software. The strategic management of the information generated by a historic city should be a key part of this process.

The development of data models based on the international CityGML standard allows GIS and BIM concepts to be integrated within the same model. The information contained in the model is single and can be used to develop various

applications that the different agents (historic heritage managers, technicians and members of the public) employ.

A web based application has been developed that aims the management and conservation of urban heritage by means of the usage of a decision making system. The decision making system will be feed by a 3D city model of a historic district, enabling the storage and presentation at data at city and building scales.

The main result presented in this article is the end user application that used the 3D city model, with standard web services that allows the decision making for the management and conservation of urban heritage.

Acknowledgments The work of this paper has been done as part of the project U3DCloud “Plataforma de servicios estadares en la nube para la gestion colaborativa del modelo digital 3D de la ciudad durante su ciclo de vida” with reference TSI-100400-2013-47.

References

- Bakar, N. N. A., Hassan, M. Y., Abdullah, H., Rahman, H. A., Abdullah, M. P., Hussin, F., et al. (2015) Energy efficiency index as an indicator for measuring building energy performance: A review. *Renewable and Sustainable Energy Reviews*, 44, 1–11.
- Chen, R. (2011). The development of 3D city model and its applications in urban planning. In *19th International Conference on Geoinformatics*, pp. 1–5.
- Chen, H. M., Hou, C. C., & Wang, Y. H. (2013). A 3D visualized expert system for maintenance and management of existing building facilities using reliability-based method. *Expert Systems with Applications*, 40, 287–299.
- Dambruch, J., & Krämer, M. (2014). Leveraging public participation in urban planning with 3D web technology. In *Proceedings of the Nineteenth International ACM Conference on 3D Web Technologies* (pp. 117–124). ACM.
- Delgado, F. J., Martínez, R., Prieto, I., Izgara, J. L., Egusquiza, A., & Finat, J. (2012). A common framework for multidisciplinary information management in historic urban districts. In *EuroGEOSS 2012 Conference*, Madrid.
- Department United Nations. (2010). Population of Economic. World urbanization prospects: The 2009 revision. Division, Social Affairs. UN.
- Döllner, J., & Hagedorn, B. (2007). Integrating urban GIS, CAD, and BIM data by service based virtual 3D city models. *Urban and Regional Data Management-Annual*, 157–160.
- Egusquiza, A., Prieto, I., Romero, A. (2014). Multiscale information management for sustainable districts rehabilitation EFFESUS and FASUDIR projects. In *eWork and eBusiness in Architecture, Engineering and Construction* (pp. 303–308). CRC Press.
- Ferretti, A. (2014). An indicator-based approach to measuring regeneration of historic cities. *Italian Journal of Planning Practice*, 4(1), 121–156.
- Goetz, M., & Zipf, A. (2012). OpenStreetMap in 3D—detailed insights on the current situation in Germany. In *City, Proceedings of the AGILE 2012 International Conference on Geographic Information Science*, Avignon (pp. 24–27).
- Gröger, G., & Plümer, L. (2012). CityGML—interoperable semantic 3D city models. *ISPRS. Journal of Photogrammetry and Remote Sensing*, 71(0), 12–33.
- Gröger, G., Kolbe, T.H., Nagel, C., & Häfele, K. H. (2012). OpenGIS city geography markup language (CityGML) encoding standard, Version 2.0.0.
- Klein, F., Spieldenner, T., Sons, K., & Slusallek, P. (2014). Configurable instances of 3D models for declarative 3D in the web. In *Proceedings of the Nineteenth International ACM Conference on 3D Web Technologies* (pp. 71–79). ACM.

- Morton, P. J., Horne, M., Dalton, R. C., & Thompson, E. M. (2012). Virtual city models: Avoidance of obsolescence. *Digital Physicality-Proceedings of the 30th eCAADe Confer-Comparative Evaluation of the Two Methodologies*.
- Navratil, G., Bulbul, R., & Frank, A. U. (2013). Maintainable city models for sustainable development. *International Journal of Sustainable Society*, 5(2), 97–113.
- OECD/IEA. (2013). International Energy Agency.
- Schilling, A., & Kolbe, T. H. (2010). Draft for candidate OpenGIS web 3D service interface standard. OpenGeospatial Consortium.
- Soave, M., Devigili, F., Prandi, F., de Amicis, R. (2013). Visualization and analysis of CityGML dataset within a client sever infrastructure. In *Proceedings of the 18th International Conference on 3D Web Technology* (p. 215). ACM.
- Steinhage, V., Behley, J., Meisel, S., & Cremers, A. B. (2013). Reconstruction by components for automated updating of 3D city models. *Applied Geomatics*, 5(4), 285–298.
- Vretanos, P. (2005). OpenGIS web feature service (WFS) implementation specification. OpenGIS Implementation Specification. OGC document (04-094).

Terrestrial Laser Scanners Self-calibration Study: Datum Constraints Analyses for Network Configurations

Mohd Azwan Abbas, Halim Setan, Zulkepli Majid, Albert K. Chong, Lau Chong Luh, Khairulnizam M. Idris and Mohd Farid Mohd Ariff

Abstract Similar to other electronic instruments, terrestrial laser scanner (TLS) can also inherent with various systematic errors coming from different sources. Self-calibration technique is a method available to investigate these errors for TLS which were adopted from photogrammetry technique. According to the photogrammetry principle, the selection of datum constraints can cause different types of parameter correlations. However, the network configuration applied by TLS and photogrammetry calibrations are quite different, thus, this study has investigated the significant of photogrammetry datum constraints principle in TLS self-calibration. To ensure that the assessment is thorough, the datum constraints analyses were carried out using three variant network configurations: (1) minimum number of scan stations; (2) minimum number of surfaces for targets distribution; and (3) minimum number of point targets. Based on graphical and statistical, the analyses of datum constraints selection indicated that the parameter correlations obtained are significantly similar. In addition, the analysis has demonstrated that network configuration is a very crucial factor to reduce the correlation between the calculated parameters.

Keywords Terrestrial laser scanner · Self-calibration · Network configuration · Datum constraints

M.A. Abbas (✉)

Faculty of Architecture Planning and Surveying, Centre of Study for Surveying Science and Geomatics, Universiti Teknologi MARA Cawangan Perlis, Arau, Malaysia
e-mail: mohdazwan@perlis.uitm.edu.my

H. Setan · Z. Majid · L.C. Luh · K.M. Idris · M.F.M. Ariff

Faculty of Geoinformatics and Real Estate, Department of Geomatic Engineering, Universiti Teknologi Malaysia, Skudai, Malaysia

A.K. Chong

Department of Geomatic Engineering, University of Southern Queensland, Darling Heights, Australia

1 Introduction

With the speed and accuracy, TLS has been widely used for numerous applications including accurate measurements. They are used for a variety of applications that demand sub-centimetre geometric accuracy such as landslide monitoring (Syahmi et al. 2011; Wan Aziz et al. 2012), structural deformation measurement (Gordon and Lichti 2007; Rönnholm et al. 2009), dam monitoring (González-Aguilera et al. 2008), automobile dimensioning (González-Jorge et al. 2012) and highway clearance measurement (Riveiro et al. 2013), among others.

TLS instruments are complex tools with many moving parts whose relative positions can change over time depending on use, handling frequency and care. Quality assurance (QA) is therefore a critical process to maximize the accuracy by investigate and model the systematic errors consisted in TLS measurement. The investigation procedure can be performed by the well-established self-calibration method (Fig. 1). Self-calibration presents a number of distinct advantages for this purpose (Mohd Azwan et al. 2014):

- i. No special calibration facilities are required apart from some form of targeting (usually signalized point targets);
- ii. It is based on a rigorous sensor model that includes the basic geometry of data acquisition as well as error models for systematic defects in the individual components and the instrument assembly (e.g. eccentricity and index errors);
- iii. It allows incorporation of stochastic models for the observations; and
- iv. It yields optimal estimates for the model variables along with their precision and reliability (i.e. accuracy) measures.

Since TLS self-calibration was developed from the photogrammetry approach, thus the datum constraints applied for TLS self-calibration are also similar to photogrammetry self-calibration. There are two types of constraints applicable:

Fig. 1 Self-calibration for the Faro Focus 3D scanner



Fig. 2 Photogrammetric camera self-calibration using Photomodeler V5.0 software



(1) ordinary minimum constraints; and (2) inner constraints. However, in photogrammetry self-calibration, the selection of datum constraints can cause different types of parameters correlation (Reshetyuk 2009). The use of minimum constraints tends to cause large correlation between object points and some of the calibration parameters. For the inner constraints, it has unfavourable property of increasing the correlations between the calibration and exterior orientation parameters.

There are two causes of parameters correlation in self-calibration: (1) weak network geometry; and (2) the type of constraint used. Lichti (2007) has found that the weak network geometry (e.g. limitation size of calibration field and distribution of range) can caused high correlations between calibration parameters and exterior orientation parameters as well as object points. According to the photogrammetry principle, the later (2) causes can lead to different types of parameters correlation. However, network configurations (e.g. targets distribution, size of calibration field and positions of the sensor) required for the self-calibration of TLS (Fig. 1) and photogrammetry (Fig. 2) are different. Thus, it is quite interesting to investigate whether the photogrammetry principle for the later (2) causes of parameters correlation is applicable for TLS self-calibration.

The network configuration for TLS self-calibration was addressed in Lichti (2007) as follows:

- i. A large variety of ranges is needed to accurately estimate the ranging error terms, in particular the rangefinder offset;
- ii. A large range of elevation angle measurements is necessary to recover some of the angular measurement error model coefficients;
- iii. The self-calibration can be conducted using a minimum of two separate instrument locations provided that they have orthogonal orientation in the horizontal plane (κ angles, rotation about Z axis); and
- iv. The calibration quality, as measured by reduced parameter correlations, is proportional to the number of targets used.

This argument regarding network configuration has initially indicated that the principle of datum constraints for photogrammetry is not relevant for TLS self-calibration. However, further investigation is a necessity to statistically verify the effect of datum constraints to the quality of TLS self-calibration. With the intention to scrutinise this issue, the study encompasses three objectives as follow:

- i. To evaluate the datum constraints effect of different network configurations;
- ii. To determine the suitability of photogrammetry principle of network design for TLS self-calibration; and
- iii. To analyse the causes that contribute to high parameter correlations in TLS self-calibration.

In order to achieve the objectives, this study performed self-calibration for two scanners, Faro Photon 120 and Faro Focus 3D. Both datum constraints were used the bundle adjustment and results were statistically analysed to determine whether there was any significant difference in correlation between the calculated parameters. Furthermore, to ensure this study has critically evaluated this issue, different network configurations were adopted during experiments. Three elements were taken into account for network configurations as follows: (1) the minimum number of scan stations; (2) the minimum number of surfaces on which targets are distributed; and (3) the minimum number of point targets. As a result, analyses of datum constraints were carried out based on full networks and minimum networks configuration according to the described three elements.

2 Geometric Model for Self-calibration

Due to the data measured by TLS are range, horizontal direction and vertical angle, the equations for each measurement are augmented with systematic error correction model as follows (Reshetyuk 2009):

$$\text{Range, } r = \sqrt{x^2 + y^2 + z^2} + \Delta r \quad (1)$$

$$\text{Horizontal direction, } \varphi = \tan^{-1} \left(\frac{x}{y} \right) + \Delta \varphi \quad (2)$$

$$\text{Vertical angle, } \theta = \tan^{-1} \left(\frac{z}{\sqrt{x^2 + y^2}} \right) + \Delta \theta \quad (3)$$

where, $[x \ y \ z]^T$ are Cartesian coordinates of point in scanner space and $[\Delta r \ \Delta \varphi \ \Delta \theta]^T$ are systematic error model for range, horizontal angle and vertical angle, respectively.

Since this study was conducted on panoramic scanners (Faro Photon 120 and Focus 3D), the angular observations computed using Eqs. (2) and (3) must be modified. This is due to the scanning procedure applied by panoramic scanner, which rotates only through 180° to provide 360° information for horizontal and vertical angles. Compared to hybrid scanner, the mechanism used is similar to total station, rotates 360° to cover horizontal and vertical views.

Based on Lichti (2010a), the modified mathematical model for a panoramic scanner can be presented as follows:

$$\varphi = \tan^{-1} \left(\frac{x}{y} \right) - 180^\circ \quad (4)$$

$$\theta = 180^\circ - \tan^{-1} \left(\frac{z}{\sqrt{x^2 + y^2}} \right) \quad (5)$$

The modified models above, Eqs. (4) and (5) are only applicable when horizontal angle is more than 180° . Otherwise, Eqs. (2) and (3) will be used, which means that panoramic scanner has two equations for both angular observations.

3 Experiments

According to Lichti (2007), the numbers and distribution of targets can affect the results of TLS self-calibration. Thus, with the aim to investigate the concrete evidence of the effect of datum constraints in TLS self-calibration, this study has employed several variations of network configurations as follows:

- i. Full network configurations using all targets (138 and 134 for Faro Photon 120 and Faro Focus 3D, respectively), all surfaces (e.g. four walls and a ceiling) and 7 scan stations;
- ii. Minimum number of scan stations (e.g. two stations);
- iii. Minimum number of surfaces (e.g. two surfaces); and
- iv. Minimum number of targets with 70 % reduction.

Through statistical analysis, the parameter correlations extracted from each network configuration were evaluated. The results obtained concluded whether the photogrammetry principle regarding datum constraints is applicable for TLS self-calibration.

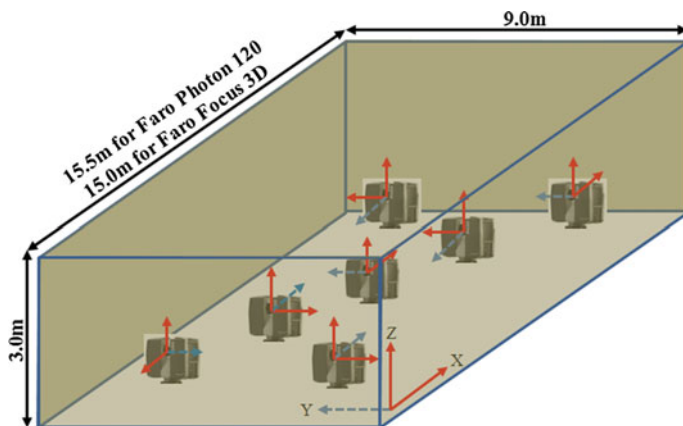


Fig. 3 Scanner locations during self-calibration

3.1 Methodology

In this study, a self-calibration was performed using two panoramic scanners, Faro Photon 120 and Faro Focus 3D. The calibration was carried out at two different laboratories with roughly similar dimensions, except the length of the room, 15.5 m (length) \times 9 m (width) \times 3 m (height) for Faro Photon 120 and 15 m length for Faro Focus 3D. The full network configurations were adopted based on Lichti (2007) conditions to ensure the quality of the obtained results is optimal.

Due to the used of different calibration laboratories, there was slightly different in number of targets used, 138 for Faro Photon 120 and 134 for Faro Focus 3D. All targets were well-distributed on the four walls and ceiling. Since the aims of the study is to evaluate the effect of datum constraints selection, which focuses on parameter correlations obtained, thus, small differences in laboratories size and targets distribution can be neglected.

Both scanners employed seven scan stations to observe the targets. As shown in Fig. 3, five scan stations were located at the each corner and centre of the room. The other two were positioned close to the two corners with the scanner orientation manually rotated 90° from scanner orientation at the same corner. In all cases the height of the scanner was placed midway between the floor and the ceiling.

With the aid of the Faroscene V5.0 software, all measured targets were extracted except for those that have high incidence angle which were not detectable. A self-calibration bundle adjustment was performed using both datum constraints (e.g. inner and minimum constraints) with precision settings based on the manufacturer's specification, which were 2 mm for distance and 0.009° for both angle measurements. After two iterations, the bundle adjustment process converged.

To perform datum constraints analyses, values of correlation coefficient were extracted from variance covariance matrix using the following formula (Abdul and Halim 2001):

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{6}$$

where,

- σ_{xy} Covariance between parameters
- σ_x Standard deviation of the parameter.

3.2 Test 1: Minimum Number of Scan Stations

Configuration of the full network was discussed in the first three paragraph of Sect. 3.1. For the second configuration, number of scan stations was reduced from seven scan stations one by one until two scan stations left as shown in Fig. 4.

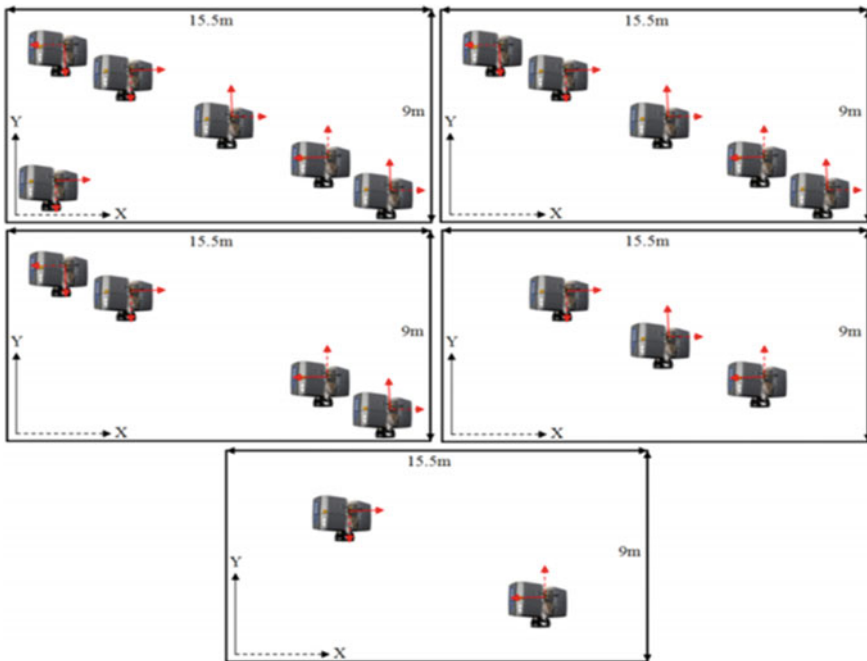


Fig. 4 Reducing number of scan stations during self-calibration

For each time the number of scan station reduced, the self-calibration bundle adjustment is performed and the datum constraints analyses were carried out. Results obtained could indicate any significant effect of datum constraints selection with variation of scan stations.

3.3 Test 2: Minimum Number of Surfaces

The subsequent network configuration focuses on reducing the numbers of surfaces used for target distribution. This is very crucial due to the difficulty to get surfaces similar as laboratory condition for on-site application. In laboratory, all targets can be distributed to the walls, a ceiling and a floor. But for on-site situation, sometimes there are only two walls and a floor available. In this study, four walls and a ceiling were used to distribute all targets. From these five surfaces, experiment was carry out by removing those surfaces one by one until two surfaces left as shown in Fig. 5. For each removing procedure, self-calibration bundle adjustment was performed and followed with datum constraints analyses.

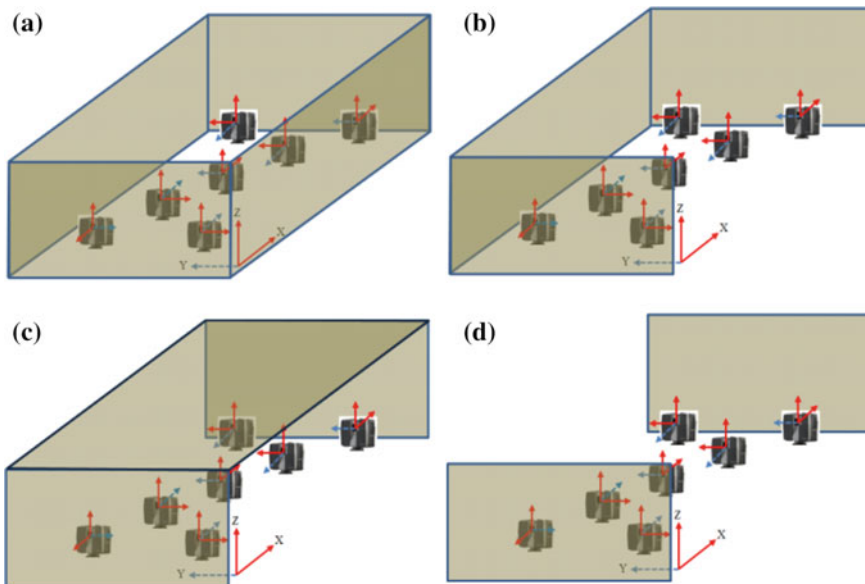


Fig. 5 Reducing number of surfaces for targets distribution. **a** Four surfaces by removing a ceiling, **b** three surfaces by removing a ceiling and a length wall, **c** three surfaces by removing both length walls and **d** two surfaces

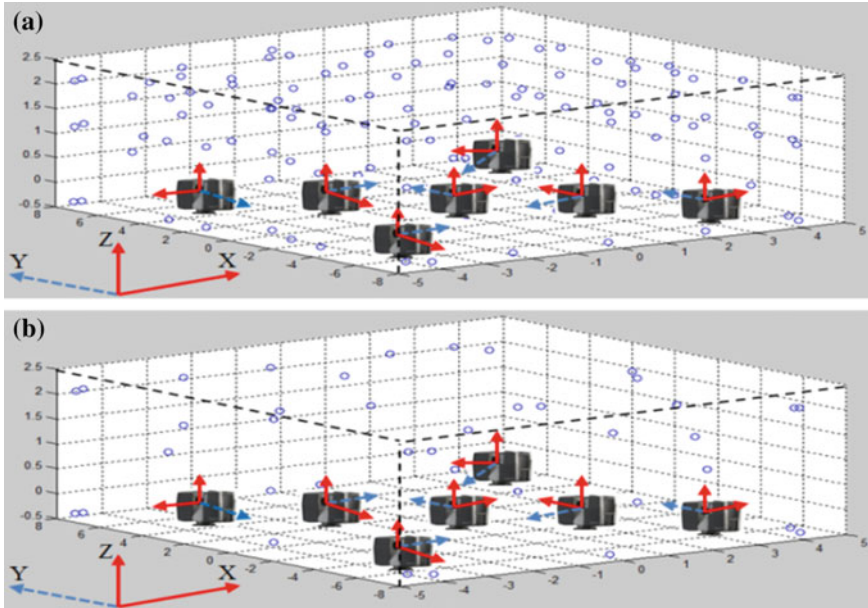


Fig. 6 Reducing the number of targets. **a** 10 % targets reduced and **b** 70 % targets reduced

3.4 Test 3: Minimum Number of Targets

The final network configuration was carried out to investigate minimum number of targets which are suitable for TLS self-calibration. This experiment was implemented by reducing the number targets from all surfaces for by every 10 % (Fig. 6a) until 70 % (Fig. 6b) were removed. As illustrate in Fig. 6, all seven scanners were used in this test to observe the targets. To maintain the quality of self-calibration result, in each trial, the targets have been well distributed. As applied in the previous experiments, each time when the targets reduced, self-calibration bundle adjustment is carried and followed with datum constraints analyses

4 Datum Constraints

Terrestrial laser scanner data involves 3D network, thus, theoretically seven datum constraints are required to remove datum defects. However, with the range observation, the scale is defined implicitly, which means that scanner network only requires six datum constraints.

To employ minimum constraints, all six datum need to be fixed. There are several procedures available to implement minimum constraints:

- i. According to Reshetyuk (2009), six fix coordinates distributed over 3 non-collinear points are required in order to use minimum constraints; or
- ii. As applied by Gielsdorf et al. (2004), position and orientation of one scanner station which represent by exterior orientation parameters were fixed to employ minimum constraints.

In order to use the minimum constraints, this study has fixed the exterior orientation parameters for the first scanner station. For the inner constraints, the method discussed in Mohd Azwan et al. (2014) was adopted.

4.1 Statistical Analysis

Correlations analyses were carried out between the calibration parameters and other system parameters (e.g. exterior orientation parameters and object points). To assess the significant difference in datum constraints selection, several graphs were plotted to visualise the difference between the parameter correlations of inner and minimum constraints. Furthermore, statistical analysis was performed to evaluate the results obtained from the plotted graphs. The F-variance ratio test was used to investigate the significance of the difference between two populations (Gopal 1999). The null hypothesis, H_0 , of the test is that the two population variances are not significantly different while the alternate hypothesis is that they are different. The F-variance ratio test is defined as:

$$F = \frac{\sigma_1^2}{\sigma_2^2} \quad (7)$$

where σ_1^2 is variance of population 1. The null hypothesis is rejected if the calculated F value is higher than the critical F value (from the F-distribution table) at the 5 % significance level. The rejection of H_0 shows that the test parameters are not equal. If the test shows no significant difference, then both datum constraints are suitable for the self-calibration bundle adjustment for terrestrial laser scanner.

5 Results and Analyses

As discussed in Sect. 1, one of the causes of parameters correlation is the type of constraints used. Furthermore, Reshetyuk (2009) mentioned that selection of datum constraints can results different types of parameters correlation in photogrammetry

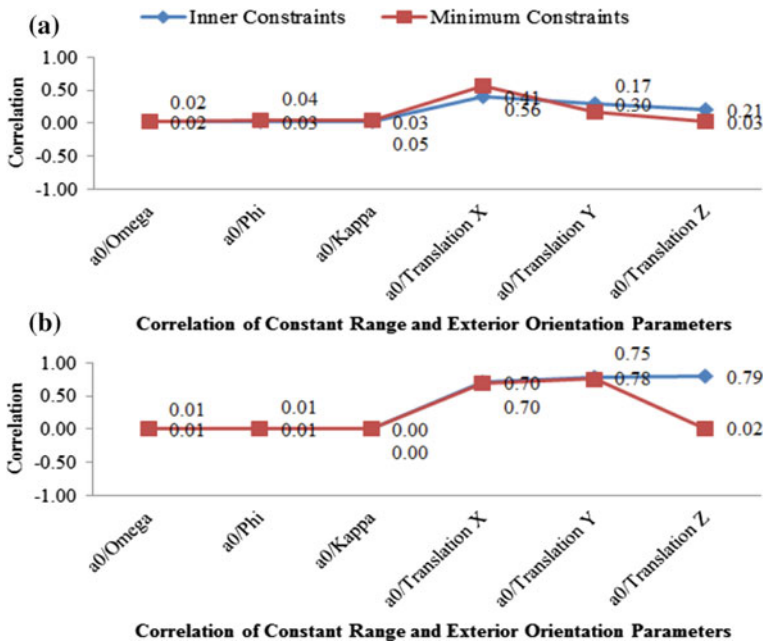


Fig. 7 Parameter correlations of constant range and exterior orientation parameters (full network configuration). **a** Faro Photon 120 and **b** Faro Focus 3D

application. Thus, investigation is carried to ensure whether that principal is applicable for TLS self-calibration. Through graphical and statistical analysis, the results obtained are discussed in detail.

Below are the plotted graphs (Figs. 7, 8, 9, 10 and 11) illustrated the comparison of parameters correlation between inner and minimum constraints for both scanners, which employed full network configuration. Due to the large number of parameters involved (e.g. seven scan stations, four calibration parameters and more than hundred targets) in variance covariance matrix, then this study has used the mean values (Fig. 8).

Lichti (2010b) in his study discussed the sources of correlation in TLS self-calibration of a basic calibration parameters (e.g. a_0 , b_0 , b_1 and c_0). The constant range (a_0) has high correlation with the scanner position (e.g. translation for X, Y and Z), while collimation axis error tends to correlate with kappa (κ). The correlations in trunnion axis error only emerge when an asymmetric target distribution exists and the vertical circle index error is highly correlated with omega (ω) and phi (ϕ).

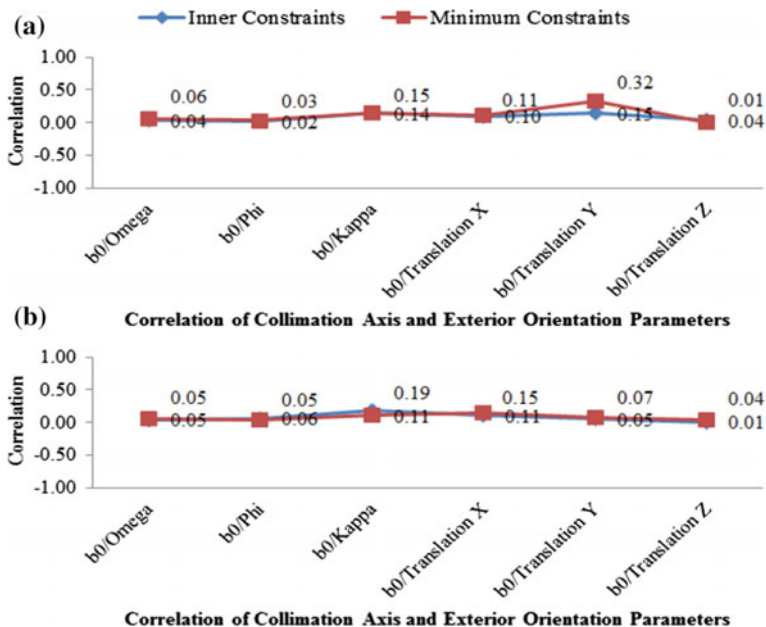


Fig. 8 Parameter correlations of collimation axis and exterior orientation parameters (full network configuration). **a** Faro Photon 120 and **b** Faro Focus 3D

Figures 6, 7, 8 and 9 represents the plotted correlation between four calibration parameters and exterior orientation (EO) parameters (e.g. omega, phi, kappa, translation X, translation Y and translation Z). According to the correlation tendency described in Lichti (2010b), similar trends have been illustrated in Fig. 6 until Fig. 9. As the aim of this study is to investigate the datum constraints effect in TLS self-calibration, each graph provides the comparison of parameter correlation yielded from using inner (blue line) and minimum (red line) constraints.

Figure 11 is depicting the correlation of calibration parameters with object points. Through the visual evaluation, initial conclusion can be made that the parameter correlation produced from both datum constraints are significantly similar. Although there are several outliers presented with maximum differences are 0.38 (between vertical circle index, c0 and translation Z in Fig. 10a) for Faro Photon 120 and 0.77 (between constant range, a0 and translation Z in Fig. 7b) for Faro Focus 3D. However, these large discrepancies can be considered as uncertainty due to the minor differences shown by the other scanner, which are 0.27 in Fig. 10b (for Faro Focus 3D) and 0.18 in Fig. 7a (for Faro Photon 120). Through statistical analysis, F-variance ratio test has mathematically proved the similarity of results obtained.

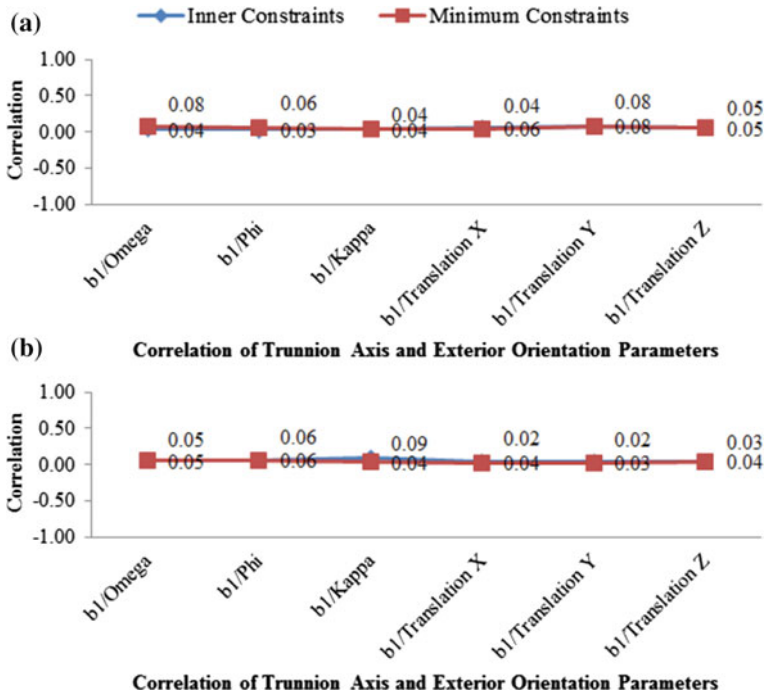


Fig. 9 Parameter correlations of trunnion axis and exterior orientation parameters (full network configuration). **a** Faro Photon 120 and **b** Faro Focus 3D

Table 1 shows that in all cases, with 95 % confidence level, the calculated F is smaller than critical F, which indicates the acceptance of null hypothesis (H0). In other words, comparison of parameters correlation calculated from using inner and minimum constraints have demonstrated a significant similarity. Since this is the results of full network which have employed very strong network geometry, thus, the good findings is expected.

With the intention to investigate the robustness conclusion regarding similarity of the correlation results yielded from both datum constraints, this study has carried out similar analysis for different type of network configurations. The first configuration is by reducing the number of scan stations. For each stations configuration, statistical analysis is performed as depicted in Table 2. For all cases, the calculated F for both scanners are smaller than critical F. In other words, the null hypothesis are accepted which mean no significant difference between both datum constraints.

Through different surfaces configurations experiment, the datum constraints analysis was again performed. Outcomes of F-variance ratio test were organised in the Table 3 for four different types of surfaces configurations. Values of calculated

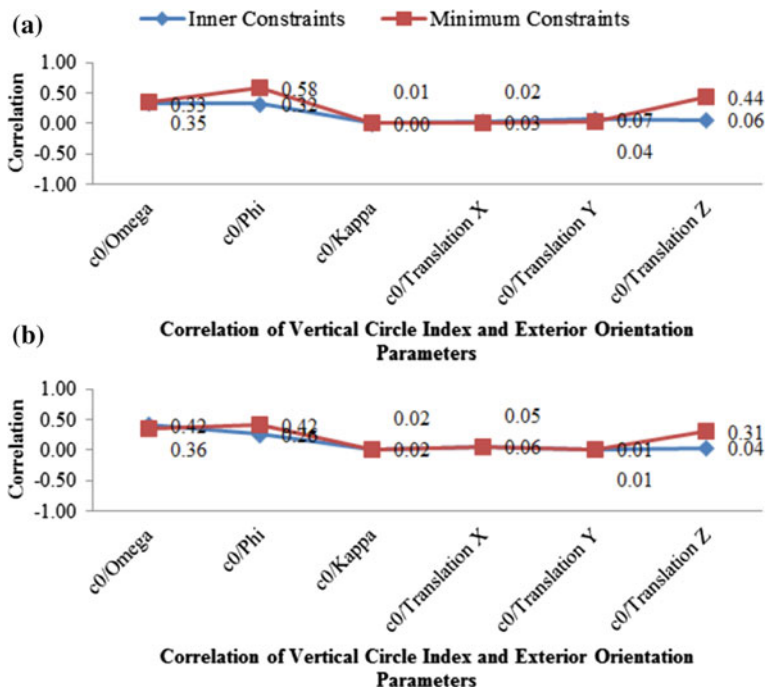


Fig. 10 Parameter correlations of vertical circle index and exterior orientation parameters (full network configuration). **a** Faro Photon 120 and **b** Faro Focus 3D

F for all circumstances have indicated the acceptance of null hypothesis, which also has increase the certainty of previous conclusion, there is no significant effect in datum constraints selection.

For the final configuration, different number of targets distribution, F-variance ratio test has concretely proved that there is no significant effect in parameter correlations from the datum constraints selection. As shown in Table 4, the null hypotheses have again statistically verified the significant similarity of both datum constraints.

As discussed in Sect. 1, according to photogrammetry self-calibration, the used of inner constraints can increase the correlations between the calibration parameters and exterior orientations. In addition, employing minimum constraints tends to cause large correlations between object points and calibration parameters. However, trend in the graphs plotted (e.g. for full network, minimum stations, minimum surfaces and minimum targets configurations) indicates different assumption. Surprisingly, for all plotted graphs, the comparisons between the parameter correlations obtained from using both datum constraints are quite similar. Since the only causes

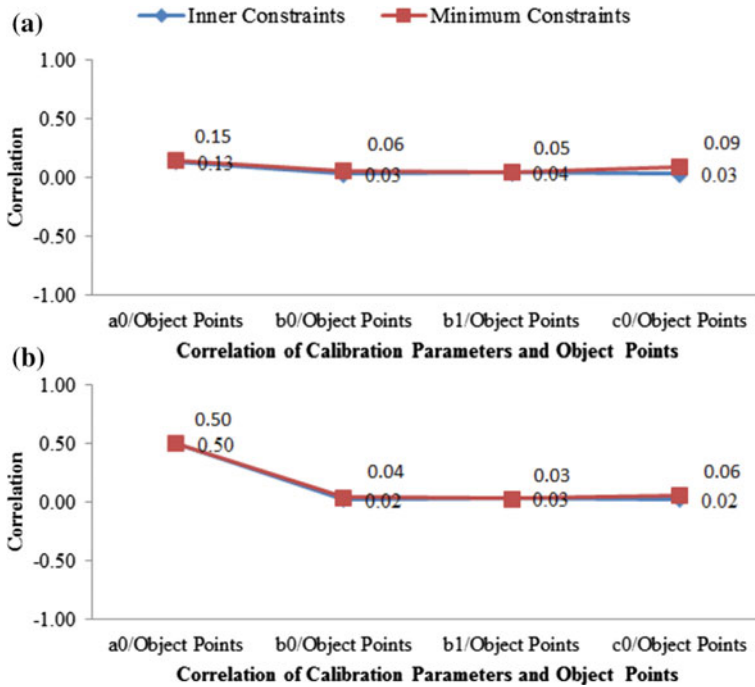


Fig. 11 Parameter correlations of calibration parameters and object points (full network configuration). **a** Faro Photon 120 and **b** Faro Focus 3D

Table 1 F-variance ratio test for full network configuration

Parameter correlations	Calculated F for Faro		>/<	Critical F
	Photon 120	Focus 3D		
a0/EO	0.09	0.35	<	5.05
b0/EO	0.42	0.001	<	5.05
b1/EO	0.01	2.50	<	5.05
c0/EO	0.69	0.33	<	5.05
CP/OP	0.01	0.01	<	9.28

for parameter correlation are network geometry and selection of datum constraints, thus, the outcomes of this study has graphically and statistically proved that the later cause is not relevant for TLS self-calibration. However, the network geometry should be made carefully, this is very crucial to ensure the quality of the results obtained (e.g. calibration parameters as well as to de-correlate the parameters).

Table 2 F-variance ratio test for different stations configurations

Configuration	Parameter correlations	Calculated F for Faro:		>/<	Critical F
		Photon 120	Focus 3D		
6 stations	a0/EO	0.07	0.22	<	5.05
	b0/EO	0.18	0.00	<	5.05
	b1/EO	0.16	2.48	<	5.05
	c0/EO	0.71	0.27	<	5.05
	CP/OP	0.86	0.01	<	9.28
5 stations	a0/EO	0.05	0.13	<	5.05
	b0/EO	0.37	0.01	<	5.05
	b1/EO	0.00	2.80	<	5.05
	c0/EO	0.63	0.28	<	5.05
	CP/OP	0.86	0.02	<	9.28
4 stations	a0/EO	0.17	0.26	<	5.05
	b0/EO	0.32	0.00	<	5.05
	b1/EO	0.21	2.08	<	5.05
	c0/EO	0.77	0.82	<	5.05
	CP/OP	0.75	0.02	<	9.28
3 stations	a0/EO	0.06	0.07	<	5.05
	b0/EO	0.00	0.33	<	5.05
	b1/EO	1.63	0.73	<	5.05
	c0/EO	0.47	1.28	<	5.05
	CP/OP	0.19	0.02	<	9.28
2 stations	a0/EO	0.14	0.27	<	5.05
	b0/EO	0.11	0.03	<	5.05
	b1/EO	0.15	0.39	<	5.05
	c0/EO	0.15	0.48	<	5.05
	CP/OP	0.11	0.11	<	9.28

According to Fig. 12, the plotted average parameter correlation (of inner and minimum constraints) with respect to the different network configurations have indicated the important things that should be considered to reduce the correlation between calculated parameters. As illustrated in Fig. 12a, b which represent the parameter correlation for Faro Photon 120 and Faro Focus 3D, respectively, the full network configuration has minimum correlation for both scanners. However, there are several results showing that minimum surfaces and targets configuration have comparable parameter correlations to full network configuration (for both scanners). This may be due to the influence of bad incidence angle caused by the used of all targets in full network configuration. In contrast, minimum surfaces and targets configurations have used reduced targets which are perpendicular to the scanner

Table 3 F-variance ratio test for different surfaces configurations

Configuration	Parameter correlations	Calculated F for Faro:		>/<	Critical F
		Photon 120	Focus 3D		
4 walls	a0/EO	0.01	0.32	<	5.05
	b0/EO	0.25	0.04	<	5.05
	b1/EO	3.18	1.53	<	5.05
	c0/EO	0.61	0.25	<	5.05
	CP/OP	0.69	0.00	<	9.28
2 walls and a ceiling	a0/EO	0.01	0.35	<	5.05
	b0/EO	0.26	0.00	<	5.05
	b1/EO	1.60	1.16	<	5.05
	c0/EO	0.56	0.37	<	5.05
	CP/OP	0.31	0.01	<	9.28
3 walls	a0/EO	0.00	0.30	<	5.05
	b0/EO	0.50	0.02	<	5.05
	b1/EO	0.81	0.55	<	5.05
	c0/EO	0.63	0.26	<	5.05
	CP/OP	0.40	0.02	<	9.28
2 walls	a0/EO	0.00	0.28	<	5.05
	b0/EO	0.07	0.44	<	5.05
	b1/EO	0.40	0.33	<	5.05
	c0/EO	0.40	0.29	<	5.05
	CP/OP	0.02	0.01	<	9.28

position. Nevertheless, the plotted graph in Fig. 12a, b for both scanners have mathematically and visually proved that network configuration has an important role compared to datum constraints, in order to reduce the parameter correlation.

6 Conclusions

A self-calibration procedure used for TLS calibration was originally adapted from photogrammetry technique, however the photogrammetry network configuration is not suitable for TLS application. This is due to the observables and measurement technique implemented by both photogrammetry and TLS are different. Therefore, further investigation was carried out to evaluate whether similar effect in datum constraints selection for photogrammetry is relevant for TLS. Graphical and statistical analyses were employed to examine any significant differences in the parameter correlations obtained from inner or minimum constraints.

Table 4 F-variance ratio test for different targets configurations

Configuration	Parameter correlations	Calculated F for Faro:		>/<	Critical F
		Photon 120	Focus 3D		
10 % targets reduction	a0/EO	0.07	0.35	<	5.05
	b0/EO	0.20	0.00	<	5.05
	b1/EO	0.53	3.07	<	5.05
	c0/EO	0.61	0.30	<	5.05
	CP/OP	0.52	0.01	<	9.28
20 % targets reduction	a0/EO	0.08	0.36	<	5.05
	b0/EO	0.27	0.00	<	5.05
	b1/EO	0.27	2.98	<	5.05
	c0/EO	0.61	0.33	<	5.05
	CP/OP	0.45	0.01	<	9.28
30 % targets reduction	a0/EO	0.10	0.36	<	5.05
	b0/EO	0.39	0.04	<	5.05
	b1/EO	0.29	2.10	<	5.05
	c0/EO	0.62	0.35	<	5.05
	CP/OP	0.61	0.02	<	9.28
40 % targets reduction	a0/EO	0.09	0.35	<	5.05
	b0/EO	0.28	0.05	<	5.05
	b1/EO	0.52	1.73	<	5.05
	c0/EO	0.56	0.29	<	5.05
	CP/OP	0.58	0.01	<	9.28
50 % targets reduction	a0/EO	0.09	0.36	<	5.05
	b0/EO	0.27	0.10	<	5.05
	b1/EO	1.22	1.21	<	5.05
	c0/EO	0.55	0.29	<	5.05
	CP/OP	0.30	0.02	<	9.28
60 % targets reduction	a0/EO	0.08	0.35	<	5.05
	b0/EO	0.18	0.01	<	5.05
	b1/EO	0.61	0.17	<	5.05
	c0/EO	0.56	0.23	<	5.05
	CP/OP	0.27	0.02	<	9.28
70 % targets reduction	a0/EO	0.10	0.34	<	5.05
	b0/EO	0.14	0.00	<	5.05
	b1/EO	0.01	0.20	<	5.05
	c0/EO	0.71	0.20	<	5.05
	CP/OP	0.30	0.01	<	9.28

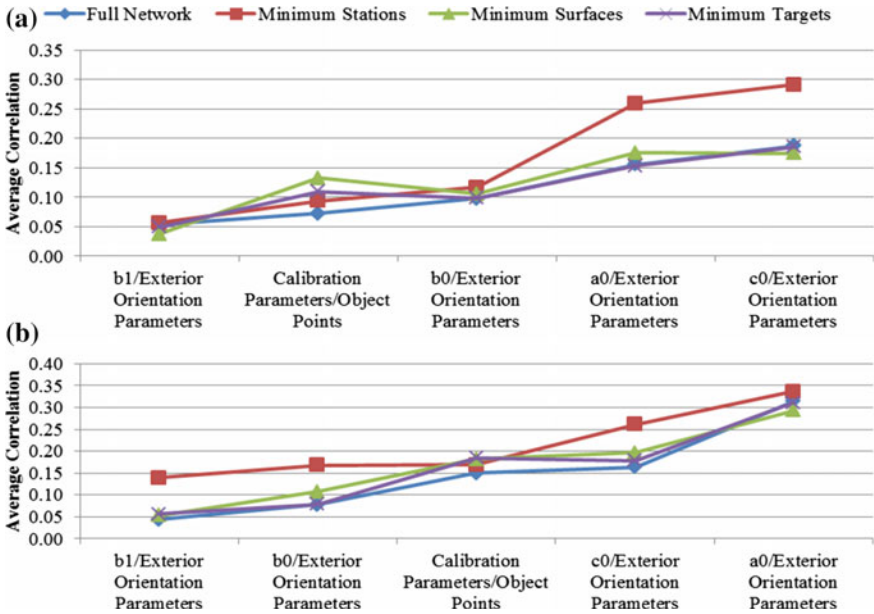


Fig. 12 Average parameter correlation (of inner and minimum constraints) with respect to the network configurations). **a** Faro Photon 120 and **b** Faro Focus 3D

To ensure that the investigation is thoroughly executed, the datum constraints analyses were carried out using three variant network configurations: (1) minimum number of scan stations, (2) minimum number of surfaces for targets distribution, and (3) minimum number of point targets. The datum constraints analyses for all network configurations have indicated that the selection of datum constraints does not affect the values of parameter correlations. Both inner and minimum constraints can provide significantly similar parameter correlations.

Nevertheless, the network configuration is a very crucial procedure to ensure that the correlation between the calculated parameters can be reduced. Since both scanners employed in this study are using panoramic system, it is quite interesting to implement similar analysis for the hybrid system scanner. Furthermore, the measurement mechanism used in panoramic and hybrid scanners are different, thus, further research will focus on the investigation of datum constraints effect for the hybrid scanner.

Acknowledgments The present research was made possible through a Vote 4L149 under MyLab grant by Ministry of Higher Education (MoHE). Special thanks goes to Photogrammetry & Laser Scanning Research Group, INFOCOMM Research Alliance, Universiti Teknologi Malaysia for the facility and technical support in this project. Authors also would like to acknowledge the Universiti Teknologi MARA (UiTM) for the financial support for my PhD study.

References

- Abdul, W. I., & Halim, S. (2001). *Pelajaran Ukur* (385 pages: 5–9). Kuala Lumpur: Dewan Bahasa dan Pustaka.
- Brian, F., Catherine, L. C., & Robert, R. (2004). Investigation on laser scanners. IWAA2004. Geneva: CERN.
- Fraser, C. S. (1996). Network Design. In K. B. Atkinson (Ed.), *Close photogrammetry and machine vision* (Vol. 371, pp. 256–279). Roseleigh House, Latheronwheel, Scotland, UK: Whittles Publishing.
- Gielsdorf, F., Rietdorf, A., & Gruendig, L. (2004). A concept for the calibration of terrestrial laser scanners. TS26 Positioning and Measurement Technologies and Practices II-Laser Scanning and Photogrammetry. FIG Working Week 2004, Athens, Greece, pp. 1–10.
- González-Aguilera, D., Gómez-Lahoz, J., & Sánchez, J. (2008). A new approach for structural monitoring of large dams with a three-dimensional laser scanner. *Sensors*, 8(9), 5866–5883.
- González-Jorge, H., Riveiro, B., Arias, P., & Armesto, J. (2012). Photogrammetry and laser scanner technology applied to length measurements in car testing laboratories. *Measurement*, 45, 354–363.
- Gopal, K. K. (1999). *100 Statistical Test* (Vol. 215, pp. 37–38). Thousand Oaks, California: SAGE Publications Ltd.
- Gordon, S. J., & Lichti, D. D. (2007). Modeling terrestrial laser scanner data for precise structural deformation measurement. *ASCE Journal of Surveying Engineering*, 133(2), 72–80.
- Lichti, D. D. (2007). Error modelling, calibration and analysis of an AM-CW terrestrial laser scanner system. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61, 307–324.
- Lichti, D. D. (2010a). A review of geometric models and self-calibration methods for terrestrial laser scanner. *Bol. Ciênc. Geod., sec. Artigos, Curitiba, 2010*, 3–19.
- Lichti, D. D. (2010b). Terrestrial laser scanner self calibration: Correlation sources and their mitigation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65, 93–102.
- Mohd Azwan, A., Halim, S., Zulkepli, M., Albert K. C., Khairulnizam, M. I., & Anuar, A. (2013). Calibration and accuracy assessment of leica scanstation C10 terrestrial laser scanner. Development in Multidimensional Spatial Data Models, Springer Lecture Notes in Geoinformation and Cartography (LNG&C), pp. 33–47.
- Mohd Azwan, A., Lichti, D. D., Albert, K. C., Halim, S., & Zulkepli, M. (2014). An on-site approach for the self-calibration of terrestrial laser scanner. *Measurement*, 52, 111–123.
- Reshetyuk, Y. (2009). Self-calibration and direct georeferencing in terrestrial laser scanning (162 pages: 68–69). Doctoral Thesis in Infrastructure, Royal Institute of Technology (KTH), Stockholm, Sweden.
- Riveiro, B., González-Jorge, H., Varela, M., & Jauregui, D. V. (2013). Validation of terrestrial laser scanning and photogrammetry techniques for the measurement of vertical underclearance and beam geometry in structural inspection of bridges. *Measurement*, 46, 184–194.
- Rönholm, P., Nuikka, M., Suominen, A., Salo, P., Hyypä, H., Pöntinen, P., et al. (2009). Comparison of measurement techniques and static theory applied to concrete beam deformation. *The Photogrammetric Record*, 24(128), 351–371.
- Syahmi, M. Z., Wan Aziz, W. A., Zulkarnaini, M. A., Anuar, A., & Othman, Z. (2011). The movement detection on the landslide surface by using terrestrial laser scanning. *Control and System Graduate Research Colloquium (ICSGRC), 2011 IEEE*, Shah Alam, Selangor.
- Wan Aziz, W. A., Khairul, N. T., & Anuar, A. (2012). Slope gradient analysis at different resolution using terrestrial laser scanner. *Signal Processing and its Applications (CSPA), 2012 IEEE 8th International Colloquium*, Melaka.

Managing Versions and History Within Semantic 3D City Models for the Next Generation of CityGML

Kanishk Chaturvedi, Carl Stephen Smyth, Gilles Gesquière,
Tatjana Kutzner and Thomas H. Kolbe

Abstract Semantic 3D city models describe city entities by objects with thematic and spatial attributes and their interrelationships. Today, more and more cities worldwide are representing their 3D city models according to the CityGML standard issued by the Open Geospatial Consortium (OGC). Various application areas of 3D city models such as urban planning or architecture require that authorities or stakeholders manage parallel alternative versions of city models and their evolution over time, which is currently not supported by the CityGML standard 2.0. In this paper, we propose a concept and a data model extending CityGML by denoting versions of models or model elements as planning alternatives. We support transitions between these versions to manage history or evolution of the city models over time. This approach facilitates the interoperable integration and exchange of different versions of a 3D city model within one dataset, including a possibly complex history of a repository. Such an integrated dataset can be used by different software systems to visualize and work with all the versions. The versions and version transitions in our proposed data model are bi-temporal in nature. They are defined as separate feature types, which allow the users to manage versioning and to perform queries about versions using an OGC Web Feature Service. We apply this data model to a use case of planning concurrent versions and demonstrate it with example instance data. The concept is general in the sense that it can be directly

K. Chaturvedi (✉) · T. Kutzner · T.H. Kolbe
Chair of Geoinformatics, Technische Universität München, Munich, Germany
e-mail: kanishk.chaturvedi@tum.de

T. Kutzner
e-mail: kutzner@tum.de

T.H. Kolbe
e-mail: thomas.kolbe@tum.de

C.S. Smyth
OpenSitePlan, Ashburn, USA
e-mail: steve@opensiteplan.org

G. Gesquière
LIRIS, Université de Lyon, Lyon, France
e-mail: gilles.gesquiere@liris.cnrs.fr

applied to other GML-based application schemas including the European INSPIRE data themes and national standards for topography and cadasters like the British Ordnance Survey Mastermap or the German cadaster standard ALKIS.

Keywords Semantic 3D city models • CityGML • Planning versions • History • City model lifecycle

1 Introduction

Semantic 3D city models describe the urban topography by decomposing and classifying the occupied physical space according to a semantic data model. The relevant real world entities are represented by objects with thematic and spatial attributes and interrelationships to other objects. However, authorities or city planners often face issues with data heterogeneity as the data is gathered from different sources at different times with differing geometric and semantic modeling techniques (Kolbe 2009). CityGML (Gröger et al. 2012) is an international standard issued by the OGC, facilitating the integration of heterogeneous data from multiple sources and the representation of the geometrical and semantic attributes of the city level objects along with their interrelationship to other objects—making it a highly semantically enriched data model in this way. As a result, today, more and more cities worldwide are representing their 3D city models according to the CityGML standard.

CityGML is an important source of information in urban planning, architecture, business development and tourism. These areas of application often address planning alternatives of buildings or other structures, e.g. for comparison by a reviewing body. The planning alternatives are not different versions of actual structures at different times but different structures that might be substituted for one another. However, CityGML currently does not support versions. Thus, the motivation behind this work is to extend CityGML by mechanisms for denoting versions of models or model elements as planning alternatives.

In the paper, we propose an approach to extend the CityGML data model and exchange format to support different versions and version transitions to allow identification and organization of multiple states in a city model. The approach is helpful in dealing with two important facets of multi-representation of semantic 3D city models (Gröger et al. 2005). The first facet is maintenance of the complete history or evolution of the city model, which is supported by version transitions having bi-temporal attributes answering the questions “How did the *city* look like at a specific point in time?” and “How did the *city model* look like at a specific point in time?”. The second facet of multi-representation is managing parallel alternative designs of the objects at the same time. Although there have been a few approaches to manage versions and history of 3D city models within 3D geo-databases (Gröger et al. 2005), our approach extends CityGML to support versions and version

transitions. This approach allows the different versions to be used in an interoperable exchange format and exchanging all the versions of a repository as one dataset. Furthermore, this single dataset can be used by different software systems to visualize and work with all the versions.

In the following sections, after a brief introduction of use cases and related work for supporting versions, we explain our proposed data model. Further, we give an illustration of the concept by demonstrating a scenario with the help of an example CityGML document.

2 Use Cases

Semantic 3D city models are used in applications like urban planning or for helping in decision making processes. As cities are constantly evolving, it is also necessary to record their changes over time. It is important that each city object may be represented with its own lifecycle. For example, for a time sequence, a building may be constructed, modified, destroyed and replaced by other ones (Pfeiffer et al. 2013) as illustrated in Fig. 1. Similar needs can also be identified in serious game projects where objects of a scene may evolve according to a given scenario. For instance, a building may be represented in different states like “destroyed”, “burned” or “partially destroyed” that can be called by the application in coordination with user actions (Chambelland et al. 2011).

Semantic 3D models also have a growing role in the documentation and reconstruction of both historical and contemporary events. Examples include crime

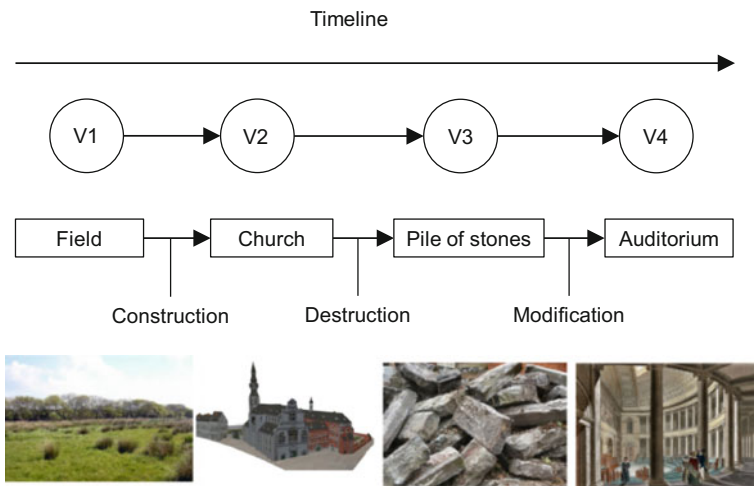


Fig. 1 An example of “historical succession”. Image adapted from Pfeiffer et al. (2013)

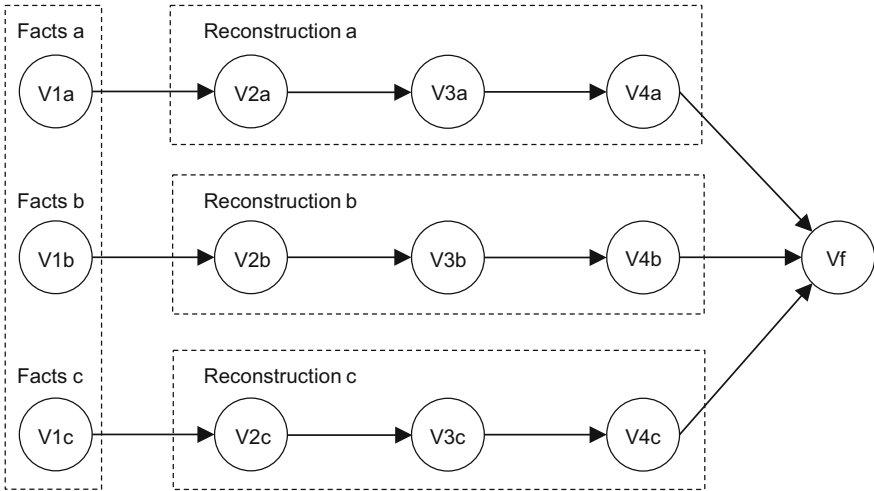


Fig. 2 Reconstruction of events to handle alternative models

scene and accident reconstructions, representation of battles and other historical events, archival descriptions of historical structures before demolition, documentation of construction and demolition of buildings. Each of these involves a sequence of versions of a “reality” at a certain time or in a certain time sequence. Events are often reconstructed from conflicting and incomplete evidence and a complete reconstruction must allow branching to handle the alternative possibilities (see Fig. 2). Modeling such approach would also allow backward compatibility to handle multiple representations of the past of a city. A given date may be a starting point to imagine the past and constructing several scenarios. Any other framework does not support such feature so far.

As shown in Fig. 3, the model should be capable of representing two different states of the city in the past, indicated here by V0 and V0'. In urban planning

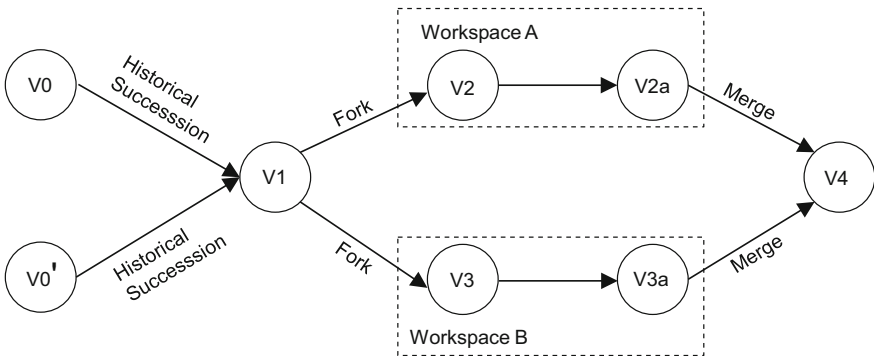


Fig. 3 History and version management

scenarios, different planning authorities can also work with alternative planned versions (V2 and V3 respectively) at the same time to insert a newly generated object or delete or update any existing object.

3 Related Work

3.1 *Related Standards and Tools*

3.1.1 AAA/INSPIRE

Two semantic information models are to be mentioned here which provide the concept of chronological versioning, i.e. the INSPIRE data specifications which define Europe-wide consistent conceptual schemas for various data themes as part of the European spatial data infrastructure initiative INSPIRE (Infrastructure for Spatial Information in the European Community) and the German AFIS-ALKIS-ATKIS (AAA) Reference Model which defines conceptual schemas for the German geospatial base data of geotopography and real estate cadaster.

INSPIRE defines several requirements and recommendations for modeling life-cycle information of spatial objects which include UML stereotypes and properties allowing for bi-temporal modeling of geospatial objects. Furthermore, a separate property exists for denoting a specific version of a geospatial object (INSPIRE Drafting Team “Data Specifications” 2014). However, currently only exchanging the last version of spatial objects is supported by INSPIRE; historic versions cannot be provided yet (and especially not within one data file).

Similarly, AAA allows for managing multiple versions by means of a specific AAA versioning schema, which makes use of life-cycle properties and an object container, where versioned objects are registered. However, the AAA versioning schema exists only as a concept: no concrete implementation is available. It is also unclear how an implementation would look like.

3.1.2 IFC

The Industry Foundation Classes (IFC) have been developed by building SMART (the former International Alliance for Interoperability), as an open standard for sharing Building Information Models (BIM) data among different software applications. To the best of our knowledge, the concept of temporal version is not currently implemented in IFC, but some extensions have been proposed in the literature. For instance, in (Beer et al. 2004), it is possible to find an approach inspired from the CAD domain.

Extensions of the IFC standard have been proposed in (Zada et al. 2014). Six existing entities from the IFC standard have been suggested to be modified to

represent as new entities within the IFC schema to support the idea of object versioning that holds the history of changes to objects of the BIM model. In (Nour and Beucke 2010), a novel approach is introduced, where both object versioning and IFC model are integrated together in an open multidisciplinary collaborative environment. Object versioning gives the possibility to have several versions of the content (attributes' values) of an object. The development of design in terms of addition of new objects, deletion of objects or modifications of attributes' values of pre-existing objects can be captured in a graph structure.

3.1.3 Tools for Supporting Versions

The concept of versions has successfully been incorporated by Oracle using the Oracle Workspace Manager (Beauregard and Speckhard 2014). The Oracle Workspace Manager allows managing multiple versions of the data in the same Oracle Relational Database Management System in the form of workspaces. A workspace is a virtual copy of the data, which separates the collection of changes in the different versions from the live (production) data. The version-enabled data are stored in separate tables with additional columns representing the version metadata. Such additional columns contain the version and workspace of each data row along with the date and time of each update. The database view is created on the version-enabled table and triggers are defined to enable SQL operations such as insert, delete, and update. This approach allows preservation of the structure of the original table and shows the data of only the respective version. The Oracle Workspace Manager also provides the management of historical data by using savepoints and the means for resolving possible conflicts during the merging of the different versions.

Similarly, versions can also be managed within ESRI ArcSDE Geodatabases (ESRI 2004). ESRI also supports managing history, performing "what-if" analysis and conflict detection and resolution. However, there are no interoperable exchange formats related to both Oracle and ESRI which would allow exchanging all versions of a repository as one dataset, nor which would allow use of the same dataset by both Oracle and ESRI.

3.1.4 GIT/SVC

Version or revision control systems (VCS/RCS) such as git (2015a), Mercurial (2015b), Concurrent Versions System (CVS) (Vesperman 2006), and Subversion (SVN) (2015c), have a structure and goals similar to the approach described in this paper. VCS were developed primarily to support parallel development within a single project. Although there are operational and architectural differences between these systems, they all maintain versions of collections of files comprising a project.

These collections are often organized in a tree structure, similar to a computer file system directory hierarchy.

A VCS has the representational power to manage changes, parallel updates, and merges of versions of CityGML models with one exception: versions always represent change in the forward direction of time. The collection maintained by a VCS is rooted in an original version, that is, the versions form a rooted directed acyclic graph (DAG). The original version is the oldest version and it is not possible to create versions earlier than the root.

In addition to the problem of forward-only temporality, a VCS also has a designated node in the DAG, usually called the head, and a distinguished path in the DAG, from the root to the head, usually called the “trunk” or “main branch”. Neither of these is required for maintaining versions of CityGML models. Although it might be possible to gain the needed representational power by piecing together multiple VCS projects, which share a common root, it would be awkward, at best.

3.2 Temporality in Semantic 3D City Models

Objects that compose the city such as buildings, bridges, vegetation, and terrain change over time. For instance, for a building, different kinds of transformations can be identified. De Luca et al. (2010) propose to define the following states for a building: creation, destruction, reconstruction, division (the building is separated in several parts), union, and variation (modifications). Other transformations are linked to the semantic part of the building, for instance when the owner of a house changes (Stefani et al. 2008). Describing the building lifecycle implies taking into account states and spatial changes along a temporal arc (Stefani et al. 2011). However, this scheme must also be extended to all objects of the city including terrain and vegetation, for example. It must take into account semantic, topological, appearance and geometric changes. Changes involving buildings can be sudden or gradual. For instance, a change of property is a sudden event. On the other hand, changes may be progressive: for example, building demolition is a short event, but the construction of a Gothic cathedral is a long event that lasts several centuries. Furthermore, historical building deteriorations may take centuries or millennia (Stefani et al. 2010).

Keeping the possibility of exchanging data with other tools and making a snapshot for a given date of the city is essential. Several papers propose methods based on CityGML. For instance in Pfeiffer et al. (2013), the CityGML scheme is modified to add temporal information on buildings. However, this method allows to register only definite states. CityGML scheme modification and possible standardized exports are not discussed. Another method proposed is based on a modification of CityGML (Morel and Gesquière 2014). In this paper the authors propose to add two additional concepts to take into account the possibility for a city

object to change and the time value which fixes this change in the city lifecycle. However, this method does not support the possibility of having different scenarios.

3.3 Requirements

Considering the limitations in related standards and tools, the following requirements have been gathered to be included in the proposed approach:

- None of the standards supports managing multiple historic versions. INSPIRE supports exchanging only the last version of spatial objects. The proposed methodology should allow supporting multiple historic versions within one data file.
- The existing approaches allow only forward-temporality. The proposed approach should allow backward compatibility to handle multiple representations of the past of a city.
- Although the DBMS systems such as Oracle or ESRI ArcSDE Geodatabases already support versions and conflict managements, the proposed approach within CityGML documents would allow exchanging all versions of a repository as one dataset and furthermore, the same dataset would be used by DBMS systems such as Oracle and ESRI.

4 Methodology

In our proposed methodology, versions become first class objects in CityGML and are modeled as feature types. Figure 4 presents the methodology as a UML model.

The city object within a specific version is assigned a stable object ID for the entire lifetime of the object—the so-called major ID. This ID is supported in GML 3.2.1 through the element *identifier* in the class *AbstractGML*, which is used for providing globally unique identifiers. Further, an extension to this major ID is given in the form of a sub ID or minor ID to distinguish different versions of the same real world entity. A separator symbol is introduced to separate the identifier from the version. For example, the specific version of a city object can be denoted as *Building1020_Version1*, where *Building1020* is the *gml:identifier* representing the stable major ID globally and *Version1* is the minor ID to represent the specific version of the building object *Building1020*. This concatenated major and minor ID (*Building1020_Version1*) is used as the *gml:id* to distinguish the different versions of the same real world object. One CityGML instance document can therefore include multiple versions of the same real world object having different *gml:id* but identical *gml:identifier* values. The idea of minor ID and major ID has been adopted from the German AAA model and the INSPIRE Data Specifications (INSPIRE

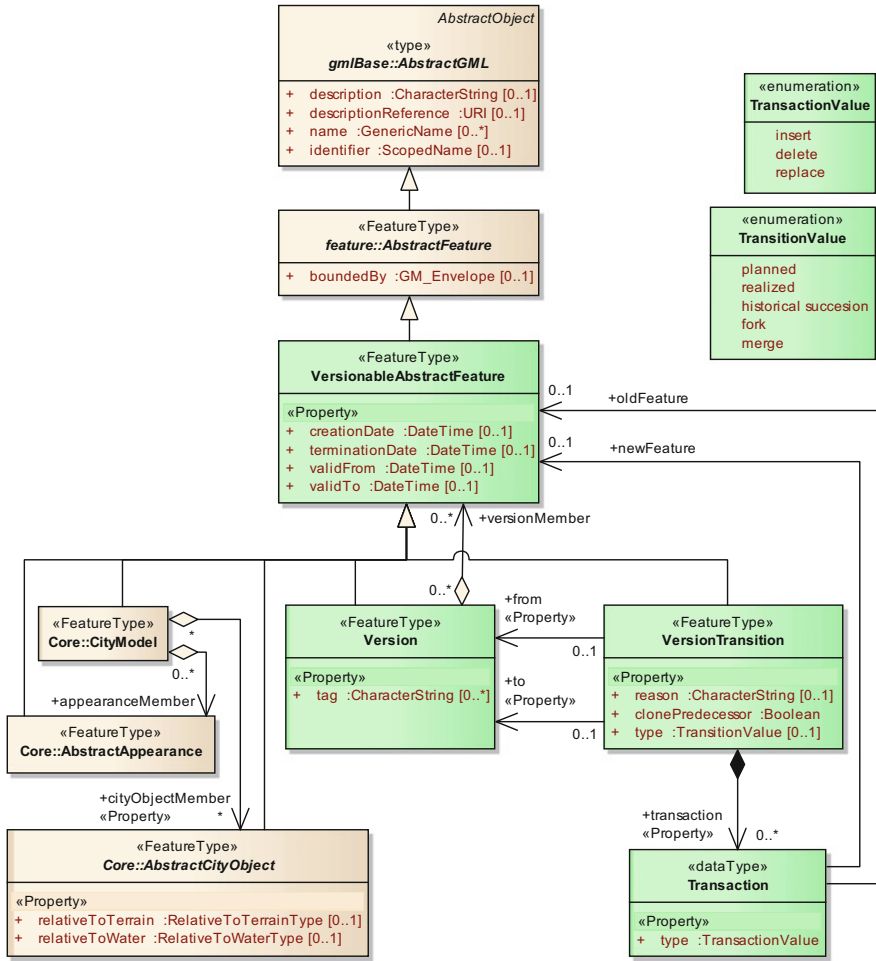


Fig. 4 Version and version transitions defined as UML model. Newly introduced classes are shown in *green*. Names of abstract classes are in *italics*

Drafting Team 2014). Referencing objects by either their minor ID or their major ID is inspired by Cox (2006).

We introduce a new abstract subclass *VersionableAbstractFeature* of class *AbstractFeature*, from which, in the future, all geo-object types shall be derived in order to become version managed. The class *VersionableAbstractFeature* contains four time attributes for expressing a bi-temporal existence model for versions. These attributes are: *creationDate* and *terminationDate*, reflecting the database transaction time, and *validFrom* and *validTo*, reflecting the actual world time. This approach is similar to the existing INSPIRE model where these attributes can be used to query how the *city model* looks like at a specific point in time and how the

actual *city* looks like at a specific point in time. These attributes can be defined as an extension to the CityGML core module and may replace the existing attributes *yearOfCreation* and *yearOfDemolition* attributes in the CityGML building module. Furthermore, the class *Version* allows each version to be denoted by a set of user-defined *tag* attributes. With the help of such *tag* attributes, the user can, e.g., search for a version developed by a specific worker. The city objects within each version can be referenced in two ways: by using a simple XLink to the *gml:id* of the referenced object which references a specific version of a real world object, or by using the XML Path Language (XPath) (W3C Recommendation 2011). XPath in conjunction with XLink allows referencing an object element in a remote XML document (or GML object repository) using the *gml:identifier* property of that object (Cox 2006). The XPath-XLink approach provides a general reference to a real world object by its major ID and does not take into account a specific version. This approach allows selecting multiple instances with the same *gml:identifier* value, but with a different *gml:id*. For example, by using a single XPath-XLink query, the user can retrieve multiple versions of the CityGML building parts within the same version of the building. However, it is necessary for the application to determine which specific version of the real world object representation should be used. The attributes *creationDate*, *terminationDate*, *validFrom*, and *validTo* can also be used to choose the appropriate version that was valid at a specific database or real world time, respectively.

In CityGML, features can also have aggregated sub-features. For example, a building feature can consist of features such as roof surface or wall surface, which

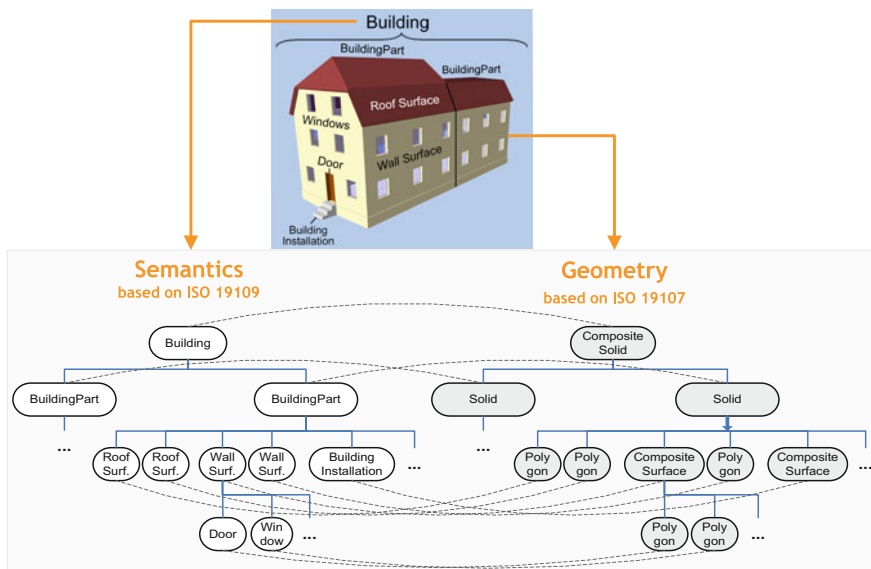


Fig. 5 Issues with versioning of aggregated features. Image taken from Stadler and Kolbe (2007)

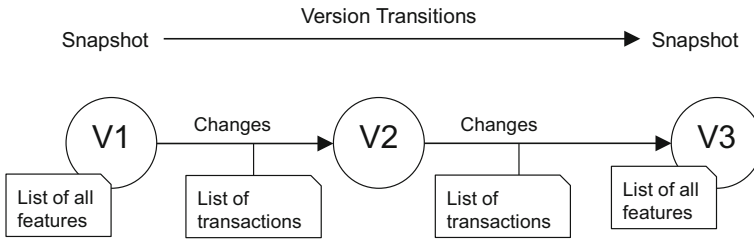


Fig. 6 Representation of version transitions

may further consist of sub-features such as window or door. However, in case of a change in any of the sub-features, the model would require changing all the parent features in the aggregation levels above because aggregate objects point to their parts. If the part is replaced by a new version with a new *gml:id*, the pointer in the aggregate object also will have to be updated. This will create a new object version also for the aggregate object, and so on, following up the aggregation hierarchy. For example, the window in Fig. 5 has been replaced by a new window with insulated glazing and a new frame. In the new version, the window will have to be changed along with its parent features. In our approach, this issue has been resolved by referencing the Major ID attribute using the XPath-XLink mechanism (see an example in Sect. 5).

Furthermore, in order to manage history or evolution of city models, version transitions are also supported. Version transitions represent the causal relationships between the version snapshots. A snapshot is a representation of the state of all features of the entire city model at a specific point in time. It explicitly links to all objects in their versions belonging to the respective city model version (see Fig. 6).

In the proposed data model shown in Fig. 4, version transition is modeled as a separate feature type *VersionTransition*, and represented by the attributes: (i) *reason*, reflecting the reason for the change in version, (ii) *clonePredecessor*, which is an indicator whether the list of features is derived from a predecessor version, (iii) *type* of transition, whether the transition is planned, realized, a historical succession, fork or merge, and (iv) *transaction*, a list of updates/transactions from a predecessor value, reflecting what types of transaction values are contained such as insert, delete or update. There are a few advantages with version transitions. This approach requires low memory or storage requirements. It is similar to the combination of full back-ups and incremental back-ups. It may also be used to stream dynamic changes.

However, a limitation with this approach is that it may be necessary to determine the actual members of a version by going back via the predecessors. One important aspect with version transitions is the merging of two different versions, which may lead to possible conflicts. For all such convergence situations, it must be ensured that the members of the converged version/state can be determined unambiguously. As shown in Fig. 7, the easiest and safest way is to require that at maximum one of the incoming transitions has transactions. Such transitions are required to be able to

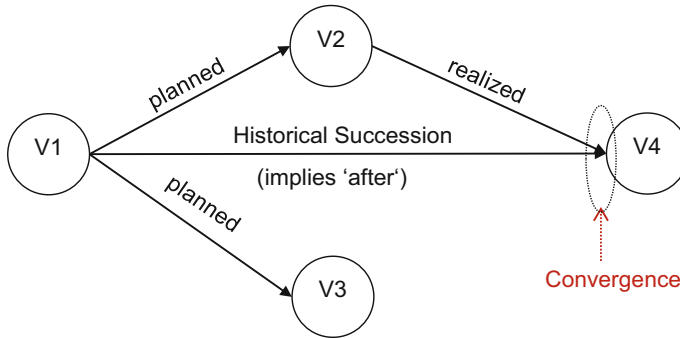


Fig. 7 Conflicts/convergence of multiple versions

detect who has changed an object and whether there are any conflicts. (Doboš and Steed 2012) have identified the methods supporting differencing and merging of 3D models. Several methods have also been identified to detect changes in CityGML files such as (Pédrinis et al. 2015) and (Redweik and Becker 2015). These methods may be useful tools to resolve conflicts in similar way as with the “diff” command in SVN.

5 Illustration of Concept

This section explains an example scenario for managing different versions within CityGML documents. Figure 8a represents the evolution of the city in the form of different versions of CityGML documents.

The successive versions represent the state of all features of the entire city at specific points in time. In addition, the authorities can work, in parallel, with different workspaces or branches to insert, delete or modify the objects. Such additions can be merged with the earlier versions of the CityGML documents to form the final versions. Figure 8b represents the evolution from initial to final versions.

However, looking more into details, the following example represents one such possibility of modification scenarios. As shown in Fig. 9, a building with the major ID *B1020* has a *function* property *Office* and one of its building parts with major ID *BP12* has a *roofType* property *Flat*. Over a period of time, the building’s *function* property is changed to *Living* which has been captured in version 2. Furthermore, at a point in time, the *roofType* property of the same building has been changed to *Saddle*.

Below is an example representing the version management in a CityGML instance data set. The building object in version 1 can be denoted as *B1020_t1* at a specific point in time *t1*. However, XPath can be used with XLink to retrieve all the instances of the same building object.

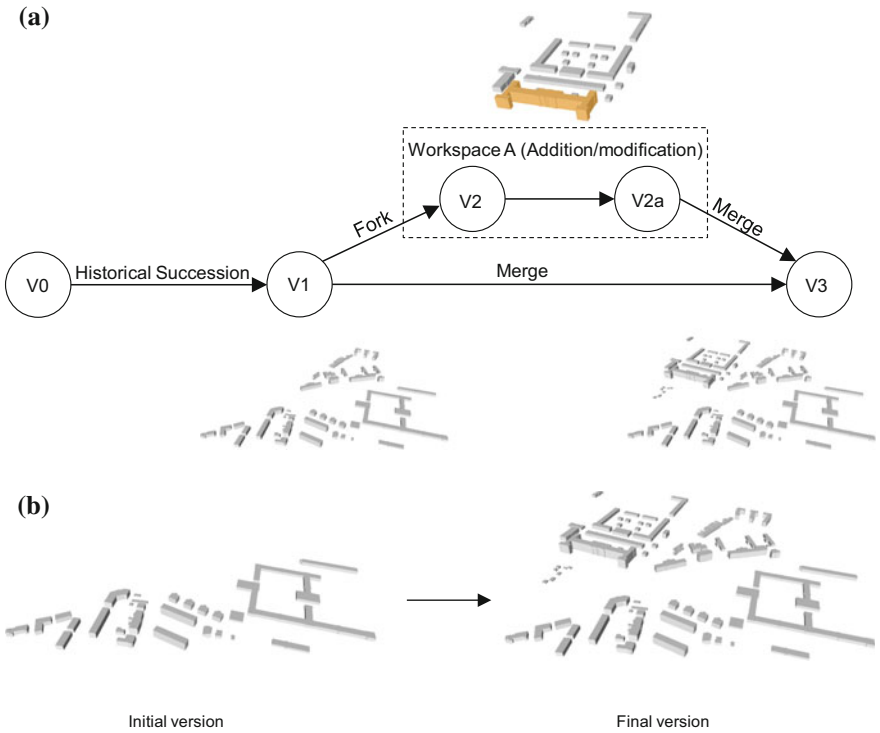


Fig. 8 a An example scenario of evolution of city in the form of different versions. b The initial version versus the final version

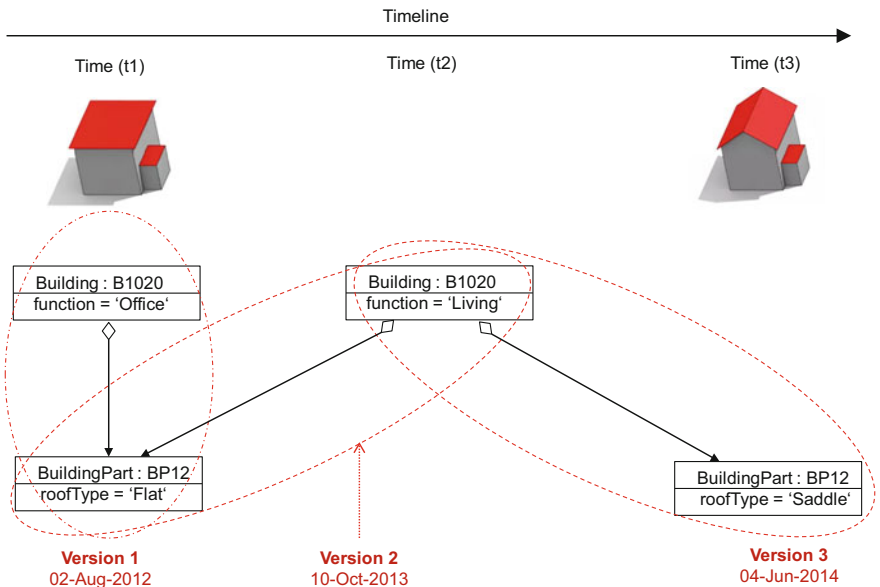


Fig. 9 An instance example of versions representing modification of a building

```

<cityObjectMember>
  <Building gml:id="B1020_t1">
    <identifier>B1020</identifier>
    <consistsOfBuildingPart>
      <BuildingPart xlink:href="//identifier[text()='BP12']"/>
    </consistsOfBuildingPart>
    <creationDate>2012-08-02</creationDate>
    <terminationDate>2013-10-10</terminationDate>
    <function>Office</function>
  </Building>
</cityObjectMember>
<cityObjectMember>
  <Building gml:id="B1020_t2">
    <identifier>B1020</identifier>
    <consistsOfBuildingPart>
      <BuildingPart xlink:href="//identifier[text()='BP12']"/>
    </consistsOfBuildingPart>
    <creationDate>2013-10-10</creationDate>
    <function>Living</function>
  </Building>
</cityObjectMember>
<cityObjectMember>
  <BuildingPart gml:id="BP12_t1">
    <identifier>BP12</identifier>
    <creationDate>2012-08-02</creationDate>
    <terminationDate>2014-06-04</terminationDate>
    <roofType>Flat</roofType>
  </BuildingPart>
</cityObjectMember>
<cityObjectMember>
  <BuildingPart gml:id="BP12_t3">
    <identifier>BP12</identifier>
    <creationDate>2014-06-04</creationDate>
    <roofType>Saddle</roofType>
  </BuildingPart>
</cityObjectMember>

```

The instance data can also include `<version>` elements for managing different versions of an object. However, due to limited space availability, this example illustrates a simple version management where it is sufficient to just use the bi-temporal time attributes and the major/minor IDs.

6 Conclusions

In this paper, we present a new modeling approach and a smart implementation for supporting the management of versions and history within CityGML. The advantage of our approach is that it not only facilitates the data model for supporting different versions, but also allows the different versions to be used in an interoperable exchange format and the exchange of all versions of a repository within one dataset. Such a dataset can be used by different software systems to visualize and work with all the versions. The approach not only addresses the implementation of

versionable CityGML models but also considers new aspects to previous work such as managing multiple histories or multiple interpretations of the past of a city. However, currently such interpretations represent slower changes; for example, change of the real property values, geometries, or ownership over time. In the future, it is also intended to support highly dynamic properties (comparatively faster changes); for example, variations of thematic attributes such as changes of physical quantities like energy demands, temperature, solar irradiation levels. The proposed UML model handles versions and version transitions as feature types, which allows the version management to be completely handled using the OGC Web Feature Service. No extension of other OGC standards is required.

The mentioned concept is going to be proposed as an official extension to the next version of CityGML (version 3.0). The concept already addresses the possibility that every feature of CityGML can be made versionable. However, in the future it might be required to also make individual geometry objects within city models versionable. It is possible to introduce the versioning of objects at a higher level in the class hierarchy of GML just below *AbstractObject*. In this way, every object of CityGML (and of GML in general) would become versionable. However, this would require changes in the GML specification, which is out of scope of the OGC CityGML standards working group (CityGML SWG). In the presented form the concept is completely modeled in the framework of the CityGML application schema and can be standardized by the CityGML SWG without changing other OGC or ISO specifications.

This concept also does not require a database or GIS with specific version management capabilities. However, in the future, it will be interesting to find out how the proposed versioning schema can be mapped onto existing version management tools of 3D GIS and databases (c.f. Sect. 3.1.3).

Acknowledgments Part of this work has been carried out within the project Modeling City Systems funded by the Climate-KIC of the European Institute of Technology and Innovation (EIT). We would like to thank Climate-KIC and the EIT for supporting this work.

References

- Beaugard, B., & Speckhard, B. (2014). Oracle Database Workspace Manager Developer's Guide, 12c Release 1 (12.1) E49170-01.
- Beer, D. G., Firmenich, B., & Richter, T. (2004). A concept for CAD systems with persistent versioned data models.
- Chambelland, J.-C., Raffin, R., Desbenoit, B., & Gesquière, G. (2011). SIMFOR: Towards a collaborative software platform for urban crisis management.
- Cox, S. (2006). Object identifiers in GML. https://www.seegrid.csiro.au/wiki/AppSchemas/GmlIdentifiers#gml:identifier_element. Retrieved June 29, 2015.
- De Luca, L., Busarayat, C., Stefani, C., et al. (2010). An iconography-based modeling approach for the spatio-temporal analysis of architectural heritage. In *Shape Modeling International Conference (SMI), 2010* (pp. 78–89). IEEE.

- Doboš, J., & Steed, A. (2012). 3D Diff: An interactive approach to mesh differencing and conflict resolution. In *SIGGRAPH Asia 2012 Technical Briefs* (p. 20). ACM.
- ESRI. (2004). Versioning! White Papers. <http://support.esri.com/es/knowledgebase/whitepapers/view/productid/19/metaid/721>. Retrieved February 26, 2015.
- Gröger, G., Kolbe, T.H., Nagel, C., & Häfele, K.-H. (2012). OGC city geography markup language (CityGML) encoding standard.
- Gröger, G., Kolbe, T. H., Schmittwilken, J., et al. (2005). Integrating versions, history and levels-of-detail within a 3D geodatabase.
- INSPIRE Drafting Team “Data Specifications”. (2014). INSPIRE Generic Conceptual Model. http://inspire.ec.europa.eu/documents/Data_Specifications/D2.5_v3.4.pdf. Retrieved April 10, 2015.
- Kolbe, T. H. (2009). Representing and exchanging 3D city models with CityGML. In *3D geo-information sciences* (pp. 15–31). Springer.
- Morel, M., & Gesquière, G. (2014). Managing temporal change of cities with CityGML. In *UDMV* (pp. 37–42).
- Nour, M., & Beucke, K. (2010). Object versioning as a basis for design change management within a BIM context. In *Proceedings of the 13th international conference on computing in civil and building engineering (ICCCBE-XIII)*, Nottingham, UK.
- Pédrinis, F., Morel, M., & Gesquière, G. (2015). Change detection of cities. In *3D geoinformation science* (pp. 123–139). Springer.
- Pfeiffer, M., Carré, C., Delfosse, V., & Billen, R. (2013). Virtual leodium: from an historical 3D city scale model to an archaeological information system. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci II-5 W 1*, pp. 241–246.
- Redweik, R., & Becker, T. (2015). Change detection in CityGML documents. In *3D geoinformation science* (pp. 107–121). Springer.
- Stadler, A., & Kolbe, T. H. (2007). Spatio-semantic coherence in the integration of 3D city models. In *Proceedings of the 5th International Symposium on Spatial Data Quality*, Enschede.
- Stefani, C., De Luca, L., Veron, P., & Florenzano, M. (2011). A tool for the 3D spatio-temporal structuring of historic building reconstructions. *Digital Media and its Application Culture Heritage*, pp. 153–168.
- Stefani, C., De Luca, L., Véron, P., & Florenzano, M. (2008). Reasoning about space-time changes: an approach for modeling the temporal dimension in architectural heritage. In *Proceedings of the IADIS International Conference*.
- Stefani, C., De Luca, L., Véron, P., & Florenzano, M. (2010). Time indeterminacy and spatio-temporal building transformations: An approach for architectural heritage understanding. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 4, 61–74.
- Vesperman, J. (2006). *Essential CVS*. O’Reilly Media, Inc.
- W3C Recommendation (2011). XML Path Language (XPath) 2.0, 2nd Edn. <http://www.w3.org/TR/xpath20/>. Retrieved February 26, 2015.
- Zada, A. J., Tizani, W., & Oti, A. H. (2014). Building information modelling (BIM)—versioning for collaborative design. In *Computing in Civil and Building Engineering* (pp. 512–519). ASCE.
- (2015a). Git User’s Manual. <https://www.kernel.org/pub/software/scm/git/docs/user-manual.html>. Retrieved March 9, 2015.
- (2015b). Mercurial SCM. <http://mercurial.selenic.com/>. Retrieved March 9, 2015.
- (2015c). Apache Subversion. <http://subversion.apache.org/>. Retrieved March 9, 2015.

Cartographic Enrichment of 3D City Models—State of the Art and Research Perspectives

Stefan Peters, Mathias Jahnke, Christian E. Murphy, Liqiu Meng
and Alias Abdul-Rahman

Abstract This paper reports on cartographic enrichments of three dimensional geovirtual environments including the representation of 3D city models. In the recent years 3D city models have become effective and powerful tools that support the simulation and visualization of our real world in a more and more realistic and detailed way. At the same time, there is a growing interest in comprising more information in the virtual living environment in addition to interior and exterior geometric features, roof and facade textures. A lot of information is related to houses, floors, flats, rooms, etc. but also to persons or specific features at certain urban locations. The paper presents the state of the art of cartographic principles in 3D city models, discusses approaches of cartographic enrichments with the aim to bring added values to the visual exploration of 3D geovirtual environments and reveals missing cartographic design rules within this area.

Keywords 3D city models · Cartographic enrichment · Information mapping

S. Peters (✉) · A. Abdul-Rahman

Department of Geoinformation, Universiti Teknologi Malaysia, Johor, Malaysia
e-mail: stefanpeters@utm.my

A. Abdul-Rahman
e-mail: alias@utm.my

M. Jahnke · C.E. Murphy · L. Meng
Chair of Cartography, Technische Universität München, Munich, Germany
e-mail: mathias.jahnke@tum.de

C.E. Murphy
e-mail: christian.murphy@tum.de

L. Meng
e-mail: liqiu.meng@tum.de

1 Introduction

For many centuries, maps are one of the prime presentation media for spatial data. The digital area during the end of the 20th century has changed cartography and brought along interactive, multidimensional, customized and context-based visual exploration methods for geoinformation. Recent developments in purchasing, managing, and visualizing 3D geodata, in particular virtual 3D cities and landscapes, reveal new potentials for 3D cartography (Pasewaldt et al. 2012). An appropriate representation and interactive use of 3D urban models is crucial for every likely application such as urban planning, 3D navigation, spatial analysis of urban data or emergency management.

A 3D geovirtual environment (3D GeoVE), such as 3D city or landscape models, attends to manage, analyze, explore and visualize geo-referenced data and information (Döllner and Buchholz 2005b). Usually, the representation of a 3D city or landscape models is composed of a database that stores the 3D geometrical, semantical as well as cartographic model, and the visualization system that renders it upon a request.

Digital 3D city models are defined as digital models of urban areas representing the relief surface, buildings, infrastructure and vegetation. Their components are described and visualized by 2D and 3D geo-referenced spatial data. 3D city models support visual exploration and analysis of urban geographical data within a single framework (Döllner et al. 2006). Furthermore, virtual 3D city models enable analysts and decision makers to investigate complex spatial and spatio-temporal processes and phenomena in urban environments. “While in 2D GIS applications exploration and analysis of thematic spatial-related objects and associated thematic information is a common practice, the potential of virtual 3D city models as a medium to communicate complex urban information spaces has not been explored extensively” (Döllner et al. 2006). This paper aims to discuss current cartographic aspects in 3D GeoVEs, including standards, limitations and possible research questions. We will introduce 3D city models in general, followed by a detailed review of cartographic rendering of 3D city models. By discussing the limitations of existing cartographic principles for 3D GeoVEs and identifying research gaps concerned with the visual improvements of 3D city models, we aim to provide an anchor point for future research on cartographic enrichment of 3D GeoVEs.

1.1 *Data Capturing and Construction of 3D City Models*

An automatic, fast as well as cost efficient construction of 3D city models is still an ongoing research topic. However, recent progresses in 3D geographic data acquisition (e.g. using LIDAR technology or image processing), data management and 3D visualization, have driven 3D city models towards widely used and effective solutions for numerous applications. Moreover, advances in cloud processing

(Döllner et al. 2012) allow to run and render large 3D city models in real time. “In practice, the creation and maintenance of virtual 3D city models is based on a number of independent data sources since the sustainable management of 3D city models requires tight links to existing administrative work flows and databases. As a major challenge, these data sources have to be integrated in a systematic and pragmatic way” (Döllner et al. 2006).

1.2 Standards for 3D City Models

However, the common information model CityGML, can be seen as the most established and widely used open data standard for 3D city models in the geoinformation community. CityGML is an official Open Geospatial Consortium (OGC) model, representing a GML-based format for storing and exchanging virtual 3D city models. Moreover, it supports thematic and semantic properties, aggregations and taxonomies. CityGML provides class definitions, normative regulations, and explanations of the semantics for essential geographic features of 3D city models. Furthermore, the model provides an easy extension by further thematic models (Kolbe 2009). The fields of architecture, engineering, construction, and facility management as well as the field of computer graphics provide their own standards. Building information models (BIM) are typically exchanged using the Industry Foundation Classes (IFC). Furthermore, CityGML is complementary to 3D computer graphics standards like X3D, VRML, or COLLADA and geovisualization formats like KML (Wilson 2008). Although advances in 3D city modeling as well as established standards allow to conglomerate distributed data from different sources, 3D data matching of overlapping but differing geometrical and semantical 3D geoinformation is still a challenge.

2 Visualization Aspects for 3D City Models

2.1 Perspective Views

The commonly used central perspective view in 3D GeoVEs offers a more natural access to geoinformation in comparison to the plane view in classic orthogonal 2D maps (Jobst and Döllner 2008). User interactions allow to reveal the visually hidden information in each static view. An alternative solution is the multi-perspective view, e.g. (Lorenz et al. 2008), deforming and distorting the view simultaneously throughout the viewer’s frame. Pasewaldt et al. (2011) extended this approach towards a view-dependent multi-perspective view.

2.2 *Photorealism and NPR*

Various applications for 3D city models, such as tourism or city planning, require a high degree of photorealistic representation of the urban objects. Such realistic impressions highly enrich the quality and quantity of the visual information content. Photorealistic representation of virtual 3D city models refers to the surface texture (respectively to textured model objects e.g. building facades and roofs) based on aerial and ground-based laser scanning data in combination with (oblique) aerial or satellite imagery. Also atmospheric effects, lighting, and shading may improve the realistic impression. Moreover, 3D photorealistic map presentations which display geographic objects in the most realistic way may facilitate mental mapping (Döllner and Kyprianidis 2010). However, photo-realistic visualization with a high degree of detail causes major problems for comprehensible visualization of 3D city models. Numerous highly detailed and textured objects might be occluded. They could also be subjected to perspective foreshortening and they may generate visual noise and overload the users with too much information (Glander 2013). Unlike photorealism, non-photorealistic rendering (NPR) abstracts the presentation from reality and offers simplified and filtered detailed elements as well as clearly encoded visualized geographic information (Döllner and Kyprianidis 2010). Visual details, such as those of buildings, are not of primary interest. Thereby an illustrative visualization achieves an effective and comprising visual display. Thematic information is encoded through an abstract visualization which allows explorative analytical functionalities.

2.3 *Level of Detail (LoD)*

Changing representation scales, performance optimization and different user- as well as usability demands require a concept for modeling and rendering objects in a 3D model at different LoDs. Most applications for virtual 3D models use the five step LoD concept as suggested within the CityGML standard (Kolbe 2009). According to Biljecki et al. (2014), the term detail can be loosely referred to the complexity and presence of geographic objects and their compartments. Thus, the term LoD is rather incomplete. Alternative terms had been introduced, such as level of completeness (Tempfli and Pilouk 1996), level of quality (Döllner and Buchholz 2005a), and level of abstraction (Glander and Döllner 2009). However, the conventional term LoD is well recognized in the GIS community. Most 3D city models use the CityGML standard for the LoD concept, which differentiates between five consecutive LoD. With increasing accuracy and structural complexity, this concept scopes from LoD0, which includes a 2½ terrain model, to LoD4, in which building details including furniture are modeled. Nevertheless, different LoD approaches have been investigated, addressing geometry, texture, semantics, the type of object, and application specific LoD concepts, e.g. (Döllner and Buchholz 2005a;

Hagedorn et al. 2009; Löwner et al. 2013). Biljecki et al. (2014) provided a comprehensive LoD analysis including major drawbacks of existing LoD concepts. Further, the authors introduced a formal and consistent framework to define discrete and continuous levels of details, enabling a finer distributions of LoDs than presently available series. Glander (2013) suggests techniques for multi-scale representations of 3D city models, which allow varying degrees of detail at the same time and support directed user attention.

2.4 Geometric and Cartographic Generalization

For an effective communication of geoinformation, cartographic symbolization and generalization are crucial tools (MacEachren 1995). For 3D city models, it is also conceivable to derive lower LoD models from higher ones. However, common 2D cartographic generalization principles (Hake et al. 2002; McMaster and Shea 1992; Slocum et al. 2009) do not adequately address all demands of 3D GeoVEs. Thus, various new generalization approaches for 3D city models have been developed, addressing in particular the geometry or façade texture of building, i.e. (Kada 2002; Mao et al. 2009; Trapp et al. 2008). According to Beck (2003) and Willmott et al. (2001), the decrease of geometric complexity ensures high and constant frame rates and thus allows to perform real-time rendering of highly complex virtual 3D city models. The multifarious and complex visualization of 3D geographic objects in perspective views demand generalization in order to represent relevant information to a user in an appropriate and efficient way (Petrovič 2003). However, existing generalization methods for 3D GeoVE are mostly feature-specific (Pasewaldt et al. 2012).

2.5 Cartographic Design

Appropriate cartographic representations increase the effectiveness, expressiveness, and readability of visualized 3D city models. Beside the conceptual phase of every cartographic product, in which user context, use situations and thematic contents are identified, Häberling et al. (2008) distinguishes between three design steps namely modeling, symbolization, and visualization. Based on this design classification, also shown in Table 1, the authors accomplished a user survey to identify altogether 19 general design guidelines that assist the cartographer to reduce visual complexity and improve comprehension in 3D GeoVEs. These guidelines address in particular the abstraction degree, the dimension degree, camera aspects, and lighting aspects.

The representation of every 3D city model can be regarded as interactive visual exploration since it involves user interactions, such as zoom, pan, rotate etc. Standard cartographic rules, valid for classic thematic and topographic 2D maps, do

Table 1 Design classification for 3D GeoVEs. *Source* (Häberling et al. 2008)

Design steps	Design aspects	Design variables
Modeling	Models of map objects	Model geometry, semantic attributes and position
Symbolization	Graphic appearance	Shape, size, color
	Textures	Pattern, pattern repetition rate and orientation
	Animations	Size and texture alteration
Visualization	Perspective	Parallel and perspective projection
	Camera settings	Viewing inclination

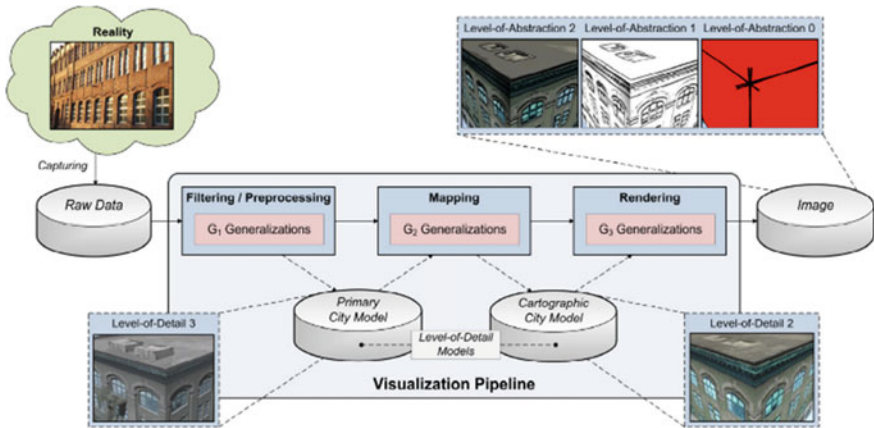


Fig. 1 Cartography-oriented visualization workflow by geometric and visual abstraction (Semmo et al. 2012a)

not exist for 3D (Häberling et al. 2008). Moreover, Pegg (2013) argued the needs of cartographic design principles for 3D city models. In the work of Semmo et al. (2012a), stylization and semantic based cartographic approaches for 3D city models are discussed. Furthermore, the authors provided an overview about the principle cartography-oriented visualization workflow for 3D city models, as illustrated in Fig. 1.

In order to consider degree-of-abstraction, depth perception and perspective distortion, new design principles are needed, and existing ones have to be extended (Pasewaldt et al. 2012). As shown in Fig. 2, Pasewaldt et al. (2012) provided a classification of cartographic design principles and visualization techniques for digital 3D GeoVEs, addressing different level-of-abstraction and LoD approaches for various feature classes.

Up to now, there are no visualization and design standards for digital 3D models. Current cartographic solutions for digital 3D GeoVEs still face a number of obstacles that influence the understanding of 3D contents, such as occlusion, visual clutter, insufficient use of screen space, and unlimited number of cartographic scales (Pasewaldt et al. 2012). According to Dykes et al. (1999), the design of 3D GeoVEs

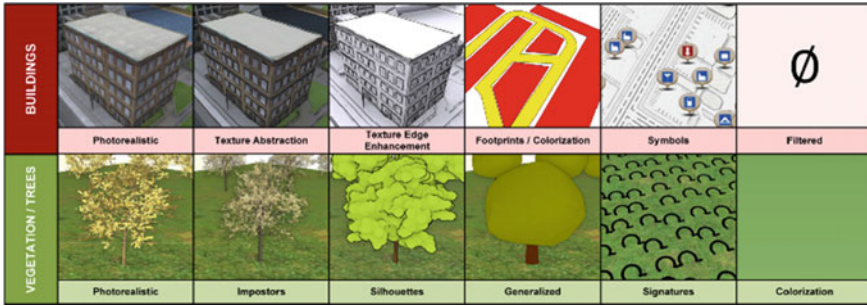


Fig. 2 Overview of cartographic techniques by the example of 3D buildings and trees. *Source* (Pasewaldt et al. 2012)

should address the needs for specific applications as well as an appropriate level of interactivity. Moreover, an appropriate typography is crucial for the understanding semantic information in 3D city models. Most of the existing studies deal with an automatic and dynamic placement of building or line labels, i.e. (Huang et al. 2007; Maass and Döllner 2007; Vaaraniemi et al. 2013). Principal design rules for texts in 3D GeoVEs, however, haven't been addressed adequately yet.

3 Cartographic Enrichment of 3D Models—Towards Interactive and Purpose Oriented Rendering

Cartographically enhanced 3D GeoVEs may improve quality and usability of the visually communicated information. On the one hand, we can use 3D city models as a platform in order to visually explore additional data or phenomena, such as the dynamics of bird swarms or the change of noise in a city during the day. On the other hand, a deeper insight into semantic information about 3D city model compartments could be provided through improved cartographic representations. One example for the latter is the thermal information of building facades, wherefore Kumke (2011) introduced various design concepts with some examples illustrated in Fig. 3. Yet, a design concept for temporal changes of thermal information, as well as the integration of those proposed design approaches into 3D city model including interactive explorative tools remain unsolved.

As discussed in Bleisch et al. (2008), the visual perception and interpretation of quantitative data, represented in an 3D GeoVEs, is not an easy task, in particular in the case of absolute data. However, due to a varying scale throughout the representation even the estimation of relative data values is not always unambiguous.

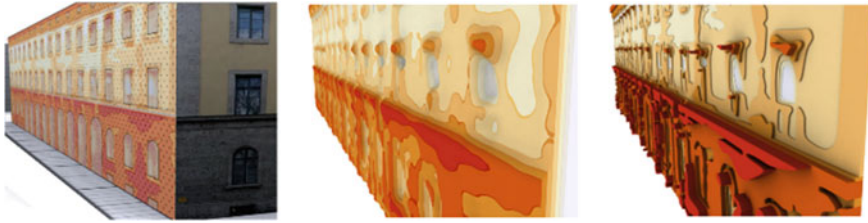


Fig. 3 Different representation of thermal information on building facades, *left* 2D isotherm surface map, *centre* 3D bi-cubic spline surface, *right* 3D isogram with discrete color scheme. *Source* Kumke (2011)

3.1 Miniaturized Non-photorealistic 3D City-Models

The visualization of 3D geoinformation is an evolving area and a challenging task in cartography. Due to the huge range of different representations of city models, ranging from block models to very detailed architectural models, and the increasing number of available city models, new approaches have to be developed to integrate additional information like semantics into the visualization. Nowadays the great advantage of 3D city models are within the city planning scenarios where the influence of a new building on the surrounded area can be very well analyzed.

Visualizing 3D spatial information is not only a rendering task, cognitive and usability aspects have to be taken into account as well as Gestalt laws (or design rules) and knowledge concerning spatial perception are the basis for converting 3D city models into an information system. Since there is still too little knowledge about how users get along with complex 3D city models (brain processes) and what kind of impact user strategies would have on the visualization design (Slocum et al. 2001). Therefore, developing new approaches and methodologies for visualizing city data is a challenging task. Within this area, the abstract and illustrative non-photorealistic approach (Strothotte and Schlechtweg 2002) seems to be promising. The non-photorealistic visualization originates from the computer graphics domain and is called non-photorealistic rendering (NPR). Typical application scenarios are ‘Cartoon Rendering’, ‘Artistic Rendering’ and ‘Sketchy Rendering’ (Gooch and Gooch 2001) as well as technical illustrations (Gooch et al. 1998). Döllner and Walther (2003) have presented a first approach of rendering a city model in a non-photorealistic way. The challenging research question is how to transfer this ‘rendering’ approach to a city model as well as including semantics into the visualization? To answer this question we have to focus on the essence of non-photorealistic and the impact it has on a 3D city model. From our point of view non-photorealistic can be described as the opposite to photorealism, which is a kind of resembling the reality. Therefore, non-photorealism must be seen as an abstraction of reality which contains illustrative, expressive and cartoon like elements.

The model and the visualization are distinct from each other (Jahnke 2013). In this way, a very detailed model can be visualized in very abstract and illustrative way. The abstract and illustrative (non-photorealistic) visualized city model can be arranged on a continuum ranging from reality to virtuality according to Milgram et al. (1995). It can be described in particular with the parameters ‘level of abstraction’, ‘information density’ and the data transferred to a user as the ‘storage capacity’ as the third parameter (Jahnke et al. 2011b). In particular the level of abstraction is of main interest as Andrew Losowsky stated that the “Visual abstraction is a human instinct and a societal necessity” (Klanten and Losowsky 2011). These parameters gave a hint on how abstract and illustrative (non-photorealistic) the visualization is. Figure 4 applied the above-mentioned three parameters to a city model visualization and showed the influence of the three parameters on the city models appearance, which can range from realism to non-realism. A highly detailed photorealistic city model incorporates a high information density as well as a high storage capacity but in contrast a low level of abstraction (Fig. 4a). On the other hand, a block model representing a city contains a high level of abstraction and less information density and storage capacity (Fig. 4d).

The degree of freedom in choosing in particular the level of abstraction that is attended by the information density is a big advantage in city model visualization. By reducing the visualization complexity, additional (semantic) information can be integrated and visualized. Moreover, it opens many possibilities of using typical cartographic design principles and graphic variables as well for three-dimensional spatial data. Table 1 shows possibilities of the applicability of different graphic variables to feature a non-photorealistic visualization. In contrast to points, lines and areas within 2D, the graphic variables can be applied to edges and the object itself (building) in 3D (Table 2).

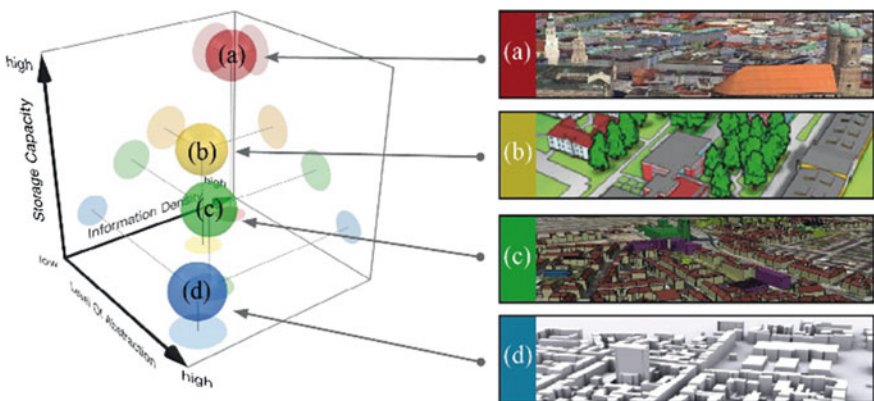


Fig. 4 Relation between level of abstraction, storage capacity and information density for different city model visualizations. *Source:* Jahnke et al. (2011b)

Table 2 Graphic variables suitable for a non-photorealistic visualization, adopted from Jahnke (2013)

Variable	Suitable for attention guiding	Suitable for edges and objects		Suitable to feature a non-photorealistic visualization	
		Edges	Objects	Edges	Objects
Color	x	x	x	x	x
Form	x	x	x	x	x
Brightness		x		x	x
Size	x	x	x	x	
Saturation	x	x		x	x

**Fig. 5** **a** A photorealistic visualization compared to non-photorealistic one; **b** color applied to objects and size applied to edges (Jahnke et al. 2011a)

The graphic variables from Table 1 are showing options to integrate semantic information or attention guiding hints into a non-photorealistic visualization. Figure 5 shows the use of color for object facades and size for edges to show the type of use for different building within Munich City Centre.

The variable “size” can be applied to objects but with the drawback of masking surrounded buildings. Therefore, the variable “form” seems to be sufficient to apply deformation to an object for coding semantic information or a more cartoon like style (Fig. 6).

Another advantage of the non-photorealistic visualization approach is the inherent change of LoD when increasing the level of abstraction and reducing the information density. Different levels of detail can be applied as well for individual buildings as shown in Fig. 7. According to (Cole et al. 2006) the varying level of detail can be used to guide the user’s attention to different areas within the city model.

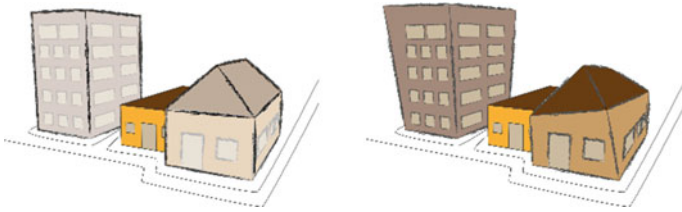
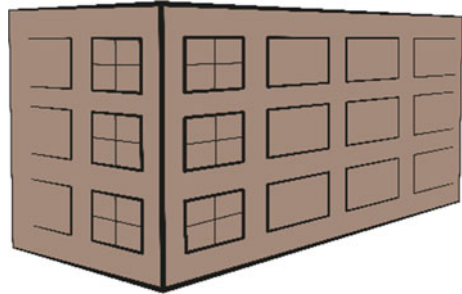


Fig. 6 The graphic variable form to apply deformation to a building (Jahnke 2013)

Fig. 7 LoD applied to an individual building (Jahnke 2013)



Within a 3D visualization the user's viewpoint influences very much the perception of the 3D city model. A viewpoint close to the ground shows only the direct surrounded buildings while a bird like viewpoint gives a good overview but less detail. To overcome this drawback annotations may be helpful. The advantages of a non-photorealistic visualized city model are the degree of freedom in choosing the design style as well as adopting the information density to the use case or purpose and the applicability of cartographic design rules. This includes as well a user or purpose oriented design. Therefore, when designing non-photorealistic visualizations usability evaluation or engineering are needed. The usability of a city model visualization is not tackled within this paper, but it is an inherent and very important area when to design feasible and suitable visualizations (Hermann and Peissner 2003; Mayhew 1999).

3.2 *Storytelling or Visual Narratives Using 3D GeoVEs*

Visual storytelling originates from the graphic design domain and is in most cases used for transferring information to a user. Losowsky stated, "The essence of visual storytelling is this combination of emotional reaction and narrative information." (Klanten and Losowsky 2011). This brings the designer to the point of not only integrating raw information but as well bringing some sense of emotion into the design and to the user. Therefore, the reaction of the user is important whether he or she is agreeing or disagreeing. Nevertheless, if visual storytelling is used in the

cartographic domain the usability plays an important role in terms of not only visualizing some information but also conveying the information in a suitable manner to gain more insights and support decision-making. Any visualization, which is intended to communicate a story, can be seen as a visual narrative while the narrative is the visual or verbal representation of a story and a story in this case can be seen as a sequence of events (Pimenta and Poovaiah 2010). The visuals and the text (story) complement each other.

An overview concerning maps and narratives is given by Caquard and Cartwright (2014) while Straumann et al. (2014) gives an example on how to construct narratives from photograph-taking-behavior. Roberts (2014) gives an example for cinematic cartography. Cinematic cartography is an emerging field covering the relation from cinema and cartographic depictions and the influence on each other, while the cinematic use of cartography has had less impact on cartographic theories (Caquard and Cartwright 2014). Based on an exhaustive literature review of online publications and magazines, Segel and Heer (2010) identified seven different genres or arrangement styles, which are common in information visualization and feature the idea of visual story telling or visual narrative. These are namely the ‘magazine style’, the ‘annotated chart’, the ‘partitioned poster’ the ‘flow chart’. The ‘comic strip’, the ‘slide show’ and the ‘film/video/animation’ (Fig. 8). These different styles refer to the ordering of elements within the visualization.

Therefore the 3D city model can play a main role within visual storytelling approach. A map or in this case a 3D representation can have two different roles in visual storytelling. The first is the role as a background information on which other information like personal trip information can be visualized. The second role is a more essential role when the map or model stays in the foreground and the narrative makes no sense without displaying the 3D city model (Fig. 9). At this point we need to distinguish between a story a map tells and the events of a story in which a map or model plays a distinct role. To sum up, a map should be seen as some sort of visual storytelling (Klanten and Losowsky 2011). According to the two defined roles the

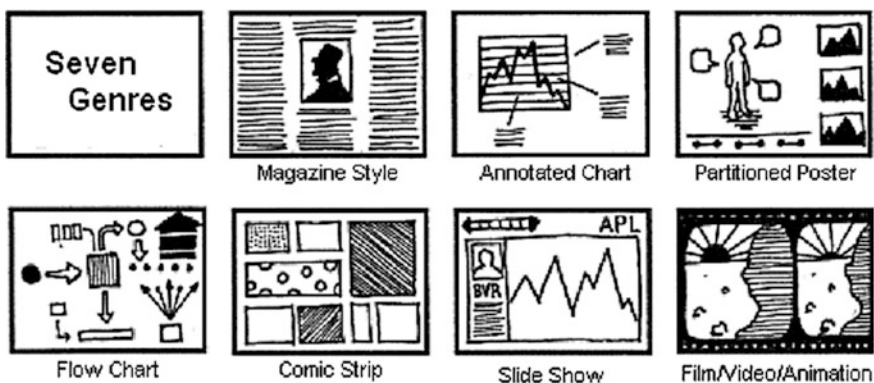


Fig. 8 Different arrangement styles in visual storytelling, from (Segel and Heer 2010)

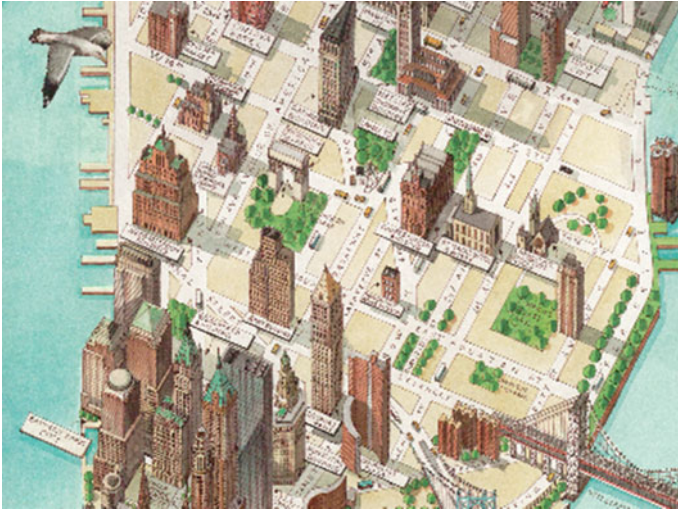


Fig. 9 A 3D model within a visual storytelling approach from Katherine Baxter to illustrate the complexity of urban areas. *Source* (Klanten and Losowsky 2011)

3D city model in the non-photorealistic visualization approach plays the second role. From this point of view visual storytelling is closely connected with the aforementioned non-photorealistic visualization of 3D city models. In particular the described level of abstraction is as well a main part in visual storytelling because the “visual abstraction is a human instinct and a societal necessity.” (Klanten and Losowsky 2011). However, the abstraction is not only outstanding in visualization it is as well very common in everyday verbal conversation (Hayakawa 1967).

3.3 *Smart Design’ of Photorealistic 3D City Models*

For many users a photorealistic 3D city model has become the holy grail of city models. Virtual city models supplemented with aerial and terrestrial imagery are therefore sometimes referred to as true 3D models. Users find the photorealistic depiction of city models an immersive and gripping experience. Their subjective feeling can be explained by hard facts. First, a photorealistic city model is much more similar to the real world. The realistic impression can help a user assigning a specific virtual camera view or an interactive walkthrough to a real place of the city. And second, the photorealistic surfaces of the city model provide an added visual value. Apart from modeling 3D building structures the textures of facades and roofs reveal detailed semantic information, such as the building type, building condition or touristic attractiveness. Furthermore, ground textures reveal information of the land use, such as traffic area or public green. However, the rich visual value of photorealistic 3D city models causes a high cognitive workload for the user. The

extremely heterogenic radiometry of photo images are very demanding to the user. It becomes more difficult to identify clear cut building edges and planes and therefore more difficult to understand the 3D shapes in a photorealistic city model than in an abstract city model presentation.

Because the photorealistic model appears visually overloaded, the composition of photorealistic 3D city models with thematic information has to be treated carefully. There are limited design guidelines that deal with the combination of imagery and cartographic symbolization. To make thematic layers visually stand out from the city model they have to be designed much differently to the imagery. This ‘pop-out’ effect described by Ware (2010) has become a visual design rule of thumb. However, creating a pop-out effect on the heterogeneous imagery is not a simple task. A holistic design approach may be missing for the design of cartographic symbols upon photorealistic surfaces, but Murphy (2014) pointed out that non-textured symbols with high color saturations have a high potential of visually standing out from imagery. The use of transparency can add to the visual segregation. Apart from a design that enables visual segregation from the visually busy city model it is recommendable to organize thematic information into themes that the user can interactively switch on and off (Kolbe 2009). This can significantly reduce the visual complexity of an interactive virtual 3D city model.

The cartographic design of virtual 3D city models should not be limited to design of map symbols. A smart design of photorealistic 3D city models should consider the design of imagery in the same way as it considers the design of map symbolization. It has been shown on image maps, that the photorealistic imagery can be manipulated in a way that photorealistic discrete objects can be designed to visually stand out from an equally photorealistic background (Murphy 2014). A number of highlighting strategies for discrete image objects (i.e. buildings) can highlight important objects such as built landmarks and make them visually more prominent without deteriorating the image legibility of both fore- and background (see Fig. 10).



Fig. 10 Visual highlighting of photorealistic objects by ‘Light Beam Guidance’ (Murphy 2014)

The user of a virtual 3D city model expects to navigate freely through it and expects a visual optimal presentation for every current view. The visual highlighting of important city model entities should be dependent on the viewing angle and distance. Different design zones could be applied as a function of distance from the virtual camera (Jobst and Döllner 2008). Building models of importance that are in the viewer's focus should be highlighted whereas distant buildings do not have to be visually highlighted. Particularly photorealistic surfaces rendered in small scales cause visual clutter (i.e. Glander 2013). This can be resolved by integrating the photorealistic design of 3D city models into a multi-scale approach. In 3D city model multi-scale approaches zones of distance (or scales) are defined in which different generalization levels apply (Brewer and Buttenfield 2007; Glander 2013; Pasewaldt et al. 2011). The generalization levels could reach from a photorealistic 3D city model over a non-photorealistic rendering to an abstract 2D map. Building blocks that stand in the focus are thereby visualized in a photorealistic view. For city model zones that are further away from the virtual camera a more abstract view is convenient.

3.4 Additional Information Within 3D City Models

Beside semantical information about geographical objects which usually shape a 3D city model, we might also incorporate additional data, such as those of people who live or work in respective urban places. Furthermore, we could consider information about distinctive events, facts, opinions, ideas, or statistical summaries, but also data of dynamic objects as individual or a group of moving people, vehicles, animals etc. After the acquisition of such additional data, they can be related to or spatially joint with certain 3D locations/addresses, buildings, streets, areas (e.g. administrative district) or landmarks. Possible data sources for those additional data include, among others, social media (Twitter, Facebook), federal statistics and census agencies, Volunteered geographic information such as OpenStreetMap (OSM) or the internet as a database for explicit knowledge processing as investigated by Rückemann (2014). When combining different geometrical and semantic data into the same 3D city model, an automatic and evaluable data matching/integration is worth striving for, which is an important research task. Challenging additional information for 3D city models are temporal changes as well as the task to visualize spatio-temporal information of moving objects or of 3D city objects (e.g. year of construction), their dynamics/temporal changes (changes regarding geometry or semantics) or predicted future situations/scenarios. Approaches for integrating time-dependent features in 3D city models had been introduced, for example by Fan (2010). Nonetheless, different representation options of such temporal information need to be investigated in more detail, including animations, dynamic effects, user interactions, adaptive and dynamic legends, as well as integrated visual analytical tools.

A further potential additional information include the ‘uncertainty’, which can refer either to the underlying data of the city model, to the above mentioned additional data, or to the data processing and visualization. According to Pang (2001), “uncertainty is a multi-faceted characterization about data, whether from measurements and observations of some phenomenon and predictions made from them. It may include several concepts including error, accuracy, precision, validity, quality, variability, noise, completeness, confidence and reliability”. Visualization guidelines and an overview about existing concepts and approaches for visualizing uncertainties of geographic data are provided by (Griethe and Schumann 2006; MacEachren et al. 2005; Peters 2014; Slocum et al. 2009). Basically, visual variables of point-, line- or polygon features correlate with the uncertainty information. Gershon (1998) distinguished between intrinsic and extrinsic visual. However, most approaches for uncertainty visualization of geodata refer to 2D solutions. A cartographic investigation of uncertainty visualization within 3D GeoVEs would involve an adaption or extension of existing 2D solutions as well as interactive tools for uncertainty-focused explorative user interactions, and a comprehensive user- and usability study.

3.5 Visual Analytics and 3D City Models (3D GeoVEs)

To take complex information within 3D city models and make them understandable for users is not a trivial task, but can be supported by visual analytical tools. According to Dykes et al. (2010), research in visual analytics may contribute directly to the exploration of 3D building models. Thomas and Cook (2005), defined visual analytics as the science of analytical reasoning supported by interactive visual interfaces. The basic concept of any visual analytical process is to combine computer advance graphics representations with human cognitive capabilities in order to provide better understanding and reasoning. The use of visual analytics for spatial and spatio-temporal processes has been extensively investigated for 2D representations. Potentials of visual analytics for urban design have been recently discussed by Batty and Hudson-Smith (2014). Visual analytical approaches are mostly data driven and application specific. Visual analytics could be used to enable users to explore and investigate urban processes within a 3D GeoVE, while additional analytical tools provide an insight into semantic information of 3D city objects respectively of the additional information. Thereby a 3D GeoVEs constitutes the frame, wherein visual analytical tools such as interactive time graphs, charts or diagrams are integrated to illustrate additional location-related qualitative and/or quantitative information. Thus, appropriate data interaction and exploring, as well as an elegant navigating through the 3D GeoVEs at the same time are crucial. Adequate solutions for such visual analytical tools need to be task- or application- respectively user-specific.

There are a limited existing works. For instance, De Amicis et al. (2009) provided a 3D web based interactive visual evaluation tool for investigating the environmental impact of new buildings. Bak et al. (2010) introduced a visual reasoning tool for investigating spatial relation between geo-referenced urban environmental variables. Moreover, Debiasi et al. (2013) developed a visual analytical tool for urban traffic simulation. Further works focus on visual decision support in flood management, such as Waser et al. (2014). A main drawback of most of these existing approaches are missing comprehensive user and usability tests. Bleisch (2012) discussed relevant tasks addressed by the field of 3D Geo-visualization, a field which we can relate to visual analysis or analytics of 3D GeoVEs. However, yet almost none of those tasks are solved with help of additional visual analytical tools. Thus we see a strong demand for further investigations of task-specific visual analytical approaches in 3D GeoVEs. Instead of using 3D GeoVE as a platform for visual analytical tasks, geometrical, semantical and topological data of a 3D city model can also be investigated via visual analytics. For a very simple example, bar charts located in the center of residential blocks could illustrate statistical data such as number of houses, windows, doors, etc.

3.6 3D Topographic Symbols

To adapt symbolization is one of the principle cartographic design tools, also for 3D GeoVEs. The lack of and need for cartographic design rules and standards for 3D symbols, in particular symbols for topographic objects, in 3D GeoVEs has been discussed in various publications, i.e. (Bandrova 2001; Petrovič and Mašera 2005). Bandrova (2001) provided a first theoretical base for 3D cartographic symbols in 3D GeoVEs, including requirements and firm tools for designing such 3D symbols. The author draws the conclusion that symbol systems have to be designed for each particular application considering purpose and end-users. According to Petrovič (2003), symbols in 3D presentations have to follow cartographic design principles as used for traditional 2D maps. Petrovič suggest to use typical realistic 3D point symbols for point-like objects, in particular for natural-made objects such as trees, bushes, and waterfalls. Furthermore, he evaluate geometrical 3D symbols suitable for man-made point objects. According to Petrovič, line symbols are mostly entirely draped over/on top of the terrain model. Applying a certain extrusion to these lines might visually emphasize these objects. Furthermore, the author distinguished between polygonal 3D area symbols and volumetric 3D symbols. For the latter, examples were provided for different scales which refer to the concept of LoD, as exemplary shown in Fig. 11.

Finally, Petrovič concludes that 3D cartographic symbols used in 3D presentations need to be further investigated and evaluated by user and usability tests. Current topographic products represent the real world mostly in two dimensions. Existing 3D approaches bond the 2D topographic map onto a 3D relief. Some of them use 3D building models and 3D objects representing landscape elements such

Fig. 11 3D symbols for different LODs (Petrovič 2003)



as trees and hedges. However, important topographic objects, represented as 2D symbol on such 3D GeoVEs, might be less or even invisible due to occlusion, viewing distance, depth perception and perspective distortion. Adequate designed and scale dependent 3D topographic symbols as well as user interactions in 3D GeoVEs such as object/layer selection or disabling, and highlighting tools (e.g. increasing symbol size) can improve the visibility of topographic objects and, thus, the communication of topographic information. Symbol concepts and standards need to be adapted to regional conventions in the same way as topographic symbolization standards differ between countries. Cartographic principles for 3D symbols are not only relevant for topographic information in 3D GeoVEs, but also for certain thematic information (layers), such as geology, tourism or urban planning.

3.7 The Problem of Invisible Objects—Revealing the Hidden

The visibility of geographic information is one of the cartographic principles, and one of the major drawbacks within 3D GeoVEs. How to deal with covered areas behind obstacles? How to minimize information loss caused by perspective distortion and large distance between object and view point? Obviously, interactive user functions, such as zoom, pan, and rotate, help to solve these tasks. However, we'd like to focus on adaptive real-time visualization solutions for the best visibility for every view perspective.

First we need to know what defines or influences the visibility of an object in a 3D GeoVE. In addition to the degree of occlusion and viewing distance and perspective, crucial visibility factors are the level of abstraction, the level of generalization as well as the cartographic symbolization of an object. A further important aspect is the visual salience of an object in comparison with its neighborhood. Instead of focusing on one object only, the task could also be to increase visual salience of several objects of interest (object group), of one or more certain locations respectively areas, or of all objects. Numerous studies were dedicated to the generalization of 3D objects and to the LoD concepts of 3D city models (Fan and Meng 2009; Kada 2002). Possible cartographic generalization methods, appropriate for increasing visibility and readability, also include object displacement or

dimensional collapse (use of 3D symbol as described in the previous section). Furthermore, view dependent multi-scale representations, as introduced in (Semmo et al. 2012b) aim to improve the visibility of selected 3D city objects. In doing so, an object of interest receives the highest LoD while with rising distance a decreasing LoD would be applied to its neighborhood objects. At the same time values of visual variables (e.g. transparency, color hue and value, size, form, fill pattern, etc.) are adapted depending on certain distances or locations using either a linear transition or discrete steps. Furthermore, to increase its visual salience, an object can be highlighted by lighting effects or by applying dynamic symbol effects (e.g. blinking). To maximize information communication for respective applications and tasks, interactive tools are needed, enabling the user to adapt visibility parameters, for instance, to increase transparency of certain objects or displace/extrude objects of interests. A real-time implementation of such view-dependent interactive visibility-optimization tasks within a 3D GeoVE poses major technical challenges, in particular in the case of a web-based distributed database.

However, existing solutions, such as the multi-perspective view approach (Pasewaldt et al. 2012) demonstrate the feasibility of those ideas. Thereby, the authors suggested a combination of cartography-oriented rendering techniques and photorealistic graphic styles with multi-perspective views in order to increase screen-space utilization while simultaneously directing viewers' eyes to important or prioritized information. An example is shown in Fig. 12.



Fig. 12 View-dependent, multi-perspective view and cartography-oriented stylization applied for a route of interest in the city center of Chemnitz. *Source* (Pasewaldt et al. 2012)

4 Conclusion

We have shown that cartographic enriched 3D city models allow us to address new application and research areas and expose visual explorative solutions for specific tasks and scenarios. Approaches in the fields of computer graphics and information technology provide powerful rendering performance for multi perspective views and multi-scale representations of 3D city models.

However, within this contribution we reveal missing cartographic design rules and standards for 3D GeoVEs. First theoretical steps in this direction have been made. The non-photorealistic visualization approaches opens the usage of well-known cartographic design rules within a 3D visualization. It reduces the information complexity, decreases the cognitive workload of the user (Bunch and Lloyd 2006) and opens up space for displaying non-geometric information. With regards to visual storytelling, 3D city models or 3D representations can have two different roles: they can serve either as additional background information or they stay in the foreground while the story wouldn't make any sense without the 3D representation. Appropriate design guidelines for cartographic symbols upon photorealistic surfaces are still an ongoing research topic. First attempts include the highlighting strategies for discrete image objects (Murphy 2014). A smart design of photorealistic 3D city models needs to treat the imagery design in the same way as it does for other map symbols.

Furthermore we discussed the enrichment of 3D city models with additional multivariate or/and multidimensional information and the need for adequate cartographic solutions while incorporating such data. In addition, cartographic design concepts as well as standards for 3D topographic symbols need to be further investigated. We have also debated that cartographic enhancements for 3D GeoVEs should involve user and usability issues. Potential applications demand customized, interactive, and smart visualization of 3D city or landscape models. That could include a visually optimized representation for every current view point in real-time. The cartographer's task is to develop better 3D GeoVEs to train users to operate them. User-friendly and intuitive interfaces should provide best communication between user and data space. The user should have the possibility to change the visualization style as well as to explore data by the use of visual analytical tools. Thus, the user himself becomes the map producer. An appropriate user friendly integration of visual analytical tools require comprehensive user- and usability investigations. Many existing applications lack comprehensive user- and usability tests. Such tests could iteratively improve an application while learning from user behaviors. More sophisticated usability investigations could include multi user interaction and communication. Last but not least, designing interactive and customized legends within 3D GeoVEs is also an up-to-date research topic.

References

- Bak, P., Omer, I., & Schreck, T. (2010). Visual analytics of urban environments using high-resolution geographic data. In M. Painho, Y. S. Santos and H. Pundt (Eds.), *Geospatial Thinking*. Springer.
- Bandrova, T. (2001). Designing of symbol system for 3D city maps. In *20th International Cartographic Conference* (pp. 1002–1010).
- Batty, M., & Hudson-Smith, A. (2014). Visual Analytics for Urban Design.
- Beck, M. (2003). Real-time visualization of big 3D city models. *International Archives of the Photogrammetry Sensing and Spatial Information Sciences*, 34.
- Biljecki, F., Ledoux, H., Stoter, J., & Zhao, J. (2014). Formalisation of the level of detail in 3D city modelling. *Computers, Environment and Urban Systems*, 48, 1–15.
- Bleisch, S. (2012). 3D geovisualization-definition and structures for the assessment of usefulness. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1, 129–134.
- Bleisch, S., Dykes, J., & Nebiker, S. (2008). Evaluating the effectiveness of representing numeric information through abstract graphics in 3D desktop virtual environments. *The Cartographic Journal*, 45, 216–226.
- Brewer, C. A., & Battenfield, B. P. (2007). Framing guidelines for multi-scale map design using databases at multiple resolutions. *Cartography and Geographic Information Science*, 34, 3–15.
- Bunch, R. L., & Lloyd, R. E. (2006). The cognitive load of geographic information. *The Professional Geographer*, 58, 209–220.
- Caquard, S., & Cartwright, W. (2014). Narrative cartography: From mapping stories to the narrative of maps and mapping. *The Cartographic Journal*, 51, 101–106.
- Cole, F., DeCarlo, D., Finkelstein, A., Kin, K., Morley, R. K., & Santella, A. (2006). Directing gaze in 3D models with stylized focus. In *17th Rendering Techniques 2006*.
- De Amicis, R., Conti, G., Simões, B., Lattuca, R., Tosi, N., Piffer, S., et al. (2009). Geo-visual analytics for urban design in the context of future internet. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 3, 55–63.
- Debiasi, A., Prandi, F., Conti, G., De Amicis, R., & Stojanović, R. (2013). Visual analytics tool for urban traffic simulation. In *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques. ICST* (pp. 51–56).
- Döllner, J., Baumann, K., & Buchholz, H. (2006). Virtual 3D city models as foundation of complex urban information spaces. In *11th international conference on Urban Planning and Spatial Development in the Information Society (REAL CORP)*, Essen, Germany (pp. 107–112).
- Döllner, J., & Buchholz, H. (2005a). Continuous level-of-detail modeling of buildings in 3D city models. In *Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems* (pp. 173–181). ACM.
- Döllner, J., & Buchholz, H. (2005b). *Non-photorealism in 3D geovirtual environments* (pp. 1–14). Las Vegas: Proceedings of AutoCarto.
- Döllner, J., Hagedorn, B., & Klimke, J. (2012). Server-based rendering of large 3D scenes for mobile devices using G-buffer cube maps. In *Proceedings of the 17th International Conference on 3D Web Technology* (pp. 97–100). ACM, Los Angeles, California.
- Döllner, J., & Kyprianidis, J. E. (2010). *Approaches to image abstraction for photorealistic depictions of virtual 3D models*. Springer.
- Döllner, J., & Walther, M. (2003). Real-time expressive rendering of city models. In *Seventh International Conference on Information Visualization* (pp. 245–250). IEEE.
- Dykes, J., Andrienko, G., Andrienko, N., Paelke, V., & Schiewe, J. (2010). Editorial–GeoVisualization and the digital city. *Computers, Environment and Urban Systems*, 34, 443–451.

- Dykes, J., Moore, K., & Fairbairn, D. (1999). From Chernoff to Imhof and beyond: VRML and cartography. In *Proceedings of the Fourth Symposium on Virtual Reality Modeling Language* (pp. 99–104). ACM.
- Fan, H. (2010). *Integration of time-dependent features within 3D city model* (p. 2010). München: Techn. Univ., Diss.
- Fan, H., & Meng, L. (2009). Automatic derivation of different levels of detail for 3D buildings modelled by CityGML. In *24th International Cartography Conference*, Santiago, Chile (pp. 15–21).
- Gershon, N. (1998). Visualization of an imperfect world. *Computer Graphics and Applications, IEEE*, 18, 43–45.
- Glander, T. (2013). *Multi-scale representations of virtual 3D city models*. Hasso-Plattner-Institute (HPI). Universität Potsdam.
- Glander, T., & Döllner, J. (2009). Abstract representations for interactive visualization of virtual 3D city models. *Computers, Environment and Urban Systems*, 33, 375–387.
- Gooch, A., Gooch, B., Shirley, P., & Cohen, E. (1998). A non-photorealistic lighting model for automatic technical illustration. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (pp. 447–452). ACM.
- Gooch, B., & Gooch, A. (2001). *Non-photorealistic rendering*. AK Peters/CRC Press.
- Griethe, H., & Schumann, H. (2006). The visualization of uncertain data: Methods and problems. In *SimVis* (pp. 143–156).
- Häberling, C., Bär, H., & Hurni, L. (2008). Proposed cartographic design principles for 3D maps: a contribution to an extended cartographic theory. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 43, 175–188.
- Hagedorn, B., Trapp, M., Glander, T., & Dollner, J. (2009). Towards an indoor level-of-detail model for route visualization. In *Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09* (pp. 692–697). IEEE.
- Hake, G., Grünreich, D., & Meng, L. (2002). *Kartographie. Visualisierung raum-zeitlicher Informationen* (8. Ausgabe). Berlin/New York: DeGruyter.
- Hayakawa, S. I. (1967). *Semantik: sprache in denken und handeln*. Verlag Darmstädter Blätter.
- Hermann, F., & Peissner, M. (2003). Usability Engineering für kartographische Visualisierungen-Methoden und Verfahren. *KN*, 2003(6), 260–265.
- Huang, Z., Feng, X., Xuan, W., & Chen, X. (2007). Causal relations among events and states in dynamic geographical phenomena. In *Geoinformatics 2007* (pp. 67531J-67531J-67538). International Society for Optics and Photonics.
- Jahnke, M. (2013). Nicht-photorealismus in der stadtmodellvisualisierung für mobile nutzungskontexte. Technische Universität München.
- Jahnke, M., Berger, T., & Krisp, J. (2011a). Nicht fotorealistische Darstellung von 3D-Stadtmodellen. *HMD Praxis der Wirtschaftsinformatik*, 48, 101–112.
- Jahnke, M., Krisp, J. M., & Kumke, H. (2011b). How many 3D city models are there?—A typological try. *The Cartographic Journal*, 48, 124–130.
- Jobst, M., & Döllner, J. (2008). *Better perception of 3D-spatial relations by viewport variations*. Springer.
- Kada, M. (2002). Automatic generalization of 3D building models. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34, 243–248.
- Klanten, R., & Losowsky, A. (2011). *Visual storytelling: Inspiring a new visual language*. Gestalten Berlin.
- Kolbe, T. H. (2009). *Representing and exchanging 3D city models with CityGML, 3D geo-information sciences* (pp. 15–31). Springer.
- Kumke, H. (2011). *Kartographische Anreicherung von Gebäudefassaden mit thermalen Bilddaten*. München, Technische Universität München, Dissertation, 2011.
- Lorenz, H., Trapp, M., Döllner, J., & Jobst, M. (2008). *Interactive multi-perspective views of virtual 3D landscape and city models* (pp. 301–321). The European Information Society. Springer.

- Löwner, M.-O., Benner, J., Gröger, G., & Häfele, K.-H. (2013). New concepts for structuring 3D city models—an extended level of detail concept for CityGML buildings. In *Computational Science and Its Applications—ICCSA 2013* (pp. 466–480). Springer.
- Maass, S., & Döllner, J. (2007). Embedded labels for line features in interactive 3D virtual environments. In *Proceedings of the 5th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa* (pp. 53–59). ACM.
- MacEachren, A. M. (1995). *How maps work: Representation, visualization, and design*. The Guilford Press.
- MacEachren, A. M., Robinson, A., Hopper, S., Gardner, S., Murray, R., Gahegan, M., et al. (2005). Visualizing geospatial information uncertainty: What we know and what we need to know. *Cartography and Geographic Information Science*, 32, 139–160.
- Mao, B., Ban, Y., & Harrie, L. (2009). A framework for generalization of 3D city models based on CityGML and X3D. *ISPRS Workshop on Quality, Scale and Analysis Aspects of Urban City Models*.
- Mayhew, D. J. (1999). *The usability engineering lifecycle: A practitioner's handbook for user interface design* (pp. 59–60). Interactions-New York.
- McMaster, R. B., & Shea, K. S. (1992). *Generalization in digital cartography*. DC: Association of American Geographers Washington.
- Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. (1995). *Augmented reality: A class of displays on the reality-virtuality continuum, Photonics for industrial applications* (pp. 282–292). International Society for Optics and Photonics.
- Murphy, C. E. (2014). Concise image maps—A design approach, Department of Cartography. Technische Universität München, Munich, p. 152.
- Pang, A. (2001). Visualizing uncertainty in geo-spatial data. In *Proceedings of the Workshop on the Intersections between Geospatial Information and Information Technology* (pp. 1–14).
- Pasewaldt, S., Semmo, A., Trapp, M., & Döllner, J. (2012). Towards comprehensible digital 3d maps. In M. Jobst (Ed.), *Service-Oriented Mapping 2012 (SOMAP2012)* (pp. 261–276). Wien: Jobstmedia Management Verlag.
- Pasewaldt, S., Trapp, M., & Döllner, J. (2011). Multiscale visualization of 3D geovirtual environments using view-dependent multi-perspective views. *Journal of WSCG*, 19, 111–118.
- Pegg, D. (2013). Design issues with 3D maps and the need for 3D cartographic design principles.
- Peters, S. (2014). Dynamics of spatially extended phenomena. Technische Universität München.
- Petrovič, D. (2003). Cartographic design in 3D maps. In *21st International Cartographic Conference (ICC)*, Durban, South Africa.
- Petrovič, D., & Mašera, P. (2005). Analysis of user's response on 3D cartographic presentations. In *Proceedings of the 22nd ICA International Cartographic Conference*, A Coruña, Spain.
- Pimenta, S., & Poovaiah, R. (2010). On defining visual narratives. *Design Thoughts*, 25–46.
- Roberts, L. (2014). The bulger case: A spatial story. *The Cartographic Journal*, 51(2), 41–151.
- Rückemann, C.-P. (2014). Knowledge processing for geosciences, volcanology, and spatial sciences employing universal classification. In *GEOProcessing 2014, The Sixth International Conference on Advanced Geographic Information Systems, Applications, and Services* (pp. 76–82).
- Segel, E., & Heer, J. (2010). Narrative visualization: Telling stories with data. *Visualization and Computer Graphics, IEEE Transactions on*, 16, 1139–1148.
- Semmo, A., Hildebrandt, D., Trapp, M., & Döllner, J. (2012a). Concepts for cartography-oriented visualization of virtual 3D city models. *Photogrammetrie-Fernerkundung-Geoinformation*, 2012, 455–465.
- Semmo, A., Trapp, M., Kyprianidis, J. E., & Döllner, J. (2012b). Interactive visualization of generalized virtual 3D city models using level-of-abstraction transitions. In *Computer Graphics Forum* (pp. 885–894). Wiley Online Library.
- Slocum, T. A., Blok, C., Jiang, B., Koussoulakou, A., Montello, D. R., Fuhrmann, S., et al. (2001). Cognitive and usability issues in geovisualization. *Cartography and Geographic Information Science*, 28, 61–75.

- Slocum, T. A., McMaster, R. B., Kessler, F. C., & Howard, H. H. (2009). *Thematic cartography and geovisualization*. NJ: Pearson Prentice Hall Upper Saddle River.
- Straumann, R. K., Çöltekin, A., & Andrienko, G. (2014). Towards (re) constructing narratives from georeferenced photographs through visual analytics. *The Cartographic Journal*, 51, 152–165.
- Strothotte, T., & Schlechtweg, S. (2002). *Non-photorealistic computer graphics: Modeling, rendering, and animation*. Elsevier.
- Tempfli, K., & Pilouk, M. (1996). Practicable photogrammetry for 3D-GIS. *International Archives of Photogrammetry and Remote Sensing*, 31, 859–867.
- Thomas, J. J., & Cook, K. A. (2005). *Illuminating the path: The research and development agenda for visual analytics*. IEEE Computer Society Press.
- Trapp, M., Glander, T., Buchholz, H., & Dolner, J. (2008). 3D generalization lenses for interactive focus + context visualization of virtual city models. In *12th International Conference Information Visualisation, 2008. IV'08* (pp. 356–361). IEEE.
- Vaaranemi, M., Freidank, M., & Westermann, R. (2013). Enhancing the visibility of labels in 3D navigation maps. In *Progress and New Trends in 3D Geoinformation Sciences* (pp. 23–40). Springer.
- Ware, C. (2010). *Visual thinking: For design*. Burlington, MA, USA: Elsevier Science.
- Waser, J., Konev, A., Sadransky, B., Horváth, Z., Ribičić, H., Carnecky, R., et al. (2014). Many plans: Multidimensional ensembles for visual decision support in flood management. In *Computer Graphics Forum* (pp. 281–290). Wiley Online Library.
- Willmott, J., Wright, L., Arnold, D. B., & Day, A. (2001). Rendering of large and complex urban environments for real time heritage reconstructions. In *Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Cultural Heritage* (pp. 111–120). ACM.
- Wilson, T. (2008). OGC KML. OGC Encoding Standard, Version 2.2.0, OGC Doc. No. 07-147r2, Open Geospatial Consortium.

Comparison of 2D & 3D Parameter-Based Models in Urban Fine Dust Distribution Modelling

Yahya Ghassoun and M.-O. Löwner

Abstract In the present study two Land Use Regression Models for the estimation of urban fine dust distribution were established and compared. The first model used 2D parameters derived from an Open Street Map project data (OSM) and the second model used 3D parameters derived from a CityGML-based 3D city model. Both models predict fine-dust concentrations by using urban morphological (2D resp. 3D) and additional semantic parameters. The models were applied to a 2 km² study area in Berlin, Germany. The 2D-LUR model explained 84 % of the variance of TNC for the full data set with root mean square error (RMSE) of 3284 cm⁻³ while the 3D-LUR explained 79 % of the variance with an RMSE of 3534 cm⁻³. Both models are capable to depict the spatial variation of TNC across the study area and showed relatively similar deviation from the measured TNC. The 3D-LUR needed less parameters than the 2D-LUR model. Furthermore, the semantic parameters (e.g. streets type) played a significant role in both models.

Keywords Fine dust distribution modelling • 3D city models • Land use regression modelling

1 Introduction and Problem Statement

Many epidemiological and toxicological studies have discussed the health effects of ultrafine particles (UFP) (WHO 2013). These studies have proved that the particulate air pollution in urban area is associated with significant impacts on human health

Y. Ghassoun (✉)

Institute for Geodesy and Photogrammetry, Technische Universität Braunschweig,
Braunschweig, Germany
e-mail: y.ghassoun@tu-bs.de

M.-O. Löwner

Institute for Geodesy and Geoinformation Science, Technische Universität Berlin,
Berlin, Germany
e-mail: m.loewner@tu-berlin.de

(Heal et al. 2012; HEI 2013; Zhang et al. 2015) especially, among children who are the most susceptible group in regard to particulate exposure compared to adults (Burtcher 2012). Ultrafine particles with diameter less than 0.1 microns often are the direct product of the combustion of fossil fuels by road transport (Geiser et al. 2005).

The UFP concentration can be estimated from measurements of the particle number size distribution. At street canyon or near traffic sites, the number of UFP generally accounts for the majority of total particle number concentrations, i.e., greater than 80–90 % (Morawska et al. 2008; Weber et al. 2013). The combustion source air pollution, especially from traffic has been considered as the most significant factor of premature death where numerous toxic materials produced by combustion processes are in the ultrafine size range (Jerrett 2011; Burtcher 2012). Therefore, detailed assessment of exposure by measurement and modelling of fine dust distribution is necessary in the field of urban planning, traffic management and city system modelling.

UFP concentrations are affected by different sources of combustion, secondary production pathways that change their number, shape, size and chemical composition (Sabaliaukas et al. 2015). The spatial distribution pattern of urban UFP concentrations are mostly affected by the local wind field and, therefore, by different factors of the urban complexity, i.e., the urban morphology that influences it.

Urban morphology can be analyzed concerning geometrical properties of street canyons (Vardoulakis et al. 2012), building density, alignment of streets towards the prevailing wind direction, and characteristics of crossing sections (Brand and Löwner 2014). Especially, the buildings structure is considered in addition of the meteorological factors the main parameters for modelling air ventilation (Wong et al. 2011). However, urban morphology has to be viewed as a 3D phenomenon.

Land use regression (LUR) models have been presented as a promising approach for the prediction of long-term, local-scale variation in traffic pollution and to obtain accurate, small scale air pollutant concentrations without a detailed pollutant emission inventory (Briggs et al. 2000; Brauer et al. 2003; Zhang et al. 2015; Ghassoun et al. 2015a). LUR models are multiple linear regression approaches that assume independent residuals and use GIS-based explanatory variables to predict pollutant concentrations at certain locations (Hoek et al. 2008; Mercer et al. 2011). They have been widely applied in cities of North America, Europe, and Asia (e.g. Arain et al. 2007; Kashima et al. 2009; Chen et al. 2010; Saraswat et al. 2013; Tang et al. 2013; Rivera et al. 2012; Abernethy et al. 2013). Different studies tried to enhance the LUR models by incorporating meteorological parameters (Arain et al. 2007; Chen et al. 2010; Kim and Guldmann 2011; Zhang et al. 2015; Li et al. 2015).

Few studies used 3D spatial data to enhance the representation of land use (or the urban morphology) and the dispersion field in LUR, such as using the 3D data of building, street canyons and porosity, i.e. the chance for the air to pass through a building block, in LUR modelling (Tang et al. 2013; Ghassoun et al. 2015b) or examining the influence of different heights from ground level on the predicted values of PM_{2.5} (Ho et al. 2015). 3D parameters not only exhibited an enhancement of the LUR models but also simplify the models by using less parameter than 2D model (Ghassoun et al. 2015b).

Today, 3D city models for semantically enriched virtual 3D city models are increasingly available due to the standardization processes like CityGML (Gröger et al. 2012). However, until today no comparative study has been performed to evaluate the benefit of 3D parameters in the development of LUR for fine dust distribution modelling in urban areas.

Here, two LUR models were established and compared, the first model used 2D parameters derived from an Open Street Map project data (OSM) and the second model used only 3D parameters derived from a CityGML-based 3D city model. Both models predict fine-dust concentrations by using urban morphological and semantic parameters.

2 Methods and Materials

2.1 Study Area and Data Source

The present study developed LUR models for 25 sites in an area of 1 * 2 km in the City of Berlin City Fig. 1. The study area is characterized by street canyons with different traffic intensities and different microenvironment. Site positions were chosen to cover the whole study area at which mobile measurements of particulate

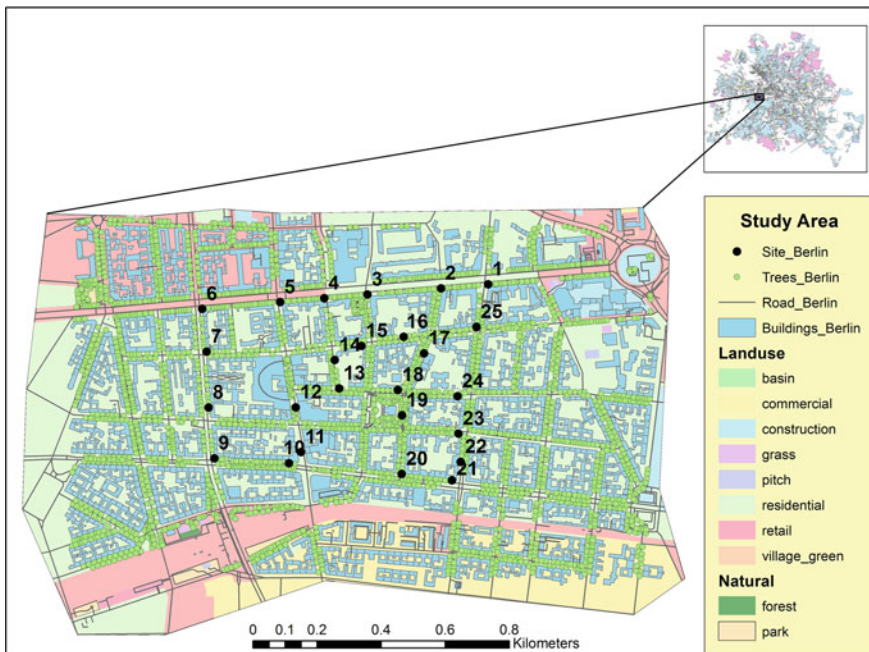


Fig. 1 Research area in “Berlin Mitte” and the 25 measurement locations

air pollutants were conducted. At each site, the average concentration of total number concentration were calculated over 1 min. the measurements were carried out during 6 campaigns in winter (January 2015) during stable weather conditions without rain and low wind speeds $<4 \text{ ms}^{-1}$. Total number concentrations (TNC) were measured with a hand-held particle counter device (TSI 3007). The TSI 3007 is a fully battery operated handheld sensor for the measurement of particle number concentrations. The instrument detects particles ranging in size from 10 to about 1000 nm. The concentration range is between 0 and 100,000 particles cm^{-3} .

To perform the parametrization of the urban morphology different data resources have been used in our study, Open Street Map and CityGML.

OpenStreetMap (OSM) has been used to extract 2D and semantically data. The latter went into both, the 2D-based and the 3D based model. OSM is one of the most well-known data of a collaborative mapping project and it receives a huge amount of contributions from across the world (Jokar et al. 2015). It provides an abundant data source for geospatial data update. Many studies have tested the quality assessment of OSM data according to its completeness, temporal accuracy, thematic accuracy and positional accuracy (Ming et al. 2013). However, OSM is a function of contributor activity. Therefore, many studies represented that the degree and nature of such activity shows significant spatial heterogeneity. A main problem of using OSM data is that we have not enough information about the people who collect these data or the patterns of data collection and the data are not complete and comprehensive (Haklay 2009). Haklay (2009) presented the satisfactory of using OSM for many application concerning the positional accuracy and completeness in major urban area and concluded that the OSM quality is beyond good enough. OSM data became not only a source for 2D data but also have been used to generate 3D city model by integrated the OSM data into the height information (Over et al. 2010). Here, OSM data was only used to get the information regarding the streets and land use information.

CityGML (Gröger et al. 2012) as an international standard of the Open Geospatial Consortium (OGC) has been used to extract 3D for the second LUR model. It is a common information model and encoding standard for the representation, storage, and exchange of virtual 3D city and landscape models. It provides 3D geometric representations next to concepts for the representation of their semantics and their relations. CityGML is commonly accepted in the field of 3D city models; the number of available city models and their applications has increased significantly in the last ten years (r.f. Löwner et al. 2013). Applications that rely on CityGML are e.g. the Energy Atlas of Berlin (Kaden and Kolbe 2013), noise simulation and mapping (Czerwinski et al. 2013) and, nowadays, fine dust distribution modelling (Ghassoun et al. 2015b). Here, CityGML data was used to extract the 3D information for the establishment of a 3D based model.

Volume of buildings have been calculated using an SQL script based on the migration script from 3DCityDB 2.x to 3DCityDB 3.x and explicitly adapted for the Berlin data (where thematically surface are used). The SQL-script cannot be applied without modification to other database with mostly only one roof surface for a whole building or no Ground- Wall- or RoofSurfaces is available. However,

data inconsistencies have been ignored in this study. The volume calculation achieved 98.23 % of the whole buildings and only 25 buildings out of 1412 were left out due to error in their geometry probably. Changing the tolerance value from 0.0005 given in the original 3DCityDB to 0.01 did not bring any improvement. Therefore, the remaining building's volumes were calculated out of their height and area information. In addition, the output table contains the information about the measured height, maximum and minimum height. The volume of the trees has also been extracted out of the 3D CityDB data. Therefore, Oracle SQL Developer (free software) was used to export the results for using them in developing the 3D-LUR model.

3 Model Developing

3.1 Geographical Parameters

In order to compare 2D and 3D parameter-based models in the field of urban fine dust distribution modelling, distinction has been made between three types of parameters. First, semantical parameters have been used to describe none geometrically properties of the urban system. In our research, semantical parameters represent the attributes of different type of streets (primary, secondary, etc.) in the study area within different radii buffer. Street types serve as a proxy for traffic intensity. Therefore, they do not stand for a geometrically property but for additionally information and are classified as semantically. Here, these parameters will be extracted out of OSM data and used in developing both, 2D-LUR and 3D-LUR (rf. Table 1). Hence, the two models developed just differ in terms of dimensions of the geometric parameters.

2D parameters are exclusively used for the development of the 2D based model that has been used to compare with the 3D based model. The potential 2D parameters were extracted from OSM data using different radii buffer around each measurement site to reflect their spatial influence on the air pollution concentrations around these sites. They are described in Table 1, also.

3D parameters are used to describe the morphology of the urban area in 3D and its impact on the fine dust distributions and urban ventilation Table 1. They are used for development of the 3D based model, only. 3D parameters have been extracted to build a model that really incorporates f.i. local wind field as a function of the built environment. Parameters are extracted from a CityGML-based database. To use City-GML-based database within GIS environment, Feature Manipulation Engine (FME) have been used to transfer the data into shapefiles and Oracle SQL Developer software to export their attributes and then the traditional LUR circular buffer have been used to extract the 3D parameters and they can be described as following (rf. Table 1):

Table 1 List of used spatial parameters and their source next to their extraction method in Esri’s ArcGIS 10.0

Category	Sub-categories	Buffer Radii [m]	Source	Methods
Semantical parameters	Length of different types of streets (Primary, Secondary, Tertiary, Residential, living)	50, 100 and 200	OSM Data	Using Esri’s Model Builder: generate buffer, intersect with the street, and summarize according the type of the street
2D parameters	Total area of building Total area of different land use	100, 200	OSM Data	Using Esri’s Model Builder: generate buffer, intersect with the buildings, and summarize according the type of the building
	Distance parameters			The distance to nearest primary roads
	Plan area ratio			$\lambda_p = A_p/A_T$
3D parameters	The height of the buildings (H) The ratio of height and width (H/W) Volumetric density (V_b) Volumetric trees density (V_{tree}) Porosity (P_{h-var}) Volumetrically averaged building height (H)	50, 100 and 200	CityGML	

- Height of the buildings adjacent to the street (H),
- Ratio of height and width of the street canyon (H/W),
- Volumetric density which describes the ratio of built volume and air (V_b),
- Volumetric tree density which describes the ratio of tree volume and the air (V_{tree}),
- Porosity, which is based on the input parameters of building volume, total areas and the height of the highest building within the buffer radii. It has calculated by the following equations of Burghardt (2014):

$$P_{h-var} = (AT * h_{UCL} - V) / AT * h_{UCL}$$

- Volumetrically averaged building height and it is calculated as following:

$$H = \sum_{i=1}^n V_i * h_i / \sum_{i=1}^n V_i$$

3.2 LUR Model Using 2D Parameters

A total of 22 2D parameters and 5 semantical parameters were included into the process of land use regression model and these parameters were grouped depending on the process impact on pollutant concentrations into three categories of emission, dilution and deposition.

In our study, a model building algorithm described by Henderson et al. (2007) was used to build the LUR model for the fine dust concentrations according to the following procedures:

1. All variables were ranked by the absolute strength of their correlation with the measured pollutant.
2. The highest-ranking variable in each sub-category was identified.
3. Afterwards the variables in each sub-category that are correlated (i.e. Pearson's $r \geq 0.6$) with the most highly ranked variables were eliminated to avoid autocorrelation and collinearity.
4. All remaining variables were implemented into robust linear regression models.
5. Hence, variables that were not significant at a 90 % confidence level or that had a coefficient with a counter intuitive sign were rejected.
6. Finally, the last two steps were repeated to convergence.

Before applying the aforementioned procedures, buffers of different radii were generated for each measured site using ESRI's ArcGIS 10.2. All the available data (streets, buildings and recreational area) were extracted and stored in order to use them in the process of LUR model. Buffer radii of 50, 100, and 200 m were used to derive the length of streets, whereas buffers radii of 100, 200, and 300 m were used to derive the building area. Buffer radii of 500 m were used to derive the area of recreational area. Specify the buffer radii reflects the scale of environmental processes appropriate for each variable. For example, effects of emission from road traffic are typically localized, so the buffer radii should be small. The effect of land use are often more extensive and more complex, therefore, larger buffer radii might be used.

The most significant parameters were selected and used in the multi regression model in order to build the models for fine dust concentrations. The LUR model was evaluated using leave-one-out cross validation (in which one observation is left out in each iteration and the model was rerun and then used to predict the excluded observation) to confirm the model fit.

3.3 LUR Model Using 3D Parameters

For the development of the 3D based model, the 3D parameters derived from CityGML-based database were extracted and transferred into ArcGIS. The 3D parameters, i.e. the height of the buildings (H), the ratio of height and width (H/W),

volumetric density (V_b), volumetric trees density (V_{tree}), porosity (P_{h-var}), and volumetrically averaged building height (H) were extracted within buffers of radii 50, 100 and 200 m for each measured site. Volumetric density was calculated as a ratio of built volume and air. Air volume was calculated as a cylinder. The top surface of this cylinder is generated as a TIN surface represents the height of the buildings lying within a buffer around the site (V_b). Porosity is one of the roughness parameters and it is a measure of how penetrable the area is for the airflow (Gàl and Unger 2009). It represents the correlation between the penetrable and impenetrable parts of an air layer over a certain area. It could be calculated as the ratio of the volume of the open air and the volume of urban canopy layer regardless the orientation. The volumetrically averaged building height can be calculated after determining the volumes and the heights of each building in each buffer. Trees are considered as passive controls in reduction of pedestrian exposure (Abhijith and Gokhale 2015). Therefore, volumetric trees density was calculated as ratio of trees volume and the air volume was calculated same in volumetric density.

In addition semantical information as a proxy for traffic intensities have been extracted from OSM data. This is the sum of length of different type of streets within LUR buffers was used. Then, all the 3D parameters were entered into multiple linear regression and same aforementioned LUR procedures were used to select the most significant variables and implemented them in building 3D model for fine dust concentrations.

4 Results

4.1 Measurements

The descriptive statistics of the measured and modeled TNC for the study area are presented in (Table 2). Mobile measurements at 25 spots were conducted to analyze the spatial variation of TNC across the study area.

Mean of measured TNC concentration was 23569 cm^{-3} in ranging from 14332 cm^{-3} at sampling spot 13 to 40972 cm^{-3} at sampling spot 9.

Generally, it is evident that the minimum concentrations were at the measurement spots close to park place while the area close to the major roads with high traffic intensity were characterized by highest concentrations.

Table 2 Descriptive statistics of measured TNC concentrations and TNC output from 2D-LUR and 3D-LUR models. Values in brackets represent corresponding sampling points

	Meas.	2D-LUR	3D-LUR
Mean (cm^{-3})	23569	23569	23292
SD (cm^{-3})	7148	6558	6262
Max (cm^{-3})	40972(9)	38016(9)	34579(9)
Min (cm^{-3})	14332(13)	14586(13)	14731(16)

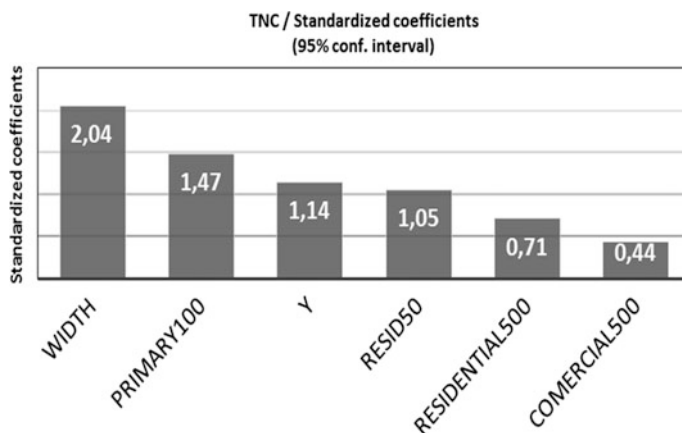


Fig. 2 Corresponding standardized regression coefficients for 2D-LUR model

4.2 Comparison of 2D & 3D Models

27 parameters were extracted from the data available and the final 2D-LUR model included parameters that incorporated site position, length of primary and residential streets, width of the streets and commercial and residential land use. Each parameter took the expected sign and matches the predefined direction of effect. Width of the streets and the length of primary streets (semantic parameters that represents high traffic intensity) within 100 m buffer resulted with higher TNC concentration, whereas commercial land use within 500 m buffer resulted in lower TNC concentrations. Figure 2 depicts the standardized regression coefficients and the influence explanatory parameters on the dependent parameter. The standardized coefficients of each parameter used in the final 2D-LUR and their errors are presented in Table 3.

Table 3 Model coefficients for 2D-LUR model

Standardized coefficients				
Source	Value	Standard error	t	Pr > t
Y	1.142	0.414	2.761	0.013
Primary100	1.480	0.578	2.560	0.020
Resid50	1.057	0.219	4.827	0.000
Comercial500	0.442	0.309	1.433	0.169
Width	-2.045	0.596	-3.429	0.003
Residential500	-0.718	0.191	-3.752	0.001
R²	0.84			

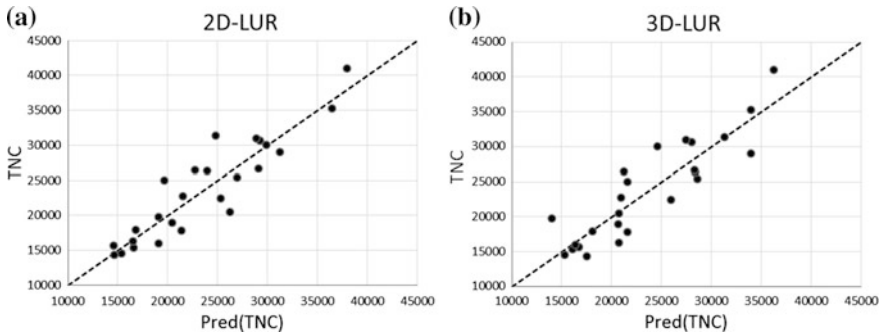


Fig. 3 Plot of residuals between measured and predicted TNC **a** 2D-LUR model, **b** 3D-LUR model

The 2D-LUR model expressed 84 % of the variance of the measured fine dust concentrations with root mean square errors (RMSE) of 3284 cm^{-3} . Figure 3a shows the plot of measured TNC concentrations against predicted one and the model gives the evidence for a linear trend with no outliers.

In contrast, 35 parameters were extracted from the 3D data and 5 semantical parameters available and the final 3D-LUR model included only the most significant parameters that incorporated length of secondary and tertiary streets, the ratio of height and width of the street canyon and volumetric trees density. Also, the parameters reflect the direction of their influence of the TNC. The secondary streets within 100 m buffer and the ratio of width and height of the streets resulted with higher TNC concentration, whereas tertiary streets and volumetric trees density within 50 m buffer resulted in lower TNC concentrations.

Figure 4 shows the standardized regression coefficients and the influence explanatory parameters on the dependent parameter. The standardized coefficients of each parameter used in the final 3D-LUR and their errors are presented in Table 4.

The 3D explanatory parameters show a high auto-correlation therefore, many parameters could not show their influence in predicting TNC concentrations and they have been deleted during the LUR processing.

The 3D-LUR model explained 79 % of the variance of the measured fine dust concentrations with root square error (RMSE) of 3534 cm^{-3} . Figure 3b shows the plot of measured TNC concentrations against predicted one and the 3D-LUR model gives the evidence for a linear trend with no outliers too. The 2D-LUR model showed slightly better performance in predicting the fine dust concentration in comparison of the 3D-LUR model. In both model, the semantical explanatory parameters show significant impact on the final 2D and 3D models.

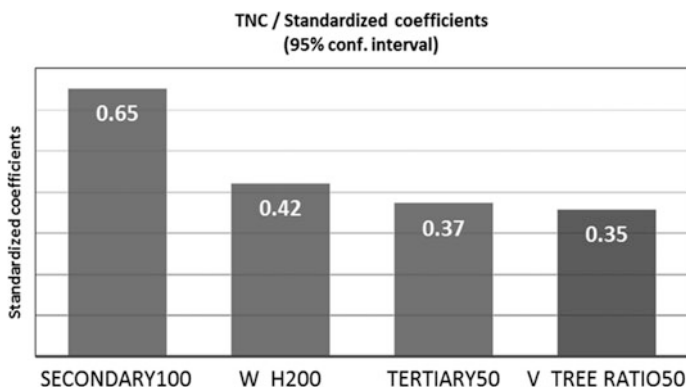


Fig. 4 Corresponding standardized regression coefficients for 3D-LUR model

Table 4 Model coefficients for 3D-LUR model

Standardized coefficients				
Source	Value	Standard error	t	Pr > t
W_H200	0.410	0.117	3.495	0.002
Tertiary50	0.313	0.105	2.973	0.008
V.tree ratio50	0.299	0.115	2.600	0.018
Secondary100	0.675	0.117	5.789	<0.0001
R²	0.79			

Standardized deviation for 2D-LUR accounts to 14 % and it was calculated by the ratio of the RMSE to the average measured TNC concentration across the study area. The visual errors were illustrated in Fig. 5a and it shows that 2D-LUR model predicts the TNC concentration very well in most of the sites and the large errors between measured and predicted TNC account on the site spots 11 and 3, where the blue area represent the pixels that have the maximum error between the measured and predicted TNC values.

The standardized deviation for 3D-LUR accounts to 15 %. 3D-LUR shows a very slight deviation in most of the measured sites comparing to 2D-LUR and Fig. 5b shows a large error of predicting TNC on the spots 1 and 14.

Mean absolute deviations between measured and modelled TNC concentrations were calculated for the 2D-LUR and 3D-LUR. Generally, 2D-LUR shows smaller deviation than 3D-LUR and is characterized by a deviation of 22 % while the mean absolute deviation for 3D-LUR turns out with 23 %. Leave-one-out cross validation was applied to validate both 2D-LUR and 3D-LUR models estimation. The deviation for 2D-LUR model accounts 17 % and it is relatively close to original model, while the deviation for the 3D-LUR model accounts to 18 %.

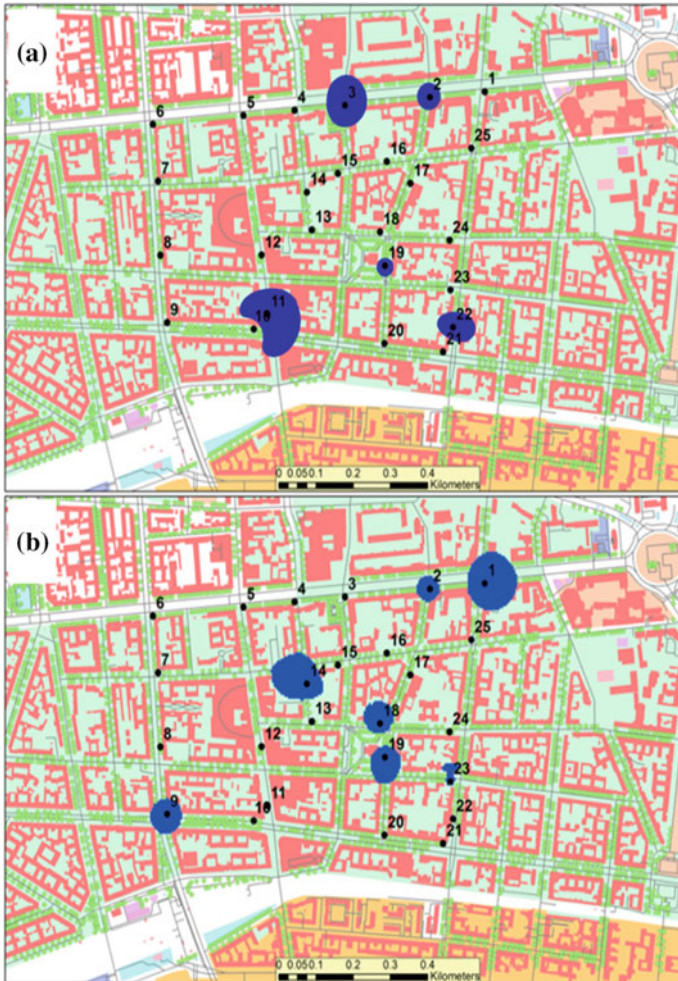


Fig. 5 The visual RMSE in each site station between the measured and predicted TNC concentration for **a** 2D-LUR model and **b** 3D-LUR model

Inverse distance weighted interpolation was carried out for the TNC data and the results indicate that both models generally show a similar spatial distribution of modeled TNC Fig. 6. The Highest TNC concentrations are distributed on the main road intersections where traffic lights and high traffic intensities.

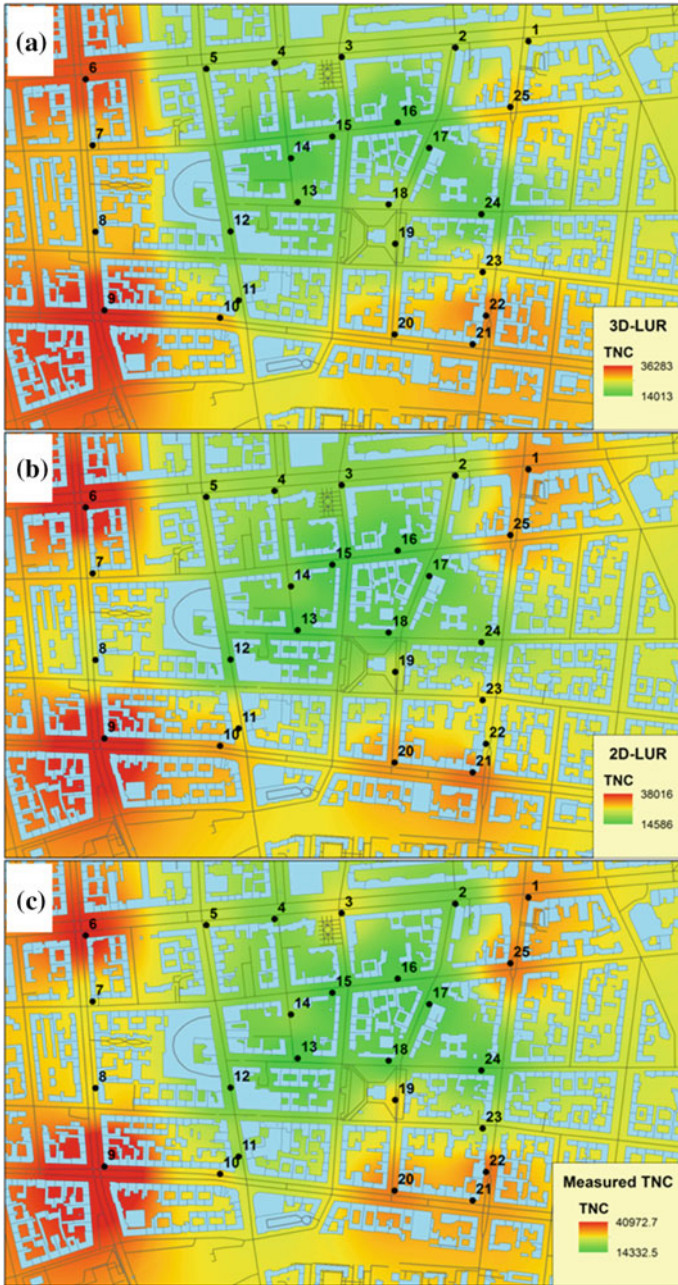


Fig. 6 IDW interpolation illustrates map of TNC concentration resulted from **a** 3D-LUR and **b** 2D-LUR models in comparison with **c** the measured TNC

5 Conclusion

To predict the TNC concentrations and their variation across the study area in Berlin, Germany three kinds of parameters were used in order to build an LUR models (the semantical parameters, 2D parameters and 3D parameters). Both models show approximately similar results and are able to depict the variance of TNC concentrations and the residuals for both models appear to behave randomly, it suggests that the model fits the data well. However, the 3D-LUR needs less parameters than the 2D-LUR and the parameters show similar significant values where the 2D-LUR parameters show big variance of influence on getting the best model. The high autocorrelation among the 3D parameters has a negative role on presenting the strength of some parameters in predicting TNC.

In this study the 3D parameters have been used as static parameters without taking into account the interaction between these parameters and the meteorological parameters such as the wind direction and its correlation with the porosity and that lead us to consider it as a dynamic parameters. We believe that the use of this approach can result an enhancement of the traditional LUR model or the LUR model that use 3D parameters. The LUR is temporally static and trained by urban morphology parameters.

As further work, we believe that LUR can be used with meteorological data or with spatial parameters that interacts with the meteorological data in order to make it dynamic. Also, the combination of traditional LUR and the urban ventilation are our promising approach for enhancing the model.

Acknowledgment We acknowledge the support of Stephan Weber from the institute of geocology of the Technische Universität Braunschweig for the supply of the measurement device that was used to perform this study.

References

- Abemethy, R. C., Allen, R. W., McKendry, I. G., & Brauer, M. A. (2013). Land use regression model for ultrafine particles in vancouver, Canada. *Environmental Science and Technology*, 47, 5217–5225.
- Abhijith, K. V., & Gokhale, S. (2015). Passive control potentials of trees and on-street parked cars in reduction of air pollution exposure in urban street canyons. *Environmental Pollution*, 204, 99–108.
- Arain, A. M., Blair, R., Finkelstein, N., Brook, R. J., Sahsuvaroglu, T., Beckerman, B., et al. (2007). The use of the wind fields in a land use regression model to predict air pollution concentrations for health exposure studies. *Atmospheric Environment*, 41, 3453–3464.
- Burghardt, R. (2014). Development of an ArcGIS extension to model urban climate factors.
- Brand, L., & Löwner, M.-O. (2014). Parametrisierung und Identifikation urbaner Straßenkreuzungen im Kontext der Feinstaubmodellierung. Parameterization an identification of street crossings in the context of fine dust modelling. Proceedings of the joint meeting of DGfK, DGPF, GfGI, and GiN, 26–28/03/2014, Hamburg, Germany.
- Brauer, M., Hoek, G., van Vliet, P., Meliefste, K., Fischer, P., Gehring, U., et al. (2003). Estimating long-term average particulate air pollution concentrations: Application of traffic indicators and geographic information systems. *Epidemiology*, 14, 228–239.

- Briggs, D. J., de Hough, C., Gulliver, J., Wills, J., Elliott, P., Kingham, S., et al. (2000). A regression-based method for mapping traffic-related air pollution: Application and testing in four contrasting urban environments. *Science of the Total Environment*, 253, 151–167.
- Burtscher, H., & Schüepp, K. (2012). The occurrence of ultrafine particles in the specific environment of children. *Paediatric Respiratory Reviews*, 13, 89–94.
- Chen, L., Du, S., Bei, Z., Kong, S., You, Y., & Han, B., et al. (2010). Application of land use regression for estimating concentrations of major outdoor air pollutants in Jinan, China. *Journal of Zhejiang University Science A (Appl Phys & Eng)*, 11, 857–867.
- Czerwinski, A., Gröger, G., Reichert, S., & Plümer, L. (2013). Qualitätssicherung einer 3D-GDI - EU-Umgebungslärmkartierung Stufe 2 in NRW. Quality management of a 3D-SDI – phase 2 of the EU ambient noise mapping in NRW. *Zeitschrift für Geodäsie, Geoinformation und Landmanagement*.
- Gál, T., & Unger, J. (2009). Detection of ventilation paths using high resolution roughness parameter mapping in a large urban area. *Building and Environment*, 44, 198–206.
- Geiser, M., Rothen-Rutishauser, B., Kapp, N., Schürch, S., Kreyling, W., & Schulz, H., et al. (2005). Ultrafine particles cross cellular membranes by nonphagocytic mechanisms in lungs and in cultured cells. *Environmental Health Perspectives*, 113, 1555–1560.
- Ghassoun, Y., Ruths, M., Löwner, M.-O., & Weber, S. (2015a). Intra-urban variation of ultrafine particles as evaluated by process related land use and pollutant driven regression modelling. *Science of the Total Environment*.
- Ghassoun, Y., Löwner, M.-O., & Weber, S. (2015b). Exploring the Benefits of 3D city Models in the Field of Urban Particles Distribution Modelling—A Comparison of Model Results. Breunig, M., Al-Doori, M., Butwilowski, E., Kuper, P. V., Benner, J., & Haefele, K.H. (2015) 3D Geoinformation Science, The 529 Selected Papers of the 3D GeoInfo, Lecture Notes in *Geoinformation and Cartography*, 530, 193–205.
- Gröger, G., Kolbe, T. H., Nagel, C., Häfele, K.-H. (2012). OGC City Geography Markup Language (CityGML) encoding standard, version 2.0, OGC Doc No. 12–019, Open Geospatial Consortium.
- Haklay, M. (2009). Beyond good enough? Spatial data quality and OpenStreetMap data. In: State of the Map Conference 2009, Amsterdam. http://de.slideshare.net/mukih/beyond-good-enough-spatial-data-quality-and-openstreetmap-data?tid=1f655291-2d6c-48ce-9ced-1af8456dc713&v=qf1&b=&from_search=12 Published on 11. Juli 2009.
- Heal, M. R., Kumar, P., & Harrison, R. M. (2012). Particles, air quality, policy and health. *Chemical Society Reviews*, 41, 6606–6630.
- HEI. (2013). Review Panel on Ultrafine Particles—Understanding the Health Effects of Ambient Ultrafine Particles. HEI Perspectives 3. Health Effects Institute, Boston, MA.
- Henderson, S., Beckerman, B., Jerrett, M., Brauer, M. (2007). Application of land use regression to estimate long-term concentrations of traffic-related nitrogen oxides and fine particulate matter. *Environmental Science and Technology*, 41, 2422–2428.
- Ho, C.-C., Chan, C.-C., Cho, C.-W., Lin, H.-I., Lee, J.-H., Wu, C.-F. (2015). Land use regression modeling with vertical distribution measurements for fine particulate matter and elements in an urban area. *Atmospheric Environment*, 104, 256–263.
- Hoek, G., Beelen, R., de Hoogh, K., Vienneau, D., Gulliver, J., Fischer, P., et al. (2008). A review of land-use regression models to assess spatial variation of outdoor air pollution. *Atmospheric Environment*, 42, 7561–7578.
- Jerrett, M. (2011). Spatiotemporal Analysis of Air Pollution and Mortality in California Based on the American Cancer Society Cohort 2011; Final Report.
- Jokar Arsanjani, J., Mooney, P., Helbich, M., & Zipf, A. (2015). An exploration of future of the contributions to OpenStreetMap and development of a Contribution Index. *Transactions in GIS*, volume and issue pending, pp. pending. Wiley. doi:10.1111/tgis.12139.
- Kaden, R., & Kolbe, T. H. (2013). City-Wide Total Energy Demand Estimation of Buildings using Semantic 3D City Models and Statistical Data. Kongress-/ Buchtitel: Proceeding of the 8th International 3D GeoInfo Conference. Band/ Teilband: II-2/W1. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 2013.

- Kashima, S., Yorifuji, T., Tsuda, T., & Doi, H. (2009). Application of land use regression to regulatory air quality data in Japan. *Science of the Total Environment*, 407, 3055–3062.
- Kim, Y., & Goldmann, J.-M. (2011). Impact of traffic flows and wind directions on air pollution concentrations in Seoul. *Korea. Atmospheric Environment*, 45, 2803–2810.
- Li, X., Liu, W., Chen, Z., Zeng, G., Hu, C., León, T., et al. (2015). The application of semicircular-buffer-based land use regression models incorporating wind direction in predicting quarterly NO₂ and PM₁₀ concentrations. *Atmospheric Environment*, 103, 18–24.
- Löwner, M.-O., Casper, E., Becker, T., Benner, J., Gröger, G., Gruber, U., et al. (2013). CityGML 2.0 – ein internationaler Standard für 3D-Stadtmodelle, Teil 2: CityGML in der Praxis. CityGML 2.0 – an international standard for 3D city models, part 2: CityGML in practice. *Zeitschrift für Geodäsie, Geoinformation und Landmanagement*, 2, 131–143.
- Mercer, D. L., Szpiro, A. A., Sheppard, L., Lindström, J., Adar, S., Allen, R., et al. (2011). Comparing universal kriging and land-use regression for predicting concentrations of gaseous oxides of nitrogen (NO_x) for the Multi-Ethnic Study of Atherosclerosis and Air Pollution (MESA Air). *Atmospheric Environment*, 45, 4412–4420.
- Ming, W., Qingquan, L., Qingwu, H., & Meng, Z. (2013). Quality analysis of open street map data. International archives of the photogrammetry, remote sensing and spatial information sciences, Volume XL-2/W1, 2013. In: *8th International Symposium on Spatial Data Quality*, 30 May-1 June 2013, Hong Kong.
- Morawska, L., Ristovski, Z., Jayaratne, E. R., Keogh, D. U., & Ling, X. (2008). Ambient nano and ultrafine particles from motor vehicle emissions: Characteristics, ambient processing and implications on human exposure. *Atmospheric Environment*, 42, 8113–8138.
- Over, M., Schilling, A., Neubauer, S., & Zipf, A. (2010). generating web-based 3D city models from OpenStreetMap: The current situation in Germany. *Computers, Environment and Urban Systems*, 34, 496–507.
- Rivera, M., Basagaña, X., Aguilera, I., Agis, D., Bouso, L., Foraster, M., et al. (2012). Spatial distribution of ultrafine particles in urban settings: A land use regression model. *Atmospheric Environment*, 54, 657–666.
- Sabaliaukas, K., Jeong, C.-H., Yao, X., Reali, C., Sun, T., & Evans, G. (2015). Development of a Land-use Regression Model for Ultrafine Particles in Toronto. *Canada. Atmospheric Environment*,. doi:10.1016/j.atmosenv.2015.02.018.
- Saraswat, A., Apte, S. J., Kandlikar, M., Brauer, M., Henderson, B. S., & Marshall, D. J. (2013). Spatiotemporal Land Use Regression Models of Fine, Ultrafine, and Black Carbon Particulate Matter in New Delhi. *India. Environ. Sci. Technol.*, 47, 12903–12911.
- Tang, R., Blangiardo, M., & Gulliver, J. (2013). Using Building Heights and Street Configuration to Enhance Intraurban PM₁₀, NO_x, and NO₂ Land Use Regression Models. *Environmental Science and Technology*, 47, 11643–11650.
- Vardoulakis, S., Gonzalez-Flesca, N., & Fisher, B. E. A. (2012). Assessment of traffic-related air pollution in two street canyons in Paris: implications for exposure studies. *Atmospheric Environment*, 36, 1025–1039.
- WHO. (2013). Review of evidence on health aspects of air pollution—REVIHAAP Project—technical report. World Health Organization—Regional office for Europe, Copenhagen p. 300.
- Wong, M. S., Nichol, J. E., NG, E. Y. Y., Guilbert, E., Kwok, K. H., & To, P. H., et al. (2011). GIS Techniques for mapping urban ventilation, using frontal area index and least cost path analysis. *Landscape and Urban Planning*, 102(4), 245–253.
- Weber, S., Kordowski, K., & Kuttler, W. (2013). Variability of particle number concentration and particle size dynamics in an urban street canyon under different meteorological conditions. *Science of the Total Environment*, 449, 102–114.
- Zhang, J. J. Y., Sun, L., Barrett, O., Bertazzon, S., Underwood, F. E., & Johnson, M. (2015). Development of land-use regression models for metals associated with airborne particulate matter in a North American city. *Atmospheric Environment*, 106, 165–177.

Investigating Semantic Functionality of 3D Geometry for Land Administration

George Floros, Eva Tsiliakou, Dimitrios Kitsakis, Ioannis Pispidikis and Efi Dimopoulou

Abstract Significance of semantic data during the recent years is growing. This trend, combined with facilitation of new 3D object modeling has led to semantically enriched 3D models, serving various applications where relations between objects' components and their environment need to be stored and presented. In the field of Land Administration, semantics can greatly contribute to optimize land management and land policies. Integration of semantics to 3D building models is currently achieved through two differently structured models: semantic-oriented CityGML and structural-oriented BIM/IFC. Integration of the semantic information of each model is still an object of intense research worldwide. In this paper, a 3D building model designed in SketchUp Pro software was transformed using FME software to a CityGML file; land use features were assigned to the model and attribute queries were executed in order to check the exported models' functionality in terms of semantics.

Keywords 3D modeling · Land use · SketchUp Pro · CityGML · Attribute query

G. Floros (✉) · E. Tsiliakou · D. Kitsakis · I. Pispidikis · E. Dimopoulou
School of Rural and Surveying Engineering, National Technical University of Athens,
9 Iroon Polytechniou Str, 15780 Zografou, Greece
e-mail: flwrosg@gmail.com

E. Tsiliakou
e-mail: eva.tsiliakos@gmail.com

D. Kitsakis
e-mail: dimskit@yahoo.gr

I. Pispidikis
e-mail: pispidikisj@yahoo.gr

E. Dimopoulou
e-mail: efi@survey.ntua.gr

1 Introduction

Semantics have gradually attracted international scientific interest due to their ability of storing data that describe relations between different object parts and their environment (Diakit  et al. 2014). Therefore, semantic based modeling has grown very popular internationally, incorporating a variety of applications of different scientific fields including energy applications, urban planning, indoor navigation, noise propagation simulation and mapping, disaster management and homeland security, cultural heritage, water management, environmental and real time simulations (Groeger and Plumer 2012). Semantic modeling is also promising for depicting relations between legal and physical space which is required to 3D Cadastre applications, where formal definition of 3D space and its containing elements is still an abstract concept, while volumetric parcels are not conceivable in reality but are established via connections to physical objects (Aien et al. 2013). 3D models' semantic enrichment allows for direct correlation between legal and physical property, improving the accuracy that legal spaces' volumes or locations are defined (Dimopoulou et al. 2014a). In the field of Land Administration, different modeling approaches such as IFC, a Building Information Modeling (BIM) standard or BIM ready software along with CityGML are exploited to attribute semantic data to 3D constructions or 3D city models. CityGML allows by the Application Domain Extension (ADE) to create extensions to the schema, to deal with special applications in 3D city modeling.  ağdaş (2013) exploits these capabilities of semantically rich CityGML information model along with its versatility to extend data features, to develop an ADE for immovable property taxation, overcoming the fact that CityGML does not model legal and administrative objects. However, each modeling approach serves different purposes (Cheng et al. 2013) while semantic description of buildings and their parts does not equal to semantic relationship between the buildings and their interior real property objects (Hu 2008). Attribution of semantic data becomes more important to the exploitation of 3D objects' models given that existing 3D modeling techniques have facilitated generation of high accuracy geometrical 3D object models. Literature provides a variety of techniques that are developed towards this approach, which however serve different objectives and cannot fully accommodate emerging issues. According to literature research, impediments of 3D objects' semantic data enrichment may be summarized as follows:

- Existing 3D object geometries created using CAAD systems lack semantic information. (Nagel et al. 2009).
- 3D data acquired from TLS/airborne laser points and images only pertain objects' exterior (Diakit  et al. 2014), while do not include high level semantics (Xiong and Huber 2013). Especially in case of already built constructions, 3D acquisition of non visible components such as slabs or pipes cannot be achieved (Nagel et al. 2009).

- Data inconsistency between BIM and CityGML data (Diakité et al. 2014), as BIM models represent the built environment as designed rather than observed (Kolbe and Plümer 2004).
- Preservation of consistency between geometry and topology of spatial data (Groeger and Plümer 2009).
- Automation of transformation procedure between different data types as presented by Nagel et al. (2009).
- Preservation of Level of Detail (LoD) during transformation process. To this, issues of generalization of objects' components also need to be considered (Geiger et al. 2014).

Topology issues also need to be considered since topology constitutes the basis for the sufficient support of correct semantics. Topology issues are analyzed in Kolbe (2009). The author notes that “solids of adjacent objects like BuildingParts must touch but their interiors are not allowed to permeate each other, because space can only be occupied by one physical object”. Different frameworks for the representation of 3D topology have been presented, most of them proposing a geometric-topological structure where coordinates are stored only within the nodes or the points associated with the nodes. Higher dimensional primitives are then constructed by connecting primitives of lower dimensions. Kolbe (2009) counterposes reusing common wall surfaces by providing the definition of the surface geometry inline within the specification of the solid geometry (bounded by a composite surface) of either building parts or the building itself. In the representation of the other solid this surface is then included by reference (and not by value) which creates the connection between both solids. The LADM also provides a conceptual description for a land administration system, including a 3D topology spatial profile. In case of the 3D topology representation, a 3D boundary face has plus/minus information included in the association to a 3D spatial unit (Dimopoulou and Elia 2012). Ying et al. (2012) address topology by introducing the real 3D geometric primitive “3D BODY” and two feature classes (3D land/legal space and 3D construction/building space) that are inherited from class “3D Parcel”. The topological elements are hierarchically interrelated, and (n-1)-dimensional geometric primitives are used as boundary of n-dimensional geometric primitives. Zhao et al. (2011) focus on the topology checking among polyhedra in 3D GIS systems stating this is a technology problem, which should be solved with help of computational geometry, topology rules, the Euler–Poincaré formula etc.

Although various tools of conversion between BIM/IFC to CityGML and vice versa have been developed, e.g. BIMserver, KIT IFCExplorer and Safe Software FME, none of the converters is currently capable of automatically creating valid geometries nor fully correct semantics, especially when high levels of detail are involved (Donkers 2013). Several approaches towards this direction have been implemented by a number of researchers internationally. Isikdag and Zlatanova (2009) define a framework automating generation of buildings from BIM to CityGML, claiming that it is possible to define rules for geometrical transformation and facilitation of semantic matching from IFC to CityGML models, De Laat and

van Berlo (2011) describe the development of GeoBIM extension on CityGML for IFC data, while El-Mekawy et al. (2011) proposed a unified building model (UBM) for integration of IFC and CityGML, allowing bilateral transformation between the two models. Dimopoulou et al. (2014b) investigate integration and interoperability between procedural modeling techniques and BIM-ready software within the CityGML framework from a semantic viewpoint using ESRI CityEngine environment and Trimble SketchUp Pro software to create 3D building models and evaluate modeling techniques. This paper aims to investigate semantic relations for Land Administration purposes through attribute querying of a building's indoor space. The paper is structured as follows: In Sect. 2 the role of semantics in Land Administration is presented, through exploitation of CityGML and BIM models. Sect. 3 describes exploitation of 3D geodatabases for 3D modeling applications specifying to integration of BIM with CityGML models and IFC with CityGML models focusing on urban applications. In Sect. 4, a case study is examined involving a 3D building model modeled in Trimble SketchUp and transformed into CityGML via FME software, in which land use features were assigned and attribute queries were executed. The paper ends with discussion and concluding remarks on Sect. 6.

2 Semantics in Land Administration

Semantic enrichment and spatio-semantic coherence could assist the integration of physical objects with their relevant cadastral attributes within robust cadastral models, although it has not been yet applied broadly for cadastral applications or the land administration domain. Evidently interoperability amongst data has triggered the need for “semantically enriched information in 3D cadastral data models meaning enriching the content and context of the available data by categorizing or classifying data in relationship to other data” as explained in Aien et al. (2013). Guo et al. (2012) argue that “semantic data in the field of land administration are used to regulate and coordinate relationships among people and property under a given legal system”. Undoubtedly, inconsistencies between the spatial extension of physical structures such as building parts and the imperceptible extension of legal rights imposed on buildings or parcels are tackled via data semantic enrichment. Guo et al. (2012) explain that current 3D GIS models capture 3D objects and their spatial and physical features, but not their semantics. One might claim though that this practice is ambiguous when referring to cadastral data models; that is registered 3D objects are defined by physical structures such as walls, floors and are described by spatial and topological relations. Actual physical boundaries distinguish the extent of RRRs within a single building or even a room, or according to (Isikdag et al. 2013) the boundary of a parcel coincides with a physical real world object. El-Mekawy et al. (2014) also claim that a 3D legal boundary surface follows the outer surface of a building in which the legal basic property unit is located.

2.1 *Semantics—CityGML*

Semantic modeling approaches along with the appliance of 3D geometry and topology of real-world objects is realized via CityGML, which constitutes the first 3D semantic standard not “only representing the shape and graphical appearance of city models but specifically addressing object semantics” (Kolbe 2009). CityGML involves a very interesting concept employed mainly for visualization efficiency, the LoD, which differs from the LoDs used in Computer Graphics. The latter facilitates the LoD concept for speeding up the rendering process of holistic 3D city models in terms of visualization, while CityGML comprises five discrete Levels of Details starting from LoD0 to LoD4 (indoor), not purely geometrical but extended to semantics as well; with increasing LoD, the semantic richness increases accordingly (Kolbe 2009). The most significant component of this model is the Building module, facilitating the depiction of buildings and their components in terms of geometry, topology and semantics.

2.2 *Semantics—BIM*

Besides CityGML, Building Information Models (BIM) as a digital version of all the substantial and functional features of a building through its entire life cycle (Isikdag et al. 2013), refer to sophisticated geometric and semantic representations of the building parts. According to Isikdag et al. (2013) each BIM building element contains comprehensive property sets on the elements’ functions or properties, while the “large set of utility elements embedded or located inside the building in 3D comprise detailed semantics”. An important difference between IFC and CityGML is related with the use of different concepts for the same semantic objects. For example, IFC addresses the buildings’ construction or design of buildings and defines construction elements such as wall, etc., while CityGML describes “how buildings are observed or used” (Nagel et al. 2009), thus providing definitions of rooms and walls (Gröger and Plümer 2012). The creation of a synergy between the strong (technology) parts of both worlds” by integrating CityGML and IFC (De Laat and van Berlo 2011) is highly recommended. Utilizing CityGML or IFC for cadastral applications, differences between geometric locations are identified; that is because BIM or CityGML are characterized by thickness information that may “be depicted by physical walls”, while cadastral representations are limited to “linear legal boundaries” (Ying et al. 2012).

2.3 Semantics-Coherence

Current trends focus on the semantic enrichment of distinctive city objects or 3D geometries which can be decomposed into their structural elements including attributes and their correlations, also addressing spatio-semantics coherence issues even for complex building models. Stadler and Kolbe (2007) discuss the spatio-semantic coherence of 3D city models with special focus on virtual 3D city models and the semantic data model CityGML. Coherence refers to solid and consistent relations between spatial and semantic features, established solely in case of structural similarity, meaning that if semantic and geometric aggregations reveal the same structure, they are considered coherent (Stadler and Kolbe 2007). The coherent thematic and spatial configuration of objects, secures that each complex (or not) geometric entity is assigned a specific semantic component, since semantics diminish ambiguities for geometric amalgamation. CityGML supports spatio-semantic coherence, since geometry entities are assigned thematic attributes while simultaneously semantic entities are accompanied by spatial information (e.g. location etc.). Spatio-semantic coherence could assist the integration of legal and physical objects within robust cadastral models (Dimopoulou et al. 2015). Table 1 summarises the key features of currently used 3D standards along with their level of compatibility to each feature according to Zlatanova et al. (2012).

3 3D Geodatabase for 3D Modeling

“3D reality needs 3D design, engineering and analyses” while modeling cities, requires a “more holistic approach to creating, building and managing infrastructure” (de Vries and Zlatanova 2011). This approach is feasible through 3D

Table 1 Comparison of 3D standards. *Source* Zlatanova et al. (2012)

Standard/criterion	X3D	KML	COLLADA	CityGML	DXF	SHP
Geometry	++	+	++	+	++	+
Topology	0	-	+	+	-	-
Texture	++	0	++	+	-	-
LOD	+	-	-	+	-	-
Objects	+	-	-	+	0	+
Semantic	0	0	0	++	+	+
Attributes	0	0	-	+	-	+
XML based	+	-	-	+	-	-
Web	++	++	+	+	-	-
Georeferencing	+	+	-	+	+	+
Acceptance	0	++	+	+	++	++

(- not supported; 0 basic; + supported; ++ extended support)

modeling and visualization, while support systems such as “geo-databases may serve as platforms to integrate 2D maps, 3D geo-scientific models, and other geo-referenced data” (Breunig and Zlatanova 2011). For example, the efficient storage of CityGML data requires both carefully optimized database schemas and data access tools (Stadler et al. 2009). El-Mekawy (2010) defines two types of 3D city models design which are usually “used for maximum level of detail in the architecture, engineering and construction (AEC)” and real world models which “are geospatial information systems representing spatial objects in GIS applications”. A 3D cadastral object is a synthesis of geometry, attributes and social and legal semantics, which is built processing 3D object construction, topological reconstruction and semantic information joining and spatial query and analysis (Ying et al. 2011) comprising both the legal space and the physical component, and may be physically represented by the Building Module of CityGML. However, according to Löwner et al. (2013) the current Level of Detail concept seems to be insufficient, since “the interior structure of a building can only be modeled when, simultaneously, there is a geometrically exact model for the exterior shell of the building, thus limiting the application of indoor models”.

3.1 Integration of CityGML and BIM Models

Based on the above, it is evident that both CityGML and IFC are flexible data models that aim at spatio-semantic coherence while enabling the visualization of 3D city models at different levels of geometric and semantic complexity (Stadler and Kolbe 2007). Isikdag et al. (2013) argue that the most evident obstacle preventing the employment of these models in real life applications have been deficiencies in “appropriate and applicable” representations of building geometry and thorough semantics for indoors, since the models developed are either too composite to query, characterized by complex geometric representations or not defined with sufficient semantics for sustaining indoor navigation. In their study, defined a new BIM based model (BO-IDM) for facilitating indoor navigation by introducing a new BIM Oriented Modeling methodology. Brown et al. (2013) discuss the concept of Topographic Space, defining it as a basic element of indoor navigation, entailing the buildings’ interior structure as well as the semantic decomposition into building elements (e.g. rooms and storeys). De Laat and van Berlo (2011) introduce Geo-BIM, which is a CityGML extension employed to obtain IFC semantic information data into a GIS framework, via a conversion process of IFC to CityGML implemented in the open source Building Information Modelserver. Zhu et al. (2011) conclude that the development of 3D city models accompanied by semantics has become a consensus and present their CityGML based semantic model, which supports the concept of Spaces (Stair, Corridor) for indoor navigation purposes.

4 Case Study

4.1 Workflow

The 3D building model was designed in Trimble SketchUp 2015. The boundaries of the entire modeling area were digitized based on the corresponding true ortho-photo using AutoCAD 2015. Digitized boundaries (in .dwg format) were imported into SketchUp. Interior and exterior parts of the building were designed based on the architectural plans forming the entire 3D building model. The final model as seen in Fig. 2, was saved as a distinct .skp file, and imported in FME Workbench producing a single GML file by using key transformers. The final GML file is viewed with FME Data Inspector module (Fig. 3). In order to check the 3D model's functionality in terms of semantics and to execute queries based on the building's CityGML properties a database (utilising postgresQL) was created and linked to the CityGML database through importer/exporter software, produced by 3DCityDB. The CityGML file was connected to this database, where various queries related to Land Use or building semantics are executed.

4.2 Trimble SketchUp

Trimble SketchUp is a user-friendly software for the design of 3D models; it provides full ability for the user, to design buildings, including their outer and inner parts. Within Sketchup, several design methods can be used, either through solids or surfaces and floors, depending mostly on user's requirements (Dimopoulou et al. 2014) (Fig. 1).

4.2.1 SketchUp as BIM Ready Software

SketchUp software allows for 3D building modeling comprising features that are able to classify objects and export files according to common Building Information Modeling (BIM) standards, constituting a BIM ready tool as presented below:

1. Sketchup provides the component feature along with geometry grouping when designing the model to form a component e.g., stairs or furniture. Different floors can also be manipulated as components, since each one can be geometrically grouped, thus generating distinctive geometries for each floor (as applied in this paper). These components can be moved and copied within the model, without affecting the rest of the model's geometry.
2. Sketchup provides the shadowing tool, which is a very useful BIM-like tool. The ability to analyze daylight on a project is a functional characteristic and very applicable for urban planning.

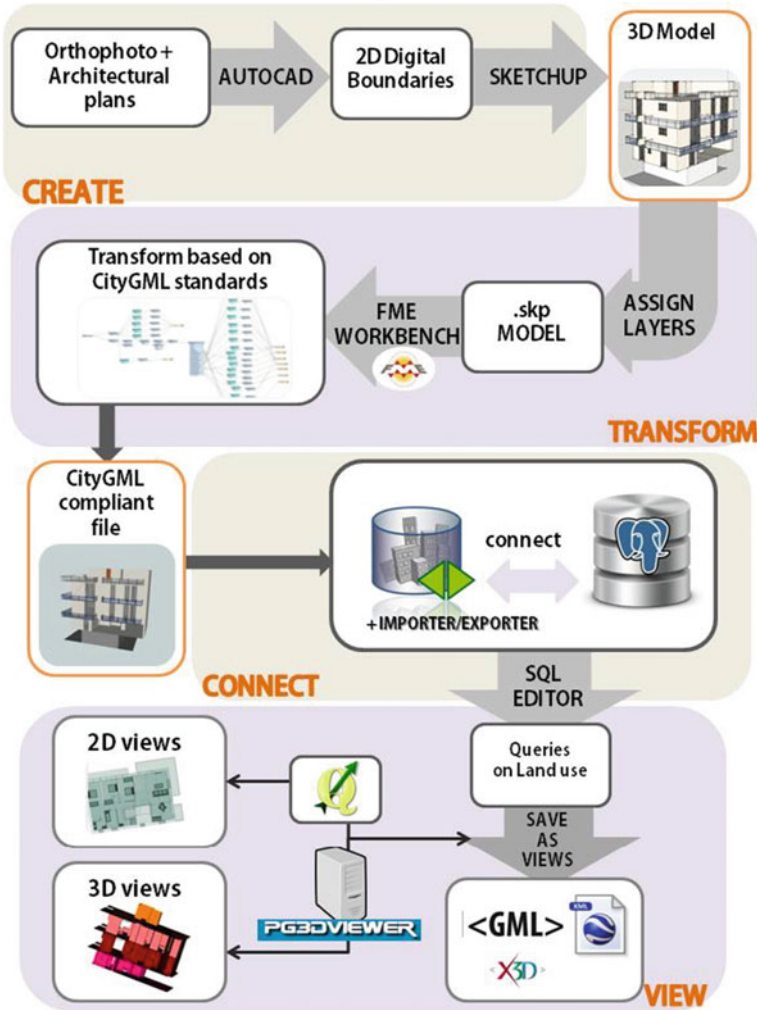


Fig. 1 Workflow

3. SketchUp can also be used to create large pieces of equipment for strictly visual purposes in various BIM software; for example SketchUp is compatible with programs such as Revit. Furthermore, connecting to the 3D SketchUp Warehouse and exploiting component libraries (such as furniture) offers a more realistic and professional model visualisation (Figs. 2 and 3).



Fig. 2 Building model designed using SketchUp Pro

4.3 Transformation via FME

FME (Feature Manipulation Engine) is a transformation engine produced by Safe Software Inc. and helps users convert data (both geometry and attributes) into several formats. FME is compatible to CityGML up to version 2.0 and includes key transformers for CityGML such as Attribute Creator and Geometry Property Setter as shown in Fig. 4. The SketchUp model can be processed within FME. The main steps for compiling CityGML from the .skp file were the addition of CityGML specific attributes (for instance, `gml_id`.) and geometry properties. CityGML entities need to be interrelated based on the standard's specifications and the appropriate geometry must be applied to all entities (e.g. Wallsurface, Door, etc.). For example, the corresponding CityGML role for all kinds of surfaces is defined as "boundedby". The "Opening" role applies for doors and windows while "int-buildinginstallation" applies for stairs and finally "building furniture" applies for furniture. Surfaces were assigned to `lod4multisurface`, doors and windows were assigned to `lod3` and stairs and furniture were assigned to `lod4geometry`. After compilation of the preferable format using FME Workbench, it is possible to optimize and represent the output in the FME Data Inspector.

4.3.1 Modeling Using Safe Software FME

Five feature types:Room were defined, each one corresponding to each building's floor. Land use was assigned to each feature type: Room (defined by 4digit codes)

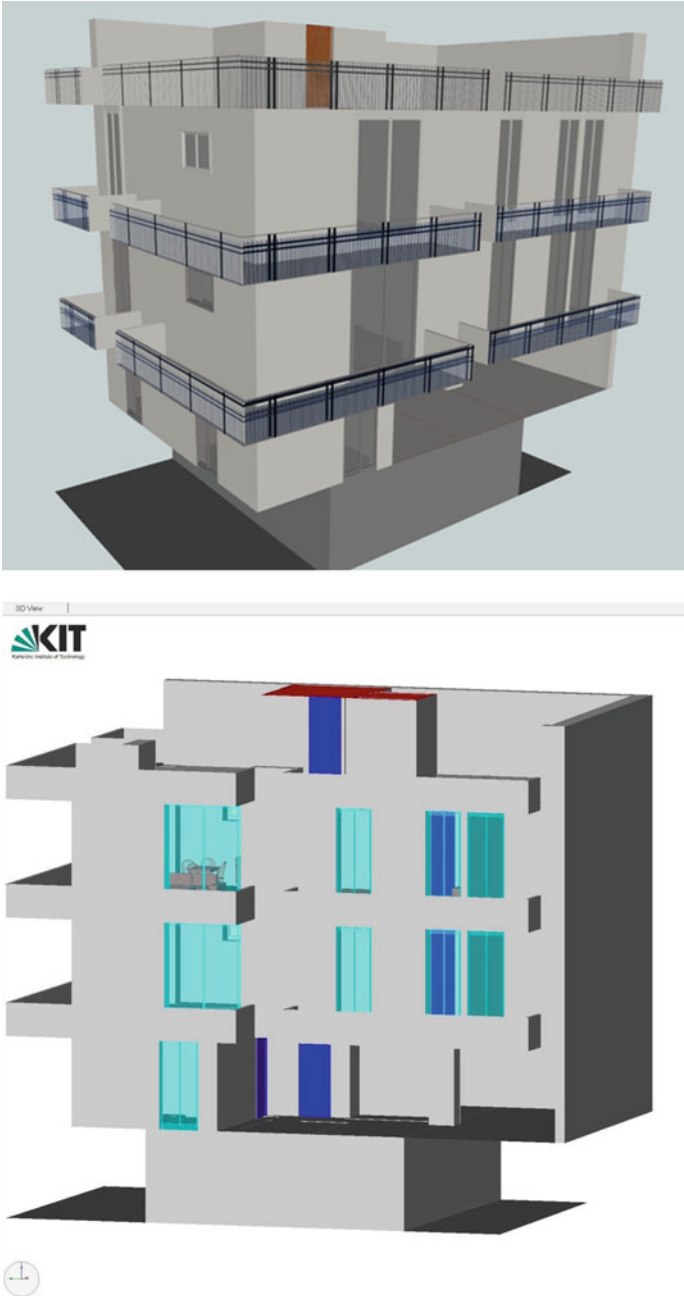


Fig. 3 *Up* 3D building model in FME Inspector *Down* 3D building model in FZK Viewer

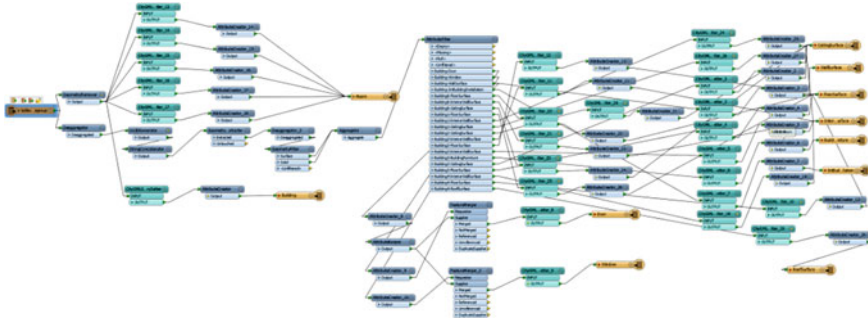


Fig. 4 Transformation via Workbench

based on the CityGML standard. Room geometries were defined by the surfaces: ceiling surface, floor surface, interior wall surface. Features' restructuring in FME was achieved through various transformers. Within the case study examined in this paper the following transformers were exploited:

1. De-Aggregator: De-aggregator transformer allows classification of each layer to its relative feature type (e.g. wall surface) through separation of the 3D plan to its constituent parts. Nevertheless, openings (doors and windows) were not decomposed in parts, since this action would considerably increase the data size.
2. Attribute filter: Using attribute filter transformer allows for manual SketchUp Pro layers' insertion. Layer names in SketchUp pro, are required to match those of the attribute filter layer.
3. Attribute creator: Attribute creator is a table generating attribute features. Entities definitions to CityGML database require creation of a new attribute `gml_name`, aiming to generate a detailed object's description. Association between the building and its constituent building parts, within the case study examined, was achieved by creating a `gml_parent_id` for each room.
4. Citygml_geometrysetter: Using this transformer allows assigning each feature's Level of Detail (LoD), from LoD2 to LoD4, along with its geometry type, (e.g. multisurface, solid, etc.) and feature role (such as opening, bounded by, building installation, etc.).

In the final CityGML file, each object was classified accordingly as wall surface, building installation, etc., with their corresponding geometries as defined in CityGML. These components were finally integrated within feature module: building.

4.4 Database Creation

The database (postgreSQL) was structured according to the 3DcityDB. Furthermore, the spatial extension of postGIS was inserted in postgresQL. The database was processed via the importer/exporter software which enables the CityGML file import in the database. The comprehension of the CityGML conceptual model is crucial; that is the accurate data analysis (meaning that the relations among the different entities need to be identified) enables execution of correct SQL queries on the database.

4.5 Querying

Queries utilize geometry data types such as points, lines and polygons and consider their spatial relationships. There are numerous examples for using queries in 3D modeling applications such as queries required for 3D display applied e.g. to computer games or queries providing options for spatial analysis in BIM data.

4.5.1 Visualization

Queries related to land use or the building’s parts were executed within SQL editor. The result of the query that is posed as “find the land use corresponding to the third floor” in natural language, can be seen below (Fig. 5). The rooms’ geometries

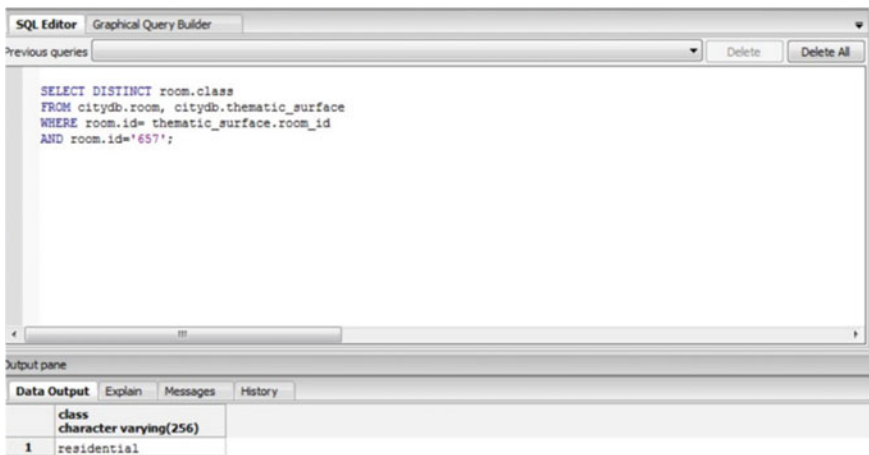


Fig. 5 Attribute query

weren't generated via FME (skp file's transformation to CityGML) but the views comprising each room's geometry were generated through postGIS.

The next SQL paradigm (Fig. 6) exports views of the executed query, meaning it exports each building floor's view (tagged as room) distinctively. The geometries can be exported in various formats such as GML, KML, X3D.

It is possible to visualize the queries' views both in 2D and 3D. Concerning the 2D approach the database was connected to QGIS, which is also open source. The pictures below (Fig. 7) present the third floor's 2D views applying different

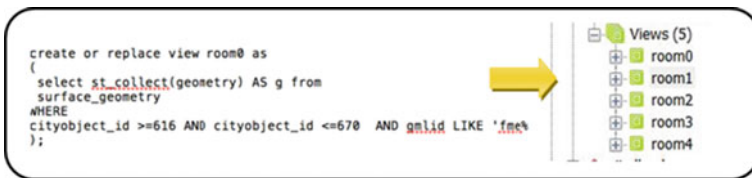


Fig. 6 Exported view of the semantic query

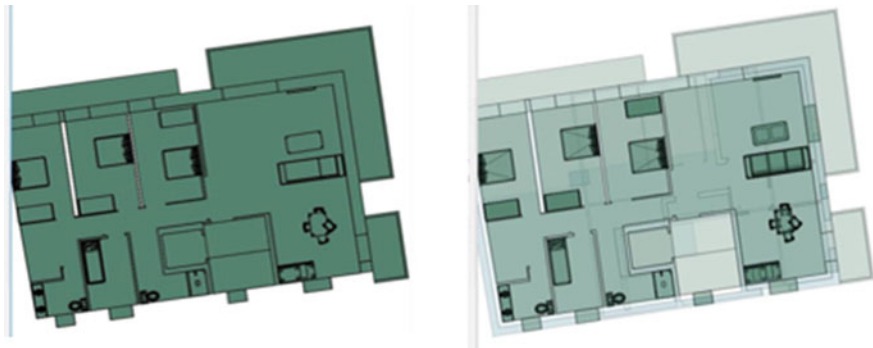


Fig. 7 2nd floor 2D views in different transparency levels

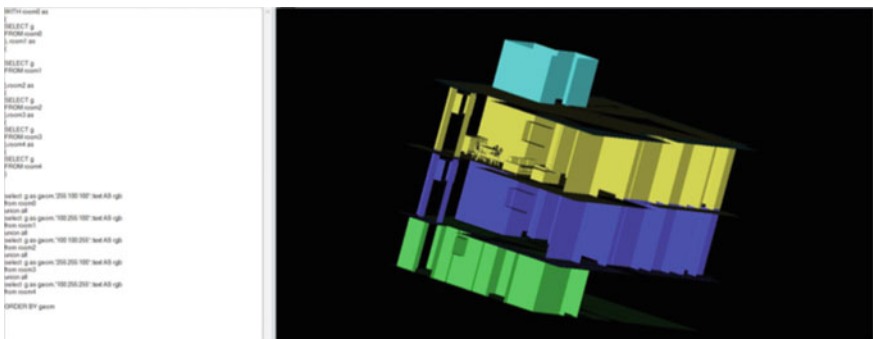


Fig. 8 3D building visualization using pg3dviewer

transparency values. All entities of the third floor (e.g. furniture) are visible because their corresponding geometry was selected from the table surface_geometry.

Concerning the 3D approach, the building was visualised through pg3DViewer (compliant with database), applying color grading per room (Fig. 8).

5 Discussion and Conclusions

The variety of solutions and different methodologies employed for the integration of semantics into 3D city models or the semantic enrichment of distinct building models, reveals the present need to define in detail and in a higher dimensionality all thematic areas or concepts within an urban environment. The various proposals examined in this paper offer pragmatic solutions for modeling urban areas within a semantic context. Current trends based on 3D models' semantics refer to the wider thematic areas of urban development, energy management, property taxation, indoor navigation, natural disasters simulation, cultural heritage registration and military operations. More specifically for this research project:

- Sketchup is a solid tool for architectural or land uses, but not that effective in environmental uses. FME is a flexible program, supporting CityGML adequately, providing transformation to CityGML format rapidly. The entities' information, such as their geometry and interrelations are maintained in high level of semantics during the translation.
- The described procedure can be characterized as cost-efficient. The PostgreSQL database, the postGIS extension and the pg3DViewer, are all open-source software, while FME offers an educational license. In terms of pre- based knowledge, the user should comprehend the CityGML structure before performing analysis. Also, a basic knowledge of databases is necessary to define the tables' connections, while basic knowledge of SQL is needed for the syntax of the queries.
- An issue still under investigation, is the limitations in adding solid geometry in the feature type:Room through the FME translation. The issue is solved manually, through the database, by creating 5 views with their including geometry that matches the 5 feature types: Room.
- According to CityGML standard, land use features are applied to volumetric spaces such as buildings or rooms instead of surface objects (e.g. groundsurface, wallsurface etc.). Feature type: Room was selected in this case study, alternatively to feature type:Building in order to assign different land use attributes to each floor, instead of the whole building. Separation of the building to its constituent floors through generation of different buildings including each of the floors assigned with different land uses was also not qualified in order to ensure indivisibility of the building object.

A further extension of this project might include the building block. The surrounding buildings could be designed as well and could be assigned with further information, such as value and view. Through a few modifications, it is feasible to perform analysis related to land use to the whole building block.

The next step would be the import, for example into Google Earth, for the whole block. The user will have the ability to right-click on every building and read the information included.

The compatibility between SketchUp and BIM programs will be investigated. Adding a building with precise architectural information into the IFC for example to further add mechanical parts, makes it a solid file for both cadastral and environmental purposes.

References

- Aien, A., Kalantari, M., Rajabifard, A., Williamson, I., & Wallace, J. (2013). Towards integration of 3D legal and physical objects in cadastral data models. *Land Use Policy*, 35, 140–154.
- Balogun, A. L., Matori, A. N., & Lawal, D. U. (2011). Geovisualization of sub-surface pipelines: A 3D approach. *Modern Applied Science*, 5(4), p158.
- Biljecki, F., Zhao, J., Stoter, J., & Ledoux, H. (2013, September). Revisiting the concept of level of detail in 3D city modelling. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Proceedings of the ISPRS 8th 3D GeoInfo Conference & WG II/2 Workshop* (pp. 63–74).
- Billen, R., Zaki, C. E., Servièrès, M., Moreau, G., & Hallot, P. (2012). *Developing an ontology of space: application to 3D city modeling*. Usage, usability, and utility of 3D city models.
- Breunig, M., & Zlatanova, S. (2011). 3D geo-database research: Retrospective and future directions. *Computers & Geosciences*, 37(7), 791–803.
- Brown, G., Nagel, C., Zlatanova, S., & Kolbe, T. H. (2013). Modelling 3D topographic space against indoor navigation requirements. In *Progress and New Trends in 3D Geoinformation Sciences* [pp. 1- 22]. Springer Berlin Heidelberg.
- Çağdaş, V. (2013). An application domain extension to citygml for immovable property taxation: A Turkish case study. *International Journal of Applied Earth Observation and Geoinformation*, 21, 545–555.
- Cheng, J., Deng, Y., & Du, Q. (2013). Mapping between BIM models And 3D GIS city models of different levels of detail. In N. Dawood & M. Kassem (Eds.), *Proceedings of the 13th International Conference on Construction Applications of Virtual Reality*, 30–31 October 2013, London, UK.
- de Laat, R., & van Berlo, L. (2011). Integration of BIM and GIS: The development of the CityGML GeoBIM extension. In *Advances in 3D Geo-information sciences* (pp. 211–225). Berlin: Springer.
- de Vries, T., & Zlatanova, S. (2011). 3D intelligent cities. *GEO Informatics*, 14(3), 6–8.
- Diakité, A. A., Damiand, G., & Gesquière, G. (2014). Automatic semantic labelling of 3D buildings based on geometric and topological information. In *Proceedings of 9th International 3D GeoInfo Conference (3D GeoInfo)*, Nov. 2014, Dubai, United Arab Emirates. Karlsruhe Institute of Technology, 3DGeoInfo conference proceedings series. <<http://nbnresolving.org/urn:nbn:de:swb:90-438043>>. <hal-01122533>.
- Dimopoulou, E., & Elia, E. (2012). Legal aspects of 3D property rights, restrictions and responsibilities in Greece and Cyprus. In *Proceedings of the 3rd International Workshop on 3D Cadastres, Developments and Practices* (pp. 25–26).

- Dimopoulou, E., Kitsakis, D., & Tsiliakou, E. (2014). Investigating correlation between legal and physical property: possibilities and constraints. In *Proceedings of SPIE 9535, Third International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2015)*, 95350A 19 June 2015. doi:[10.1117/12.2192572](https://doi.org/10.1117/12.2192572).
- Dimopoulou, E., Tsiliakou, E., Kosti, V., Floros, G., & Labropoulos, T. (2014). Investigating integration possibilities between 3D modeling techniques. In *Proceedings of 9th International 3D GeoInfo Conference (3D GeoInfo), Nov. 2014*, Dubai, United Arab Emirates.
- Donkers, S. (2013). Automatic generation of CityGML LoD3 building models from IFC models. MSc thesis, Delft University of Technology, Department of GIS Technology, OTB Research Institute for the Built Environment.
- El-Mekawy, M. (2010). *Integrating BIM and GIS for 3D city modelling: The case of IFC and CityGML*.
- El-Mekawy, M., Östman, A., & Shahzad, K. (2011). Towards interoperating cityGML and IFC building models: A unified model based approach. In *Advances in 3D Geo-information sciences* (pp. 73–93). Berlin: Springer.
- Frédéricque, B., Raymond, K., & Van Prooijen, K. (2011, May). 3D GIS as applied to cadastre—A benchmark of today’s capabilities. In *FIG Working Week*.
- Geiger, A., Benner, J., & Haeefe, K. H. (2014). Generalization of 3D IFC building models. In M. Breunig, M. Al-Doori, E. Butwilowski, P. V. Kuper, J. Benner, & Haeefe, K. H (Eds.), *3D Geoinformation sciences, The Selected papers of the 3D GeoInfo 2014* (pp. 19–35). Berlin: Springer.
- Gózdź, K., Pachelski, W., & Poland, P. V. (2014). The possibilities of using CityGML for 3D representation of buildings in the cadastre. In *Proceedings 4th International Workshop on 3D Cadastres*, 9–11 Nov 2014, Dubai, United Arab Emirates. International Federation of Surveyors (FIG).
- Gröger, G., & Plümer, L. (2009). Updating 3D City models—How to preserve geometric-topological consistency. In W. G. Aref, D. Agrawal, M. F. Mokbel, C.T. Lu, C. Shahabi, P. Scheuermann, et al. [Eds.], *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2009)* Seattle, Washington, 4–6 Nov 2009, ACM Press, New York, pp. 536–539.
- Gröger, G., & Plümer, L. (2012). CityGML—Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71, 12–33.
- Guo, R., Li, L., Ying, S., Luo, P., He, B., & Jiang, R. (2013). Developing a 3D cadastre for the administration of urban land use: A case study of Shenzhen, China. *Computers, Environment and Urban Systems*, 40, 46–55.
- Guo, R., Yu, C., He, B., Zhao, Z., Li, L., & Ying, S. (2012). Logical design and implementation of the data model for 3D cadastre in China. In *Proceedings 3rd International Workshop 3D Cadastres: Developments and Practices* (pp. 113–136).
- Hu, M. (2008). Semantic based LoD models of 3D house property. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (Vol. XXXVII, Part B2), Beijing.
- Isikdag, U., & Zlatanova, S. (2009). Towards defining a framework for automatic generation of buildings in CityGML using building information models. In J. Lee, & S. Zlatanova (Eds.), *3D geoinformation sciences* (pp. 79–96). Berlin: Springer.
- Isikdag, U., Horhammer, M., Zlatanova, S., Kathmann, R., & van Oosterom, P. J. M. (2014). Semantically rich 3D building and cadastral models for valuation. In *Proceedings 4th International Workshop on 3D Cadastres*, 9–11 Nov 2014, Dubai, United Arab Emirates. International Federation of Surveyors [FIG].
- Isikdag, U., Zlatanova, S., & Underwood, J. (2013). A BIM-oriented model for supporting indoor navigation requirements. *Computers, Environment and Urban Systems*, 41, 112–123. INSPIRE, D2.8.III.2 Data Specification on Buildings—Draft Technical Guidelines.
- Jazayeri, I., Rajabifard, A., & Kalantari, M. (2014). A geometric and semantic evaluation of 3D data sourcing methods for land and property information. *Land Use Policy*, 36, 219–230.

- Karki, S., McDougall, K., & Thompson, R. J. (2010). An overview of 3D Cadastre from a physical land parcel and a legal property object perspective. In *Proceedings of the XXIV FIG International Congress 2010: Facing the Challenges-Building the Capacity*, Sydney, 11–16 April 2010. FIG.
- Kolbe, T. H. (2009). Representing and exchanging 3D city models with CityGML. In *3D geoinformation sciences* (pp. 15–31). Berlin: Springer.
- Kolbe, T. H., & Plümer, L. (2004). Bridging the Gap between GIS and CAAD. *GIM International* 2004, 18(7).
- Li, Y., & He, Z. (2008). 3D indoor navigation: A framework of combining BIM with 3D GIS. In *44th ISOCARP Congress*.
- Löwner, M. O., Benner, J., Gröger, G., & Häfele, K. H. (2013). New concepts for structuring 3D city models—an extended level of detail concept for CityGML buildings. In *Computational Science and Its Applications—ICCSA 2013* (pp. 466–480). Berlin: Springer.
- Nagel, C., Stadler, A., & Kolbe, T. (2009). Conceptual requirements for the automatic reconstruction of building information models from uninterpreted 3D models. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (Vol. XXXIV, Part XXX).
- Rönsdorff, C., Wilson, D., & Stoter, J. E. (2011). Integration of land administration domain model with CityGML for 3D Cadastre. In *Proceedings 4th International Workshop on 3D Cadastres, 9-11 November 2014*, Dubai, United Arab Emirates. International Federation of Surveyors [FIG].
- Shojaei, D., Kalantari, M., Bishop, I. D., Rajabifard, A., & Aien, A. (2013). Visualization requirements for 3D cadastral systems. *Computers, Environment and Urban Systems*, 41, 39–54.
- Shojaei, D., Rajabifard, A., Kalantari Soltanieh, S. A. E. I. D., Bishop, I., & Aien, A. (2012). *Development of a 3D ePlan/LandXML visualisation system in Australia*.
- Shojaei, D., Rajabifard, A., Kalantari, M., Bishop, I. D., & Aien, A. (2014). Design and development of a web-based 3D cadastral visualisation prototype. *International Journal of Digital Earth*, [ahead-of-print], 1–20.
- Stadler, A., & Kolbe, T. H. (2007). Spatio-semantic coherence in the integration of 3D city models. In *Proceedings of the 5th International Symposium on Spatial Data Quality, Enschede*.
- Stadler, A., Nagel, C., König, G., & Kolbe, T. H. (2009). Making interoperability persistent: A 3D geo database based on CityGML. In *3D Geo-information sciences* (pp. 175–192). Berlin: Springer.
- Van Berlo, L., & de Laat, R. (2009). Integration of BIM and GIS: The development of the CityGML GeoBIM extension. In: T. Kolbe, G. König, & C. Nagel (Eds.), *3D Geo-information sciences*. Lecture Notes in Geoinformation and Cartography (pp. 211–225). Berlin: Springer.
- van Oosterom, P. (2013). Research and development in 3D cadastres. *Computers, Environment and Urban Systems*, 40, 1–6.
- Xiong, X., & Huber, D. (2013). Using context to create semantic 3D models of indoor environments. In *Proceedings of the British Machine Vision Conference*, BMVA Press (2010), pp. 45.1–45.11. <http://dx.doi.org/10.5244/C.24.45>.
- Ying, S., Guo, R., Li, L., & He, B. (2012). Application of 3D GIS to 3D cadastre in urban environment. In *3rd International Workshop on 3D Cadastres: Developments and Practices*, Shenzhen, China (pp. 25–26).
- Ying, S., Li, L., & Guo, R. (2011). Building 3D cadastral system based on 2D survey plans with SketchUp. *Geo-spatial Information Science*, 14(2), 129–136.
- Zhu, Q., Zhao, J., Du, Z., Zhang, Y., Xu, W., Ding, Y., et al. (2011). Towards semantic 3D city modeling and visual explorations. In T. Kolbe, G. König, & C. Nagel (Eds.), *3D Geo-information sciences* (pp. 275–294). Berlin: Springer.
- Zlatanova, S., Stoter, J., & Isikdag, U. (2012, June). Standards for exchange and storage of 3D information: Challenges and opportunities for emergency response. In *Proceedings of the Fourth International Conference on Cartography and GIS, Albena, Bulgaria* (pp. 17–28).

3D Complete Traffic Noise Analysis Based on CityGML

Lu Lu, Thomas Becker and Marc-Oliver Löwner

Abstract Nowadays, transportation plays a more and more significant role in our daily life but produces noise. Noise not only causes annoyance and health problems, but also shows effects on economics. In 2002, the European Union published the Common Noise Assessment methods (Kephalopoulos et al. 2012). The objective of this paper is to present a method for simulating the noise propagation in 3D and calculating traffic noise on building façade level with different height by using 3D city model and integrating all noise coming from individual traffic such as cars or motorcycles as well as planes and railroad based vehicles. Since noise sources are located in our 3D urban environment—the analysis and the mapping has to cover the 3D aspect as well. A neighborhood of Berlin was chosen as research area. Currently, we propose a semi-automatic solution for 3D noise mapping: generating 3D observer points from CityGML building data; modeling 3D propagation path and calculating different kinds of traffic noise level. The total noise levels are then calculated by estimating the total annoyance based on effect equivalent sound pressure levels for different types of traffic source. The results are presented as a 3D map. In the future this approach can be further developed to an on-the-fly tool, that makes use publicly available data and processes to determine the noise for one building to a certain point in time. Besides that we found out that more investigation and evaluation on noise calculation methods are needed. Thus the development of near real time calculation methods together with noise measurements is required.

Keywords Noise · Traffic analysis · CityGML · Noise mapping

L. Lu (✉) · T. Becker · M.-O. Löwner

Institute for Geodesy und Geoinformation, Technische Universität Berlin,
Straße des 17. Juni, 10623 Berlin, Germany
e-mail: lulu5399@gmail.com

T. Becker
e-mail: thomas.becker@tu-berlin.de

M.-O. Löwner
e-mail: m.loewner@tu-berlin.de

1 Introduction

Transportation plays an important role in our daily life. As part of the transportation systems, users on one side enjoy the benefited of transportation systems, individual or public, but, on the other hand also suffer from noise produced by transportation. Noise not only causes annoyance and health problems (rf. Bluhm et al. 2007), but also shows effects on economic, f.i. on house prices (Rahmatian and Cockerill 2004; Pennington et al. 1990). In modern cities, especially Megacities, noise is one of the most serious pollution next to air, littering and water. Nowadays, noise pollution is a hot issue not only for specific individuals but also for urban society calling for the assessment of noise emitters and infrastructural setups.

The whole noise estimation process can be divided into two main stages: noise generation and noise propagation. Basically, noise is produced by a single vehicle. However, on a road, vehicles run and generate noise continuously, thus, the noise level of a road is the collective contribution of those individual noises. Therefore, for road traffic noise estimation, the road is viewed to be the source for noise generation while noise level is mainly depending on the number of vehicles, vehicle type and speed. After noise generated from the source, sound waves propagate through the air in the free field and are received by human-beings. During propagation, noise level can be reduced by distance or increased by reflection. If objects existed between noise source and receiver, sound wave can be reflected, refracted or attenuated which not only change the propagation path, but also may affect sound spectrum and sound pressure. Therefore, it is necessary to analysis the spatial relationship between them.

Traffic is the main sources for noise pollution and traffic noise can be classified into three categories: road traffic noise, railway traffic noise and aircraft noise. All of these noises have their own specific characteristics and points of origin in the three dimensional space. Plenty of noise prediction model have been developed to model traffic noise pollution like RLS-90 noise prediction algorithm and ECAC aircraft noise contour model. Those prediction models are widely applied for noise mapping. In 2002, the European Union published the Environmental Noise Directive (END) for the assessment and management environmental noise (European Union 2002). According to this, strategic noise maps for major roads, railways, airports and agglomerations has to be documented every 5 years. To improve the reliability and comparability between different countries and to support the implementation of END, the Common Noise Assessment methods (Kephelopoulous et al. 2012) was introduced and presented.

Limited by the data quality and quantity, most of the noise prediction models are still based on empirical equations. Noise maps are generally calculated and represented in 2D. Nowadays, increasing quantity of semantically enriched 3D geodata are available by the development of 3D city models. 3D noise maps with higher resolution and differentiated results, f.i. vertical noise propagation are of high value for the urban environment. 3D noise maps do not only represent the noise level result in 3D, moreover simulation and calculation of the noise propagation paths in

3D space are required. In comparison to 2D noise mapping, the results are more reliable because noise spread to all direction are modelled.

The main achievements that should be made by this hot topic research are:

- Simulation of noise propagation in 3D
- Calculation of traffic noise on building façades with different height by using 3D city model
- simulation of noise propagation paths between noise source and an arbitrary observer point in 3D
- on-the-fly determination of observer points on building façades
- determination of complete noise levels at these observer points

This contribution is structured as follows. Background information about noise and previous research works regarding noise prediction and noise mapping is given in Sect. 2. The input data and methodology are described in Sect. 3 including the algorithm for noise calculation. In Sect. 4 results of the noise level calculation are presented in 3D maps and discussed. We end up with a conclusion and an outlook for future work in Sect. 5.

2 Theoretical Background of Noise Mapping

Unwanted sound is called “Noise”. Noise, same as sound, can be described in terms of amplitude, frequency and time pattern. The amplitude is normally perceived as loudness, measured by sound pressure level (SPL) in decibels (dB). SPL is a logarithmic measure that describes the strength of a noise event. The noise frequency is perceived as pitch, which is the number of pressure variations per second and is measured in hertz (Hz). Human beings are less sensitive to very low or high frequency. The frequency significantly influences air absorption and refraction during propagation. According to the tones or changes of sound level, the noise can be categorized into 3 types:

- continuous noise which the frequency or tones changed slightly for long period like blower noise;
- intermittent noise which noise level increases and decreases rapidly as in an event of alarm;
- and impulsive noise which is brief and abrupt like explosions noise.

From this point of view, nearly all of the transportation noises are intermittent noise. To measure noise, both subjective and objective factors are involved in noise assessing and some indicators are introduced in 3 different steps: leveling, scaling and rating.

Leveling refers to the human hearing, which is frequency selective and very sensitive in the range of 500 Hz–6 kHz. However, frequency response of human hearing changes with amplitude. To eliminate noise variations and to isolate the

noise from other influencing factors, weighting levels have to be introduced and named with ‘A’, ‘B’, ‘C’ and ‘D’ (Kjellberg et al. 1997; Nilsson 2007).

Scaling is the combination of level and time that is needed because noise levels are not steady but fluctuate with time. The scaling parameters can be the level for a given proportion of the time as in the statistical level (L_{A50}), or an integration of level with respect to duration as in equivalent level (L_{eq}), single event noise exposure level (L_{AX}), disturbance index (Q) or effective perceived noise level (EPNL). The equivalent continuous sound level (L_{eq}) is the essential averaged parameter that uses a steady level in a fixed time period representing the amount of energy of the fluctuating sound level. The time period T is normally defined as long time from 1, 4, 8 to 24 h. The mathematical definition of L_{eq} respect to ‘A’ weighted is:

$$L_{eq} = 10 \log_{10} \left[\frac{1}{T} \int_0^T 10^{L_A/10} dt \right] dB(A)$$

Rating is needed to define a time period for the scale to evaluate a specific type of noise in particular circumstances because human reaction to noise is different from day to night. It may be an aggregate of different scales applied to day time and night time. The day-night average sound level (L_{DN}) results from L_{eq} in 24 h but a 10 dB(A) penalty is applied to night-time period (22:00–07:00). The formula is used as:

$$L_{DN} = 10 \log \left[\frac{1}{24} \left(15(10^{L_d/10}) + 9(10^{(L_n + 10)/10}) \right) \right] dB(A)$$

where

L_d is the day-time equivalent sound level

L_n is the night-time equivalent sound level

2.1 Calculation Methods for Noise

There are various sources of noise and normally they can be grouped into three classes (Chambers 2004):

- Transportation noise caused by cars, trucks, bus, trains, tram, motorcycles and aircrafts,
- Industrial noise caused by industrial operations and equipment
- Residential noise caused by air conditioners, television, stereos, children, pets.

As one of the most important noise source, traffic noise comes from three primary transport modes: road traffic, railway and aircraft. Each of them has their own noise emission characteristics and should be treated in a different way.

2.1.1 Road Traffic Noise

Nowadays, there are 6 commonly used prediction models which are applied by different countries: The FHWA TNM version 1.0, The CoRTN model, RLS-90, MITHRA, StL-86 version 1, ASJ Method—1993. These models can be used to predict sound pressure levels (L_p) or equivalent continuous sound pressure level (L_{eq}). They are designed for roadway engineers to check whether it meets requirement of noise constraints, but not for other users of traffic noise models (Steele 2001). RLS-90 (Richtlinien für den Lärmschutz an Straßen) is the anonymous legal standard for noise prediction in Germany. This prediction model incorporates several variables: vehicle flow, percentage of heavy vehicles, parking lots, etc. The mean A-weighted level is calculated by each nearer or further lane:

$$L_m = L_m^{(25)} + D_v + D_{StrO} + D_{Sig} + D_E + D_s + D_{BM} + D_E$$

where

- $L_m^{(25)}$ is the A-weighted mean level,
- D_v is the correction for speed level,
- D_{StrO} is the correction for road surfaces,
- D_{Sig} is the correction for rises and falls,
- D_E is the correction for the absorption characteristics of building surfaces,
- D_s is the attenuation due to distance and air absorption,
- D_{BM} is the attenuation due to ground and atmospheric effects,
- D_E is the attenuation due to the topography and building dimensions.

Moreover, $L_m^{(25)}$ is decided by the traffic flows for different kinds of roads, the corresponding share of heavy vehicles and time period for day or night. Specially, RLS-90 is unique in the parking lots algorithm:

$$L_{m,E}^* = 37 + 10 \lg(N, n) + D_p$$

where

- N is the number of vehicle movements per hour, by parking spot,
- n is the number of such parking spots,
- D_p is a correction for the type of car park.

2.1.2 Railway Noise

In Europe, several prediction models have been developed: calculation of railway noise (CRN) from UK, the Nordic prediction method for train noise (NMT) for Norway and Sweden, Mithra for French, Schall 03 for Germany. The differences between them are correction variables, source position and sound radiation

characteristics. For example, the Schall 03 and CRN take the train source position as the height of the railhead (0 m) while Mithra take 0.8 m ahead the railhead (Lui et al. 2006). The Schall 03 was updated from 1990 to Schall 03 2006 and implemented in 2008 (Schall 03 2014). In Schall 03 2006, calculation of noise emission is a method using soundpower-levels in octave bands (Moehler et al. 2008):

$$L_{W'_{A,f,h,m,Fz}} = a_{A,h,m,Fz} + \Delta a_{f,h,m,Fz} + 10 \lg \frac{n_Q}{n_{Q,0}} \text{ dB} + b_{f,h,m} \lg \left(\frac{v_{Fz}}{v_0} \right) \text{ dB} \\ + \sum c_{f,h,m} + \sum K$$

where:

- $a_{A,h,m,Fz}$ is A-weighted overall level of the sound power per unit track length emitted from the height range h . r the source type m of one vehicle unit Fz , equipped with the reference number $n_{Q,0}$ of sound sources, running at the reference velocity $v_0 = 100 \text{ km/h}$, on a ballasted track with average condition of the rail surface, neither on a bridge nor on a curved track, in dB,
- $\Delta a_{f,h,m,Fz}$ is level difference between overall level and octave band level in the octave band f , in dB,
- n_Q is number of sound sources of a vehicle unit,
- $b_{f,h,m}$ is velocity factor,
- v_{Fz} is train velocity, in km/h,
- $\sum c_{f,h,m}$ is level corrections for type of track and rail surface condition, in dB,
- $\sum K$ is level corrections for bridges and particular nuisance of noise, in dB.

2.1.3 Aircraft Noise

Air traffic has low frequency and more flexible route in contrast to road traffic. Hence, traffic volume is not helpful to calculate the aircrafts noise. There are several prediction models being used in Germany: DIN 45684, AzB, ECAC Doc. 29, 3rd edition, etc. The first two methods are applied in Germany whereas ECAC Doc.29 is the European method. Different from the AzB method which only the segments perpendicular from the receiver point is considered, in ECAC Doc.29 all segments of an air route contribute to the overall A-weighted level (ECAC 2005). Besides, with speed and thrust the aircraft route can be divided into “sub-track” with straight lines in 3D by ECAC Doc.29. The flight path segments are constructed by 6 parts: ground roll, take-off, constant speed climb, power cutback, accelerating climb, descent and landing. To model the aircrafts noise, an individual flyover of aircraft is viewed as an event. The most commonly used parameter is the Single Event Sound Exposure Level (L_E). It is a metric of time-integration which describes the amount of sound energy in the events (ECAC 2005):

$$L_E = 10 \lg \left(\frac{1}{t_o} \int_{t_1}^{t_2} 10^{L(t)/10} dt \right)$$

where

t_o is a reference time,

$[t_1, t_2]$ is the time period that (nearly) all significant sound in the event is encompassed.

Because all segments contribute to the overall noise level, for each flight path segment is:

$$L_{E, seg} = L_{E\infty}(P, d) + \Delta_V + \Delta_I(\varphi) - (\beta, \iota) + \Delta_F$$

where

Δ_V is the duration correction,

$\Delta_I(\varphi)$ is the installation effect,

$\Lambda(\beta, \iota)$ is the lateral attenuation,

Δ_F is the segment correction.

Based upon the mentioned methods above, three preliminary calculation methods for environmental noise were published in 2006 in the Federal Gazette No.154: VBUS for roads, VBUSch for railways and VBUF for aircrafts. These three methods are designed for the EU Environmental Noise Directive (END) which aims to “define a common approach intended to avoid, prevent or reduce on a prioritized basis the harmful effects, including annoyance, due to the exposure to environmental noise” (Directive 2002/39/EG). The noise indicators for all methods are the A-weighted equivalent continuous sound level in decibels according to ISO 1996-2. The assessment period is one year and every day is divided into the following periods:

- L_{Day} : 12 h, from 6:00 to 18:00
- $L_{Evening}$: 4 h, from 18:00 to 22:00
- L_{Night} : 8 h, from 22:00 to 6:00

And the noise index L_{DEN} is defined by:

$$L_{DEN} = 10 \cdot \lg \frac{1}{24} \left(12 \cdot 10^{\frac{L_{Day}}{10}} + 4 \cdot 10^{\frac{L_{Evening} + 5}{10}} + 8 \cdot 10^{\frac{L_{Night} + 10}{10}} \right)$$

3 3D Complete Traffic Noise Mapping Approach

Typically the results of noise mapping are presented in 2D maps or are just calculated for a fixed height, normally 4 m above ground. However, noise propagates through media in all directions (3D) and the sources of noise are located in 3D space as well. Thus, 2D noise maps cannot provide relevant information to inhabitants regarding noise levels at different heights nor gradient effects regarding the source of noise. To extend the noise mapping from 2D to 3D, spatial relationships between the noise emitter and the noise receiver should be discussed in detail.

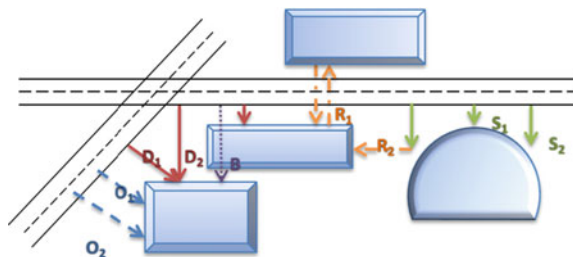
In horizontal direction, orientation and distance of buildings to the road or flight path as well as the shape of buildings are the main influence factors to the noise effect. Figure 1 illustrates the different noise effects in horizontal direction: If a building is not parallel to the road, the noise effect will be decreasing with the increasing of the distances (O_1 & O_2). Besides, the shape of a building may not have great influence in vertical but in horizontal direction. Therefore, the noise should be calculated based to the real orientation and distance to the road (S_1 & S_2).

Buildings that are located close to an intersection receive noise from two crossing roads. Normally, the wall is parallel to the one road and perpendicular to another road. In this situation, the noise from the facing road takes the lead. Otherwise, if it is an acute angle between the road and the wall, the noise from both of the roads could be of similar level and both of them should be calculated and integrated (D_1 & D_2). The reflection (R_1 & R_2) and screening effect (B) should be taken into consideration.

The noise difference in vertical direction (rf. Fig. 2) mainly depends on the building height. Take high-rise buildings for example. Road traffic and aircrafts noise propagate from different directions and produce different effects to the ground and top floor of the buildings. As shown in Fig. 2, the distance from the aircraft to the top floor and that to the ground floor are different ($D_1 < D_2$) which results in different noise levels.

If the altitude of aircrafts is below 500 m during landing or departing, such difference cannot be ignored. Besides, different heights can also influence the road reflection and absorption effect (F_1 & F_2). Furthermore, the front side and back side of the same building also have different noise levels. Because most of the noise is absorbed or reflected by the front building, noise pollution of the building behind is significantly be reduced. (B_1 & B_2). To simplify complex reflection and refraction

Fig. 1 Road traffic noise in horizontal direction



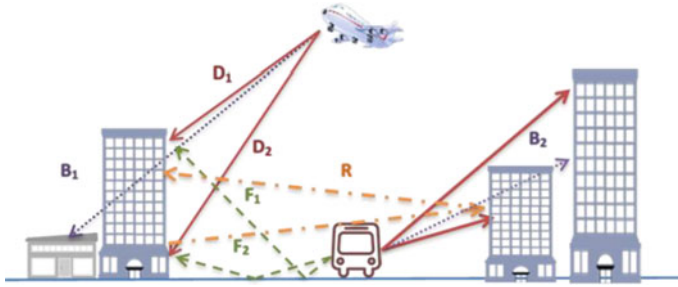


Fig. 2 Road traffic and aircraft noise in vertical direction

physics, propagation paths are classified into 4 types which are described in Common Noise Assessment Methods in Europe (CNOSSOS-EU) (Kephelopoulos et al. 2012).

3.1 Noise Calculation

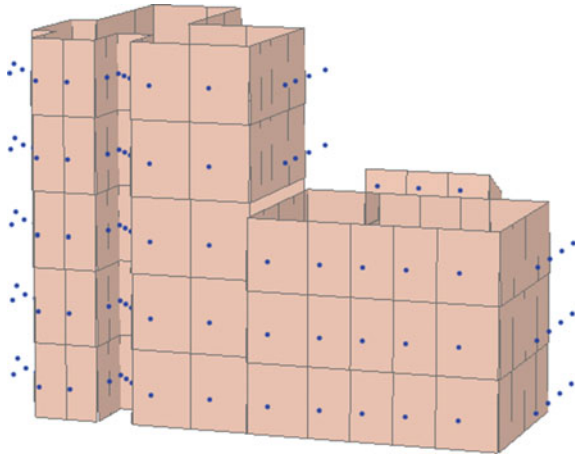
The method for 3D traffic noise estimation consists of 5 steps: source noise level acquisition, observer points on building facades generation, noise propagation path modeling, observer point noise level calculation, and multiple sources substitute levels calculation. Because the parameters for estimating railway and aircraft noise are not available, noise level from 2012 Berlin Strategy Noise Map were used here, where the noise indicator is L_{DEN} (weighted 24-h mean value) and the average noise level for each traffic mode in the research area is: road (76 dB(A)), railway (74 dB(A)) and aircraft (98 dB(A)). The aim of Berlin Strategy Noise Map is to create strategic noise maps and related statistical evaluations (for burdened people, flats, schools and hospitals). The results were prepared in terms of the following points:

- Basis for reporting to the EU, including information to the public
- Basis for the continuation of the Noise Action Plan 2013 (Noise abatement Planning Berlin)
- Basis for managing the output data (maintenance of the data model)
- Basis for new calculations and evaluations of spatially limited areas.

3.1.1 Observer Point Generation

An important step for noise estimation is to generate the observer points for which the modelling results will be produced. For 3D noise mapping, the observer points will be placed along the building facades. In specific, the observer points will be placed according to CNOSSOS-EU (Kephelopoulos et al. 2012, VII.2.2—Assigning receiver points to the facades of buildings). However, interval between observer

Fig. 3 Observer points and wall polygons in 3D



points is different in horizontal and vertical direction. In vertical direction (see Fig. 3), the interval is depended on the height of each floor that is calculated by using the CityGML property *measuredHeight* and is divided by the property *StoreysAboveGround* as a first approximation.

3.1.2 Propagation Path Modeling

When the sound wave transmitted in the medium, it can be reflected and diffracted because of barriers. Both of them will change the propagation path. To simplify, the propagation paths are classified into 4 types which is described in Fig. 4 (Kephalopoulos et al. 2012). Type 1 path is a straight path in plain view and only includes diffractions on the horizontal edges of barriers. It is the easiest scenario. Type 2 path is the path reflected on vertical barriers. The diffractions also happen on the lateral edges of barriers that are represented by path type 3. Type 4 path is the mixed path including diffraction by the lateral edges of barriers and reflection by vertical surfaces. Type 2 and type 4 paths are hard to determine, since they include diffraction that again requires knowledge about surface material and thus not part of the following calculation. Besides that, the laterally diffracted propagation paths (type 3) make negligible contribution to the total sound levels and can be omitted acc. to the CNOSSON-EU. Therefore, in this research only type 1 paths are modeled as propagation paths (Fig. 4).

Although four different types of propagation paths are known in 3D space, here only type 1 path is used. A method for modelling the type 1 paths and estimating noise levels in 3D was already develop by Farcaş for his master thesis (Farcaş 2008). Figure 5 demonstrates the propagation paths before (blue) and after (red) the use of 3D space and 3D objects. As input for our 3D environment we have choose

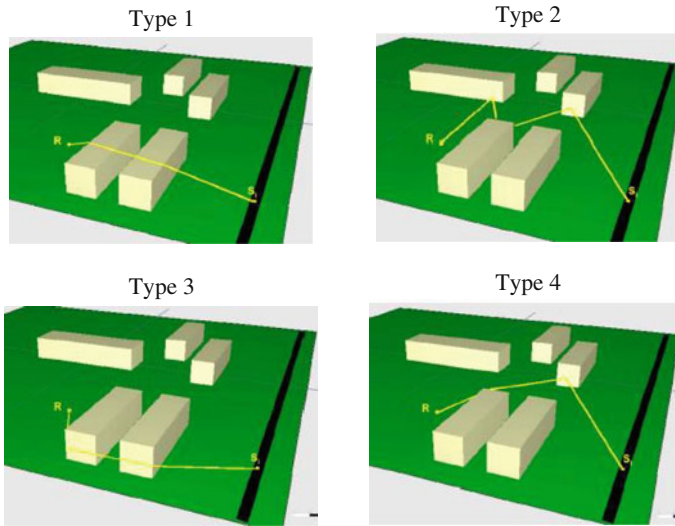


Fig. 4 Four different types of propagation paths

CityGML LoD 2 buildings, as well as a terrain dataset. The method for modeling noise propagation paths can be applied to all three traffic modes. Take road traffic as an example, the modeling method includes the following steps:

1. Select one observer point.
2. Construct the sight line to road shapefile, the output including directions.
3. Join the road shapefile with sight line shapefile to get the noise level on road.
4. Check the visibility for sight line through buildings.
5. If sight line is visible, keep it as the new sight line.
6. If sight line is not visible, intersect sight line with building multipatch to get intersection points. Then elevated them to the maximum height of the building. These points along with the original vertexes of sight line compose the new sight line.
7. Check the visibility of new sight line through the topography.
8. If sight line is visible, it is the propagation path.
9. If sight line is not visible, intersect sight line with topography surface to get the intersection polyline. Then add the surface information to intersection polyline. Compare the Z value in polyline and elevated the vertexes to the maximum Z value. All the vertexes compose the propagation path.

As it can be seen from the Fig. 5, in the right part, new noise paths (red) are pointing from road cross over the roof of the buildings and arrive at the observer point. The same happens on the left side, the paths also cross the railway subgrade and reach the observer point.

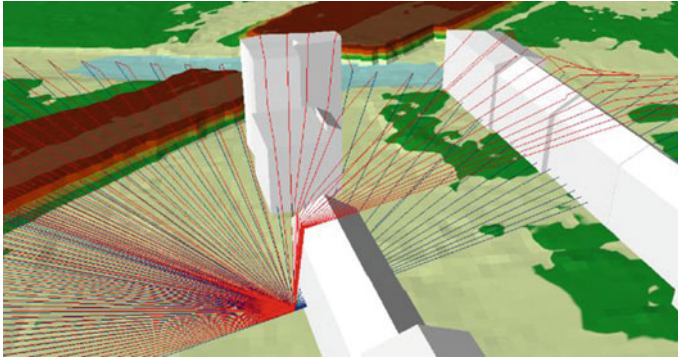


Fig. 5 Propagation paths before and after using the 3D space

3.1.3 Noise Level at Observer Point Calculation

After modeling the noise propagation paths, the noise level at observer point can be calculated by energy summing contribution from all N paths:

$$L_{Aeq} = 10 \times \lg \left(\sum_i^N 10^{L_i/10} \right)$$

where:

- L_i is the noise level for one propagation path,
- N is the number of path to observer point.

For each path, the following corrections are applied for noise calculation:

- Geometric attenuation A_{geo}
- Ground correction A_{ground}
- Angle of view correction A_{angle}

The noise level for each path is calculated from:

$$L_{path} = L_w - A_{geo} - A_{ground} - A_{angle}$$

$$A_{geo} = 10 \lg(r) + 8 \text{ dB}$$

where L_w is the sound power level, L_p is the sound pressure level, r is the distance from source to receiver.

$$A_{ground} = -3(1 - \bar{G}_m)$$

where \bar{G}_m is the average G value in the path without diffraction, for example for hard surfaces the value is 0. That means all the sound waves are reflected by the ground without any attenuation.

$$A_{angle} = -10 \lg \left(\frac{\alpha}{180} \right)$$

where α is the angle of view for one line source segment. For one observer point, noise propagated from all directions contributes equally to the total sound pressure level. The weight of each segment in a line noise source is decided by the distance to the observer point and the length of the segment.

3.1.4 Multiple Sources Substitute Levels Calculation

Noise with different amplitude, frequency and time pattern have different impacts on human-beings. In most of areas, inhabitants are not only exposed to one traffic noise but to two or even to all three traffic noises. The sound pressure level cannot be simply summed up; because each of these three traffic noises has its own characteristics and the impact to people are different even with the same sound pressure level.

The Association of German Engineers (VDI) published a guideline in 2013 This guide line proposes procedures to determine characteristics for evaluating in case of impact of different types of noise sources with regard to annoyance and self-reported sleep disturbance (VDI 2013). It includes the method to estimate the self-reported total sleep disturbance or total annoyance based on effect equivalent continuous sound pressure levels for the different types of sources. The indicator in this guideline is renormalized substitute level (L_x^*) in dB which is the value of a source specific exposure resulting in reference to road traffic noise to the same value of impairment.

The substitute levels is calculated for four impairment groups: annoyance (%A), high annoyance (%HA) as well as self-reported (%SD) and highly self-reported sleep disturbance (%HSD). If the noise level in range from 37 to 75 % dB, the following equations can be applied to calculate the effect related substituted level for annoyance:

Road traffic noise:

$$\begin{aligned} \%A &= 1.795 \times 10^{-4} (L_{\gamma, TAN} - 37)^3 + 2.110 \times 10^{-2} (L_{\gamma, TAN} - 37)^2 + 0.5353 \\ &\times (L_{\gamma, TAN} - 37) \end{aligned}$$

Rail traffic noise:

$$\begin{aligned} \%A &= 4.538 \times 10^{-4} (L_{\gamma, TAN} - 37)^3 + 9.482 \times 10^{-3} (L_{\gamma, TAN} - 37)^2 + 0.2129 \\ &\times (L_{\gamma, TAN} - 37) \end{aligned}$$

Aircraft noise:

$$\begin{aligned} \%A &= 8.588 \times 10^{-6} (L_{\gamma, TAN} - 37)^3 + 1.777 \times 10^{-2} (L_{\gamma, TAN} - 37)^2 + 1.221 \\ &\times (L_{\gamma, TAN} - 37) \end{aligned}$$

where:

$L_{\gamma, TAN}$ is A-weighted equivalent long term average level with adjustments during the three partial time intervals for the rating time $T = 24$ h.

For the determination of $L_{\gamma, TAN}^*$, corrected on road traffic noise, applies:

$$L_{\gamma, TAN}^* = 37 + 0.0003(\%A)^3 - 0.0313(\%A)^2 + 1.4982(\%A)$$

The last step is calculating energetic addition of A-weighted renormalized substituted levels of types: rail traffic and air traffic as well as the A-weighted level of road traffic noise L_{AES} :

$$L_{AES} = 10 \lg \left[\sum_j 10^{0.1(L_{x,j}^*)} \right]$$

4 Results and Analysis

Applying the aforementioned methods and formulae the noise propagation can be determined to a specific neighborhood of Berlin. There are two main factors for choosing this specific neighborhood. Firstly, it is influenced by all three different kinds of traffic noise. Secondly, the building should be high enough to measure the different noise level in vertical direction. Therefore, a residential area nearby Tegel Airport has been chosen which suffers from aircraft noise (>60 db), road noise and railway noise. This is in detail: a major street called Provinzstraße; the footprint of the flight path for aircrafts which are departing from or arriving at Tegel airport is only 200 m away; the distance to city railway station(S-Bahn) Schönholz is only 50 m away.

At least there is a residential building with 8 floors located in this neighborhood that can be used to analyze the relationship between noise source and 3D environment. Figure 6 shows the selected neighborhood and all 3 traffic noise sources.

The noise level (Fig. 7) ranges from 45 to 58 dB(A) for road traffic. As expected noise level decreases by distance from road in horizontal direction. The noise level grows rapidly if buildings are close to road. In vertical direction, starting from first floor to the top floor, the noise level is decreasing as well.

Examining Fig. 8, which is a view from south to north, reveals that again the noise level horizontally decreases by distance from noise source but in vertical

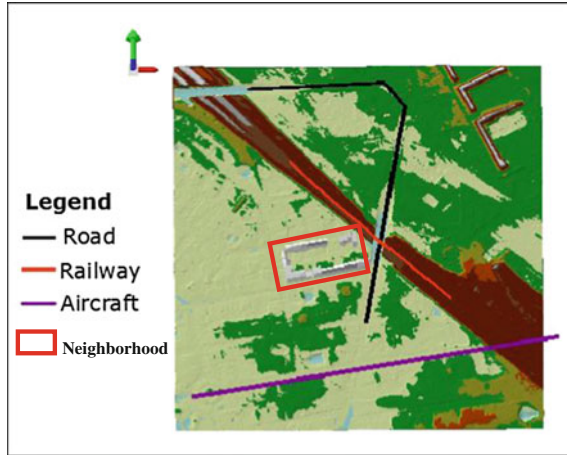


Fig. 6 Noise sources within the investigated neighborhood

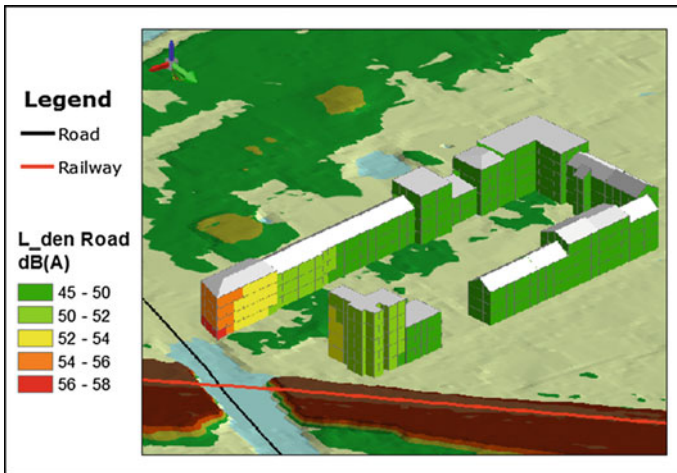


Fig. 7 Road noise map

direction it increases. There are two main reasons causing this result. The first one is the lower building next to the highest building closest to the railway road obstructs the noise propagating to the lower floors, but it has no influence to the higher floor. The second reason is the propagation path which goes over the building top. For observer points close to building roof, the length of the path is shorter than the observer points nearby ground. Thus, the noise level is increasing by height.

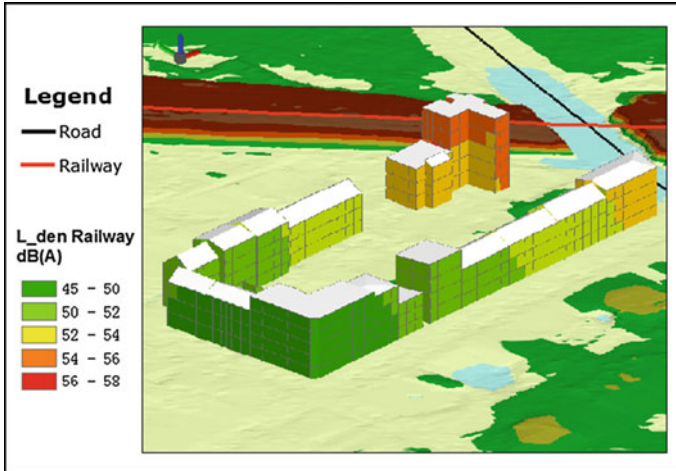


Fig. 8 Railway noise map

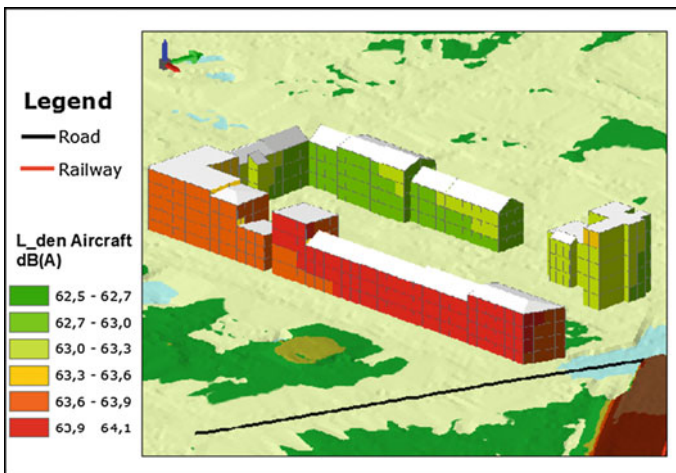


Fig. 9 Aircraft noise map

Figure 9 (aircraft) reveals that the noise level is ranging from 62.5 to 64.1 dB(A) decreasing from north-east to south-west. Tegel airport is located approx. 6 km south-west from our neighborhood.

However, the flightpath is just 250 m away and aircrafts are typically flying in a height of 300 m with a speed of 250 km/h and a descent rate of 2.5 m/s. Thus, the noise decreases by height (top-down) and by distance from flight path. Investigating the levels for total impairment by multiple sources reveals that in this neighborhood (see Fig. 10) most of the inhabitants are annoyed by aircraft noise even on the

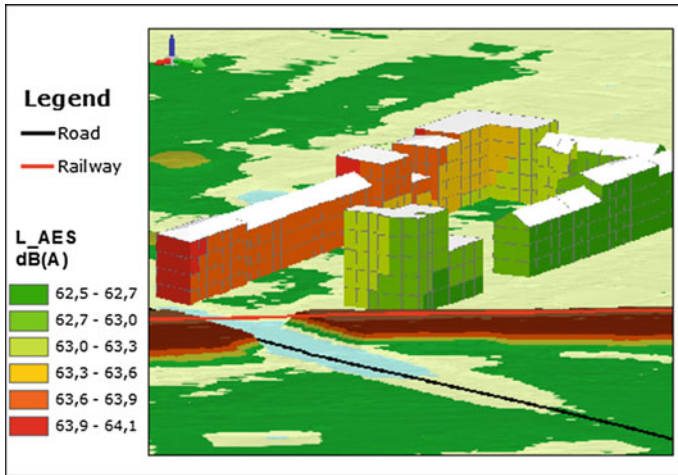


Fig. 10 3D map for effect related substituted levels L_{AES}

backside of the buildings. Especially buildings with the highest noise level are not close to a road or railway nor are flats with high noise closely located to the ground neither to the top of the building. The composition of noise and the existence of multiple 3D noise sources determine the presence or absence of high noise.

5 Conclusions

In order to check the validity of the results, the noise level for each of 3 different traffic modes as well as for the sum of all traffic modes are compared with the results from the Strategic Noise Map of Berlin 2013. In Strategic Noise Map of Berlin, noise levels were measured at 4 m height above the ground. Meanwhile, noise level for observer points at ground floor (2 m) are selected for comparison. Table 1 shows the noise levels at the same location from different noise maps.

As it can be seen from Table 1, at this observer point, the road and railway noise levels of our approach is obviously lower in comparison to the Strategic Noise Map of Berlin. Especially for railway, the noise level is more than 10 dB(A) lower. Aircraft noise levels are approximate but significantly smaller than the noise levels provided by metafly. Firther, slightly difference for the sum of all traffic noise can be observed.

Table 1 Compare noise levels from different source (dB(A))

	Road	Railway	Aircraft	Sum
3D traffic noise map	51	53	62	67
Strategic Noise Map of Berlin	58.8	65.7	61.1	66
Metafly			74.8	

As mentioned before, each noise prediction method has its own parameters and algorithms. It is hard to say which method is more accurate than the other because almost all prediction methods declare that they had the best results. The results calculated from different methods may have huge difference to each other which sometimes are ignored by researchers. According to Kevin Carr’s research (Carr et al. 2011), in the situation of reflective ground, no barrier and receptor placed with 1.5 m height, the sound level for different noise prediction methods can reach to 10 dB(A) for 300 m and 13 dB(A) for 600 m.

The main component of noise propagation is the geometric correction. Table 2 shows an overview of 5 different geometric correction algorithms.

As it can be seen from Table 2, the main difference between these algorithms besides the algorithm itself—is the source type. If the noise source is viewed as a point source, the algorithm uses $20 * \lg(s)$. For line source, it is $10 * \lg(s)$. Therefore, noise level shows different increasing ratio when the distance is increasing. It leads to totally different results: at the distance of 100 m, noise level calculated by VBUS is similar to the method in this paper, while it is almost doubled at 300 m. Definitely, noise prediction model is not only consisted of geometry correction. Other corrections like air absorption, ground reflection may reduce the noise level difference between different prediction methods. However, those algorithms illustrate that nowadays a clearly consensus of noise prediction is still missing and it is hard to calibrate noise level results with other prediction methods.

The SPL (sound pressure level) is a logarithmic measurement, which means that the weight of higher values is much higher than small values when integrating noise level from each path. For the selected observer point, the shortest distance to railway is around 20 m and to road is 50 m. According to the algorithms above, at the distance of 20 m, geometric correction value by VBUS is about 6 dB lower than the method presented in this paper, which causes that the noise level at the observer point, is 6 dB higher. Furthermore it can be proved as well by the aircraft noise: the Strategic Noise Map of Berlin uses VBUI for aircraft noise prediction and metafly uses ECAC prediction model; the difference of noise level results here are more than 10 dB(A). Therefore, the reason for a lower noise levels in this paper might be different prediction methods especially different geometry correction algorithms. To be able to evaluate the results noise measurements should be done at specific points within the city collecting data at source points, observer points and some

Table 2 Overview of different geometric correction algorithms (s in [m])

Method	Source	Algorithm	s = 20	s = 100	s = 300
			[dB]		
VBUS	Point	$20 * \lg(s) + s/200 - 11.2$	14.9	29.3	39.8
RLS	Line	$-15.8 + 10 * \lg(s) + 0.0142 * (s)^{0.9}$	-2.6	5.09	11.3
Nodic	Line	$10 * \lg(s/10)$	6.9	23	34
CNOSSON-EU	Point	$20 * \lg(s) + 11.2$	37	51	60.5
In this paper	Line	$10 * \lg(s) + 8$	21	28	32.7

intermediate points in vertical and horizontal direction. Those measurements can be compared with the results of the different determination results in order to find out which noise determination best fits. Only based on this a decision can be made which model should be used to calculate the 3D noise effects in our urban environment.

References

- Bluhm, G. L., Berglund, N., Nordling, E., et al. (2007). Road traffic noise and hypertension. *Occupational and Environmental Medicine*, 64(2), 122–126.
- Chambers, J. P. (2004). *Handbook of environmental engineering*, Volume 2: Advanced Air and Noise Pollution Control (p. 445). The Humana Press, Inc.
- Carr, K., Penton, S., & Li, M. (2011). Evaluation of road transportation noise modelling algorithms and software packages. *Canadian Acoustics*, 39(3), 84–85. Retrieved from <http://jcaa.caa-aca.ca/index.php/jcaa/article/view/2422>.
- ECAC. (2005). CEAC Doc 29; 3rd; Report on Standard Method of Computing, Noise Contours around Civil Airports, Volume 2: Technical Guide.
- European Union (Ed.). Directive 2002/49/EG of the European Parliament and of the Council of 25 June 2002 relating to the assessment and management of environmental noise, in Official Journal of the European Communities.18.7.02.
- Farçaş, F. (2008). Road traffic noise: a study of Skåne region, Sweden[D]. MSc thesis, Linköping University, Sweden.
- Kephalopoulos, S., Paviotti, M., & Ledee, F. A. (2012). Common noise assessment methods in Europe (p. 180)
- Kjellberg, A., Tesarz, M., Holmberg, K., & Landström, U. (1997). Evaluation of frequency-weighted sound level measurements for prediction of low-frequency noise annoyance. *Environment International*, 23(4), 519–527.
- Lui, W. K., et al. (2006). A comparative study of different numerical models for predicting train noise in high-rise cities. *Applied Acoustics*, 67(5), 432–449.
- Metafly. TU-Berlin <http://metafly.info/en/index.html>.
- Moehler, U., Kurze, U. J., Liepert, M., & Onnich, H. (2008). The New German Prediction Model for Railway Noise “Schall 03 2006”: An Alternative Method for the Harmonised Calculation Method Proposed in the EU Directive on Environmental Noise. *Acta Acustica united with Acustica*, 94(4), 548–552.
- Nilsson, M. E. (2007). A-weighted sound pressure level as an indicator of short-term loudness or annoyance of road-traffic sound. *Journal of Sound and Vibration*, 302(1), 197–207.
- Pennington, G., Topham, N., & Ward, R. (1990). Aircraft noise and residential property values adjacent to Manchester International Airport. *Journal of Transport Economics and Policy*, 49–59.
- Rahmatian, M., & Cockerill, L. (2004). Airport noise and residential housing valuation in southern California: A hedonic pricing approach. *International Journal of Environmental Science and Technology*, 1(1), 17–25.
- RLS-90 Richtlinien für den Schallschutz an Straßen, in BGBl. I 1990, 1037–1044.
- Schall 03 (2014). Richtlinie zur Berechnung der Schallimmissionen von Schienenwegen. Ausgabe 1990, bekannt gemacht im Amtsblatt der Deutschen Bundesbahn Nr. 14 vom 4. April 1990. Retrieved June 23, 2015 from http://www.gesetze-im-internet.de/bimschv_16/anlage_2.html.
- Steele, C. (2001). A critical review of some traffic noise prediction models. *Applied Acoustics*, 62(3), 271–287.
- VDI 3722 Blatt 2. Effects of traffic noise—Part 2: Characteristic quantities in case of impact of multiple sources.2013.05.

Highly Efficient Computer Oriented Octree Data Structure and Neighbours Search in 3D GIS

Noraidah Keling, Izham Mohamad Yusoff, Habibah Lateh
and Uznir Ujang

Abstract Three-dimensional (3D) visualization has given a new perspective in various fields such as urban planning, hydrology, infrastructure modelling and geology. This is due to its capability of handling real world object in more realistic manners, rather than the two-dimensional (2D) approach. However, implementation of 3D spatial analysis in the real world situations has proven to be difficult to comprehend due to the complexity of the algorithm, computational process and time consuming. The existing Geographical Information Systems (GIS) enable 2D and two-and-a-half-dimensional (2.5D) spatial datasets, but less capable of supporting 3D data structures. Recent development in Octree showed that more effort was given to improve the weakness of Octree in finding neighbouring nodes by using various address encoding scheme with specific rule like matrix, lookup table and arithmetic to eliminate the need of tree traversal. Therefore, the purpose of this paper is to propose a new method to speed up the neighbouring search by eliminating the needs of complex operation to extract spatial information from Octree by preserving 3D spatial information directly from the Octree data structure. This new method will be able to achieve $O(1)$ complexity and utilizing Bit Manipulation Instruction 2 (BMI2) to speed up address encoding, extraction and voxel search 1000x compared to generic implementation.

1 Introduction

Recent developments in the field of three-dimensional (3D) spatial data representation have led to a renewed interest in Geographic Information Systems (GIS). Application of 3D technologies in GIS developments is one of the issues that has

N. Keling (✉) · I. Mohamad Yusoff · H. Lateh
School of Distance Education, Universiti Sains Malaysia, George Town, Malaysia
e-mail: noraidahkeling@gmail.com

U. Ujang
Faculty of Geoinformation and Real Estate, Universiti Teknologi Malaysia,
Johor Bahru, Malaysia

been given special attention in this platform and has come to the limelight among various geo-related professionals such as land practitioners and environmentalists (Berry et al. 2008; Goodchild 2009). Perspective in third dimension is really helpful in some fields such as urban planning and management (Ujang et al. 2014), geological sub surface modelling (Li et al. 1996), mining and oil exploration (Pouliot et al. 2008; Jin et al. 2011), hydrology (Izham et al. 2011), infrastructure modelling, and geology (Zhu et al. 2012, 2013).

Apart from visualization task, 3D has given a new perception in this field due to its ability to handle complex real-world phenomena in a more realistic manner for a better recognition of geographical and geological space and to master the inherent laws in earth-science and engineering application (Tianding 2010). Currently, most of the direct and indirect approaches to effectively utilize GIS is by using GIS as a spatial database for storing, displaying and updating the input data for site-specific measurement and experiment for a better understanding of the interaction of the conditioning factor (Rolf 2004).

However, the major problem with this kind of application is the existing commercial GIS software only support 2D and 2.5D spatial datasets which is only useful to visualize adjacent surfaces with variations of height such as terrain, but it is not suitable to represent real 3D data structures. So far, the 3D functionality of GIS software which is derived from the 2.5D data extrapolation is only being used as visualization (Abdul-Rahman and Pilouk 2008). Currently the usage of 2.5D for the 3D data representation is only useful as an approximation and it does not satisfy the current need anymore (Lixin 2004; Rogers and Luna 2004).

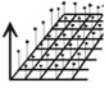
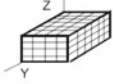





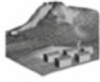

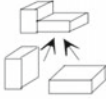






Therefore, the demand on a fully functional 3D GIS is increasing and became more and more important to the geological analysis and engineering application to understand geological phenomena and complex structure of any kind of information related with their 3D environment in a more realistic way and to carry out the complicated 3D spatial analysis (Wenzhong 2000).

There are various data representations that has been proposed for GIS and it can be categorized into three different models; surface model, volumetric model and hybrid model (Li et al. 1996; Lixin 2004). Table 1 summarizes some examples of the data representation and its classification (Wenzhong 2000a; Shen et al. 2003, 2006, 2013; Huixin and Huifeng 2006; Shen and Takara 2006; Abdul-Rahman and Pilouk 2008).

For 2D GIS, there are two types of data representations that are regularly used; Triangulated Irregular Network (TIN) and Grid. TINs is usually used by GIS software to store and display surface and boundaries due to the accuracy and relatively small database size. However, TIN is not suitable for some application, such as spatial analysis like slope stability, topographic wetness index and hydrology infiltration model. For those application, Grid data representation is usually used.

In 3D data representation, Tetrahedral Network (TEN) is actually equivalent to 3D-TIN while 3D Array is equivalent to 3D-Grid. However, 3D array is not widely

Table 1 Various data model representation

Surface Model	Volumetric Model		Hybrid Model
	Regular	Irregular	
 Grid	 3D array	 TEN	 TIN-Octree
 Shape	 Octree	 Pyramid	 TIN-CSG
 Facet	 CSG	 TP	 Octree-TEN
 B-Rep		 3D Voronoi	
 TIN		 Geocellular	

used in practical application due to the amount of memory needed to keep the data and its role is usually being replaced by the Octree. In this paper, we tried to focus on improving the Octree to be better and more practical in 3D spatial analysis by improving the neighbour search performance.

2 Methods

2.1 Related Research

Octree is a method where 3D space is being represented in a hierarchical tree structure as visualized in Fig. 1. Classical Octree is originated from Klinger (1971) as quadtree and subsequently expanded to cover 3D space as Octree. The earlier implementation of Octree used tree traversal as suggested by Samet (1989), Ballard and Brown (1982), Besançon and Faugeras (1988). Neighbours search can be done by backtracking upwards from selected node until the common ancestor is found and then travel down the tree until the neighbours are found. The advantage of this implementation is that the algorithm is simplified by exploiting recursive nature of octree, but the drawback is it require long tree traversals.

Recent research development in Octree see more effort to improve weakness of Octree in finding neighbour node which is required for spatial analysis by using various address encoding scheme with the specific rule to eliminate the need of tree traversal.

Vörös (2000) suggested that each leaf node is to be described by a unique locational code corresponding to a sequence of directional codes encoding the path from the root to that node as Fig. 2 and represented in matrix formed as shown in Fig. 3. The neighbour nodes are located by using matrix based approach.

Payeur (2006) suggested a generic set lookup table that determines the address offset from a given cell to its neighbour for each possible direction based on the observation of the addressing structure as shown in Fig. 13. 26 neighbours voxel is categorized into 3 different categories, which are face neighbour, edge neighbour and vertex neighbour. Each of these categories has its own lookup table as shown in Figs. 4, 5, and 6.

Kim and Lee (2009) suggested another method by relying on the combination of arithmetic algorithm as shown in Fig. 7 and also lookup table as in Fig. 8. The lookup table is derived based on the observation of his address scheme as shown in Fig. 14.

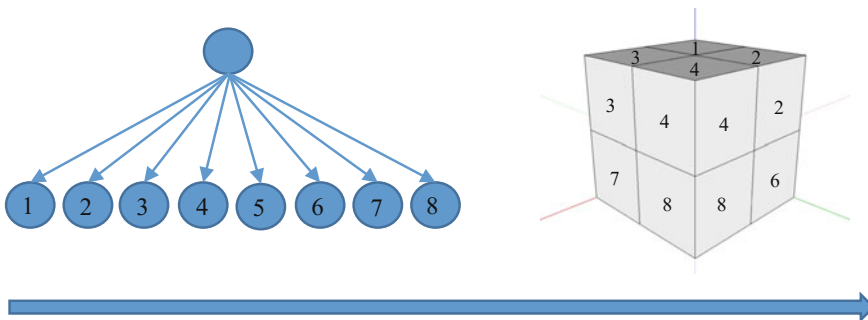


Fig. 1 Octree node to 3D space mapping

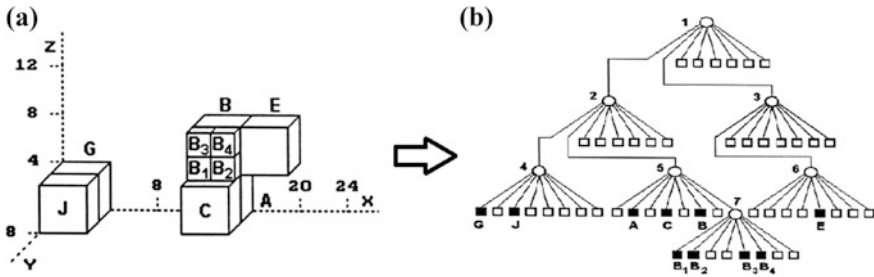


Fig. 2 Voxel/leaf locational code (Vörös 2000)

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} \quad C = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

Fig. 3 Leaf matrix representation (Vörös 2000)

		Direction					
		N	S	E	W	F	R
Initial Digit Value	0	2	2+S	1	1+W	4	4+R
	1	3	3+S	0+E	0	5	5+R
	2	0+N	0	3	3+W	6	6+R
	3	1+N	1	2+E	2	7	7+R
	4	6	6+S	5	5+W	0+F	0
	5	7	7+S	4+E	4	1+F	1
	6	4+N	4	7	7+W	2+F	2
7	5+N	5	6+E	6	3+F	3	

Fig. 4 Face neighbours definition and look-up table (Payeur 2006)

		Direction											
		NW	NE	SW	SE	FN	RN	FS	RS	FE	FW	RE	RW
Initial Digit Value	0	3+W	3	3+SW	3+S	6	6+R	6+S	6+RS	5	5+W	5+R	5+RW
	1	2	2+E	2+S	2+SE	7	7+R	7+S	7+RS	4+E	4	4+RE	4+R
	2	1+NW	1+N	1+W	1	4+N	4+RN	4	4+R	7	7+W	7+R	7+RW
	3	0+N	0+NE	0	0+E	5+N	5+RN	5	5+R	6+E	6	6+RE	6+R
	4	7+W	7	7+SW	7+S	2+F	2	2+FS	2+S	1+F	1+FW	1	1+W
	5	6	6+E	6+S	6+SE	3+F	3	3+FS	3+S	0+FE	0+F	0+E	0
	6	5+NW	5+N	5+W	5	0+FN	0+N	0+F	0	3+F	3+FW	3	3+W
7	4+N	4+NE	4	4+E	1+FN	1+N	1+F	1	2+FE	2+F	2+E	2	

Fig. 5 Edge neighbours definition and look-up table (Payeur 2006)

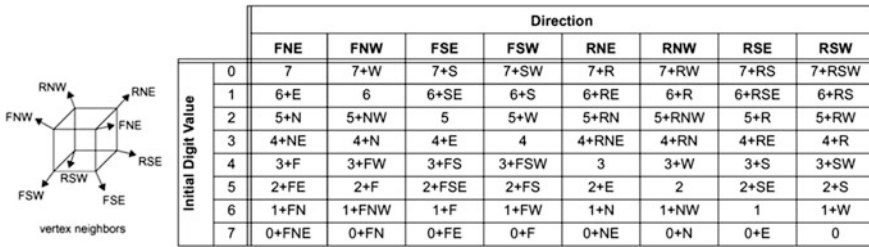


Fig. 6 Vertex neighbours definition and look-up table (Payeur 2006)

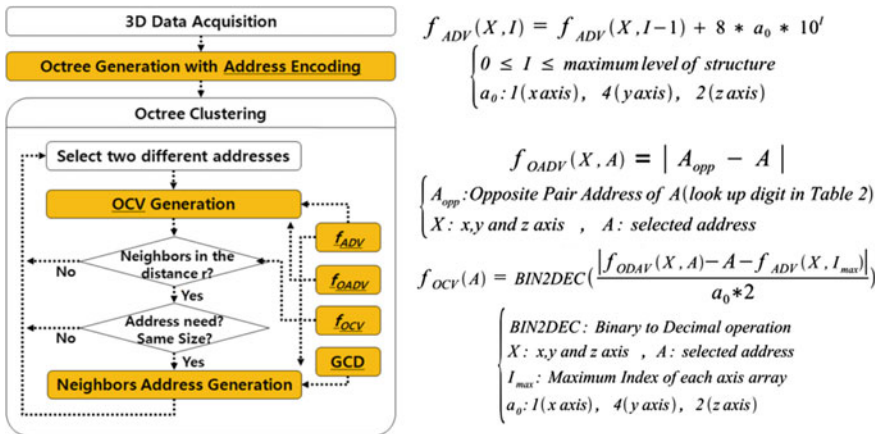


Fig. 7 Neighbour search flow and arithmetic involved (Kim and Lee 2009)

Fig. 8 Opposite pair address lookup table (Kim and Lee 2009)

Digit	X axis	Y axis	Z axis
1	2	5	3
2	1	6	4
3	4	7	1
4	3	8	2
5	6	1	7
6	5	2	8
7	8	3	5
8	7	4	6

2.2 Proposed Octree Structure

In this paper, the author is also proposing another address encoding scheme and data structure which bases on binary ordering which efficiently preserve 3D spatial information from Octree structure and carefully designed in such way to fully take

advantage of the advancement of computer CPU accelerator instruction to speed up neighbour search and eliminating the need of complex operation to derive voxel spatial information like suggested by Kim and Lee (2009), the matrix solution like Voros (2000) or lookup tables like Payeur (2006).

2.2.1 Binary Address Encoding

The basis of the proposed Octree encoding scheme is to assign each node of Octree a unique address in binary form with 3D spatial information embedded. Octree in 3D space is binary by nature. Voxel progression along with the axis can be represented by only 1 bit as shown in Fig. 9. For example, the voxel which is close to axis X, the address is set as 0 and for the voxel located further from the axis the address is set as 1. By incorporating this nature into address encoding, we can preserve all 8 voxels spatial data information by using only 3 bits.

These 3 bits can be arranged using $X_p Y_p Z_p$ standard, where P is Octree level or depth. For example, for the voxel which located in coordinate $x = 0, y = 0, z = 0$, the voxel address is 0b000 (0d0) and voxel located at coordinate $x = 1, y = 1, z = 1$ then the voxel address will be 0b111 (0d7). The addressing for the voxel is straight forward and easy to identify voxel physical location just based on Octree address. The Fig. 10 shows the complete address for proposed Octree in binary form.

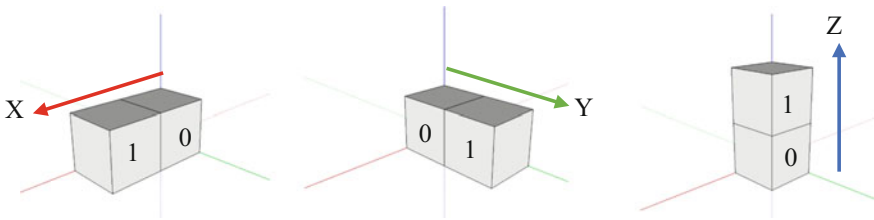
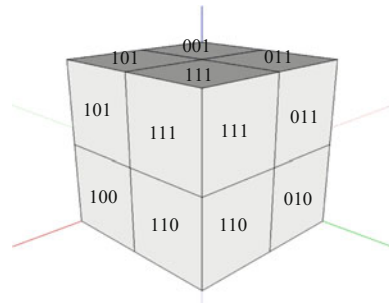


Fig. 9 Voxel numbering progression

Fig. 10 Proposed octree addressing scheme



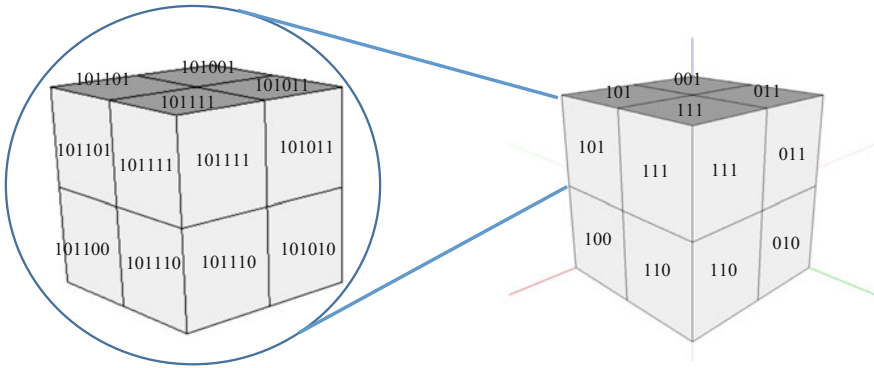


Fig. 11 Proposed octree subdivision addressing scheme (binary)

2.2.2 Subdivision Address Encoding

$$\text{Voxel Address} = (X_p Y_p Z_p) \cap (X_{p+1} Y_{p+1} Z_{p+1}) \cap \dots \cap (X_{p+n} Y_{p+n} Z_{p+n}) \quad (1)$$

The address encoding for subdivision Octree is similar with parent voxel except the parent address is directly concatenating into child voxel address as shown in (1). The main difference between proposed encoding with previous methods that was introduced by Voros (2000), Payeur (2006), Kim and Lee (2009) is this method strictly uses 3 bits binary encoding per level as shown in Figs. 11 and 12 while others are using decimal based as shown in Figs. 13 and 14. Decimal representation is only make sense for human readable for manual processing because parent address can be easily identified concatenated with child address, but to represent those number in decimal will require 25 % more space and without any advantage for computer processing.

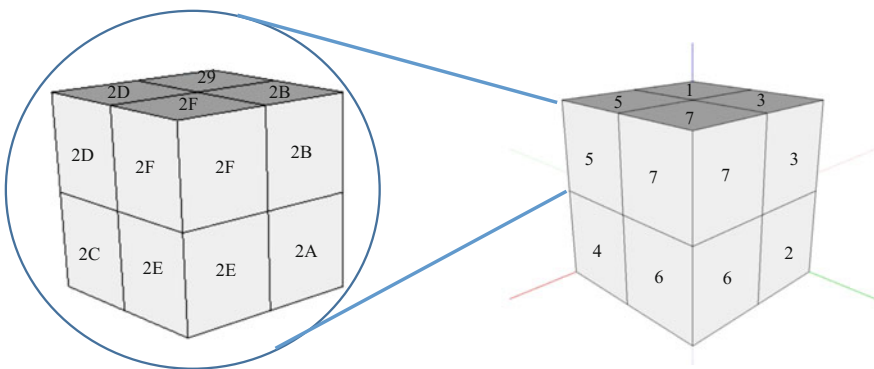


Fig. 12 Proposed octree subdivision addressing scheme (hexadecimal)

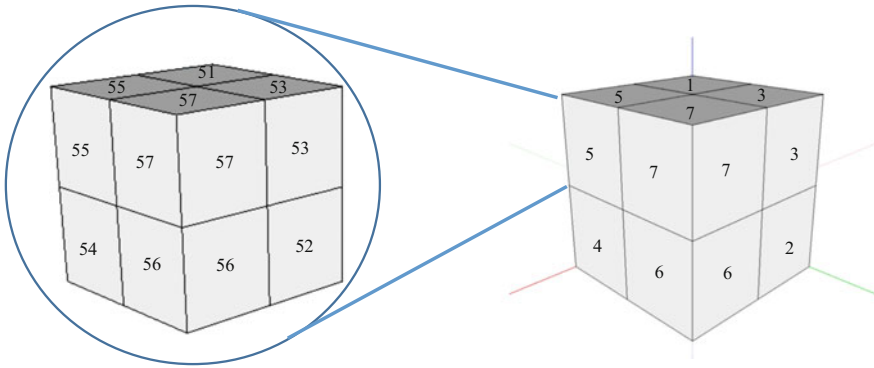


Fig. 13 Payeur and Voros octree subdivision addressing scheme

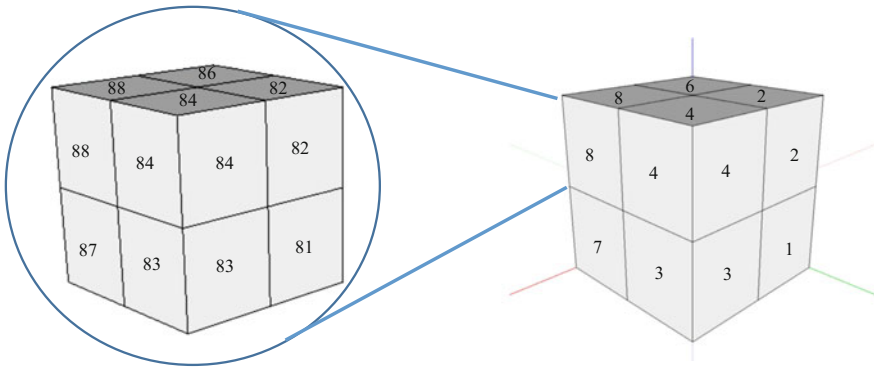


Fig. 14 Kim and Lee octree subdivision addressing scheme

2.3 Data Structure

In order to take advantage of modern computer (explained in Sect. 2.4), each voxel should be kept either in 32 bits or 64 bits. By using Eq. (2), 32 bits data structure are able to hold maximum of 10 levels of Octree and for 64 bits data structure will be able to hold up to 21 levels of Octree.

$$Maximum\ Octree\ Maximum\ level\ (P) = \lfloor \frac{Data\ Structure\ Bits}{3} \rfloor \tag{2}$$

By using Eq. (3), with the assumption of minimum resolution of surface area that we want to cover is 1 m², 32 bits data structure only able to cover up to 1,048,576 km² but 64 bits data structure is able to cover up to 4,398,046,511.104 km². With the total earth surface area is estimated only around 510,072,000 km², 64 bits data structure is practically more than enough to cover the

Fig. 15 Proposed data structure

MSB									LSB				
X ₁	Y ₁	Z ₁	X ₂	Y ₂	Z ₂	X ₃	Y ₃	Z ₃	...	X ₂₁	Y ₂₁	Z ₂₁	1

Fig. 16 Proposed data structure with partially fill up bits

MSB									LSB				
X ₁	Y ₁	Z ₁	X ₂	Y ₂	Z ₂	1	0	0	...	0	0	0	0

whole earth thus there is no need for more than 64 bits data structure. 1 m² resolution generally considered as very high resolution and enough for detail spatial analysis, for example in hydraulic and hydrological modelling (Vaze et al. 2010).

$$Covered\ Area = 2^{P*2} * Resolution \tag{3}$$

For this paper, we focus on 64 bits data structure but the same method also can be applied for 32 bits data structure. The Octree node address X_pY_pZ_p for each level should be start concatenated from MSB to LSB with 0b1 appended after last 3 bits address as terminator as shown in Figs. 15 and 16.

Numbers of Octree level can be determined by (4) and since each level requires 3 bits of information embedded into the data structure, the total of bits that needs to be used is calculated according to (5).

$$Numbers\ of\ Octree\ level\ (P) = \frac{\log_2 \left(\left(\frac{Area\ Size\ to\ Cover}{Smallest\ unit@resolution} \right) \right)}{2} \tag{4}$$

$$Bits\ Required = 3P + 1 \tag{5}$$

2.4 CPU Accelerated Instruction

Bit Manipulation Instruction 2 (BMI2) is a series of new instructions introduced by Intel as part of Advance Vector Extensions 2 (AVX 2) in 2013 (Intel 2011). The same instruction and also supported by AMD in Excavator 2015. Intel and AMD is the market leader in computing with 80 % of the total computer sold is based on Intel or AMD CPU. There are 3 instructions from BMI2 that can be used to accelerate Octree address generation and manipulation; TZCNT, PDEP and PEXT.

2.4.1 Count the Number of Trailing Zero Bits (TZCNT)

TZCNT is a single instruction to count trailing 0 in the registers. This instruction eliminates the need of bit extraction-compare loops as shown in Fig. 17. TZCNT

```

temp ← 0
DEST ← 0
DO WHILE ( (temp < OperandSize) and (SRC[ temp] = 0) )
    temp ← temp +1
    DEST ← DEST+ 1
OD
IF DEST = OperandSize
    CF ← 1
ELSE
    CF ← 0
FI
IF DEST = 0
    ZF ← 1
ELSE
    ZF ← 0
FI

```

Fig. 17 TZCNT pseudo code (Intel 2011)

can be used to identify Octree level from voxel address as demonstrate in Sect. 2.5.1.

2.4.2 Parallel Bits Extract (PEXT)

PEXT is a single instruction to insert bits into different position from sequences as shown in Fig. 18. PEXT is being used in the proposed method to reconstruct voxel address as demonstrate in Sects. 2.5.2 and 2.5.4.

```

TEMP ← SRC1;
MASK ← SRC2;
DEST ← 0 ;
m← 0, k← 0;
DO WHILE m< OperandSize
    IF MASK[ m] = 1 THEN
        DEST[ k] ← TEMP[ m];
        k ← k+ 1;
    FI
    m ← m+ 1;
OD

```

Fig. 18 PEXT pseudo code (Intel 2011)


```

TEMP ← SRC1;
MASK ← SRC2;
DEST ← 0 ;
m← 0, k← 0;
DO WHILE m< OperandSize
    IF MASK[ m] = 1 THEN
        DEST[ m] ← TEMP[ k];
        k ← k+ 1;
    FI
    m ← m+ 1;
OD
    
```

Fig. 19 PDEP pseudo code (Intel 2011)

2.4.3 Instructions, Parallel Bits Deposit (PDEP)

PDEP is a single instruction to gather bits from different position into sequences as shown in Fig. 19. PDEP is being used in the proposed method to extract X Y and Z 3D spatial coordinate from voxel address as demonstrate in Sects. 2.5.3 and 2.5.4.

2.5 Manipulation

2.5.1 Determine Octree Level from Voxel Address

Using proposed voxel address encoding and data structures, we can easily determine the Octree level from voxel address alone without requiring the whole Octree traversal. Here TZCNT can help to determine the location of termination bit as shown in Fig. 20.

After getting the termination bit location, Octree level can be determined by using (6)

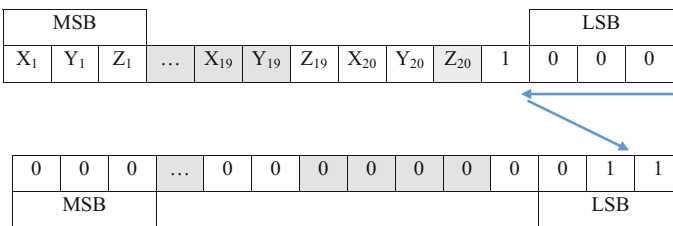


Fig. 20 Counting trailing zeros

Fig. 21 PDEP and PEXT mask generation

Xmask:
 $0x4924924924924924 \ll (TZCNT(\text{Voxel Address Register})+1)$
 Ymask:
 $0x2492492492492492 \ll (TZCNT(\text{Voxel Address Register})+1)$
 Zmask:
 $0x1249249249249249 \ll (TZCNT(\text{Voxel Address Register})+1)$

$$\text{Voxel Octree Level} = 21 - \frac{TZCNT((\text{Voxel Address Register}))}{3} \tag{6}$$

Octree level determination is important to determine the area/volume covered by respective voxel. Unlike voxel data representation, Octree voxel size is variable because Octree keeps a high resolution data only when needed in order to keep the data structure size optimized. The area covered by voxel is the inverse proportion of voxel level.

TZCNT value also will be used to generate Xmask, Ymask and Zmask for PDEP and PEXT instructions as shown in Fig. 21.

2.5.2 Voxel Address to 3D Spatial Coordinate

Our proposed method is to make the process to determine 3D spatial coordinate from voxel address simple and fast. With PEXT instruction, the extraction will be much faster. The coordinate can be easily extracted by using (7):

$$\text{Coordinate} = (PEXT(X\ mask), PEXT(Y\ mask), PEXT(Z\ mask)) \tag{7}$$

Figure 22 visualize how X axis coordinate can be extracted from voxel address. The same process is also true for Y and Z axis as shown in Figs. 23 and 24.

Fig. 22 Extracting voxel X coordinate

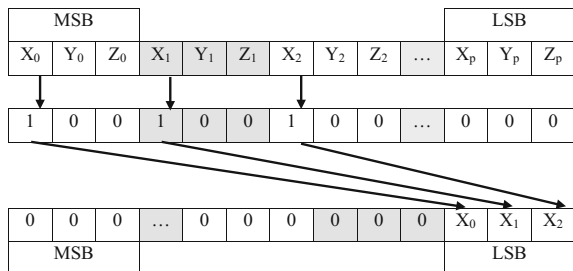


Fig. 23 Extracting voxel Y coordinate

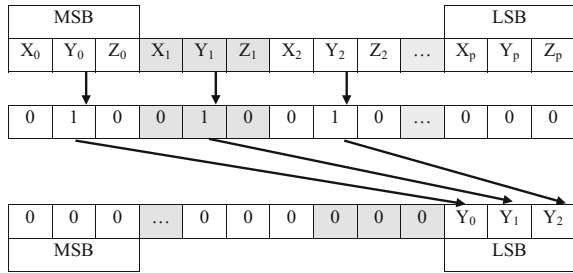
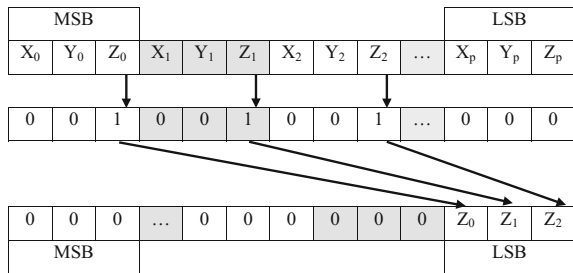


Fig. 24 Extracting voxel Z coordinate



2.5.3 3D Spatial Coordinate to Voxel Address

For application that needs voxel address generation for certain voxel coordinate like voxel neighbour discovery, PDEP can be used efficiently to reconstruct voxel address as shown in (8).

$$Voxel\ address = PDEP(X\ mask) \cup PDEP(Y\ mask) \cup PDEP(Z\ mask) \quad (8)$$

Figure 25 visualize how X axis coordinate can be converted into voxel address. The same process is also true for Y and Z axis as shown in Figs. 26 and 27.

Fig. 25 Voxel address generation for X axis

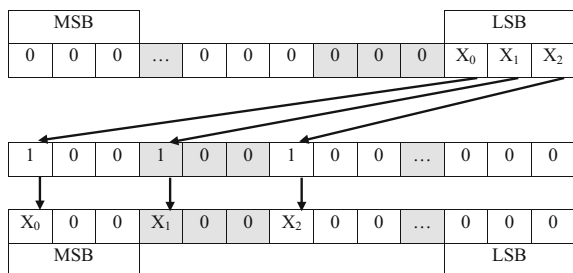


Fig. 26 Voxel address generation for Y axis

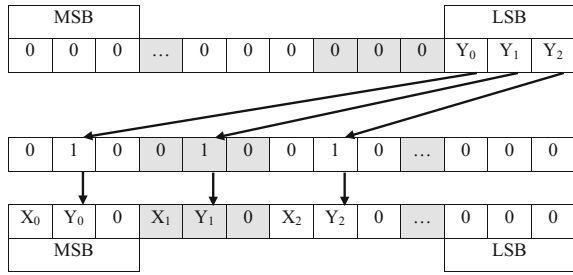
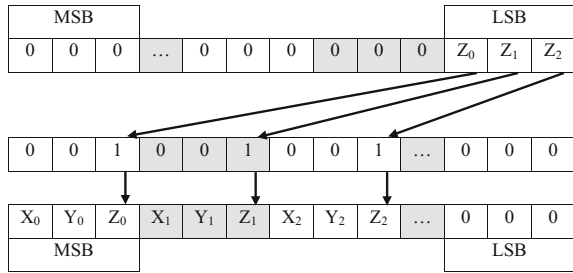


Fig. 27 Voxel address generation for Z axis



2.5.4 Finding Neighbour Voxel

Neighbour search is commonly used for spatial navigation and analysis. Finding the neighbouring search in classical Octree very resource extensive since it requires full node traverse to find each neighbour so in order to practically use Octree in those situation, address encoding like by proposed Voros (2000), Payeur (2006), Kim and Lee (2009) is favoured.

There are total of 26 neighbour voxels in an Octree and the offset of XYZ coordinate is shown in Fig. 28. Our proposed method is able to find each of them with 3 simple generic steps without involving complicated math or lookup table. For example, to get neighbour address at offset +X+Y + Z:

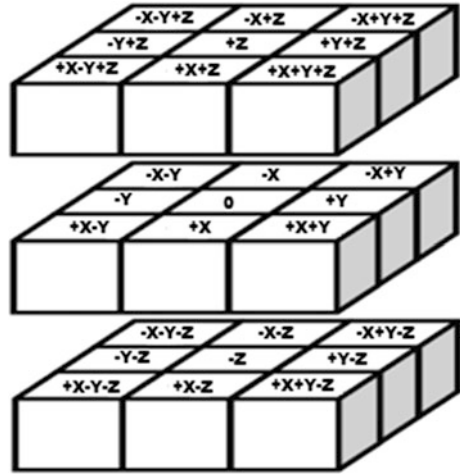
- (1) Extract voxel reference coordinate

$$\begin{aligned}
 X_{ref} &= \text{PEXT}(X \text{ mask}, \text{Voxel}_{ref}) \\
 Y_{ref} &= \text{PEXT}(Y \text{ mask}, \text{Voxel}_{ref}) \\
 Z_{ref} &= \text{PEXT}(Z \text{ mask}, \text{Voxel}_{ref})
 \end{aligned}$$

- (2) Add neighbour offset

$$\begin{aligned}
 X_{+X+Y+Z} &= X_{ref} + 1 \\
 Y_{+X+Y+Z} &= Y_{ref} + 1 \\
 Z_{+X+Y+Z} &= Z_{ref} + 1
 \end{aligned}$$

Fig. 28 Neighbours voxels



(3) Reconstruct Voxel_{+X+Y+Z} address

$$\text{Voxel}_{+X+Y+Z} = \text{PDEP}(\text{X Mask}, X_{+X+Y+Z}) \cup \text{PDEP}(\text{X Mask}, Y_{+X+Y+Z}) \cup \text{PDEP}(\text{X Mask}, Z_{+X+Y+Z})$$

Our proposed method also has a fixed complexity regardless Octree level so it is suitable for simple and huge Octree structure.

3 Results and Discussions

Classical Octree is inefficient to find neighbour address and has a complexity $O(2^{3(p-1)})$ where p represents the maximum number of resolution levels in Octree. Payuer’s (2004) method has a maximum complexity of $O(26(p-1))$ if there is no direction information given. Kim and Lee’s (2009) method has the worst case complexity $O(6(p-1))$. Gargantini (1982), Schrack (Schrack 1992) both have $O(p)$ complexity. Our proposed method is independent of Octree level and has a fixed $O(1)$ complexity. Figure 29 shows comparison of the complexity neighbour search for those alternative implementation and our proposed method.

Another advantage of having data structure and encoding that is computer-friendly is the ability to speed up address encoding, extraction and neighbour search with BMI2 instruction. It is very fast and simpler to implement compared to generic implementation. Figure 30 shows comparison of performance between generic implementation and BMI2 accelerated instruction. By average, BMI2 implementation is 1000 times faster than generic implementation.

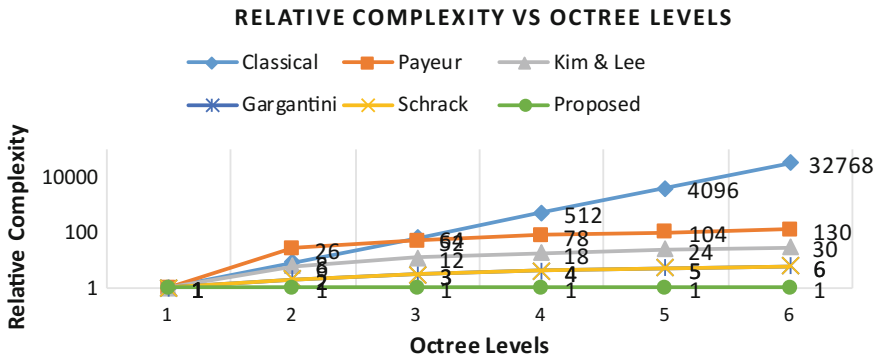


Fig. 29 Worst case complexity of neighbour search

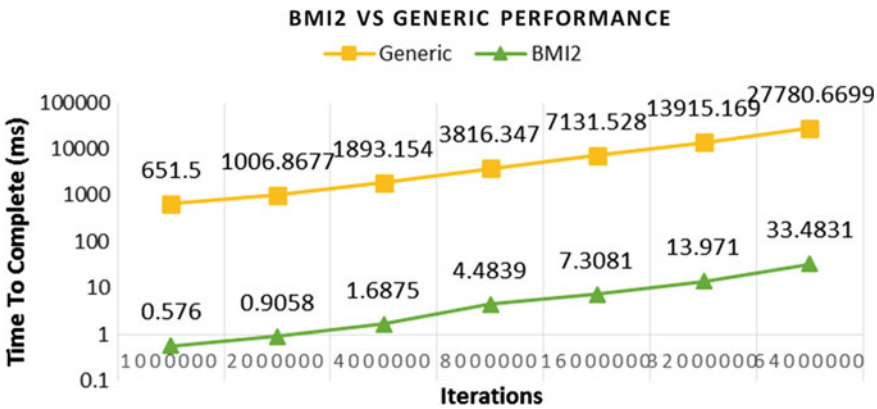


Fig. 30 Octree neighbour finding performance improvement with BMI2

4 Concluding Remarks

The key for faster and simple Octree address encoding scheme is the use of binary form. Octree in 3D space is binary compatible and by using the clever rule, 3D spatial information can be easily embedded inside the address itself and can be constructed, extracted and manipulated without using complex post processing. We have proposed a new 64 bits data structure which is designed with BMI2 compatible and introduced a simple and fast method for neighbouring search which is having constant complexity regardless Octree level. This method can be expanded in the future for neighbouring search aimed at different intra Octree level.

Acknowledgments The authors would like to thank the reviewers for their valuable comments and suggestions significantly improving this paper. The author also gratefully acknowledges financial support from the Ministry of Higher Education, Malaysia—Japan International

Cooperation Agency (MOHE-JICA) via grant scheme No. 203/PJJAUH/6711279 and UTM Research University Grant, Vote J.130000.2427.02G77 for their support and funding for this research work.

References

- Abdul-Rahman, A., & Pilouk, M. (2008). *Spatial data modelling for 3D GIS*. Springer.
- Ballard, D., & Brown, C. (1982). Computer vision.
- Berry, B. J. L., Griffith, D. A., & Tiefelsdorf, M. R. (2008). From spatial analysis to geospatial. *Science*, *40*, 229–238.
- Besaçon, J., & Faugeras, O. (1988). Vision par ordinateur en deux et trois dimensions.
- Gargantini, I. (1982). Linear octrees for fast processing of three-dimensional objects.
- Goodchild, M. F. (2009). Geographic information systems and science: today and tomorrow. *Annals of GIS*, *15*, 3–9.
- Huixin, W.U., & Huifeng, X.U.E. (2006). A new hybrid data structure for 3D GIS. In *First international conference on innovation in computing and information control* (Vol. 1, pp. 162–166). doi:[10.1109/ICICIC.2006.15](https://doi.org/10.1109/ICICIC.2006.15).
- Intel, I. (2011). *Advanced vector extensions programming reference*.
- Izham, M. Y., Muhamad Uznir, U., Alias, A. R., et al. (2011). Influence of georeference for saturated excess overland flow modelling using 3D volumetric soft geo-objects. *Computers and Geosciences*, *37*, 598–609. doi:[10.1016/j.cageo.2010.05.013](https://doi.org/10.1016/j.cageo.2010.05.013).
- Jin, B., Fang, Y., & Song, W. (2011). 3D visualization model and key techniques for digital mine. *Transactions of the Nonferrous Metals Society of China*, *21*, s748–s752. doi:[10.1016/S1003-6326\(12\)61674-4](https://doi.org/10.1016/S1003-6326(12)61674-4).
- Kim, J., & Lee, S. (2009). Fast neighbor cells finding method for multiple octree representation. *IEEE International Symposium on Comput Intelligence Robotics and Automation, 2009*, 540–545.
- Klinger, A. (1971). *Patterns and search statistics*.
- Lixin, W. (2004). Topological relations embodied in a generalized tri-prism (GTP) model for a 3D geoscience modeling system. *Computers and Geosciences*, *30*, 405–418. doi:[10.1016/j.cageo.2003.06.005](https://doi.org/10.1016/j.cageo.2003.06.005).
- Payeur, P. (2004). An optimized computational technique for free space localization in 3-D virtual representations of complex environments. In *2004 IEEE Symposium on Virtual Environ Human-Computer Interfaces Measurement Systems 2004 (VCIMS)* (pp. 1–7). doi:[10.1109/VECIMS.2004.1397175](https://doi.org/10.1109/VECIMS.2004.1397175).
- Payeur, P. (2006). A computational technique for free space localization in 3-D multiresolution probabilistic environment models. *IEEE Transactions on Instrumentation and Measurement*, *55*, 1734–1746. doi:[10.1109/TIM.2006.881028](https://doi.org/10.1109/TIM.2006.881028).
- Pouliot, J., Bédard, K., Kirkwood, D., & Lachance, B. (2008). Reasoning about geological space: Coupling 3D geomodels and topological queries as an aid to spatial data selection. *Computers and Geosciences*, *34*, 529–541. doi:[10.1016/j.cageo.2007.06.002](https://doi.org/10.1016/j.cageo.2007.06.002).
- Li, R., Chen, Y., Dong, F., & Qian, L. (1996). 3D data structures and applications in geological subsurface modeling. In *International archives of photogrammetry and remote sensing* (Vol. XXXI, Part B4, pp. 508–513). Vienna.
- Rogers, J.D., & Luna, R. (2004). *Impact of geographical information systems on geotechnical engineering* (pp. 1–23).
- Rolf, A.D. (2004). *Principles of geographic information systems—An introductory textbook*.
- Samet, H. (1989). Neighbor finding in images represented by octrees. *Computer Vision, Graphics and Image Processing*, *45*, 400. doi:[10.1016/0734-189X\(89\)90099-6](https://doi.org/10.1016/0734-189X(89)90099-6).
- Schrack, G. (1992). *Finding neighbors of equal size in linear quadtrees and octrees in constant time*.

- Shen, D., & Takara, K. (2006). A modelling and 3-D simulation system for water erosion on hillslopes—M3DSSWEH. *Journal of Hydraulic Research*, *44*, 674–681. doi:[10.1080/00221686.2006.9521716](https://doi.org/10.1080/00221686.2006.9521716).
- Shen, D., Wong, D. W., Camelli, F., & Liu, Y. (2013). An ArcScene plug-in for volumetric data conversion, modeling and spatial analysis. *Computers and Geosciences*, *61*, 104–115. doi:[10.1016/j.cageo.2013.08.004](https://doi.org/10.1016/j.cageo.2013.08.004).
- Shen, D. Y., Ma, A. N., Lin, H., et al. (2003). A new approach for simulating water erosion on hillslopes. *International Journal of Remote Sensing*, *24*, 2819–2835.
- Shen, D. Y., Takara, K., Tachikawa, Y., & Liu, Y. L. (2006). 3D simulation of soft geo-objects. *International Journal of Geographical Information Science*, *20*, 261–271. doi:[10.1080/13658810500287149](https://doi.org/10.1080/13658810500287149).
- Tianding, H. (2010). 3D GIS interactive editing method: Research and application in glaciology. *Science and Engineering (ICISE)*, 2010 2nd, (pp. 1–4).
- Ujang, U., Rahman, A. A., Anton, F. (2014). *An Approach of Instigating 3D City Model s in Urban Air Pollution Modeling for Sustainable Urban Development in Malaysia An Approach of Instigating 3D City Model s in Urban Air Pollution Modeling f or Sustainability Urban Development in Malaysia* (pp. 1–22).
- Vaze, J., Teng, J., & Spencer, G. (2010). Impact of DEM accuracy and resolution on topographic indices. *Environmental Modelling and Software*, *25*, 1086–1098. doi:[10.1016/j.envsoft.2010.03.014](https://doi.org/10.1016/j.envsoft.2010.03.014).
- Vörös, J. (2000). Strategy for repetitive neighbor finding in octree representations. *Image and Vision Computing*, *18*, 1085–1091. doi:[10.1016/S0262-8856\(00\)00049-4](https://doi.org/10.1016/S0262-8856(00)00049-4).
- Wenzhong, S. (2000a). Development of a hybrid model for three-dimensional GIS. *Geo-Spatial Information Science*, *3*, 6–12.
- Wenzhong, S. H. I. (2000b). Development of a hybrid model for three-dimensional. *GIS Geo-Spatial Information Science*, *3*, 6–12.
- Zhu, L., Zhang, C., Li, M., et al. (2012). Building 3D solid models of sedimentary stratigraphic systems from borehole data: An automatic method and case studies. *Engineering Geology*, *127*, 1–13. doi:[10.1016/j.enggeo.2011.12.001](https://doi.org/10.1016/j.enggeo.2011.12.001).
- Zhu, L. F., Li, M. J., Li, C. L., et al. (2013). Coupled modeling between geological structure fields and property parameter fields in 3D engineering geological space. *Engineering Geology*, *167*, 105–116. doi:[10.1016/j.enggeo.2013.10.016](https://doi.org/10.1016/j.enggeo.2013.10.016).

Framework for on an Open 3D Urban Analysis

Marc-O. Löwner and Thomas Becker

Abstract The Open Geospatial Consortium (OGC) is the driving organization for the specification of Web Processing Services (WPS) that enables distributed execution of computing processes. However, WPS are not widely distributed in the field of urban analysis. We identified economic relevance of projects that hinder an open source like publication, complexity of projects and, therefore, limited reusability of code and algorithms from these projects, very specific and geographically bound projects, and, a lack of publication culture for the reluctance of publishing open accessible WPS. Here we present a general framework for the publication or reusable WPS that enables both, the full reusability of algorithms and the possibility to develop business models. A small scaled WPS, the *AbstractKeyProcesses* like *Algorithm*, *Transformer* and *Inquirer* are introduced for the publication of open accessible, atomic and reusable services. Further aggregated and more advanced analysis can be published by the use of *KnowledgeBasedChainedProcesses* that may be published either as open access or represent already business models. In a last step even complete projects may be implemented as so called *ProblemSolvers*. Being aware not to present a fully elaborated standard, we give information about the components, transfer values and communication in a conceptual way. Further, we discuss the benefits of this approach along an example of relevance in the field of urban analysis.

1 Introduction and Problem Statement

Today, 54 % of the world's population lives in urban areas all over the world. This proportion is expected to increase to 66 % by 2050 (United Nations 2014). Urbanization in general is accompanied by environmental and energy problems

M.-O. Löwner (✉) · T. Becker
Institute for Geodesy and Geoinformation Science,
Technische Universität Berlin, Berlin, Germany
e-mail: m.loewner@tu-berlin.de

T. Becker
e-mail: thomas.becker@tu-berlin.de

resulting from different needs of people living, working and moving in a relatively small area. However, cities can be viewed as a feed backed system of different interacting players as well as the natural and build up environment. Geoformation technology can play a crucial role for the analysis and management of such systems. Hereby, analysis can be viewed as a supporting method to understand spatial interrelationships of human actors, infrastructure and natural resources. Therefore, urban analysis should cover the components of data preparation, database modelling and system description as well as spatial simulation processes. A precondition for the analysis of complex urban systems is the exchange of data and services that are processing those data.

The Open Geospatial Consortium (OGC) is the driving organization for the specification of international standards for the seamless exchange of 3D geo data. Including languages, data models, and middle ware technology specification, OGC has released 46 accepted international standards, until now. As a base language for the exchange of geographical data the Geography Markup Language (GML) has been released (Portele 2007). In the field of urban analysis one could name the application schema CityGML (Gröger et al. 2012) for the exchange of semantically enriched virtual 3D city models in a row with the Keyhole Markup Language (KML) (Wilson 2007) for the exchange of 3D geometries. Technologies supporting the exchange of data include the Web Map Service (WMS) (de la Beaujardiere 2006) and the Web Feature Service (WFS) (Vretanos 2010). While WFS needs to be applied using GML, CityGML is the most important application model in the field of 3D city modelling and a good starting point for urban analysis.

The OGC provides specification of the Web Processing Service (WPS) that ‘enables the execution of computing processes and the retrieval of metadata describing their purpose and functionality [...] from quick to complex computation scenarios’ (Mueller and Pross 2014). Therefor it provides rules how a client can request the execution of a process, and how the output from the process is handled. It serves as an interface that facilitates the publishing and client’s discovery of geospatial processes and the binding to those processes. Data required by a specific WPS can be delivered across a network by the client or another remote resource or it can be available at the host server of the WPS. As a result, a WPS serves as an important component of the Service Oriented Architecture (SOA) that is one main concept of the Internet 2.0 (Krafzig et al. 2004).

Using a Web Processing Service offers major and general advantages that are desirable for both, geospatial analysis and generic software design. A WPS exploits the power of distributed computing by the efficient utilization of distributed computing and storage resources. It offers an “up to date” calculation and is reusable as the code is not directly visible but the service can be reused. As a web based service it offers kind of flexibility as it allows to change internal code without affecting client applications. Further, it is meant to be scalable from a simple operation to model calculations that requires a super computer.

Advantages of the general definition of a Web Processing Service and its architecture can be brought to bear in the field of urban analysis. We identified three main reasons for that.

1. Until now, no stand alone software system is available that would be able to comprehensively deal with semantically enriched 3D city models as a precondition for urban analysis or, that are capably to extensively support the work chain associated with those models. Traditional, commercial systems have been developed from a 2D point of view and are either not able to deal with GML based data or support just a part of the solution. The establishment of linkable tools as a service oriented architecture could counterbalance this disadvantage.
2. Since CityGML as a common information model and encoding standard for the representation, storage, and exchange of virtual 3D city and landscape models (Gröger et al. 2012) has been commonly accepted, the number of available city models and their applications has increased significantly in the last ten years (Löwner et al. 2013). These models normally are provided either as a web resource via an OGC Web Feature Service (WFS), or as an internet capable GML-File. The same accounts for geometric models that are supported in form of Keyhole Markup Language (KML) as valuable base data. Having web resources available the obstacle of using web based services is not excessively high.
3. Next to 3D city models an increasing number of semantical or 2D geometrical datasets are provided by public administrations and others via the internet. As a main source for data refinement and urban analysis, this data could easily be integrated in a WPS based process chain.

Despite the added value of OGC Web Processing Services and the good preconditions in the field of urban analysis, there is a great reluctance to apply these technologies and to publish solutions in this field using OGC related standards like WFS and WPS. A general standard process chain for urban analysis tasks is not in sight.

The question arises why we do not make use of open services or do not create open services for recurring methods? Which reason prevents the community to create and open up such services? Is there a lack in OGC open standards or at least in the best practice when applying them?

In this contribution we reflect why we do not make use of open services or do not create open services for recurring methods in the field of urban analysis by reviewing recently published projects (Sect. 2). General requirements for an integrated urban analysis platform are defined in Sect. 3 followed by the development of a general architecture of a 3D urban analysis platform, its components, transfer values, and communication in Sect. 4. We will end up with a discussion in Sect. 5.

2 Reasons for the Refusal of WPS and Its Consequences

Over the last years many research papers were presented on the 3D Geoinfo conference covering a lot of hot topics related to urban analysis, city modeling, validity checks of CityGML documents, change detection in city models and much more.

Each of them represents a need in the field of 3D geoinformation science. However, most of them act like a proprietary system. They make use of well-known algorithms in the field of GIS but they do not offer them as a service. They are implemented on its own again and again or are used inside commercial software systems such as FME, ESRI ArcGIS and so on. The benefit for the whole community would be much higher if at least key processes in the field of GIS would be reusable. To give some examples for such key processes we have evaluated and investigated some papers presented and published on the 3D Geoinfo conferences over the last years.

Wagner et al. (2013) introduce a geometrical-semantical consistency check for CityGML Models. Their approach covers a lot of geometric validity checks, such as validity of LinearRing and Polygon, checks for planarity, checks for self-intersection and more. On top of that the healing of those inconsistent geometries would be possible if those checks would be made available as key processes. Everybody would be able to make use of them without implementing them again and again. That would save development time and would push a lot of research projects significantly.

Atila et al. (2013) as well as Becker et al. (2009) and Arendholz and Becker (2015) presented approaches for the design and the application of indoor navigation. All these papers and some others as well make use of a GIS method to transform a Solid into a point for using it with their graph structure for navigation purposes. However, they just use proprietary software tools for generating the points and the graph structure as well. If Becker et al. (2009) would have created a service for their approach the two following papers could have made use of it and would have saved a lot of time.

Geiger et al. (2015) presented a generalization process from IFC to CityGML using a lot of GIS related operations and algorithms such as extrusion, solid to boundary conversion and subtraction. Implementing these operations as single processes and services would boost further developments going in the same direction and would open the development steps to a broader community.

Furthermore, Butwilowski et al. (2015) presented an approach for Modeling and managing topology for track planning that makes use of a decomposition concept that decomposes connected 3D objects into connected simplicial complexes of lower dimension such as triangle nets, polylines and point sets. The same is valid the other way round. They presented an approach to aggregate single connected cells into a single cell by their common outer boundary. Both processes would be helpful to a lot of applications in 3D application and analysis if they would be open and easy accessible.

Pedrinis et al. (2015) and Redweik and Becker (2015) presented methods to detect changes in CityGML documents or City models, respectively. Both papers make use of spatial operations to detect those changes. Redweik and Becker (2015) make use of overlapping and Hough transformation to detect the changes. Pedrinis et al. (2015) use a way more spatial algorithms such as projection into xy-plane, merging algorithm, intersection, calculation of Hausdorff distance, extrusion from

footprint to 3D solid, splitting polygons, etc. to detect the changes between two timely different instance of a city model.

Next to the above discussed contributions, more examples could be evaluated in the field of urban environmental studies like noise propagation (Czerwinski et al. 2007; Lu et al. in press.), fine dust distribution modelling (Ghassoun et al. 2015b), Energy consumption and production estimation (Kaden et al. 2013; Baumanns and Löwner 2009; Strzalka et al. 2012) or urban specific navigation issues (Löwner et al. 2010). However, it is obvious that a lot of research activities especially in the field of 3D geoinformation sciences make use of frequently recurring routine algorithms. Even by reviewing the aforementioned papers we reveal that some methods are reused again and again. Some of them are *extrusion from footprint to 3D solid*, *3D solid to footprint*, *solid to point*, as well as merging operations and intersection operations. The question arises why recurring methods are not published as open services like the OGC WPS. Since the OGC open standard initiative seems technically to be successful, a best practice of applying service oriented architecture for free usage seems to be questionable.

We identified four possible reasons for the refusal of publishing the discussed projects, and, naturally further approaches, in the form of a Web Processing Service.

1. Projects in the field of urban analysis as well as almost all large-scaled projects in the field of GI have more or less economic relevance. This accounts for both, the working groups, which carried out the project and the principal. Since the working groups are often concerned with keeping their exclusive expert knowledge, principals constitute conditions for the transfer of results or base data to other parties. However, since data and algorithms are separated, the last point may account only for a part of the problem.
2. In complex projects a lot of additional expertise from experts or literature must be provided. Sometimes also simplifying assumptions are required. However, this additional information could be interpreted in a different way if another project should be carried out. Therefore, reusability of code and algorithms from these complex projects is limited.
3. In general, research questions and projects of a certain complexity require data that fit to this problem or they need preparation as well as intermediate procedures. Furthermore, sometimes objectives are very specific and geographically bound to a specific location or city.
4. There is still a lack of publication culture. Next to the analysis results this also incorporates the implemented algorithms and processes in form of a WPS. Not only solutions for the whole complex research question are not being published, also intermediate states and processes are hidden.

To sum up, the main reasons for the refusal of publishing OGC WPS is a commercial nature of the projects and the fact that these processes are of quite a complexity. Furthermore, a best practice to solve those problems is missing in the scientific and engineering community as well as in the OGC.

Refusal of publishing OGC Web Processing Services is not without consequences, especially in the field of urban analysis since this field is a new and very synoptically one. Here we give an overview of the most severe consequences in the field of urban analysis.

First of all, results of complex urban analysis projects and their possible solution remain opaque and incomprehensible. Results, data, as well as code and algorithms are just handed over to the principal but not to the scientific community. This results in first, an unclear situation for the customer and the scientific community and second, in a slowed development.

Reusability of results is severely restricted. Because some of the results are achieved by chaining many small scaled tools together, this accounts for the investigating organization as well as, of course, for the scientific community outside this organization. Furthermore, reusability is strictly coupled to expert knowledge and experience of the researcher or engineer of the investigating organization. As a result, repeatability has a very poor half-life even in the investigating organization.

Results are published in the form of a report, if published at all, and not in the form of reusable code or detailed algorithm description. The sentence “can easily be reprogrammed” is not uncommon in technical publications ignoring especially negative results, setbacks and simplifications (Fanelli 2012). As a result, threshold for the conduction of urban analysis studies is comparably high. Because analyses that already have been carried out and code that already has been programmed, have to be reproduced in a time consuming and cost intensive way. That accounts for advanced analysis steps as well as for simple but specific geometrically algorithms.

As a consequence, urban analysis is knowingly slowed down with all the negative impacts including open questions of organizing an increasingly urbanized society.

3 Requirements for an Integrated Urban Analysis Platform

An integrated urban analysis platform that is based on a web based publication of distributed OGC Web Processing Service is a desirable condition in an increasingly urbanized society of the 21th century. As discussed above, provision of analysis tools that would constitute a substantial part of such a web based platform is subject to many restrictions. However, resulting consequences are unacceptable. This is in particular the case as with the publication of the OGC Web Processing Service basic technology is already available and partly used.

The call for an integrated urban analysis platform based on OGC WPS/WFS is worthless, if it is not based on conditions that are required to help this call to success. These general requirements are both, technically and economically motivated and are as follows:

1. Published processes of an integrated urban analysis platform have to be strictly generic and reusable. They have to be designed in a way that expert knowledge of the project in which they have been designed is not needed for using them. Furthermore, they need to support the user's process chain without limiting its purpose. It follows that published processes need to be organized in as smallest units as possible.
2. A technology or protocol needs to be established that supports the possibility to aggregate the aforementioned, smaller process units to more powerful tools. Therefore, existing technical protocols or standards must be eventually adopted and further be developed.
3. Next to freely accessible smaller tools under f.i. the GNU Lesser General Public License (LGPL), business models still must be possible in the overall framework of an integrated urban analysis platform. In particular this accounts for complex projects that require extensive expert knowledge next to GIT algorithms and are financed by private or municipal stakeholders.
4. All these concepts must be developed on the basis of well-known and accepted international standards, i.e. the OGC WPS, WFS and, as the main language, the Geography Markup Language (GML). However, as an extension to a generally accepted data exchange format, a more general data transfer logic has to be accepted to overcome semantical inconsistencies (r.f. Bishr 1998).

To sum up, general concepts like the OGC WPS are already available to fulfill the above mentioned needs. However, it seems that the general definition of OGC WPS is somehow to open to support an integrated urban analysis platform.

4 General Framework of a 3D Urban Analysis Platform

We now broadly outline the architecture of an integrated urban analysis platform that fulfils the requirements that have been developed from problems discussed in Sects. 2 and 3. The proposed architecture is not intended to be a formalized protocol or standard but can be viewed as an essential step towards a memorandum of understanding in the field of chained web processing architecture. It defines basic aspects of such an architecture that helps to fulfill the above discussed needs. Please note that all of the introduced components are Web Processing Services according to the OGC.

4.1 Components

AbstractKeyProcess: An *AbstractKeyProcess* (rf. Fig. 1) is the smallest possible entity in the integrated urban analysis platform. It should be neither capable to cache intermediate data nor to process complex requests, i.e. process chains.

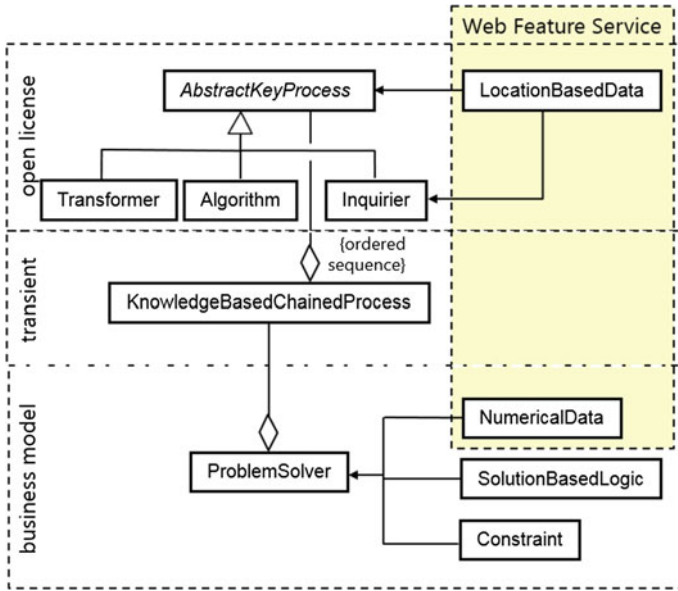


Fig. 1 General architecture of a 3D urban analysis platform (see text for further explanation)

Furthermore, it must not hold references to other *AbstractKeyProcesses* or their instantiations. As an input, location based data has to be provided, either as data encapsulated in the request or as reference. As meant to be the instantiation of an OGC WPS, an *AbstractKeyProcess* has to be either read or write GML. Furthermore, an *AbstractKeyProcess* should be reusable for any complex operation without restricting it. It is conceived to fall into the domain of open source. Thus, it should be assessable for all other users.

Note that below the specialisations of *AbstractKeyProcesses*, i.e. *Transformer*, *Algorithm* and *Inquirer* may abbreviatory be called *KeyProcesses*.

Transformer: A *Transformer* as a specialisation of an *AbstractKeyProcess* is a tool that ensures the use of GML in the proposed urban analysis platform. Since GML is the only accepted data exchange format for an algorithm (see below) its only purpose is to either transform non GML based data into or non GML based data from GML to ensure data communication. Please note that only one of these transformations should be provided by one transformer to keep the constraint of reusability. An example of this would be the transformation of a 2D shape file into a *GM_MultiSurface*. Because GML2 geometry concepts differ from that of GML3 the transformer should clearly state the geometry type of its transformation, f.i. whether it transforms to a *GM_Aggregate*, *GM_Complex* or a *GM_CompositeSurface*. In no case a transformer should leave the user in abeyance about that. Besides, all of this would be different instances of a *Transformer*.

Algorithm: An *Algorithm* as a specialization of an *AbstractKeyProcess* is the smallest possible process with relation to GIS. It should answer one and only one

simple question in the field of spatial analysis, irrespective how complex the performing algorithm will look like.

An example of this would be the creation of a 3D buffer analysis that has been implemented by Ghassoun et al. (2015a) in the field of fine dust distribution modelling but may be usable for other applications (rf. Fig. 2). It needs a point as a location and two parameters determine the radius and the height as necessary input and GML3 geometry as an output. No further analysis like clipping or data query should be provided by this *Algorithm* ensuring perfect reusability.

Inquirer: Service oriented spatial analysis needs to query data from other resources, e.g. 3D City models. An *Inquirer* as a specialization of an *AbstractKeyProcess* is meant to support such a query that can be implemented in whatever query language that is supported by the service providing the data. Although for instance a Web Feature Service provides advanced spatial query operations, an *Inquirer* should supply only the most atomic query that is suitable. Therefore, only a data source and needed parameter should be enter an *Inquirer*, which makes it largely comparable with an *Algorithm*. Take the 3D Buffer as an example (Fig. 2). If the height of this buffer is meant to be the average of the building heights of the clipped buildings within the 2D buffer (actually this is a demand concerning a parametrization of porosity in the field of fine dust distribution modelling) only building features are requested and send to an *Algorithm* that is performing the average height from building features as an input for the 3D buffer algorithm.

KnowledgeBasedChainedProcess: A *KnowledgeBasedChainedProcess* is a component that organizes the chaining of *AbstractKeyProcesses* or its specializations. It is therefore realized as an aggregation of an ordered sequence of *AbstractKeyProcesses*. It does not provide any *AbstractKeyProcesses* itself, which also excludes *Transformers*. However, only references to already published *KeyProcesses* are provided. This will be implemented as links to WPS dealing as such a *KeyProcesses*. Next to this, input and output data is defined.

The calculation of a 3D buffer with respect to the heights of the surrounding buildings could serve as an example of a *KnowledgeBasedChainedProcess*. As a first step, reference to a 2D buffer calculation algorithm (from *AbstractKeyProcess*) is needed. Then a clipping operation with a 2D buffer and a (CityGML) dataset is

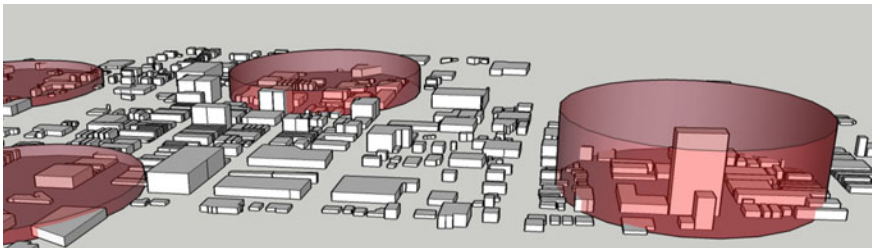


Fig. 2 Example of 3D buffers that could be the result of an *Algorithm* (Ghassoun et al. 2015a)

performed applying an *Inquirer*. After this, calculation of a mean or average height of the buildings needs to be performed, followed by the calculation of the aforementioned calculation of a 3D buffer. Hence, at least four instances of *AbstractKeyProcesses*, i.e. three *Algorithms* and one *Inquirer* are involved and aggregated by this specific *KnowledgeBasedChainedProcess*.

A *KnowledgeBasedChainedProcess* is expected to fall into the domain of open source but may also be part of a business model if run by a commercial company. It performs analysis steps that needs the specific knowledge of trained GI experts but may look like simple algorithms.

ProblemSolver: A *ProblemSolver* defines the level of a project that solves problems of social relevance like complex fine dust distribution modelling or urban energy consumption estimation. In addition to aggregated *KnowledgeBasedChainedProcesses* it also involves further numerical data, solution based logic and constraints that may come from experts, internal software systems, and data sources. In contrast to a *KnowledgeBasedChainedProcess* a *ProblemSolver* does not only aggregate specializations of *AbstractKeyProcesses* that are published over the internet but may also link to internal expert systems and software components.

It is hardly conceivable that a *ProblemSolver* will be published as a Web Processing Service, but is fully be viewed as a business model with the already discussed restrictions on publication. Nevertheless, by naming the incorporated and freely available *AbstractKeyProcesses* and *KnowledgeBasedChainedProcesses* used, results may be comprehensible in a better way. Therefore, a *ProblemSolver* also serves as an important component for the analysis of urban environments. In addition, it is expected that *KeyProcesses* and *KnowledgeBasedChainedProcesses* will be developed throughout the project that can be reused by other researchers.

4.2 Transfer Values

Next to the well-known capabilities of the OGC Web Processing Service additional considerations on communication issues have to be made. Again, this is not a proposal for a well-defined standard.

All the communication should be based on international standards as WPS, WFS and GML. Therefore, all already existing techniques should be used, even if not referenced here. However, general components of the data transferred are described below.

Aspirant (Multiplicity: 1...*): Feature (geometry) that has to be analyzed, changed or checked against a (spatial) constraint by an *Algorithm*. As an output, the manipulated feature is provided, f.i. a changed surface or the feature or features that successfully match against a condition.

Take an unclosed surface as an example. If an *Algorithm* is requested to close the surface, the closed surface will be returned. If, as an additional example, 20 points have to be checked whether they lie in a polygon, only the points are returned for which this request is positive. That ensures a further processing by subsequent

Algorithms. Aspirants can also be provided to *Transformers* and be returned as GML features if they were not before or vice versa. Note that, again, only these features should be returned for which an error free transformation is possible.

Geodata (multiplicity: 0...1): Geodata on which an analysis should be performed by an *Algorithm*. Geodata are not meant to be changed or manipulated and, therefore can easily be referenced by any resource.

A polygon in a Point-in-Polygon algorithm would be a valid example. It has to be referenced or transferred together with the request but will not be changed or returned.

Parameter (multiplicity 0...*): Parameters normally are numeric values that influence the *Algorithm*. They can be provided within the specification of a WPS. Radius for a buffer calculation or thresholds for the fixing of unclosed polygons are two examples. However, since more than just one *Algorithm* will be performed when requesting a *KnowledgeBasedChainedProcess*, tuples of parameter have to be provided.

Resource (multiplicity 0...*): A resource is an essential part of the communication between the components of an integrated urban analysis platform. In general it is represented by an URL that refers to either to geodata or to an additional *AbstractKeyProcesses*. The latter of cause is prohibited when used in an *AbstractKeyProcess* itself. It may be used to refer to all the components discussed above. A list of resources is the main added value of a *KnowledgeBasedChainedProcess*.

4.3 Communication

The *Keyprocess* communication is very similar to the well-known OGC WPS specification (Mueller and Pross 2014). At first a *Process* that fits our needs has to be found. Afterwards we are able to get the well-known capabilities document and are able to get the DescribeProcess document delivered by that service. The WPS Describe Process document provides a list of process inputs and outputs, the IO and parameters descriptions, a list of accepted formats, and a list of output formats. In this framework, the input and output format is restricted to GML. It is recommended to use the transformers for feeding in different input formats and delivering different output formats than GML.

Once a suitable *KeyProcess* is found, needed parameters, such as buffer distance, weight, buffer type etc., have to be defined as Aspirant (rf. Sect. 4.2) as well as the geodata. If everything is defined, an XML Document will be sent to the server, the server processes the request and sends a resulting XML-file back to the requestor (Fig. 3).

In contrast to the aforementioned *KeyProcess* communication, the *KnowledgeBasedChainedProcess* has to be defined differently. First of all the chain of processes has to be defined by applying our knowledge about valuable *KeyProcesses* and chaining them in a feasible and meaningful manner. Knowledge in that

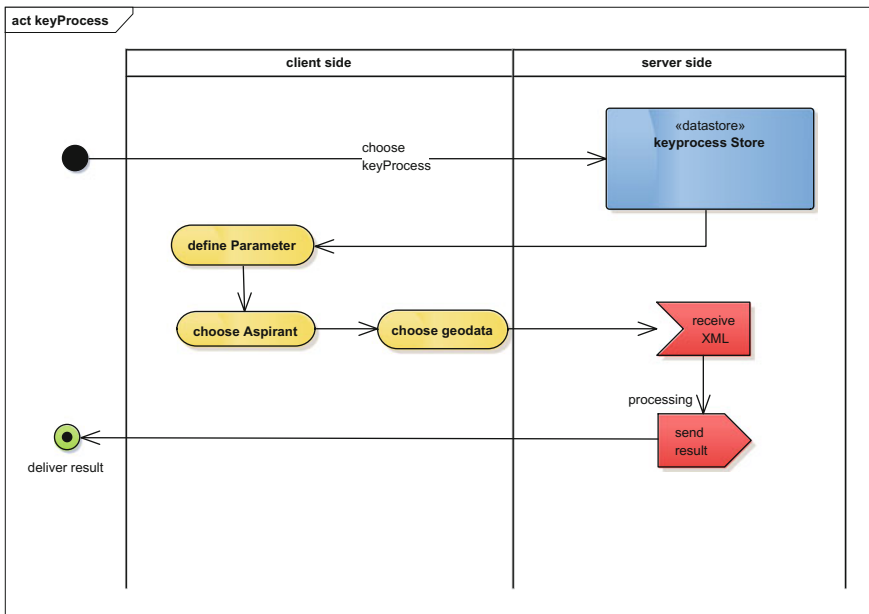


Fig. 3 Simple keyprocess communication

case means a meaningful knowledge about how to solve our specific spatial related problem and finally how to bring the processes in correct order.

As it is depicted in Fig. 4 the first activity is to define the chain and then send the XML document to the server. We have to define needed parameters, Aspirants and geodata for each *Keyprocess*. If the data and the produced results are simply piped through the whole chain, only parameters have to be defined. The server than parses the XML document, chooses the first *Keyprocess* and than processes each *Keyprocess* step-by-step and simply feeds the result of *Keyprocess* $n - 1$ into *Keyprocess* n as long as the end of the chain is reached. Afterwards, the result will be send to the requestor.

4.4 Example

Here, we demonstrate the propagation of data and results using the above suggested architecture in a figurative way. As a use case a 3D Buffer analysis serves as an example assuming an accidental gas pipeline rupture caused by an excavator. Now immediately the gas blows out and a huge gas balloon develops (Fig. 5). The question that arises is how many buildings or inhabitants are affected by this gas balloon and whether an evacuation should be planned. Assuming a desirable

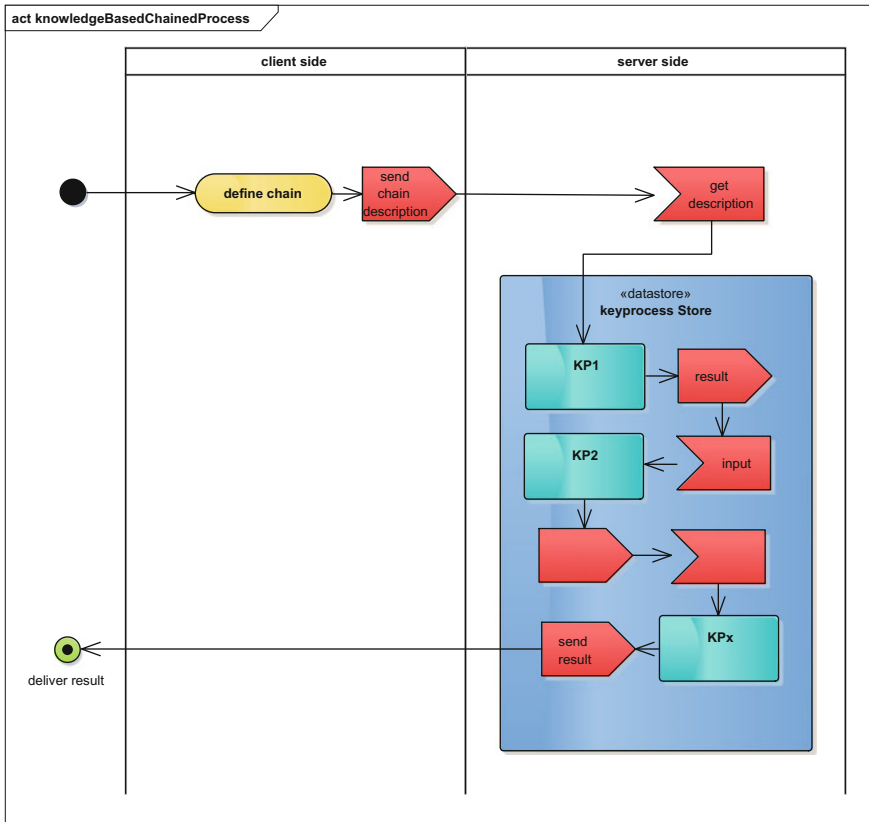


Fig. 4 Activity diagram for processing a chained process

Keyprocess store exists, we can use the 3D Buffer *Keyprocess*, already mentioned in Sect. 4.1, to calculate the 3D Buffer around the position of the excavator. The parameters needed by the *keyprocess* are determined by additional simulation software.

The *keyprocess* (see Fig. 6) requires the position of the excavator and the buffer distance coming from the simulation software. As a result we receive the GML solid of the balloon which is feeded into the next *Keyprocess* Solid_in_Solid which needs a *Solid* as Aspirant and further *Solids* as Geodata (probably buildings from CityGML). As a result we receive the buildings affected by the balloon.

However, this process can be used by many other scenarios such like the ones mentioned in Sect. 4.1 or any other scenario where a point is used to define a 3D region (buffer) and that, again, is used to identify the objects (solids) that are inside the 3D region. Thus, many other spatial tasks can be solved by only two existing public processes.

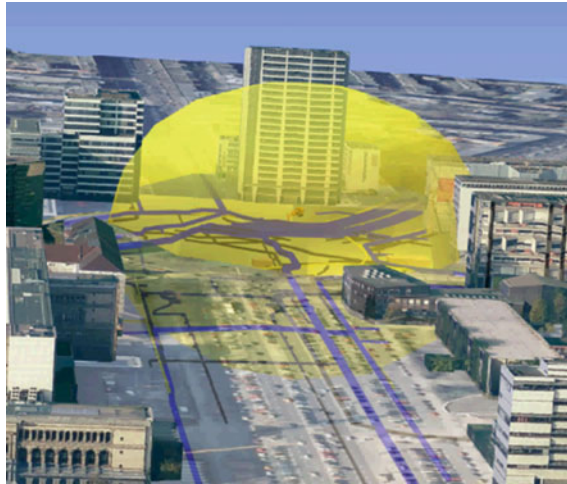


Fig. 5 Gas balloon caused by an excavator accident

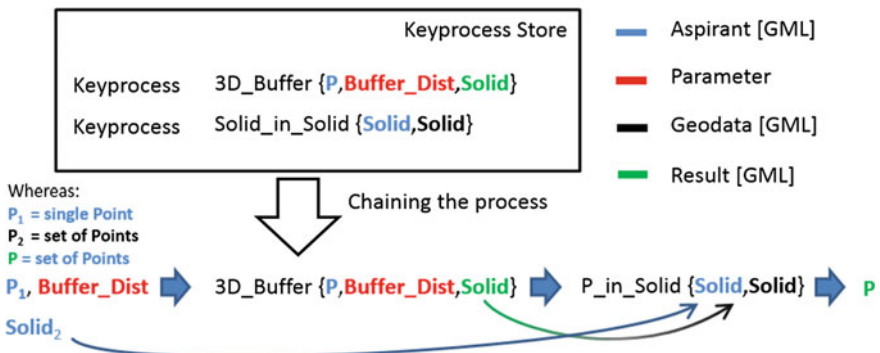


Fig. 6 Example chain of keyprocesses

5 Challenges and Outlook

Urban analysis today represents both, a major challenge and a very good chance for the science of geoinformation technology. The knowledge of virtual 3D city models, spatial analysis and algorithm development as well as the ability to formulize problems and work chains are highly needed skills in a fast urbanizing society. Many of already performed projects demonstrated this ability and were reviewed, here. However, progress and knowledge exchange could be reasonably enhanced by applying more service oriented approaches that are already developed, standardized and accepted. The Open Geospatial Consortium provides the

specification of a Web Feature Service for the advanced exchange of geographic features as well as the Web Processing Service for service oriented and distributed analysis. However, at least WPS seems not to be used in a sufficient extend.

Reviewing recently published projects, we identified different reasons for the reluctance of publishing open accessible WPS by the community of geoinformation science. These are the economic relevance of projects that hinder an open source like publication, complexity of projects and, therefore, limited reusability of code and algorithms from these complex projects. Further, we identified very specific and geographically bound projects and, last but not least, a lack of publication culture. While the first recognition may not easily be resolved, other reasons may be overcome by a general framework that enables the publication of reusable WPS for urban analysis approaches.

We discussed a general framework for the publication of reusable WPS that enables both, the full reusability of algorithms and, the possibility to develop business models. Thereby, very small scaled WPS, the *KeyProcesses* like *Algorithm*, *Transformer* and *Inquirer* are dedicated for the publication of open accessible, atomic and reusable services. Further aggregated and more advanced analysis can be published by the use of *KnowledgeBasedChainedProcesses* that may be published either as open access or already represent business models. In a last step even complete projects may be implemented as *ProblemSolvers*. Agreeing the discussed restrictions and terms of this concept could support urban analysis in a proper way.

However, to make the vision of shared and service oriented analysis power true, more than just technical definitions are needed. A new or renewed commitment has to be archived to publish key processes as WPS and to share analysis power over the internet. This can only be performed by the members of the GIS community. We as a community should make the progress to make use of publicly available GIS processes to benefit from each other and to safe development and research time and power. Furthermore, an OGC Testbed should be initialized to verify the expected profits of the structure presented here.

References

- Arendholz, K., & Becker, T. (2015). Requirements on building models enabling the guidance in a navigation scenario using cognitive concepts. In: *3D Geoinformation Science* (pp. 157–173). Springer International Publishing.
- Atila, U., Karas, I. R., & Rahman, A. A. (2013). A 3D-GIS implementation for realizing 3D network analysis and routing simulation for evacuation purpose. In: *Progress and New Trends in 3D Geoinformation Sciences* (pp. 249–260). Berlin, Heidelberg: Springer.
- Baumanns, K., & Löwner, M.-O. (2009). Refined estimation of solar energy potential on roof areas using decision trees on CityGML-data. *Geophysical Research Abstracts* (Vol. 11).
- Becker, T., Nagel, C., & Kolbe, T.H. (2009). A multilayered space-event model for navigation in indoor spaces. In: *3D Geo-Information Sciences* (pp. 61–77). Berlin Heidelberg: Springer.
- Bishr, Y. A. (1998). Overcoming the semantic and other barriers to GIS interoperability. *International Journal of Geographical Information Science*, 12(4), 299–314.

- Butwilowski, E., Thomsen, A., Breunig, M., Kuper, P.V. & Al-Doori, M. (2015). Modeling and managing topology for 3-D track planning applications. In: *3D Geoinformation Science* (pp. 37–53). Springer International Publishing.
- Czerwinski, A., Sandmann, S., Stöcker-Meier, E., Plümer, L. (2007). Sustainable SDI for EU noise mapping in NRW—best practice for INSPIRE. *International Journal for Spatial Data Infrastructure Research (IJS DIR)*. <http://ijsdir.jrc.it>, 2007:1.
- de la Beaujardiere, J. (2006). OpenGIS® Web Map Server Implementation Specification, Version 1.3.0, OGC Document 06–042, Open Geospatial Consortium.
- Fanelli, D. (2012). Negative results are disappearing from most disciplines and countries. *Scientometrics*, 90, 891–904.
- Geiger, A., Benner, J., & Haefele, K.H. (2015). Generalization of 3D IFC building models. In: *3D Geoinformation Science* (pp. 19–35). Springer International Publishing.
- Ghassoun, Y., Löwner, M.-O., & Weber, S. (2015a). Exploring the Benefits of 3D city Models in the Field of Urban Particles Distribution Modelling—a Comparison of Model Results. In: M. Breunig, M. Al-Doori, E. Butwilowski, P. V. Kuper, J. Benner, K. H. Haefele (Eds.), *3D Geoinformation Science, The 529 Selected Papers of the 3D GeoInfo, Lecture Notes in Geoinformation and Cartography* (Vol. 530, pp. 193–205).
- Ghassoun, Y., Ruhts, M., Löwner, M.-O., & Weber, S. (2015b). Spatio-temporal patterns of ultrafine particles in urban environment—A comparison between process-related geostatistical regression model and aerosol model. *Science of the Total Environment*.
- Gröger, G., Kolbe, T.H., Nagel, C., & Häfele, K.-H. (Eds.) (2012). OGC City Geography Markup Language (CityGML) Encoding Standard, Version 2.0, OGC Document 12-019, Open Geospatial Consortium.
- Kaden, R., & Kolbe, T.H. (2013). City-Wide Total Energy Demand Estimation of Buildings using Semantic 3D City Models and Statistical Data. Kongress-/Buchtitel: Proceedings of the 8th International 3D GeoInfo Conference. Band/Teilband: II-2/W1. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences.
- Krafzig, D., Banke, K., & Slama, D. (2004). Enterprise Soa: Service-Oriented Architecture Best Practices. Retrieved June 22, 2015 from <http://entropy.soldierx.com/~kayin/archive/ebooks/Prentice%20Hall%20-%20Enterprise%20SOA.%20Service-Oriented%20Architecture%20Best%20Practices.pdf>.
- Löwner, M.-O., Sasse, A., & Hecker, P. (2010). Needs and potential of 3D city information and sensor fusion technologies for vehicle positioning in urban environments. T. Neutens, & P. De Maeyer (Eds.), *Developments in 3D geo-information sciences. Lecture Notes in Geoinformation and CARTOGRAPHY* (Vol. 27, pp. 143–156. doi:10.1007/978-3-642-04791-6_8.
- Löwner, M.-O., Benner, J., Gröger, G., & Häfele, K.-H. (2013). New concepts for structuring 3D city models—an extended level of detail concept for CityGML buildings. In: B. Murgante, S. Misra, M. Carlini, C. M. Torre, H.-Q. Nguyen, D. Taniar, B. Apduhan, O. Gervasi (Eds.), ICCSA 2013, Part III, LNCS 7973 (pp. 466–480), Heidelberg: Springer.
- Lu, L., Becker, T., & Löwner, M.-O. (in press). Complete Traffic Noise Analysis Based on CityGML. In *Advances in 3D Geoinformation, Lecture Notes in Geoinformation and Cartography*. ISBN: 978-3-319-25689-4
- Mueller, M., & Pross, B. (2014). OGC WPS 2.0 Interface Standard. OGC Document 14–065, Open Geospatial Consortium 2014.
- Pédrinis, F., Morel, M., & Gesquière, G. (2015). Change Detection of Cities. In *3D Geoinformation Science* (pp. 123–139). Springer International Publishing.
- Portele, C. (Eds.) (2007). OpenGIS Geography Markup Language (GML) Encoding Standard, Version 3.2.1, OGC Document 07-036, Open Geospatial Consortium.
- Redweik, R., & Becker, T. (2015). *Change detection in CityGML documents*. In: *3D Geoinformation Science* (pp. 107–121). Springer International Publishing.
- Strzalka, A., Alam, N., Duminil, E., Coors, V., & Eicker, U. (2012). Large scale integration of photovoltaics in cities. *Applied Energy*, 93(2012), 413–421.

- United Nations (2014). 2014 revision of the World Urbanization Prospects, Retrieved July 22, 2015. <http://www.un.org/en/development/desa/publications/2014-revision-world-urbanization-prospects.html>.
- Vretanos, P.A. (Ed). (2010). OpenGIS Web Feature Service 2.0 Interface Standard. 09-025r1.
- Wagner, D., Wewetzer, M., Bogdahn, J., Alam, N., Pries, M., & Coors, V. (2013). Geometric-semantic consistency validation of CityGML models. In: *Progress and New Trends in 3D Geoinformation Sciences* (pp. 171–192). Berlin Heidelberg: Springer.
- Wilson, T. (Ed.) (2007). OGC® KML. OGC (R) Standard, Version 2.2.0, OGC Document 07-147r2, Open Geospatial Consortium.

Usability Assessment of a Virtual Globe-Based 4D Archaeological GIS

Berdien De Roo, Jean Bourgeois and Philippe De Maeyer

Abstract Acquired using 3D technologies, archaeological data is increasingly represented via 3D visualizations. For analysing, interpreting and exchanging, these data are unfortunately mostly reduced to two dimensions. Therefore, a 4D archaeological GIS that integrates 3D representations and analytical functionalities will contribute to different parts of the archaeological workflow from fieldwork preparation over analysis to reporting. Such a 4D approach will facilitate better and more integrated insights and allow more complex analyses and interpretations. Incorporating such a 4D archaeological GIS in a web-based environment will even increase the benefits as this could function as a virtual workspace. Since virtual globes have proven their capabilities to manage and visualize 3D data in non-expert applications, a prototypical 4D archaeological GIS was developed based on the virtual globe Cesium. This paper demonstrates by means of a usability test with employees of a Flemish archaeological organization that the concept of such a low-threshold application is supported by the intended end-users. Although some usability problems were encountered and the functionalities of the prototype are rather limited, extending and further developing the system could result in a valuable research tool for archaeology.

B. De Roo (✉) · P. De Maeyer
Department of Geography, Ghent University, Ghent, Belgium
e-mail: berdien.deroo@ugent.be

P. De Maeyer
e-mail: philippe.demaeyer@ugent.be

J. Bourgeois
Department of Archaeology, Ghent University, Ghent, Belgium
e-mail: jean.bourgeois@ugent.be

1 Introduction

Geographical Information Systems (GIS) have proven to serve archaeology well. Since the 1990s, spatial analyses and cartographic visualizations have been used to examine and interpret archaeological finds and their relationships while integrated with other spatial data. The kind of applications for which GIS is used, varies from basic data management and map-making to complex network analysis and predictive modelling (Wheatley and Gillings 2002; Scianna and Villa 2011). In GIS, factoring in the third dimension, which is inextricable from archaeological data, is often restricted or even ignored (Wheatley and Gillings 2002; Breunig and Zlatanova 2011; von Schwerin et al. 2013).

The spatial data of archaeological finds are however acquired using modern 3D technologies such as total stations and GPS (Stal et al. 2014). Increasingly, laser scanning and photo modelling are used to create 3D models of excavation and architectural objects (De Reu et al. 2013; Lonneville et al. 2014; Stal et al. 2014). These models offer a better understanding of the object by the ability to zoom, rotate and change colours (Losier et al. 2007). Notwithstanding these considerable advantages, analytical tools are for the most part lacking in the virtual environments in which these models are visualized (von Schwerin et al. 2013).

Integrating both the 3D representations and the analytical functionalities will result in a valuable research tool for archaeology. Furthermore, a web-based environment for this tool to belong to could even increase the benefits. Such a geodata infrastructure has been suggested by several authors (Snow et al. 2006; von Schwerin et al. 2013; De Roo et al. 2015a).

In the current research project, an attempt is made to develop such a web-based virtual environment. Taking into account the previously gathered user and organizational requirements, a prototypical 4D archaeological GIS was designed based on a virtual globe (De Roo et al. 2015b). Besides their strength in visualizing spatial data, virtual globes have proven their capabilities in non-expert applications such as public participatory GIS because of their intuitive nature (Butler 2006; Minghini 2013). However, their use in archaeology as true 4D GIS, involving 3D and temporal data, has not yet been investigated (De Roo et al. 2015b). Therefore, this paper probes the potential feasibility of extending a virtual globe towards a 4D archaeological GIS by means of a usability test.

In this paper, a prototypical application is used to assess the concept and usability of a virtual globe, which is extended towards a 4D archaeological GIS. Section 2 describes the design of the prototypical application against the background of the concepts of usability engineering. The methodology for the usability assessment is outlined in Sect. 3. Then, the results are described and discussed in Sects. 4 and 5 respectively.

2 Design of the 4D Archaeological GIS Prototype

The development of the 4D archaeological GIS prototype fits in with the methodological framework suggested by De Roo et al. (2013). As this framework integrates theories on human cognition and user-centred design, usability of the system towards the end users is at the centre stage. Usability is a multifaceted concept involving learnability, efficiency, memorability, errors and satisfaction (Nielsen 1993). Employing this concept as central point in the development of a product may increase productivity, reduce errors, lower needs for training and support, and improve acceptance (Maguire 2001).

The methodological framework proposed by De Roo (2012) focusses on three pillars—user-oriented, data-oriented and analysis-oriented—while continuously considering the end-users. The first pillar centres on the user requirements and the context of use of the intended system. This way, the first pillar agrees with the first stage of the usability engineering lifecycle, ‘know the user’, explained by Nielsen (1993). In the current research project, analyses of scientific literature, user surveys and interviews with experts and non-experts in GIS have revealed the main user requirements and potential pitfalls of a 4D archaeological GIS (De Roo et al. 2013, 2015a). Besides ease of use and avoiding complexity for the true user requirements, organizational (e.g. minimal cost), data (e.g. 4D, temporal) and functional (e.g. 3D spatial) requirements could also be specified (De Roo et al. 2015b).

Next, the second pillar concentrates on the particularities of the archaeological data, which will be used in the system. It is acknowledged that people experience difficulties in expressing their opinion on more abstract, conceptual and technological matters (Nielsen 1993; Anastassova et al. 2007). Therefore, a prototypical application was designed in which case study data was implemented. This prototype is based on the virtual globe Cesium, which ensures a realistic and multi-dimensional visualization (Minghini 2013). The functionalities of the virtual globe were extended to suit archaeologists’ needs and a connection is established with a PostGIS database. Although a basic overview of the prototype is given in Sect. 3.2, a complete overview of the system design, components and functionalities are described in De Roo et al. (2015b).

The implementation of some (basic) data management and analysis functionalities in the prototype indicates the interconnectedness with the third pillar, which focusses on the archaeological analyses.

Next, usability engineering comes into operation again. At the end of the first cycle of the design process, the concept of extending a virtual globe towards a 4D archaeological GIS and the usability of such an application will be evaluated. This evaluation, in the early development stage of the application, forms the subject of the current paper.

When this first usability testing has finished, subsequent cycles in the iterative design process will take place: adapting the user requirements, matching the user needs to the archaeological data particularities, optimizing the prototypical application and its functionalities and performing user tests. The central position of the

user is, this way, maintained during the development and each cycle will further strengthen the application.

3 Study Design

3.1 Participants

As this study intends to determine the attitude of archaeologists towards a low threshold 4D archaeological GIS, potential end-users of the system were chosen as participants for the test. Therefore, the Flemish archaeological organization GATE (<http://www.gatearchaeology.be>) was contacted to participate in the project by providing data and allowing some employees to take part in the usability test. Although only a small number of test users will be reached this way, this will be sufficient in this early development stage to uncover usability problems and the general attitude towards the system (Nielsen 1993).

3.2 The Prototypical Application

The application used during the usability tests is a high-fidelity prototype (HFP). This means that users can fully interact with the system as if it was a real application (Rudd et al. 1996). The considered application (4D GIS) is innovative and an emerging technology. In such cases, it is important to show in detail how the application functions to potential end-users and therefore, a HFP was preferred above a low-fidelity prototype (LFP) (Anastassova et al. 2007). HFPs are however questioned in literature to be used early in the design phase of a system because of their higher costs and less imaginative character compared to LFPs (Rudd et al. 1996; Anastassova et al. 2007). To limit the programming workload in this early phase of the development, the number of features in the prototype was narrowed down. While the functionalities are limited in number, they were chosen to represent still the intended set of functions (De Roo et al. 2015b). Thus, a vertical prototype was developed (Nielsen 1993; Rudd et al. 1996).

Figure 1 shows the interface of the prototype at initialization. The standard Cesium viewer buttons (e.g. geolocator, changing base maps, etc.) are available at the top right of the user interface. Similar to traditional desktop GIS, at the left side of the screen, a layers panel allows the user to switch visibility of the data layers, to change the active layer, and to view the layer symbology. Furthermore, the developed 4D GIS functionalities are accessible via the buttons at the top of the screen. In order to enable data management and analysis besides Cesium's visualization, the following functions are implemented for now:



Fig. 1 User interface of the prototype application

- display of attribute information on click;
- consulting and updating attribute info in table form;
- filter on attribute;
- temporal filtering;
- distance calculation;
- 3D Buffer around point.

For the purpose of the usability test, two versions of the prototype were created. The first one, hereafter called prototype A, uses the data that were provided by the participants’ organization. As these data were 2D, but contained a depth attribute, the 3D display was created by extruding the polygons (Fig. 1). The second version, prototype B, contains archaeological data that was unknown for the participants. In this application, the Cesium Ground-Push Plugin was used to excavate the terrain (NICTA 2015) and the data were shown at the depth that they were found (Fig. 2). All functionalities from the first version were available except for the extrude option, which allows switching between flat 2D polygons and extruded ones. In addition, a timeline was introduced in this prototype and the data were only visualized during the period they actually date back to. (Figure 2).

3.3 Usability Test Setup

Making use of the two prototypes, a usability test was set up to perform a formative evaluation of the 4D archaeological GIS. This assessment served a duplicate goal. On the one hand, the objective of the test was to gain insight in the level of interest and acceptance; on the other hand, it would reveal usability problems in the current

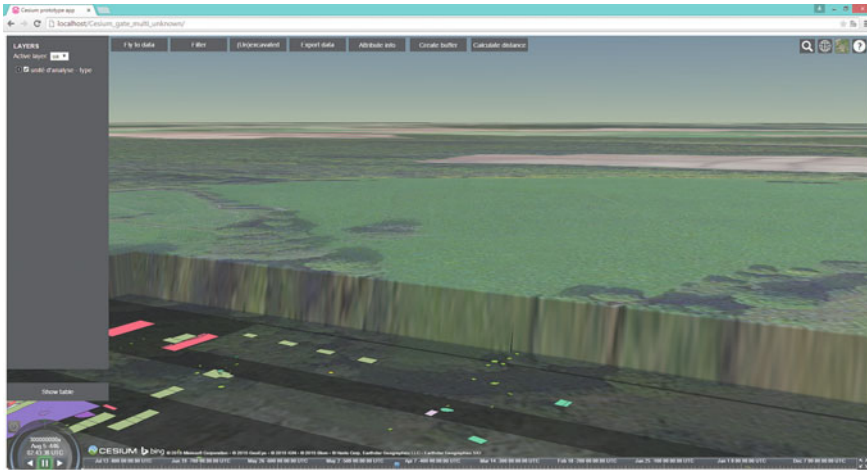


Fig. 2 Second version of the prototype with excavated terrain and timeline

design. Regarding the usability of the prototype, the current user test focused on two aspects: efficiency and learnability. The test was designed according to the principles and procedures described by Nielsen (1993).

The test took place at the temporary office of the organization on the archaeological site in Kerkhove (Avelgem, Belgium), where they have been excavating. Taking place during the participants' working hours, the complete test was restricted in time to approximately 30 min per test user.

For the test, the experimenter's laptop was used as the prototypes were running on a local web server on that machine. To facilitate easier navigation in the virtual globe a mouse was provided in addition to the laptop's touchpad. Furthermore, an internet connection was required to fetch in the (aerial) images for the virtual globe's background maps. The prototype was loaded in Google Chrome, although other internet browsers could be used because of the cross-browser nature of Cesium.

The user test consisted out of three parts: an introduction, the actual usability test, and a final questionnaire (Nielsen 1993). First, the user was given an introduction on the broader research context, the design of the system and the goal of the test to put them at their ease. During this introduction, some questions on the user's background, such as age and experience with GIS, were asked. Second, the user was shown the application and asked to perform some test tasks with both prototypes. Third, a questionnaire was given to probe the participant's opinion on the applications as means of debriefing.

During the actual usability test, the user was requested to accomplish a series of tasks with prototype A first, followed by some comparable tasks with prototype B. The starting task in both series aimed at familiarizing the participant with the application, its navigation tools and functionalities. Then, both sets were completed

Table 1 Overview of the test tasks and their objectives

Task	Short description	Objective
A1	Explore the application; navigate to complete globe and change background map.	Getting acquainted with the application
A2	Change visibility and get attribute information	Using data management functions
A3	Filter on attribute	Using query functions
A4	Change extruded to flat polygons	Using visualization function
B1	Explore the timeline and time animation	Familiarizing with the temporal functions
B2	Filter on attribute and time	Using query function and timeline

with respectively three and one additional tasks. These tasks are more GIS-related, i.e. data management and analysis, and have an increasing expected difficulty level. Table 1 gives a short overview of the test tasks. The users were requested to think aloud during the test and to indicate when they think a task has been completed. While performing the usability test, questions from users were not answered and no aid was provided. This help was not allowed in order not to influence the users and the results of the usability test regarding the intuitiveness of the prototypes. To avoid, however, that a user gets very demoralized, a hint was provided when the user was completely stuck.

For each of the six test tasks, the following usability measures were used:

- the time the user takes to complete the task;
- the kind of commands utilized by the user;
- the number of errors.

Furthermore, per task indications of satisfaction and confusion/frustration were collected.

Before the usability test was performed with the test participants, a few pilot tests were executed with colleagues to try out the test procedure and have an idea of the required time. This has led to decreasing the initial number of ten test tasks to six and reformulating some questions of the final interview.

4 Results

4.1 Test Users

Five persons (4 male, 1 female) employed by the Flemish archaeological organization GATE and working on the excavation site of Kerkhove (Belgium) at that moment, participated in the usability test. All test users held at least a Master's

Table 2 Experience of the test participants

Participant	Experience			
	Computer	GIS	Virtual globe	3D
1	Experienced	Daily	Weekly	No
2	Experienced	Daily	Sporadic	Yes
3	Experienced	Daily	Monthly	Yes
4	Average	No	Weekly	No
5	Average	Sporadic	Weekly	No

degree in Archaeology. Their age ranged between 28 and 41. Table 2 indicates the participants' experience with computers, GIS, virtual globes and 3D applications as they have indicated during the test introduction. All test users who assessed their computer experience as experienced, daily use GIS, while the persons with average computer experience only sporadically use GIS. The commercial ArcGIS is the most used GIS software package. All participants stated that they have already used the virtual globe Google Earth. Most of them use it in their work to get an impression of future site locations. Only two participants had experience with 3D applications or reconstructions.

4.2 Usability Test

The usability measures could not reveal differences between the participants with regard to their previous experience with GIS, virtual globes or 3D applications. For the first task with prototype A (A1), the users took their time to become acquainted with the application and especially its navigation options (e.g. zoom, rotate). They took up about 5 min to finish this task, except for participant 1 who performed this in less than 2 min. The first task with prototype B (B1), to explore the temporal functionalities, was finished in about 2 min. Participant 1 again used less time for this task, approx. 1 min.

Considering the kind of commands and the number of errors occurred during the tasks, the participants could have acted more efficiently. Participant 3 seemed to be the most efficient by completing three tasks with the minimal number of commands/actions. This, however, does not imply that no confusion or frustration could be detected. No relation could be made between the number of indications of satisfaction or confusion on the one hand, and the time needed for the task or the previous GIS experience on the other hand. The filtering tasks (A3 and B2) have shown the most confusion indications. Although the test users were unaware, task A4 was not always correctly understood and thus done wrong. Four participants changed the view of the globe to a 2D map view instead of changing the visualization of the features themselves from extruded to flat polygons. Furthermore, multiple participants encountered problems when they had to change the view angle. This involved pushing down the scroll wheel while dragging the mouse,

which is an uncommon operation. The task that seemed to give the most satisfaction was A2 because semantic information of a clicked feature was shown by the application.

Thanks to the encouragement to think aloud, additional usability problems became apparent during the tests. A first issue that was pointed out by several participants, concerned the zoom operation. In the current prototypes, Cesium's standard navigation operations are maintained, which implies using the scroll wheel to zoom. Unfortunately, dragging a rectangle with the mouse from top left to bottom right to zoom to that region is not supported by the default Cesium Viewer. Furthermore, a participant noticed that the application zooms to the middle of the screen and not to the position, to which the mouse is pointing. Another related remark made by two participants is that the zoom level is not maintained when changing views (e.g. from 3D globe to 2D map). A second issue that became clear by the thinking aloud and the indications of frustration has to do with the filter functionality. Although the filter box that pops up when clicking the 'filter' button contains controls to select or unselect all options, no participant has noticed these at first. As only two options should be checked for the given tasks (A3 and B2), the participants clicked multiple times to uncheck all other options. Some of the users, however, found the 'unselect all' button at the end of the task. Furthermore, a real-time filtering was preferred by the test users. Now, after the user has indicated the desired options, he has to push the filter button to execute the filtering. A third issue is concerned with the timeline. Several participants indicated that the labelling on the timeline is unclear: too long and specific, and vaguely pointed. In addition, task B1 has demonstrated that the navigation operations for the timeline, i.e. zooming and scrolling, are not intuitively found.

4.3 Final Questionnaire

First, the final questionnaire was used to assess the user's general opinion on the application. Four of the five participants indicated that they especially preferred the visualization as most interesting feature of the system. The query functionalities and the connection with the database are placed second. The concept of a 4D archaeological GIS that has a low threshold was received rather positively (1 neutral, 1 positive, 3 rather positive). All test users would consider using such an application when available in the future, and especially for communication, report production and fieldwork preparation. No user had problems with accessing the application via an internet browser; on the contrary, the participants experienced it as an advantage for dissemination purposes.

Second, to gain a more detailed insight in the opinions on the design and functionality of the prototypes, some questions with a 5-point scale were used. The participants' assessment of the design, the working of the current functions, the supply of functions, the extrude function and the timeline feature are given in Table 3. The overall design is found rather well, while some participants gave some

Table 3 Assessment of the participants on the design and functionality of the prototypes

Participant	Design	Function working	Function supply	Extrude	Timeline
1	+	+	+	+	+
2	–	0	0	–	+
3	0	0	–	0	++
4	+	+	0	+	++
5	+	0	0	0	++

++: positive, + rather positive, 0: neutral, – rather negative, — negative

suggestions, e.g. using icons instead of text for the buttons and replace the ‘filter’ button to the bottom of the filter box. Although the current supply of functions is considered sufficient for visualization, for using the application as true research tool it is too limited. The opinions on the extrude function differed between the participants. Some of them found this useless for the current type of data (prospection), but thought it could be beneficial for excavations that are more complex. In this regard, the preferred visualization of the data was this of prototype B, in which the data were shown on their found depth and the ground was push downward simulating an excavation. As shown in Table 3, all test users were enthusiastic on the timeline feature. They believed this could hold many advantages for the research, both for communication towards the public and fellow researchers and for analysis.

Finally, participants were asked what functionalities should be added to the prototypes. For supporting the interpretation of excavation results and the creation of reports, the prototype should be expanded to allow:

- connections of spatial features with drawings and photographs;
- exports of reports;
- integration of other data, such as environmental data;
- views from below;
- spatial analyses, such as density analysis.

5 Discussion and Conclusion

The presented usability test results show a rather positive attitude towards the concept of a low-threshold 4D archaeological GIS. This may be due to the current absence of tools that integrate both three-dimensionality and analytical functions (von Schwerin et al. 2013).

Although the most preferred aspect of the developed prototypes was the visualization, the application can be widely deployed in the future by considerably extending the functionality. Especially, about the integration of the timeline feature the participants were highly enthusiastic. The test participants thought of integrating connections to drawings and photographs, more extended spatial analyses, and more elaborated export options. In the current state, participants would use the

application for preparing fieldwork, supporting the creation of reports and communication.

For this communication purpose, most test participants recognize the advantages the web-based nature of the application can hold. They indicated that sharing archaeological information and maps with both colleagues and the public via the Internet could be beneficial. This finding agrees with the suggestions for a web-based geodata infrastructure for archaeological research reported by several authors (Snow et al. 2006; von Schwerin et al. 2013; De Roo et al. 2015a). Moreover, this idea is consistent with the concept of ‘shared virtual workspaces in which researchers can collaborate virtually on larger tasks as proposed by Snow et al. (2006).

Since multiple persons with diverse computer and GIS experience will collaborate in such an environment, the application has to be intuitive, easily understandable, rapidly learnable and efficient. The two prototypes used in the presented usability test seem to support the hypothesis that a virtual globe can function as basis for a low-threshold GIS. Although some participants encountered difficulties with rotating and changing the viewing angle, they became rapidly familiar with the navigation options of the virtual globe. Furthermore, the implemented functionalities have found to work intuitively and have a good learnability factor. For example, the ‘uncheck all’ option in the filter box was not noticed at first, but once remarked the participants used it in the following tasks.

However, the test has also demonstrated some usability problems. The standard zoom functionalities of the Cesium Viewer does not enable all operations, which are known from popular web mapping application such as Google Maps. Zoom to an area dragged over by the mouse or centering the zoom to the mouse pointer is not supported by Cesium. Regarding the developed filter function, participants expected a real-time filtering. Finally, the design and implementation of the timeline feature should be reconsidered, even though the participants were already positive about it. Mainly the labeling was found both too small and too exact. Currently, the labels display positions in time accurate to the second (in UTC), e.g. May 26-600 00:00:00 UTC. Such an indication contrasts with the more imperfect and fuzzy temporal data used in archaeology, such as periods, Terminus Post Quem dates, etc.

Elaborating the timeline feature is one of the major future development issues to be tackled. Besides, spatial analyses similar to the ones existing in current 2D GIS and frequently used in archaeological research form a second area for development of the prototypes shown in this usability test. Third, the spatial and temporal dimensions should be combined to allow 4D analyses. The easiest type of such analysis is a combined spatial and temporal query. However, more advanced analyses such as temporarily extended geostatistical analyses or even spatiotemporal simulations (e.g. site evolution in both time and space) are not unthinkable to be developed in the future. Before arriving at such an advanced stage, the application will have passed through a number of development cycles in which the user requirements will be refined and rematched each time.

Finally, this study has revealed a positive attitude towards the concept and has indicated some usability issues with the current prototype. However, implementing

more advanced functionalities and subsequent user tests will allow us to develop a fully-fledged user-centered 4D archaeological GIS.

Acknowledgments The authors are very grateful to the participants of the usability tests. Financial support from the Special Research Fund (BOF) of Ghent University is gratefully acknowledged.

References

- Anastassova, M., Mégard, C., & Burkhardt, J.-M. (2007). Prototype evaluation and user-needs analysis in the early design of emerging technologies. In J. Jacko (Ed.), *Human-Computer Interaction* (pp. 383–392). Berlin: Springer. Interaction Design and Usability.
- Breunig, M., & Zlatanova, S. (2011). 3D Geo-database research: Retrospective and future directions. *Computers & Geosciences*, *37*, 791–803.
- Butler, D. (2006). Virtual globes: The web-wide world. *Nature*, *439*, 776–778. doi:10.1038/439776a.
- De Reu, J., Plets, G., Verhoeven, G., et al. (2013). Towards a three-dimensional cost-effective registration of the archaeological heritage. *Journal of Archaeological Science*, *40*, 1108–1121. doi:10.1016/j.jas.2012.08.040.
- De Roo, B. (2012). Archeologische data en analyses: Factoren die de implementatie van een 4D-GIS beïnvloeden. Ghent University.
- De Roo, B., Bourgeois, J., & De Maeyer, P. (2015a). Information flows as bases for archaeology-specific geodata infrastructures: An exploratory study in Flanders. *Journal of the Association for Information Science and Technology*,. doi:10.1002/asi.23511.
- De Roo, B., Bourgeois, J., & De Maeyer, P. (2015b). *Extending virtual globes to a 4D web-based GIS for archaeology*. Submitted for publication.
- De Roo, B., Bourgeois, J., & De Maeyer, P. (2013). A survey on the use of GIS and data standards in archaeology. *International Journal of Heritage in the Digital Era*, *2*, 491–507. doi:10.1260/2047-4970.2.4.491.
- Lonneville, B., De Roo, B., Stal, C., et al. (2014). Accurate and cost-efficient 3D modelling using motorized hexacopter, helium balloons and photo modelling: A case study. In M. Ioannides, N. Magnenat-Thalmann, E. Fink, et al (Eds.), *Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection SE-39*. Springer International Publishing, pp 410–417
- Losier, L. M., Pouliot, J., & Fortin, M. (2007). 3D geometrical modeling of excavation units at the archaeological site of Tell 'Acharneh (Syria). *Journal of Archaeological Science*, *34*, 272–288. doi:10.1016/j.jas.2006.05.008.
- Maguire, M. (2001). Methods to support human-centred design. *International Journal of Human-Computer Studies*, *55*, 587–634. doi:10.1006/ijhc.2001.0503.
- Minghini, M. (2013). *Multi-dimensional geoweb platforms for citizen science and civic engagement*. Politecnico di Milano.
- NICTA. (2015). Cesium ground-push plugin. Retrieved February 25, 2015, from <https://github.com/NICTA/cesium-groundpush-plugin>.
- Nielsen, J. (1993). *Usability engineering*. San Francisco: Morgan Kaufmann Publishers Inc.
- Rudd, J., Stern, K., & Isensee, S. (1996). Low vs. high-fidelity prototyping debate. *Interactions*, *3*, 76–85. doi:10.1145/223500.223514.
- Scianna, A., & Villa, B. (2011). GIS applications in archaeology. *Archeol e Calc*, *22*, 337–363.
- Snow, D. R., Gahegan, M., Giles, C. L., et al. (2006) Cybertools and Archaeology. *Science* (80-) *311*:958–959. doi:10.1126/science.1121556.

- Stal, C., Van Lieferinge, K., De Reu, J., et al. (2014). Integrating geomatics in archaeological research at the site of Thorikos (Greece). *Journal of Archaeological Science*, 45, 112–125. doi:[10.1016/j.jas.2014.02.018](https://doi.org/10.1016/j.jas.2014.02.018).
- Von Schwerin, J., Richards-Rissetto, H., Remondino, F., et al. (2013). The MayaArch3d project: A 3D WebGIS for analyzing ancient architecture and landscapes. *Literary and Linguistic Computing*, 28, 736–753. doi:[10.1093/lc/fqt059](https://doi.org/10.1093/lc/fqt059).
- Wheatley, D., & Gillings, M. (2002). *Spatial technology and archaeology. The archaeological applications of GIS*. London: Taylor & Francis.

Temporal and Spatial Database Support for Geothermal Sub-surface Applications

M. Jahn, M. Breunig, E. Butwilowski, P.V. Kuper, A. Thomsen,
M. Al-Doori and E. Schill

Abstract Geothermal energy production from the deep subsurface requires a detailed knowledge of the relevant static and transient parameter distribution in the reservoir and host rock. In reservoir exploration, engineering and operation phases, both temporal and spatial subsurface parameters are acquired, evaluated and monitored in order to improve the reservoir performance. To support temporal and spatial data access to geothermal data sources, an efficient 3D/4D GIS is proposed in this study. We discuss theoretical and first practical approaches for the management of such temporal and spatial geothermal data. A first practical example is provided using the data acquired in the Soultz-sous-Forets geothermal project (France). A distributed software architecture, database design, and the concept for advanced query component with embedded simulations are presented. Finally, we give an outlook on the planned future research in 3D data management of subsurface, near-surface, and above-surface installations in other projects.

M. Jahn (✉) · M. Breunig · E. Butwilowski · P.V. Kuper · A. Thomsen
Geodetic Institute, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany
e-mail: markus.jahn@kit.edu

M. Breunig
e-mail: martin.breunig@kit.edu

E. Butwilowski
e-mail: edgar.butwilowski@kit.edu

P.V. Kuper
e-mail: paul.kuper@kit.edu

A. Thomsen
e-mail: andreas.thomsen@kit.edu

M. Jahn
Universit de Strasbourg, EOSt, 5 rue Ren Descartes, 67084 Strasbourg Cedex, France

M. Al-Doori
School of Engineering, American University in Dubai, Dubai, United Arab Emirates
e-mail: maldoori@aud.edu

E. Schill
Institute of Nuclear Waste Disposal, Karlsruhe Institute of Technology, 76344 Karlsruhe, Germany
e-mail: eva.schill@kit.edu

© Springer International Publishing AG 2017
A. Abdul-Rahman (ed.), *Advances in 3D Geoinformation*,
Lecture Notes in Geoinformation and Cartography,
DOI 10.1007/978-3-319-25691-7_19

1 Introduction and Related Work

Geothermal energy production from deep reservoirs faces the high complexity of the subsurface due original geological inhomogeneities and to multi-phase modification by post-sedimentary processes or tectonic deformation. Against this background, it is a challenge to identify productive reservoir zones with optimum performance or condition to enhance the system by reservoir engineering. Additionally, reservoir engineering and operation introduce temporal variation of parameters as a function of the original static condition. Typically, simulation of related processes require fully coupled thermo-hydraulic-mechanic-chemical simulations. It is obvious that data management of geothermal projects requires advanced concepts such as presented in this study.

The experience of the Soultz geothermal project and the subsequent development of geothermal energy in the adjacent fields in the Upper Rhine Graben (URG) underline this necessity to apply an experience and analogy based approach during exploration and exploitation. During 20 years of research, different options of exploitation for geothermal energy of the subsurface have been tested in Soultz. Three reservoirs at different depth levels with different initial hydraulic condition have been developed since 1988 (Table 1; Schill et al. 2015). Apart from differences in equilibrium temperature condition, the differences in depth of the three reservoir zones imply also differences in the ambient stress field (e.g. Tenzer et al. 1992; Evans et al. 2005; Dorbath et al. 2010). In addition to this reservoir scale, spatial variations down to very small-scale (m-scale) were found to influence the performance of the reservoir (e.g. Meller Carola and Kohl Thomas 2014).

The intense reservoir engineering using different techniques such as hydraulic and chemical stimulation and operation in Soultz introduce irregular temporal changes in the subsurface condition (Table 1).

Furthermore, experience from Soultz has been transferred to projects in the adjacent geothermal fields such as Landau (Schindler et al. 2010). With this in mind, recently there is a strong interest to save and make available the scientific results from the Soultz project, since they serves as reference for the development in the entire URG. From a of computer sciences point of view there are several reasons to store data in a database: A database avoids redundancy and inconsistency of the data as well as high development costs for data loss. It provides data integrity by automatically checking during the insertion and update of the data as well as multiple users access with security rights. Furthermore, it does not limit the possibility of linking data. Thus this database enhances scientific productivity on geothermal data for the future. Geothermal applications generate sophisticated requirements to geo-databases such as spatial, temporal and spatio-temporal queries, management of multi-dimensional and multi-scale data, navigation through geometric, topological models etc. Additionally, geothermal applications need the comparison and integration of different data sources in different phases of geothermal projects. Phase I includes all the steps prior to drilling operation such as the evaluation of existing geological data and initial feasibility studies on the application. In Phase II—based

Table 1 Overview of the natural reservoir conditions and the number of stimulation operations in the three different reservoir levels (upper, intermediate and deep) at the Soultz-sous-Forets EGS site (e.g. Schill et al. 2015; Jung 1992; Tischner et al. 2007). The reservoir levels are indicated as follows: I: upper, II: intermediate, III: deep. The vertical extension of the reservoir has been determined by the extension of the seismic clouds

Reservoir	Depth range (m)	Mean temperature range (°C)	Natural productivity/injection ($\text{m}^3 \text{Pa}^{-1} \text{s}^{-1}$)	Number of hydraulic stimulations	Number of chemical stimulations	Number of circulation periods
I (GPK1)	1400–2200	132–143	9×10^{-10}	3	0	0
I (GPK2)			3×10^{-8}			
II (GPK2)	3000–3900	147–166	3×10^{-10}	4	0	7
II (GPK1)	2000–3900		$5\text{--}7 \times 10^{-10}$			
III (GPK2)	4000–5400	168–201 (in 5097 m)	2×10^{-10}	4	7	18
III (GPK3)			n/a			
III (GPK4)			1×10^{-10}			

on the legal authorization—comprises the drilling and exploration of reservoirs with e.g. geophysical and hydrochemical methods. Finally, Phase III concludes with the commissioning of the power plant or the heating network. The choice of power plant technology and the design of the plant among others takes place on the basis of hydraulic and hydrochemical parameters of the thermal water. Thus geo-databases play different roles such as database archive, data analyzer, data integrator, and data visualizer. To account for this, we propose a database-support by providing suitable analysis functions and in the future also database supported simulations.

Some data transfer standards of the OGC (Open Geospatial Consortium) to be integrated in our future work provide mechanisms and models to describe spatial objects in terms of geometry, topology and thematics with temporal and measurement management e.g. CityGML (Grger et al. 2012), GeoSciML (Boisvert et al. 2007) or the O&M (Observation and Measurements) (Cox 2013) standards. All of them are based on XML. As well known CityGML concentrates on aboveground objects, GeoSciML has a focus on subsurface objects with digital geoscientific information e.g. geological maps, drilling or sampling. The O&M standard provides models and mechanisms used in sensor observation acts.

Furthermore the development of a Web Geological Feature Server (WGFS) for sharing and querying of 3D objects by Jacynthe Pouliot in (Pouliot et al. 2008) provides a workflow with Gocad as front-end and an application server (Apache Tomcat and Degree) with MySQL as back-end database. The work of S. Zlatanova et. al. described in (Tegtmeier et al. 2014) provides a 3D geotechnical extension model to handle subsurface geological and geotechnical objects.

2 Databases for Geothermal Projects

2.1 Database Requirements

A challenging requirement is the need of the access to several data sources:

Data in phase I of geothermal projects are acquired at the surface either along profiles or in the surface plane. They image, when interpreted, 1D to 3D structures of the subsurface. In phase II, data are mainly acquired along the bore-hole or at certain measurement points such as e.g. the well-head. Since seismicity is a measure of the reservoir area modified by engineering, a large number of seismic data are acquired at different points and provide 3D distributions. A distinction between time coordinates and location coordinates is to be made since seismicity is always a function of space and time. Furthermore, the amount of data e.g. from the seismicity monitoring is relatively high.

In contrast to seismicity data, hydraulic measurements (flow rate, pressure, ...) are fixed at constant geometries, i.e. the wells, that rarely change in shape over time. They are thematic attribute data at fixed geometries. The thematic data are collected as function of time. Thus, a geothermal database needs both, to manage highly time-dependent geometries and to provide an efficient thematic model to manage hydraulic measurements.

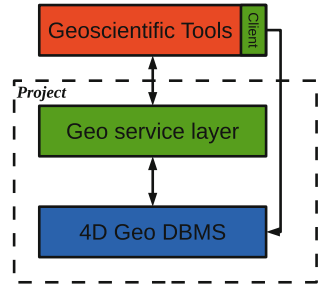
Data in geothermal projects are often acquired by different companies and institutions and therefore the database has additionally to account for different legal property types (e.g. industrial, university, free ownership, ...). This requires a user management system. Thus, beside an ordinary database there is interest to develop a meta-data database on the different datasets to provide information such as the source of the datasets, contact information of the data owner or finally to provide information on possible permission.

2.2 Database Architecture

To support full spatio-temporal data availability in geothermal projects over decades, geo-scientific tools (see blue box in Fig. 1) should have direct access to a 4D geo database management system (see red box in Fig. 1). In a first approach, the object-relational geo database management system PostGIS has been used to manage time series as records in the DBMS. In future, it will be examined if an object-oriented geo database management system with 4D operations such as DB4GeO (Breunig et al. 2010, 2012, 2013) may be used in addition for special 4D operations on complex 4D geometric data types such as simplicial complexes (triangle and tetrahedron networks).

At the top level of Fig. 1 the geo-scientific tools that have been used in the Soultz project over the years by the researchers are summarized. Besides software such as GOCAD, Petrel, WellCad, Tough 2, ArcGIS, Quantum GIS or ParaView, some tools

Fig. 1 Database architecture in geothermal applications



have been developed from scratch. Each software imports and exports different types of data formats, but internally it is mostly using its own data format.

A two-layer model is preferred in which a geo-scientific tool has its own client interface which can connect directly to the geo database. On the basis of its own internal data structure it then is able to interpret and integrate the results of database queries. It has a number of advantages if a geo-scientific tool can connect directly to a database especially concerning data synchronization. The geo-scientific tool can then be viewed as a client of the database.

Because the number of geo-scientific tools is high, it is assumed that not every software developer will program his or her own client. If the database does not provide specific functionalities for synchronization with these tools, the data formats have to be converted for import and export and the specific services of the database have to be made available in an intermediate step as well. The middle layer summarizes these services as a user interface to the database. The functionalities should be kept as thin as possible to be executable on different systems without major adjustments. Furthermore, special database queries can be provided to query some basic geothermal secondary data. The lower layer in the proposed architecture is the 4D geo database management system (4D Geo DBMS). Specific requirements have to be complied with the Geo DBMS to enable efficient geothermal data handling.

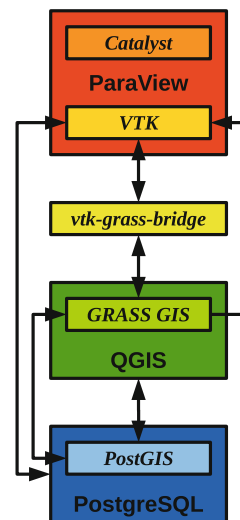
2.3 *Concept for an Advanced Query Component with Embedded Simulations*

In addition to the storage of raw data or the geological interpretation in form of sub-surface models, the archiving of scientific work plays a distinguished role. Many analysis can be supported by advanced database queries. However, the quality and the possibilities of analysis is highly dependent on the query language used by the DBMS and the provided services of the DBMS. For example, modern DBMS provide various aggregated statistical functions to run statistical analysis directly within the DBMS. The advantage is that the data does not need to be exported and then to be imported again into a scientific tool to calculate the results. The data transfer,

data transformation and the individual scientific tool can cause problems if past scientific work needs to be checked or tracked. In other cases the data itself must not be accessed by the public. Thus only some statistics about the data should be available. It makes sense to offer similar opportunities to scientists who increasingly purchase their findings with simulations. In the field of geophysics, it would be just as useful to check existing subsoil models on a mobile device to collect missing information of a simulation model. A centralized system for filling and processing large amounts of data could provide these needs.

In geothermal modeling there are many different simulation types. The main types are finite element method, finite difference method and agent-based modeling (particle systems). All types of simulations are based on specific initial conditions. The initial conditions are geometries with certain properties. Together with an underlying simulation model (partial differential equations, cellular automata, ...) and a set of parameters new geometries or properties are calculated. A database system can offer numerical services. In order to reach the target, it is necessary that these initial conditions are derivable by a DBMS from the raw data or geological subsurface model. A DBMS must provide certain geometry types with certain characteristics based on the corresponding simulation at least. In addition to the geometry, all parameters of the simulation model must be able to be archived with the simulation model itself. In Fig. 2 first considerations of practical implementation to couple simulations and databases with open source products are presented. This approach will be tested in our future work. The drawback of this approach is that the visualization of simulations is ensured by an external module (in this example ParaView with Catalyst). Paraview is based on a server-client principle. It can use two types of servers, the data server and the render server or the combination in one process at one server. The clients get the finished rendered results directly on the screen. Furthermore Par-

Fig. 2 Open source approach



aView is strongly based on parallel processing. The data server instance and the render server instance are realized in a cloud in which each sub-computer/node provides a partial section of the entire data set or renders a partial section of the whole scene. Again, the data flow between visualization and simulation and all the connected data transformations needs to be low for maximal performance. This is achieved by not storing individual time steps as the results of a simulation on the data server but once the visualization receives the initial and boundary conditions from the data server a co-processing unit transforms the data according to the simulation. Unfortunately ParaView is based on files which need to meet certain conditions. Thus, a dataset must be dismembered into manageable files. In future it will be advantageous to develop a DBMS, which uses this paradigm to relieve the DBMS clients not only in the memory usage but also in utilization and processing of the data. The output will be streamed to the mobile clients through a visualization unit of the DBMS such as the paradigm of ParaView.

3 Application in the Soultz Geothermal Project

3.1 Experiences with Data Collection

As mentioned above the Soultz geothermal project started in 1987 and a large number of scientific and industrial groups have been working for different periods on the project. However, hydraulic and thermal data were continuously collected at the site and archived on zip-drives, JAZ/Syquest drives and CDs/DVDs in ascii-format. Since these data are of major importance, they have been used as a reference to test the database in this study.

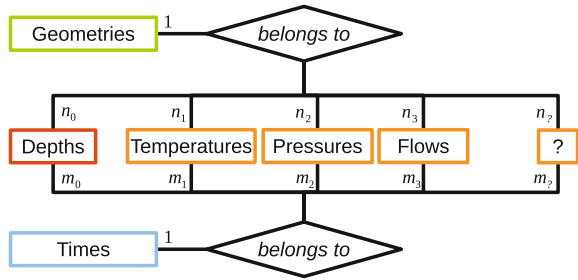
A number of 761 out of 2500 possible readable files could be analysed and imported through an OpenSource GIS to an OpenSource Database. A multithreaded Importer-PlugIn for OpenJump had been developed. Any file imported into the database is done by a single thread. Some files were corrupted. Other data were extracted and transformed into Microsoft Excel tables for scientific usages.

These data are connected to 27 different geometries, the drilling holes. Some data depend on depths, but most are measured at the well-head as a function of time in different period and periodical time steps.

3.2 Conceptual Database Schema

If possible, geothermal spatial data should be managed in a standardized way. To be compatible with today's geo-databases, we propose to use for simple geometries the General-Feature-Model which extends the Simple-Feature-Model of the Open Geospatial Consortium (OGC)—by the temporal dimension and related operations.

Fig. 3 Conceptual schema for geothermal hydraulic tests



This specification which has already become an ISO standard seems to be very suitable for mapping geothermal data into a database representation.

A feature is a real-world object with a certain geometry and a series of thematic attributes. Figure 3 shows a draft of the conceptual database schema for hydraulic data represented as Entity Relationship Model (ERM). The real world object (feature) e.g. could be—in connection with hydraulic data—a measuring device with a unique geometry or location and the corresponding measured data which were recorded over a specific period of time at predetermined time points. For hydraulic data only simple geometries are needed such as points which define the locations of the measuring devices. The trajectories of the wells can also be defined as polylines. Furthermore, the ISO standard (ISO 19107 or ISO 19109) provides certain 3D operations for single geometries. A typical operation would be to calculate distances or intersections with other geometries or geometry specific attributes such as the bounding box, length or volume. Other spatial operations are given in accordance with this standard. The measurements of a measuring device can be divided into two tables: Not only one time point, but a whole time series is stored per record. An example is the series of temperature values. The intrinsic time points are also summarized in an equally long series of time values. If e.g. the temperatures, pressures and some flow rates are measured synchronously, they are referenced to the same time series in the times table and the specific geometry in the Geometries table (see Fig. 4). This structural concept is a simple workaround to manage 4D geo-objects in a 3D plus nD environment. A proper 4D management is desirable, because real 4D operations on complex geometries including 3D spatial operations during time intervals could significantly improve the analysis of geothermal data.

3.3 A PostgreSQL/PostGIS Implementation

To be compatible with today's geo-databases, the prototypical database implementation uses—for simple geometry types such as points and lines—standardized geometry objects of the Open Geospatial Consortium (OGC). OGC's Simple-Feature-Model is pursued by PostGIS, an extension of PostgreSQL object relational database management system (ORDBMS) that communicates with Paraview via Open-

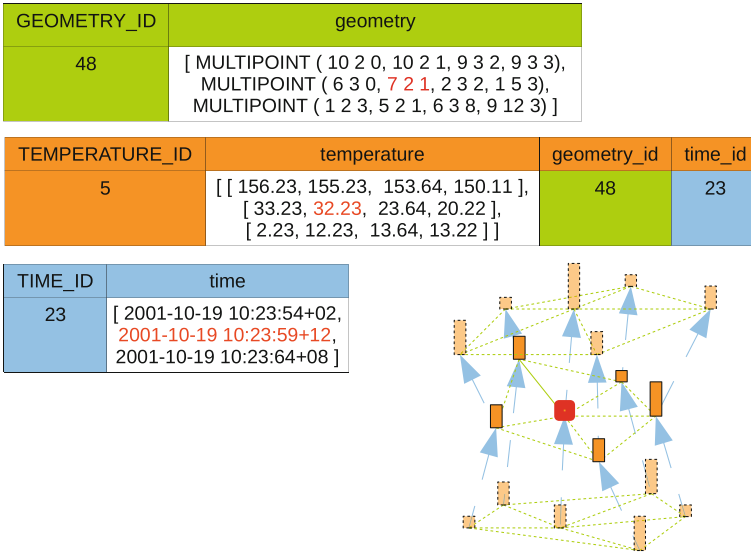
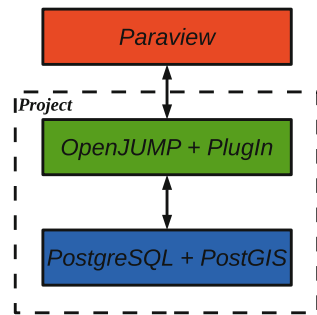


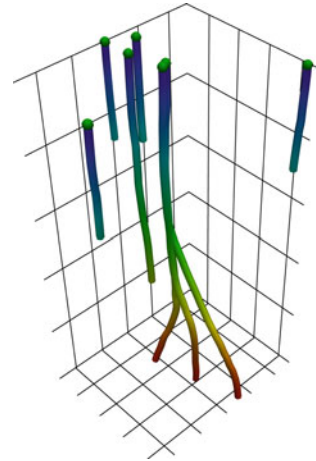
Fig. 4 Example with multidimensional data-arrays usage for thematic attributes connected to a spatio-temporal geometry

Fig. 5 Database architecture implementation with OpenSource products



JUMP, an easy-to-use GIS, which uses the Java-Topology-Suite (JTS) as its topology and geometry core. The spatial object types from PostGIS and JTS are largely ISO compliant, thus they are compatible and bring some geometric calculations with them. For these reasons and the straight forward ability to extend those open source products by new requirements of geothermal projects, the choice was made to use PostGIS and OpenJUMP. Figure 5 shows the database architecture for the practical approach. Figure 6 shows the location and form of the wells at Soultz site managed by PostGIS and visualized by Paraview via the Open JUMP connection and a VTK exporter PlugIn. To archive hydraulic test data this approach is well suited. The database conceptual schema was implemented with only one dimensional attribute data arrays because the geometries were assumed to be constant over time. Spatial and temporal analysis were performed through OpenJumps PostGIS PlugIn by some

Fig. 6 Visualization of the drilling holes through Paraview from an Exporter-PlugIn for OpenJump



test queries on the geothermal database developed in the Soultz geothermal project. However, for complex geometry types an extension is required, because most high-performance simulation as well as visualization tools use complex geometries such as simplicial complexes rather than simple geometric features. A geo-database architecture such as DB4GeO is well suited for this challenge.

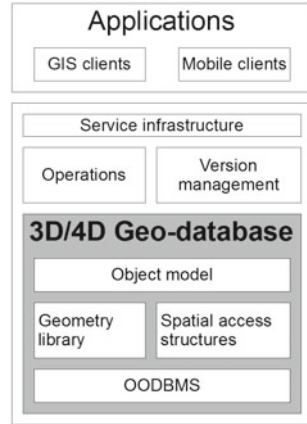
4 DB4GeO to Optimize 3D and 4D Data Access

The mathematical foundation of the concepts implemented in DB4GeO project (Brenig et al. 2010) has been published by the group of Helmut Schaeben (Le et al. 2013). An overview of the general system architecture of DB4GeO is shown in Fig. 7. DB4GeO has been successfully tested in several geological applications.

The basic optimization was done in the object model of DB4GeO. The Java source code is divided into the API interfaces and the resulting implemented classes and organized in different packages concerning the theme. The Java interface Object3D/-4D serves as an interface for all database objects available in 3D or 4D. Database objects are collected in a Space. Spaces define the reference system, the accuracy constant and manage constraints all objects need to fulfill. Spaces are managed by a Project and Projects are managed by an implementation of the DBMS class which handles the connection and transactions.

There has been only one implementation for each dimension of the Object3d/-4D interface. The spatial part of the DB4GeO objects could only be complex nets. The spatial part in 3D and 4D will be described in detail in the following two sections. The thematic part is formed by the interface Thematic3D/-4D. Attributes are handled by an instance of the implementation of the Thematic3D/-4D interface. They are collected in one Record. Records were linkable to an instance of the implementation

Fig. 7 Database architecture of DB4GeO



of the Object3D/-4D interface as the object thematic and to the spatial elements contained by an instance of the implementation of the Object3D/-4D interface as the simplex thematic. Beside the object and simplex thematic records were linkable to the points of a complex net as point thematic, too. The type of each record element is defined through an instance of a RecordDefinition class.

Beside the object model, its connection to the geometry core of DB4GeO, the operation package provides many services which operate on top of the geometry core or the objects themselves. Importers and exporters to GoCad and WebGL had been developed. A ParaView/VTK importer and exporter is under development at the moment. The data can be visualized through those tools by querying DB4GeOs export services. Beside importers and exporters the operations package offers services e.g. cutting through three dimensional geometries or triangulation.

4.1 Spatial 3D Model of DB4GeO

The Spatial3D interface is the base class of the geometry API for all complex geometry data types implemented in DB4GeO which is the geometric kernel of DB4GeO (see Fig. 8, left). The 3D geometries are divided into their dimension—0D, 1D, 2D or 3D. A database object contains either multiple point clouds (Sample3D), multiple curves (Curve3D), multiple surfaces (Surface3D) or multiple volumes (Volume3D). Volumes may be hulls (Hull3D) or solids (Solid3D). ResultGeometry3D objects are multidimensional objects with unknown content. A class that implements this interface must provide methods to filter its content by certain Spatial3D types or by the dimension. Thus, this type is the most complex case as a mixed type.

The data model is based on simplicial complexes. Solids, for example, are represented as tetrahedral grids. They consist of a set of tetrahedral clusters, the components which constitute their own topological structure. While networks of compo-

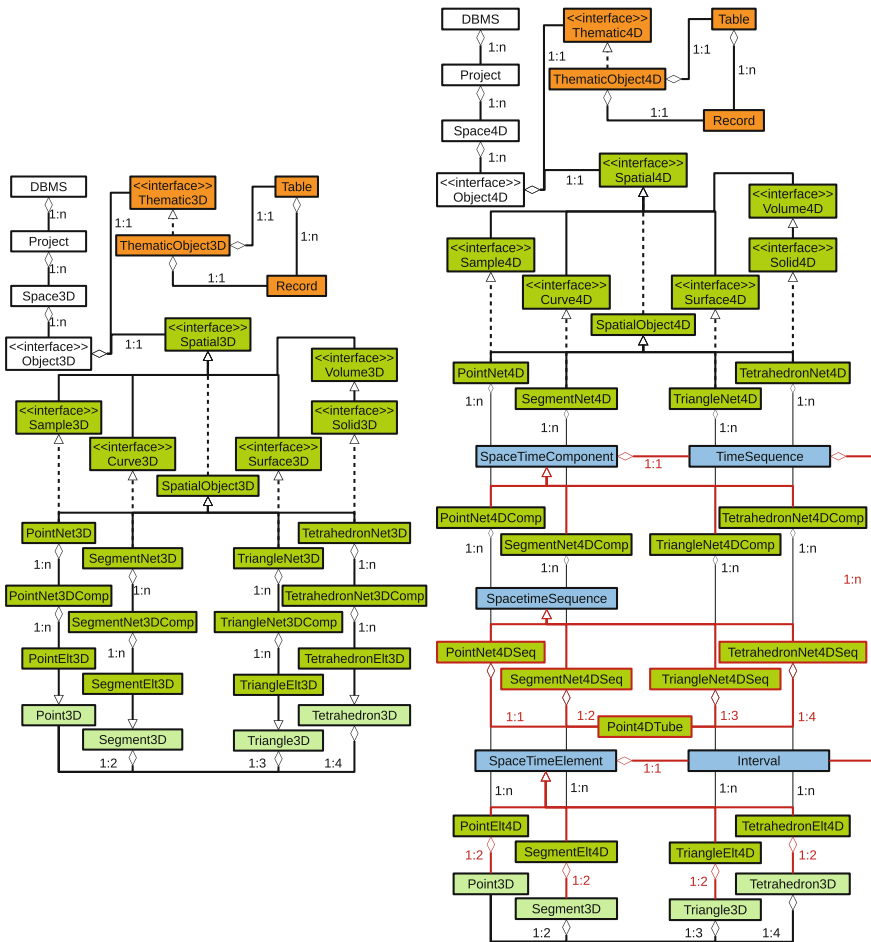


Fig. 8 Excerpt of DB4GeOs historical structure of its 3D model (left) and 4D model (right)

nents may be made in all topological relationships to each other, all d-dimensional elements of the components must be interconnected through $(d - 1)$ -dimensional elements. The second constraint of a component concerns the manifold. A curve component, for example, does not split in any point. To model a net of rivers each river has to be modeled as one curve component. The only exception are point cloud components whose points can not fulfill these conditions. The components contain a spatial access method that allows them to retrieve their elements spatially. The API of spatial access methods allows a free choice of implementation. The R-tree, a hash map and a hash set are available.

The simplex types are implemented through the implementation of the geometry API. The basic four types of simplexes are—point (Point3D), segment (Segment3D), Triangle (Triangle3D) and tetrahedron (Tetrahedron3D). These classes are summa-

rized in geometry package “geom” and define the spatial aspects of an element. Simplexes are defined in DB4GeO by their points. Thus, a segment consists of two points, a triangle of three and a tetrahedron of four. The actual boundary objects are created at runtime.

4.2 *Spatial 4D Model of DB4GeO*

The 4D model of DB4GeO is structurally almost the same as the 3D model (see Fig. 8, right, major differences to the 3D model are colored in red). The spatial types are divided by the dimension and from complex to more simple types. It was developed to model morphing and moving objects. Therefore, the basic elements of a net are simplex intervals with a three dimensional start simplex and end simplex called SpaceTimeElements. The time is saved through the Interval class. Each of those elements are collected in a set through the SpaceTimeSequence classes. The elements need to be connected. Every end of an element needs to refer to a start of the following element in time except the last one. Thus the sequences are topologically ordered in time and space within a four dimensional component. Therefore, the collection of time intervals, e.g. TimeSequences, does not change within a component and so called Point4DTubes can be calculated. They represent the spatial trajectories of each simplex point in time. DB4GeO allows to implement different interpolators to retrieve three dimensional objects at a certain time. An example of a tetrahedron net with only one moving or morphing tetrahedron over three time steps is given in Fig. 9.

4.3 *Redesign of DB4GeO*

To fulfill the needs of geothermal database requirements and to be able to integrate standards from the OGC like GeoSciML, O&M and CityGML the object model of DB4GeO had to be revised. First investigations on the incorporation of such a new object model into DB4GeO have been done on the basis of prototypical implementations (Breunig et al. 2013; Lienhardt 1989 et seqq.). The prototype topology module is based on the concepts of cell-tuple structure (Brisson 1989) and Generalized-Maps (G-Maps) (Lienhardt 1989). The experimental module integrates the cell types Node, Edge, Face, and Solid as a topological abstraction of the DB4GeO types Point3D, Segment3D, Triangle3D, and Tetrahedron3D. All cell types are then aggregated to create a cell-tuple. Thus, a CellTuple object is a composition of one cell object of each dimension. Additionally, a CellTuple object is defined to be a composition of all cell-tuples that can be reached by an involution step of dimensions d , ($d \in \mathbb{N} \mid 0 \leq d \leq 3$). With this model, build on top of the existing DB4GeO geometry model, the algorithmic navigation capabilities inside DB4GeO net components could be improved and the model now provides greater flexibility.

Fig. 9 Example of Tetrahedron4DNet

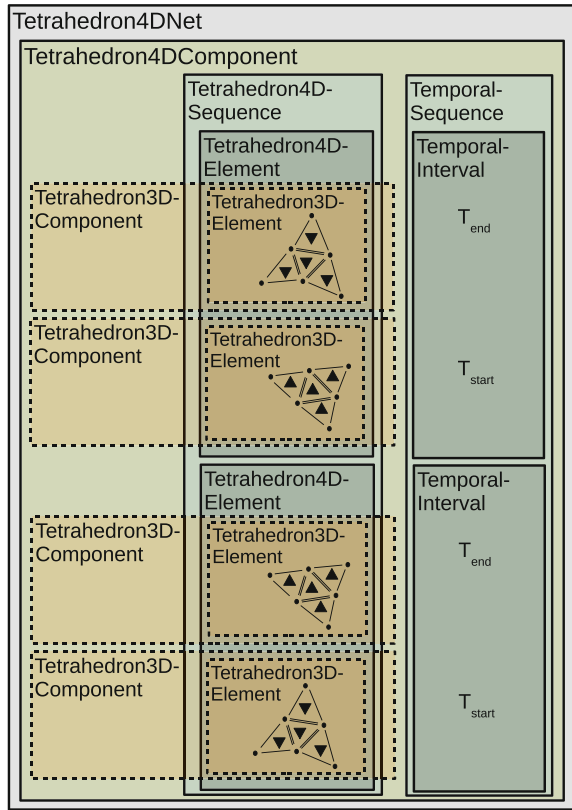
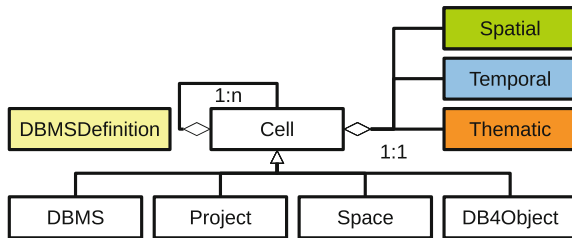
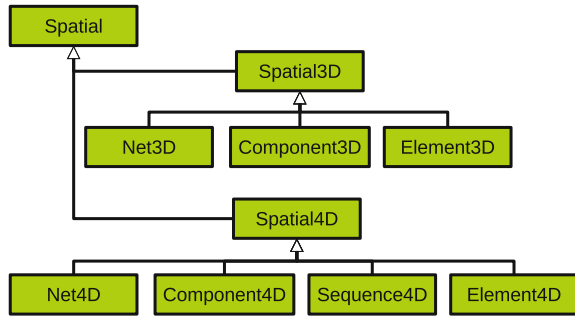


Fig. 10 Cell class hierarchy



Motivated by the experiences of the G-Maps module of E. Butwilowski a new object model was introduced to DB4GeO. It is based on cells (see Fig. 10) which are organized in a tree-structure. Cells in DB4GeO set up and manage the connection between the spatial model, temporal model and the thematic model seamlessly. Furthermore, it allows the implementation of general multidimensional algorithms. The four dimensional spatial types were changed to be able to implement dynamic 4d cell types. A strict separation between the temporal classes and the spatial classes was performed to reach those needs. Besides temporal intervals and sequences, indi-

Fig. 11 Spatial interface hierarchy



vidual timestamps are now available and linkable with the spatial model. Therefore, a cell or an object in DB4GeO consists consequently of three parts, the spatial (see Fig. 11), the temporal and the thematic part (see Fig. 10). Special cell types are able to be implemented by linking those parts carefully to each other. A cell could be interpreted as a feature of the common paradigm Simple or General Features by the OGC with additional functionalities.

Besides the object model of DB4GeO the main classes which manage a set of geometries had been derived from this new abstract cell type. These classes include the DBMS class, Project class, the Space3D and Space4D classes, the Object3D and Object4D classes (see Fig. 10). Every function and attribute of those classes were transferred to the new paradigm to ensure backward compatibility.

A DBMSDefinition class defines for each cell type which combinations of spatial- and temporal-types are possible and which standard attributes each cell type is going to provide. The cell hierarchy is determined by the cell type itself. An instance of the DBMS class will build instances of the Project class, an instance of the Project class will build either an instance of the Space3D or Space4D class and an instance of a Space3D/4D class will build an instance of an Object3D/4D respectively. The standard implementation of the Object3D/4D classes does not determine how the structure of the object tree needs to be. Any combinations of object hierarchies are possible. With this approach the database system is dynamically adaptable to the practical situation just by building child objects with the help of the DBMSDefinition.

Many object tree structures are conceivable. Imagining a tetrahedral mesh as soil with thematic attributes assigned at all levels of complexity and dimensions a object tree has to be developed to provide an easy to use way of data handling. E.g. a root is set up, the mother object with the tetrahedral network as spatial part. It provides a record for its meta-data and thematic tables managing the records of the child cells through its Theamtic3D object. All components, sequences, elements and elements of lower dimension could be collected as child cells of this mother object and the attributes of each child with the same spatial type could be collected through an own table (see Fig. 12). This was not possible before in DB4GeO, because only the d-dimensional elements of a d-dimensional component and its Point3Ds could own

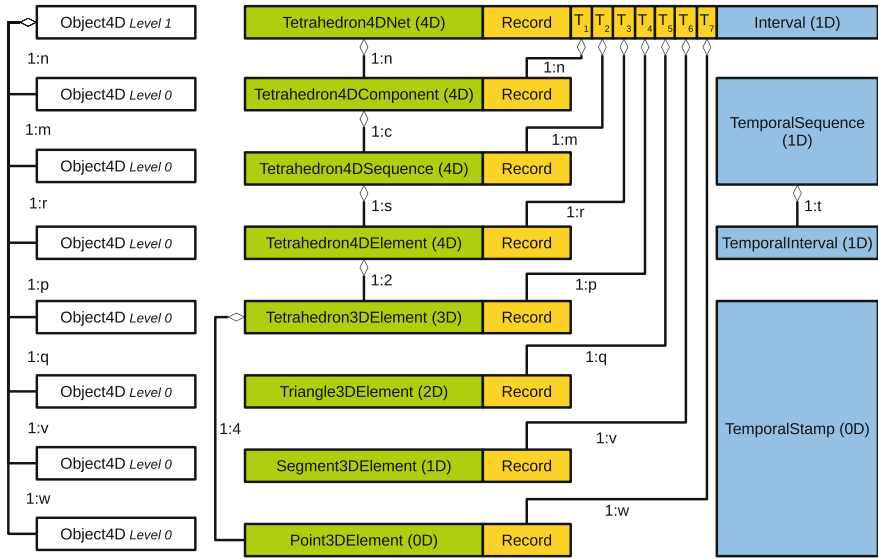


Fig. 12 Object hierarchy, example for thematic attributes connected to all geometries available in a Tetrahedon4DNet

thematic attributes and other elements of lower dimensions are generated at runtime and had no index to rely on. The objects hierarchy will only be the size of two levels and the mother object has to organize a large number of child objects with different spatial types.

If needed, a more complex object hierarchy will reduce the number of child objects per object level. Therefore, it is also possible to create an object with d -dimensional spatial part which may consist only of child objects with $(d - 1)$ -dimensional spatial parts. Even the complexity can be hierarchically organized. Both approaches together are provided in Fig. 13.

The example hierarchies, e.g. Figs. 12 and 13 lead to spatial, temporal and/or thematic redundancy when d -dimensional neighboring elements in time and space share a $(d - 1)$ -dimensional element, e.g. two tetrahedrons share a triangle, the triangles share a segment or the segments share a point. If this is not desired, the spatial, temporal and/or thematic redundancies can be removed by re-referencing those parts throughout the child tree.

All kinds of flexible referencing are available for topological relationships. Figure 13 represents part wise the standard case. Objects with d -dimensional spatial parts contain child objects with $(d - 1)$ -dimensional spatial part. The dual space, in which a object with d -dimensional spatial part contains child objects with $(d + 1)$ -dimensional spatial parts are linkable. A combination of the standard case and its dual space is possible but lead to high memory requirements and the algorithms on the data structures are usually much more complex. Which way may be the best representation of adjacency relationships on this basis is due to further research.

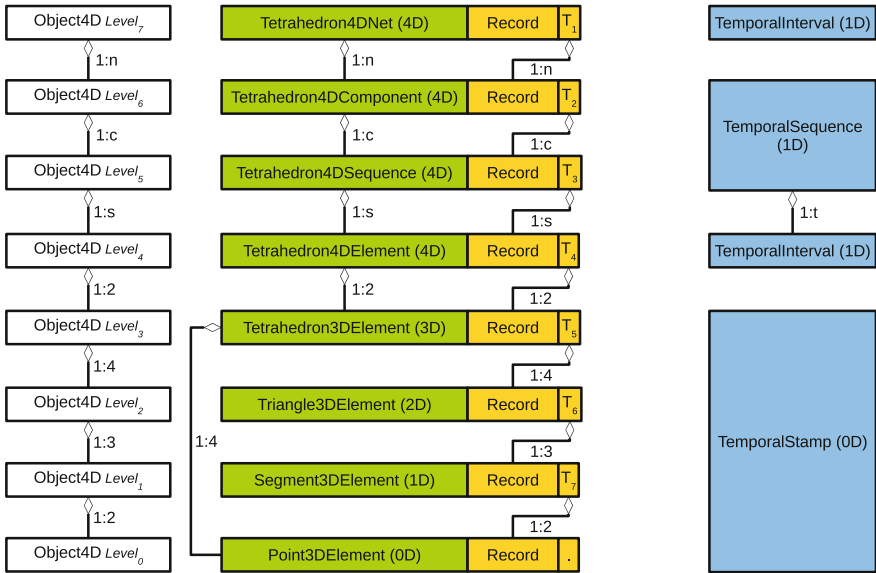


Fig. 13 Object hierarchy, example for topological use inside a Tetrahedon4DNet

The last interesting application is the level of detail. The object hierarchy is independent from the spatial or temporal type. For example, the child tree of a mother object is not limited to represent the spatial sub-parts of its own spatial type like the example in Fig. 12. Even the example of Fig. 13 only uses the spatial sub-parts from the spatial type of the mother object for the spatial part of the child cells. The spatial part of the mother object could simply refer to a simple version of the spatial part of its child objects. As an example, a mother object may represent a city as a `Point3D` with only one child object representing the area of the city as `Triangle3DComponent` and so on (see Fig. 14). These object hierarchies form the complete tree of level of detail.

Thus, the new model has significant advantages. The generalization to cells and the intelligent combination of spatial, thematic and temporal parts results in a greater variety of object types. Object tree structures allow a very dynamic variant to depict certain applications. It is nevertheless clear which object types are available by the `DBMSDefinition`. By specializations of the `Cell` or `Object` classes many specific cell/object types are able to run certain tasks. In case of archiving interconnected base geometries of simulations this data structure is of great help. Even multidimensional algorithms and paradigms such as `GMaps` are able to be developed and tested which lead to quick and robust results.

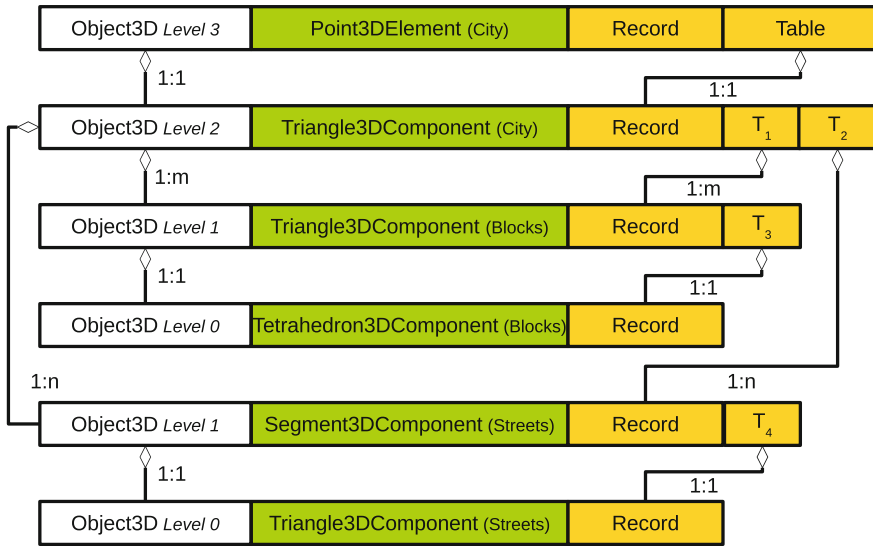


Fig. 14 Level of detail, example object hierarchy

5 Conclusion and Outlook

In this paper we presented theoretical approaches and first practical results on the temporal and spatial database support for geothermal applications. A distributed software architecture, database schemas, and the re-design of DB4GeO, our service-based geo-database architecture have been discussed.

In geothermal applications, the fluid transport through fault systems plays a central role. Therefore, in our future research we intend to explore a new concept called “fluid topology” that not only considers topological relationships between static objects such as fault systems and strata, but also topology in dynamically changing space such as time-dependent spaces filled with water. Furthermore, we intend to extend our activities to 3D applications in the Dubai area. Dubai is going through a second stage of geospatial revival through a surge of massive projects that involve sub-surface and above surface 3D geospatial applications. These projects span many areas of geospatial industry including oil and gas industry, aviation, construction and water authorities. The preparations for EXPO2020 resulted in the approval of Mohammed Bin Rashid city which is a massive project that will accommodate twenty million visitors, and includes the development of 100 hotels, the largest mall in the world, another airport and a huge park. A sub-surface project under construction is “The Water Discus Hotel”, which is the world largest underwater hotel which will include two massive discs above and under the water. Finally, the current massive project should be mentioned linking Dubai’s creek to the sea in a full circle going through the heart of Dubai and generating massive waterfront areas. Obviously all these projects potentially may profit from 3D geo-spatial technology. However, in

close cooperation with Dubai Municipality, it will have to be investigated in detail which project or special field will be suitable as a starting point to involve 3D database technology as added value.

Acknowledgments The authors acknowledge the GEIE EMC for providing Soultz data and Albert Genter for fruitful discussion. A part of this work was done in the framework of the Labex G-Eau-Thermie Profonde which is co-funded by the French government under the program Investissements d'Avenir. We also thank the German Research Foundation (DFG) by supporting the research work with grant BR 2128/14-3.

References

- Boisvert, E., Brodaric, B., Cox, S., Duffy, T., Holmberg, J., Johnson, B., et al. (2007). GeoSciML—A GML Application for Geoscience Information Interchange.
- Breunig, M., Butwilowski, E., Golovko, D., Kuper, P.V., Menninghaus, M., & Thomsen, A. (2012). Advancing DB4Geo. In *Progress and New Trends in 3D Geoinformation Sciences*. Lecture Notes in Geoinformation and Cartography (pp. 193–210). Springer.
- Breunig, M., Butwilowski, E., Kuper, P. V., Golovko, D., & Thomsen, A. (2013). Topological and geometric data handling for time-dependent geo-objects realized in DB4Geo. In *Advances in Spatial Data Handling*. Lecture Notes in Geoinformation and Cartography (pp. 1–13). Springer.
- Breunig, M., Schilberg, B., Thomsen, A., Kuper, P.V., Jahn, M., et al. (2010). DB4Geo, a 3D/4D geodatabase and its application for the analysis of landslides. In *Geographic Information and Cartography for Risk and Crisis Management*. Lecture Notes in Geoinformation and Cartography (pp. 83–102). Springer.
- Brisson, E. (1989). Representing geometric structures in D dimensions: Topology and order. In *Proceedings of the Fifth Annual Symposium on Computational Geometry. SCG 1989* (pp. 218–227). New York, NY, USA: ACM.
- Cox, S. (2013). Geographic information? Observations and measurements.
- Dorbath, L., Evans, K., Cuenot, N., Valley, B., Charlthly, J., & Frogneux, M. (2010). The stress field at Soultz-sous-Forts from focal mechanisms of induced seismic events: Cases of the wells GPK2 and GPK3. *Comptes Rendus Geoscience*. doi:10.1016/j.crte.2009.12.003.
- Evans, K., Moriya, H., Niitsuma, H., Jones, R.H., Phillips, W.S., et al. (2005). Microseismicity and permeability enhancement of hydrogeologic structures during massive fluid injections into granite at 3 km depth at the Soultz HDR site. *Geophysical Journal International*, 160, 388–412. doi:10.1111/j.1365-246X.2004.02474.x.
- Grger, G., Kolbe, T.H., Nagel, C., & Hfele, K.H. (2012). OGC City Geography Markup Language (CityGML) Encoding Standard.
- Hettkamp, T., Baumgrtner, J., Baria, R., Grard, A., Gandy, T., Michelet, S., et al. (2004). Electricity production from hot rocks. In *Twenty-ninth Workshop on Geothermal Reservoir Engineering: Proceedings*, 26–28 January 2004. Stanford, CA: Stanford University.
- Jung, R. (1992). Hydraulic fracturing and hydraulic testing in the granite section of borehole GPK1, Soultz sous Forets. In J. C. Bresse (Ed.) *Geothermal Energy in Europe: The Soultz Hot Dry Rock Project*. Montreux, Switzerland: Gordon and Breach Science Publishes SA.
- Jung, R., Willis-Richard, J., Nicholls, J., Bertozzi, A., & Heinemann, B. (1995). Evaluation of hydraulic tests at Soultz-sous-Forts, European HDR Site. In *Proceedings of the World Geothermal Congress, 1995* (Vol. 4, pp. 2671–2676). Florence, Italy: International Geothermal Association.
- Le, H.H., Gabriel, P., Gietzel, J., & Schaeben, S. (2013). An object-relational spatio-temporal geoscience data model.

- Lienhardt, P. (1989). Subdivisions of N-dimensional spaces and N-dimensional generalized maps. In *Proceedings of the Fifth Annual Symposium on Computational Geometry. SCG 1989* (pp. 228–236). New York, NY: ACM.
- Meller, C., & Kohl, T. (2014). The significance of hydrothermal alteration zones for the mechanical behavior of a geothermal reservoir. *Geothermal Energy*, 2, 12, 21.
- Pouliot, J., Badard, T., Desgagne, E., Bedard, K., & Thomas, V. (2008). Development of a Web Geological Feature Server (WGFS) for sharing and quering of 3D objects.
- Schill, E., Cuenot, N., Genter, A., & Kohl, T. (2015, April). 2015 Review of the hydraulic development in the multi-reservoir/multi-well EGS project of Soultz-sous-Forets. In *Proceedings World Geothermal Congress 2015* (pp. 19–25). Melbourne, Australia.
- Schindler, M., Baumgrtner, J., Gandy, T., Haufler, P., Hettkamp, T., Menzel, H., et al. (2010). Successful hydraulic stimulation techniques for electric power production in the Upper Rhine Graben, Central Europe. In *Proceedings World Geothermal Congress 2010*, Bali, Indonesia, 25–29 April 2010.
- Tegtmeier, W., Zlatanova, S., van Oosterom, P.J.M., & Hack, H.R.G.K. (2014). 3D-GEM: Geotechnical extension towards an integrated 3D information model for infrastructural development.
- Tenzer, H., Mastin, L., & Reinemann, B. (1992). Determination of planar discontinuities and borehole geometry in the crystalline rock of borehole GPK-1 at Soultz-sous-Forts. In J. C. Bresse (Ed.), *Geothermal Energy in Europe: The Soultz Hot Dry Rock Project* (pp. 31–68). Montreux, Switzerland: Gordon and Breach Science Publishers SA.
- Tischner, T., Schindler, M., Jung, R., & Nami, P. (2007). HDR project Soultz: Hydraulic and seismic observations during stimulation of the three deep wells by massive water injections. In *Thirty-second Workshop on Geothermal Reservoir Engineering: Proceedings, 22–24 January 2007*. Stanford, CA: Stanford University.
- Weidler, R. (2001). *Slug test in the non-stimulated 5 km deep well GPK2*. Internal BGR report. Hannover, Germany.

Automatic Semantic and Geometric Enrichment of CityGML Building Models Using HOG-Based Template Matching

Jon Slade, Christopher B. Jones and Paul L. Rosin

Abstract Semantically rich 3D building models give the potential for a wealth of rich geo-spatially-enabled applications such as cultural heritage augmented reality, urban planning, radio network planning and personal navigation. However, the majority of existing building models lack much if any semantic detail. This work demonstrates a novel method for automatically locating subclasses of windows and doors, using computer vision techniques including the histogram of oriented gradient (HOG) template matching, and automatically creating enriched CityGML content for the matched windows and doors. Good results were achieved for class identification with potential for further refinement of subclasses of windows and doors and other architectural features. It is part of a wider project to bring even richer semantic content to 3D geo-spatial building models.

Keywords Semantic • Geometric • CityGML • HOG • Template matching

1 Introduction

The built environment is of fundamental importance to society: 80 % of GDP is generated within its walls (Hampson et al. 2014) and its buildings variously provide shelter and a sense of home or belonging, not to mention a source of pleasure. On this latter note it has, for example, been estimated that Britain's built heritage alone

J. Slade (✉) · C.B. Jones · P.L. Rosin
School of Computer Science & Informatics, Cardiff University, Cardiff, UK
e-mail: sladejd@cardiff.ac.uk

C.B. Jones
e-mail: jonescb2@cardiff.ac.uk

P.L. Rosin
e-mail: rosinpl@cardiff.ac.uk

attracts £6.5 billion of foreign spending per year (Dawe 2013). Potential applications for computerised 3D building models include those for cultural heritage, leisure (think how many attractions, such as shopping centres, now have interactive plans for visitors to navigate), urban planning such as the emergent field of Building Energy Modelling or BEM e.g. Ham and Golparvar-Fard (2015), radio network planning, cultural heritage tours, personal navigation systems and augmented reality or AR (Döllner and Hagedorn 2007; Sivic and Efros 2014).

Whilst detailed Architecture, Engineering and Construction industry (AEC) building models have existed in the world of Computer-aided Design (CAD) for some decades, the arrival of Web 2.0 has also led to the advent of building models generated by members of the public. The Trimble *3D Warehouse* is one such freely available repository, while *TurboSquid* provide commercially sourced models for a fee.¹ CAD building models, increasingly conforming to the standards of Building Information Modelling (BIM), contain both structured geometry and semantic content, although the semantic content tends to be focussed on the needs of the AEC industry rather than those of other disciplines. Moreover, CAD content is often the intellectual property of AEC companies and not generally in the public domain. Meanwhile, those models outside of the sphere of CAD lack much if any semantic content (Jones et al. 2014).

Why is semantic content important? Imagine a mobile application which allows the user to explore a city, neighbourhood or building via augmented reality learning of important cultural heritage and architectural features through superimposed annotations. Or, as a radio network planner, having access to a city model which contains semantic content representing the materials for the individual building's façade components thus enabling a more refined network planning model which takes account of the effect of different materials on the amount of radio wave attenuation (de Fornel and Sizun 2006).

National Mapping Agencies, such as Ordnance Survey in Great Britain, have traditionally been concerned with the two-dimensional measurement of topographic features. As a potential data provider for some of the applications listed above, there is now increasing demand for such agencies to provide richer content, not just in the geometry (e.g. moving to the collection of 3D data) but also in the attribution of topographic features. This semantic content is likely to become ever more important, as it enables users to answer more complex queries than can be answered by geometry alone.

The method presented in this paper uses automated computer vision techniques, allied to 2D-3D geometry conversion, to identify different architectural features in the texture maps associated with a CityGML building model and to auto-create new semantic and geometric CityGML content for those features.

¹<https://3dwarehouse.sketchup.com/>, <https://turbosquid.com/>.

2 Previous Work

2.1 Architectural Object Detection in Images

As early as 1993, Koutamanis and Mitossi (1993) outlined the potential for computer vision to determine architectural components, specifically from scanned architectural drawings, including the use of simple template matching on line drawings. Whilst subsequent work such as that from Debevec et al. (1996) reconstructed buildings in 3D—using a form of structure from motion (SfM)—it did not focus on recognising façades or architectural features. The determination of façades, floors and architectural features are inextricably linked. As Johansson and Kahl (2002) assert, if one can delineate a row of windows then floors might be inferred.

An early foray into the identification of architectural components came from Iqbal and Aggarwal (2002) who used edge-detection. Johansson and Kahl (2002) attempted to detect windows in city scenes for AR city tours, learning window shapes using a Support Vector Machine (SVM). An alternative approach from Dick et al. (2004) used Markov Chain Monte Carlo (MCMC), generating possible building model forms and then comparing those possibilities to actual buildings.

Mayer and Reznik (2005) extended Dick's work by using MCMC to reconstruct buildings to obtain façade planes as Johansson and Kahl had done previously. They also matched windows, using an implicit shape model. Mayer and Reznik made assumptions about aspect ratios, rectangularity and that window panes had lower pixel intensities. They extended this approach in later work (Reznik and Mayer 2007) by considering the principle of repeatability to define columns and rows for windows. More recently, Meixner et al. (2011) used Harris corners, K-d trees and match-cost histograms to obtain an accuracy of 87 % for dormer identification.

The idea of using generative modelling concepts to infer façades in real images (related to the aforementioned repeatability) was picked up by Ok et al. (2012) who created possible façade-component combinations and compared test image façades to them via pixel, local and patch-based descriptors such as difference of Gaussian (DoG) and scale-invariant feature transform (SIFT). In contrast, Dick's façade--component combinations came from manual work carried out by architects. As with Mayer and Reznik's work, possible matches from Ok et al's work were refined using RANSAC (Fischler and Bolles 1981), an iterative parameter estimation approach.

Koziński and Marlet (2014) used graph grammars and Markov random field (MAP-MRF) to infer the positions of architectural components. Finally, Dore and Murphy (2014) created shape grammars for architectural styles, as a library for the *ArchiCAD* BIM software. These grammars can then be used in façade reconstruction, the parameterised templates providing a speed-up in the reconstruction process, reducing manual intervention. However, at the time of publication only a few architectural styles (including a few window types) were included in the library.

In summary, whilst there is a rich history of façade interpretation, identifying individual subclasses of architectural features e.g. structural descriptions of a window such as ‘3 × 3 panes’ or styles such as ‘Baroque’, is far less researched.

2.2 Semantic Content for Building Models

The XML-based CityGML from the Open Geospatial Consortium (Gröger et al. 2012) appears to be becoming the format for storing both geometric, semantic and geo-spatial building content, catering for a number of levels of detail (LOD) and for the inclusion of semantic attributes (van den Brink et al. 2013). This is in contrast to KML and many of the formats used in graphic design (Stadler and Kolbe 2007; Smart et al. 2011). Indeed Ross et al. (2009) suggest that GML (on which CityGML is based) should become the standard for the interchange of data for all involved in the realm of urban planning whilst Döllner and Hagedorn (2007) and Zhu et al. (2009), in researching the practice of 3D modelling in the field, conclude that CityGML is the most suitable standard for semantic attribution. With respect to interchange of data with the world of BIM and CAD Isikdag and Zlatanova (2009), Kolbe (2009) and Gröger and Plümer (2012) outline how CityGML needs to become and is increasingly becoming interchangeable with the standard BIM data format namely Industry Foundation Classes (IFCs).

Here we present a new method to auto-generate new CityGML content for architectural features on existing 3D building models, using the building’s texture maps. We focus particularly on styles (sub-types) of windows that could be used in cultural heritage applications that allow the user to explore the architectural details of a building.

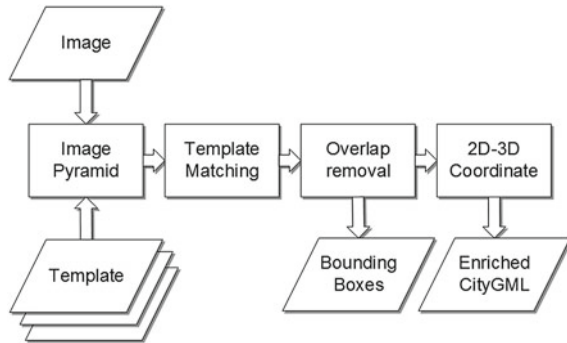
3 Methodology

There were two stages in the method for this work: (i) performing histogram of oriented gradients (HOG) based template matching, using the approach of Zhang et al. (2013) and Xiao (2013); and (ii), taking the output of the first step to create new CityGML content containing the matched architectural features, representing the semantic and geometric enrichment. Figure 1 shows a summary of the method.

3.1 HOG-Based Template Matching

Within computer vision, template matching is a standard method for identifying objects in images (Sonka et al. 2014). A template image (e.g. containing a single window object) is scanned across a test image (in our case a texture map from a

Fig. 1 Summary of methodology



building model) in order to find matching occurrences in the test image of the object contained in the template. Traditionally good matches are identified as locations producing large correlation values between the template and the corresponding test image windows. We use a recent alternative (Zhang et al. 2013; Xiao 2013) that is based on comparing HOG descriptors (Dalal and Triggs 2005) instead of directly comparing pixel intensities. The HOG method is based on edges, i.e. gradients or changes in pixel brightness. A template is decomposed into a grid of cells, and within each cell the orientations of edges are computed and histogrammed. The histogram is subsequently normalised, and forms the HOG descriptor. In a similar manner, the HOG descriptor is extracted from windows in the test image. HOG descriptors are not sensitive to variations in lighting or to small local geometric deformations, and so the HOG-based template matching approach can be more robust than standard template matching, i.e. it is not so sensitive to the exact location and appearance of the particular features of an object. This means that objects in a test image are more likely to be matched when they are stretched, compressed or where their component parts (e.g. individual window panes within a window object) are distributed with differing proportions, when compared to the template. Such a scenario is likely to occur in the real world, considering the potential variety in architectural design and in building methods. Prior to using HOG-based template matching we trialled a standard template matching approach, that of (Kroon 2011). However its sensitivity to the location of the features within an object meant that superior results were obtained using HOG-based matching. The aim of our study is to assign generic labels to objects in the texture maps on the building models. Methods such as SIFT are seen as too instance-based for such an aim, though may be used in a later stage of our work. It was therefore determined that HOG-based template matching should be used instead.

Preliminary runs of the HOG-based approach used a test image of the British Classical style heritage property Belton House, in Lincolnshire, UK (Pevsner et al. 1989) taken from *Wikipedia*. Belton House was chosen as a useful starting point due to the repeatable nature of architectural components within the building. The initial template used was of a tightly cropped sash window common to the period, but from a different building, found via a Google *Image* search. Three further *Flickr*

test images were used, each of a different UK heritage property and of similar architectural period and style (1660–1714). Corresponding templates were obtained from Google *Image* and *Flickr* searches, again from entirely different buildings. Test images and templates photographed square-on to the façade were chosen and manually corrected for perspective distortion in advance. The centroids of the windows on the test images were recorded manually as ground truth (verification data) for the purposes of evaluation.

The match-scores are the result of running convolution on the HOG descriptors (Felzenszwalb et al. 2010, on which HoG-based template matching is partly based) for a template and a test image. To obtain a more meaningful value our work then divides the match-score by the area of the input template, multiplied by 100. Note that whilst match-scores tend to be less than one, a value close to zero does not necessarily represent a poor result, nor are they always less than one. How the match-score compares to other match-scores is more important. The HOG-based method was further extended by applying a pre-determined match-score threshold and by recording the true and false positives and computing precision, recall and F-measure. We use micro-averaging rather than macro-averaging to calculate precision and recall (based on the recommendation of Sebastiani (2002), their text classification focus being equally applicable to computer vision). During early runs the F-measure allowed the threshold value to be determined empirically and the starting size of the template to be optimised, using the size of the ground truth windows in the respective test images as a starting point. Initially, the threshold and template size were set per building.

The HOG-based method matches over a number of scales using an image pyramid. Our work extends the approach by non-maximally suppressing the results thus removing overlapping matches. This was achieved by reducing to zero those match scores which were less than any within a neighbourhood the size of the template. Matches were recorded as bounding boxes, centred on the pixel location which achieved the match and sized according to the template size which achieved the match.

The method was further extended to accommodate multiple templates (from Google *Image*, *Flickr* and the Pevsner *Architectural Glossary App*²), eliminating lower scoring overlapping matches by a further round of non-maximum suppression, *regardless* of the class of template that achieved the match. In order to obtain match-scores which were comparable across multiple templates the starting dimensions of the templates were made similar. 115 ground truths in total, across all four building test images were recorded in advance. This study only recorded ground truths which discriminated between windows and doors (classes), between the numbers of panes horizontally and vertically within the window class (subclasses) and between ‘single’ and ‘double’ doors (subclasses) and did not attempt to distinguish the named architectural styles, such as the Classical styles *Palladian* or *Baroque*, explicitly. The same set of templates was then used with all four

²From <http://www.yalebooks.co.uk>.

buildings, the starting size of the template being the same, regardless of building. The threshold was set per class ('door' or 'window') not per building as had been the case previously. Fundamentally, these two steps test the ability of the solution to work for many buildings and many templates.

Finally, the approach was trialled with texture maps, taken from a Trimble *3D Warehouse* model of Belton House, the model having been previously converted from the Trimble *SketchUp* format into CityGML using 3DIS' *CityEditor* plug-in for *SketchUp*, noting that the converted model needs to be a minimum of LOD 3 to accept window and door elements.

3.2 *Semantic and Geometric Enrichment of CityGML*

To ensure that a CityGML model is truly geo-spatial, a CityGML viewer will obtain real-world coordinates for its geometric components via a transformation from their model-space 3D Cartesian coordinate system to that in the spatial reference system attribute *srsName* (Whiteside 2009). The matching process previously described has obtained the 2D pixel coordinates for each vertex of the bounding boxes of the matches. Those coordinates then need to be converted to 3D Cartesian coordinates of the building model, which is achieved as follows.

CityGML specifies that texture map files (images) appear as covers for each *linearRing*, each *linearRing* representing a polygon in the building model (Gröger et al. 2012). It is feasible that the person who created the model cropped (or masked) the image file, perhaps only selecting part of that file to represent the texture map for the corresponding *linearRing*. CityGML records such cropping within 2D 'texture space', using *textureCoordinates* in the [0, 1] interval regardless of aspect ratio (Gröger et al. 2012). An example of such masking is shown in Fig. 2a. Consequently, matches that fell outside of this space were removed. Currently our work is only able to address rectangular polygon shaped *linearRing* and textures which do not repeat (or wrap) within a *linearRing* since the elimination of any cropped matches is more complex for non-rectangular polygons and repeating textures.

Remaining matches are then converted from pixel coordinates into texture space [0, 1]. This is achieved by converting the pixel coordinates into a value proportional to their position along their respective axes. The *gml:posList* for the *linearRing* contains the coordinates representing its vertices in 3D space. CityGML specifies that a *linearRing* must sit in a plane. Three points define a plane so we choose the coordinates of three of the *linearRing*'s vertices, denoting the corresponding points as Q, H and K, where \vec{H} and \vec{K} are the vectors for points H and K respectively and Q the origin. A 2D match coordinate in texture space is denoted as (u, v). This arrangement is illustrated in Fig. 2b. Thus, we are able to linearly interpolate to transform the 2D match coordinate (u, v) into 3D coordinates for point P using (1–3).

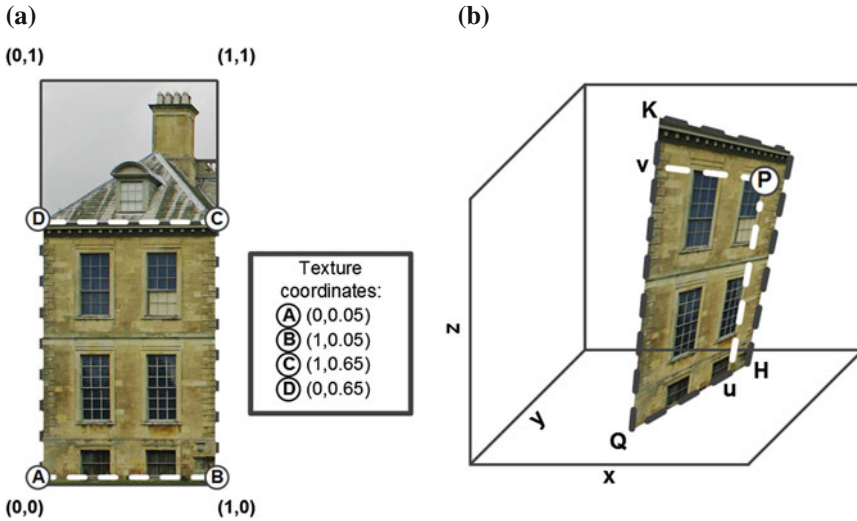


Fig. 2 **a** Example of how CityGML allows masking of an image file used as a texture map. The image is represented in ‘texture space’ in the range $[0, 1]$ regardless of aspect ratio. The *mask*, defined by the `textureCoordinates` for points *A*, *B*, *C* & *D* crops the image. The cropped image then maps on to a `linearRing` in the building model. Based on Gröger et al. (2012). **b** Visual representation of the terms in (1–3), as used to transform a 2D texture space match coordinate (u, v) into 3D Cartesian coordinates of the corresponding point *P*. *Q*, *H* and *K* define the plane and correspond to the vertices of the `linearRing` on which a texture map sits. The texture map has been *cropped* following that in (a)

$$P_x = Q_x + u\vec{H}_x + v\vec{K}_x \quad (1)$$

$$P_y = Q_y + u\vec{H}_y + v\vec{K}_y \quad (2)$$

$$P_z = Q_z + u\vec{H}_z + v\vec{K}_z \quad (3)$$

The results were then validated by computing precision, recall & F-measure for each texture map and by checking qualitatively (visually) in a 3D graphing tool that the coordinates of the matches lay in the plane of the corresponding `linearRing`.

Each match then generates a CityGML `<blgd:opening> <blgd:Window>` or `<blgd:opening> <blgd:Door>` text string, containing a `gml:pos` for each 3D Cartesian coordinate for the vertices of the match and a `gml:name` for `blgd:WindowType` or `blgd:DoorType` containing the type of template and the match-score. The latter is recorded as ‘Match-score’ followed by the corresponding score. The match-score is intended to give a confidence measure for the match that resulted in the geometry. The process is repeated for each texture map in the CityGML and for each `linearRing`.

4 Results

4.1 HOG-Based Template Matching

Operating with just one template per building and using the test images from Flickr and Wikipedia, the results of the HOG-based template matching on the five test images can be seen in Fig. 3. Strong F-measures between 0.74 and 0.88 were recorded, with one outlier, Blenheim Palace, of 0.34. Looking at the outlier result in detail it is clear that the template has in fact achieved a successful partial match, identifying ground truth windows which have more panes than the template. It proved possible to refine the results further through fine-tuning of the threshold per test image (per building). However, crucially, the approach allowed for the use of the same threshold (0.3) for all test images of the same class of architectural feature used (windows). This demonstrated early on that HOG-based template matching represented a method which could extend to multiple test images and multiple templates, something which is fundamentally advantageous.



Fig. 3 HOG-based template matching over varying scales on *Flickr* images of UK heritage properties Welford Park (“Welford Park, Newbury, Berkshire” by Amanda Slater is licensed under CC BY-SA 2.0; Template: from a Google *Image* Search result) (*top-left*) and Holyrood Palace (“Holyrood palace” by Asif Musthafa is licensed under CC BY-NC-ND 2.0; Template: “IBM Hursley” by Alexis Birkill is licensed under CC BY-NC-SA 2.0) (*top-right*), a *Wikipedia* image of Belton House (“South facing front to Belton House” by Wehha is licensed under CC BY-SA 3.0; Template: “IBM Hursley” by Alexis Birkill is licensed under CC BY-NC-SA 2.0) (*bottom-left*) and a *Flickr* image of Blenheim Palace (“Blenheim_Palace: Unprocessed (img_8298_hdr)” by Peter Gawthrop; Template: “Winslow Hall” by Lesley; both licensed under CC BY-NC 2.0) (*bottom-right*). For each image the template used is shown *top-left*, with matches as *thick white bounding boxes*



Fig. 4 Template *corpus* used for multi-template matching (From *top-left* to *bottom-right*, derivatives of 1–11: “Winslow Hall” by Lesley, is licensed under CC BY 2.0; 12: “Arched window (8034234792).jpg” by russavia is licenced under CC BY 2.0; 13 and 14: “St Chad” & “Lytham Hall” from Pevsner Architectural Glossary App by Yale University Press are copyright, used with permission; 15: “Portico” by Arthur John Picton is licenced under CC BY-NC 2.0; 16: combination of “Dublin yellow and red Georgian doors” by hugovk is licensed under CC BY-NC-SA 2.0; 17: “Dublin Doors” by Jim McDougall is licensed under CC BY 2.0, and 16)

Successful tests with intentionally incorrect templates (architectural styles not appearing in the test images) and those not optimally sized for the ground truths in the test image demonstrated further potential for the method to match multiple templates on one test image. Figure 4 shows the templates used for the multi-template matching stage. Note the relative morphological, brightness and colour similarity of many of the templates, illustrating the challenge for the method to correctly match at a subclass level, e.g. distinguishing a 3×3 pane window from a 3×4 pane window.

The results of matching against one of the four buildings, the *Flickr* image of Welford Park, can be seen in Fig. 5, where the text beside each bounding box represents the subclass of the template and the ‘match-score’. Note that for texture maps one can trivially remove/ignore matches in the sky or off the building.

Extending the method to multiple templates per test image, and using the same *corpus* of templates to run matching on multiple test images, proved feasible. The method demonstrated a low sensitivity to choice of template between different test images. This, alongside the low sensitivity to the match-score threshold, meant that, importantly, the HOG-based template matching has real potential for operating effectively across multiple templates and buildings. Incidentally, further refinement



Fig. 5 HOG-based template matching on a Flickr image of Welford Park using multiple templates, bounding box text represents the subclass of the template (line 1) and the ‘match-score’ (line 2)

for the removal of false positives could potentially be carried out with heuristics utilising the likely number of doors and their likely relative height.

With a successful implementation of multi-template matching against the Flickr test images, our work then moved on to matching the same templates to the texture maps of a Trimble 3D Warehouse building model of Belton House (containing twenty-four texture maps). Table 1 shows the results by F-measure, in detail. With the results grouped by class the F-measure of 0.92 indicates a high level of success in distinguishing between classes (window from a door). With the results grouped by subclass the F-measure of 0.61 demonstrates that the ability of the method to distinguish between a subclass was less successful. However, improvements could be gained using a wider range of templates. Figure 6 shows the matches displayed on the model itself.

Table 1 Multi-template matching *F-measure* for Belton House 3D model

Correct subclass identification e.g. '3 × 3 Pane Window'		Correct class identification e.g. 'Window'						
Building aspect	Total relevant	Total retrieved	True positives	False positives	Total relevant	Total retrieved	True positives	False positives
Front	50	61	37	24	50	61	49	12
Rear	44	52	30	22	44	52	42	10
Left	31	31	17	14	31	31	31	0
Right	30	30	16	14	30	30	29	1
All	155	174	100	74	155	174	151	23
			Precision	0.57			Precision	0.87
			Recall	0.65			Recall	0.97
			F-measure	0.61			F-measure	0.92



Fig. 6 Results of multi-template matching (*thick white bounding boxes*) on Belton House 3D mode 1 (“Belton House” by Johan is licenced for public use by Trimble)

4.2 *Semantic and Geometric Enrichment of CityGML*

Figure 7a and b show examples of the auto-generated CityGML from the matching.

This represents the semantic and geometric enrichment of the model. Future work will determine the correct location to auto-inject the corresponding textual descriptors into the existing CityGML model.

5 Conclusions

Using multiple buildings and with multiple templates this work has shown impressive results for automatically identifying different classes of architectural features (windows and doors) on the texture maps of CityGML building models, using a HOG-based template matching approach. Results were measured quantitatively using the precision and recall of matches, based on ground-truthing. The results were better than standard template matching.

Our study has also presented promising potential for identifying subclasses of architectural feature (different types of windows or doors). We have introduced a method for automatically converting the 2D pixel coordinates of window and door matches into 3D CityGML coordinates. Our approach also auto-generates the corresponding new CityGML.

(a)

```

<bldg:opening>
  <bldg:Window>
    <bldg:WindowType>
      <gml:name>3x3 Pane - Match-score 0.33</gml:name>
      <bldg:lod3MultiSurface>
        <gml:MultiSurface>
          <gml:surfaceMember>
            <gml:Polygon gml:id="_Belton_House_BD.0_PG.5">
              <gml:exterior>
                <gml:LinearRing gml:id="_Belton_House_BD.0_PG.5_LR.0">
                  <gml:pos>8261.667712 11532.527224 268.647945</gml:pos>
                  <gml:pos>7552.562582 11706.445573 268.647945</gml:pos>
                  <gml:pos>7552.562582 11706.445573 1187.170911</gml:pos>
                  <gml:pos>8261.667712,11532.527224,1187.170911</gml:pos>
                  <gml:pos>8261.667712 11532.527224 268.647945</gml:pos>
                </gml:LinearRing>
              </gml:exterior>
            </gml:Polygon>
          </gml:surfaceMember>
        </gml:MultiSurface>
      </bldg:lod3MultiSurface>
    </bldg:WindowType>
  </bldg:Window>
</bldg:opening>

```

(b)

```

<bldg:opening>
  <bldg:Door>
    <bldg:DoorType>
      <gml:name>Single Arch - Match-score 0.28</gml:name>
      <bldg:lod3MultiSurface>
        <gml:MultiSurface>
          <gml:surfaceMember>
            <gml:Polygon gml:id="_Belton_House_BD.0_PG.5">
              <gml:exterior>
                <gml:LinearRing gml:id="_Belton_House_BD.0_PG.5_LR.0">
                  <gml:pos>8496.369581 11474.963176 5294.921752</gml:pos>
                  <gml:pos>6616.547947 11936.016780 5294.921752</gml:pos>
                  <gml:pos>6616.547947 11936.016780 7827.194552</gml:pos>
                  <gml:pos>8496.369581 11474.963176 7827.194552</gml:pos>
                  <gml:pos>8496.369581 11474.963176 5294.921752</gml:pos>
                </gml:LinearRing>
              </gml:exterior>
            </gml:Polygon>
          </gml:surfaceMember>
        </gml:MultiSurface>
      </bldg:lod3MultiSurface>
    </bldg:DoorType>
  </bldg:Door>
</bldg:opening>

```

Fig. 7 **a** Example auto-generated CityGML for *window* matches. **b** Example auto-generated CityGML for *door* matches

Future work will focus on the refinement of subclass identification to improve matching scores, the addition of new classes and subclasses, machine learning of optimized templates, richer semantic descriptions of the architectural features and automatic injection of the enriched CityGML content into the models.

Acknowledgments Funded by an EPSRC Industrial CASE studentship with Ordnance Survey, GB; special thanks go to Isabel Sargent and David Holland from Ordnance Survey. Aside from templates 13 and 14 (see Fig. 4) all data used in this work are already publicly available at the locations referenced in the text.

References

- Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *Paper presented at the 17th IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, USA 20–26 June 2005.
- Dawe, S. (2013). King of the castles: Britain's built heritage rules Huffington post. <http://www.huffingtonpost.co.uk>. Retrieved July 11, 2014.
- de Fornel, P., & Sizun, H. (2006). *Radio wave propagation for telecommunication applications*. Berlin: Springer.
- Debevec, P. E., Taylor, C. J., & Malik, J. (1996). Modeling and rendering architecture from photographs: A hybrid geometry-and-image-based approach. *Paper presented at the 23rd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, New Orleans, USA, 04–09 August 1996.
- Dick, A. R., Torr, P. H. S., & Cipolla, R. (2004). Modelling and interpretation of architecture from several images. *International Journal of Computer Vision*, 60(2), 111–134.
- Döllner, J., & Hagedorn, B. (2007). *Integrating urban GIS, CAD, and BIM data by service based virtual 3D city models. Urban and regional data management—annual*. Leiden: Taylor & Francis.
- Dore, C., & Murphy, M. (2014). Semi-automatic techniques for generating BIM façade models of historic buildings. *Journal of Information Technology in Construction*, 19(2), 20–46.
- Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(9), 1627–1645.
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 726–740.
- Gröger, G., Kolbe, T., Nagel, C., & Häfele, K. (2012). OGC city geography markup language (CityGML) en-coding standard. Open Geospatial Consortium.
- Gröger, G., & Plümer, L. (2012). CityGML—interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71(July), 12–33.
- Ham, Y., & Golparvar-Fard, M. (2015). Mapping actual thermal properties to building elements in gbXML-based BIM for reliable building energy performance modeling. *Automation in Construction*, 49(Part B), 214–224.
- Hampson, K., Kraatz, J. A., & Sanchez, A. X. (2014). *The global construction industry and R&D. R&D Investment and Impact in the Global Construction Industry*. Abingdon: Taylor & Francis.
- Iqbal, Q., & Aggarwal, J. K. (2002). Retrieval by classification of images containing large manmade objects using perceptual grouping. *Pattern Recognition*, 35(7), 1463–1479.
- Isikdag, U., & Zlatanova, S. (2009). Towards defining a framework for automatic generation of buildings in CityGML using building information models. *Lecture Notes in Geoinformation and Cartography—3D Geo-Information Sciences*. Berlin: Springer.
- Johansson, B., & Kahl, F. (2002). Detecting windows in city scenes. *Lecture Notes in Computer Science—Pattern Recognition with Support Vector Machines*. Berlin: Springer.
- Jones, C. B., Rosin, P. L., & Slade, J. (2014). Semantic and geometric enrichment of 3D geo-spatial models with captioned photos and labelled illustrations. *Paper presented at the 25th International Conference on Computational Linguistics (COLING)—3rd Workshop on Vision and Language (VL)*, Dublin, Ireland, 23 August 2014.

- Kolbe, T. H. (2009). Representing and Exchanging 3D City Models with CityGML. Lecture Notes in Geoinformation and Cartography—3D Geo-Information Sciences. Berlin: Springer.
- Koutamanis, A., & Mitossi, V. (1993). Computer vision in architectural design. *Design Studies*, 14(1), 40–57.
- Koziński, M., & Marlet, R. (2014). Image parsing with graph grammars and Markov Random Fields applied to facade analysis. *Paper presented at the 14th IEEE Winter Conference on Applications of Computer Vision (WACV)*, Steamboat Springs, USA, 24–26 March 2014.
- Kroon, D.-J. (2011). Fast/robust template matching. MathWorks Inc. <http://uk.mathworks.com>. Retrieved September 01, 2014.
- Mayer, H., & Reznik, S. (2005). Building facade interpretation from image sequences. *Paper presented at the ISPRS Workshop on Object Extraction for 3D City Models, Road Databases, and Traffic Monitoring—Concepts, Algorithms, and Evaluation (CMRT)—WG III/4–5 IV/3*, Vienna, Austria, 29–30 August 2005.
- Meixner, P., Leberl, F., & Brédif, M. (2011). Interpretation of 2D and 3D building details on facades and roofs. *Paper presented at the 3rd Conference on Photogrammetric Image Analysis (PIA)—ISPRS Technical Commission III Symposium*, München, Germany, 5–7 October 2011.
- Ok, D., Kozinski, M., Marlet, R., & Paragios, N. (2012). High-level bottom-up cues for top-down parsing of facade images. *Paper presented at the 2nd Joint 3DIM/3DPVT International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, Zürich, Switzerland, 13–15 October 2012.
- Pevsner, N., Harris, J., & Antram, N. (1989). *Lincolnshire*. London: Yale University Press.
- Reznik, S., & Mayer, H. (2007). Implicit shape models, model selection, and plane sweeping for 3D facade interpretation. *Paper presented at the 2nd Conference on Photogrammetric Image Analysis (PIA)—ISPRS Technical Commission III Symposium*, München, Germany, 19–21 September 2007.
- Ross, L., Bolling, J., Döllner, J., & Kleinschmit, B. (2009). Enhancing 3D city models with heterogeneous spatial information: Towards 3D land information systems. *Lecture Notes in Geoinformation and Cartography—Advances in GIScience—12th AGILE Conference*. Berlin: Springer.
- Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, 34(1), 1–47.
- Sivic, J., & Efros, A. A. (2014). Urban-scale quantitative visual analysis. *ERCIM News—Special Theme: Smart Cities*, 98, 43–44.
- Smart, P. D., Quinn, J. A., & Jones, C. B. (2011). City model enrichment. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(2), 223–234.
- Sonka, M., Hlaváč, V., & Boyle, R. (2014). *Image processing, analysis, and machine vision* (4th ed.). Boston: Cengage Learning.
- Stadler, A., & Kolbe, T. H. (2007). Spatio-semantic coherence in the integration of 3D city models. *Paper presented at the 5th International Symposium on Spatial Data Quality (ISSDQ)*, Enschede, The Netherlands, 13–15 June 2007.
- van den Brink, L., Stoter, J., & Zlatanova, S. (2013). Establishing a national standard for 3D topographic data compliant to CityGML. *International Journal of Geographical Information Science*, 27(1), 92–113.
- Whiteside, A. (2009). Definition identifier URNs in OGC namespace. OpenGIS Best Practice document.
- Xiao, J. (2013). HOG-based template matching. <http://vision.princeton.edu/code.html#templateMatching>. Retrieved January 10, 2015.
- Zhang, Y., Xiao, J., Hays, J., & Tan, P. (2013). FrameBreak: Dramatic image extrapolation by guided shift-maps. *Paper presented at the 25th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, USA, 23–28 June 2013.
- Zhu, Q., Hu, M., Zhang, Y., & Du, Z. (2009). Research and practice in three-dimensional city modeling. *Geo-spatial Information Science*, 12(1), 18–24.

Stochastic Buildings Generation to Assist in the Design of Right to Build Plans

Mickaël Brasebin, Julien Perret and Romain Reuillon

Abstract The design of documents impacting potential new constructions, such as Right to Build plans, is a complex issue. New tools need to be proposed in order to systematically assess the impact of regulations on the building potential of the concerned areas. Furthermore, it is often not directly the morphology of new constructions that administrations and citizens would like to regulate but their properties with regard to other phenomena (solar energy potential, etc.). In order to tackle these issues, we propose in this article to explore building configurations and regulations using a stochastic building generator and a workflow engine. The workflow we propose for such an exploration will produce important amounts of data that we intend to release as OpenData in order for administrations, urban planners and citizens to be able to freely visualize and collectively choose the regulations that best suit their territory. Such amount of 3D geographical data also suggests new issues in geovisualization.

1 Introduction

The development of cities is usually regulated by a set of plans designed by local administrations that concerns different objects (i.e. construction, environment, transportation). These plans offer administrations a control over city evolutions supported by non public actors (for example, citizens, and promoters). Generally, the scope of

M. Brasebin (✉) · J. Perret
Université Paris-EST, IGN, COGIT, 73 Avenue de Paris,
94165 Saint-mandé, France
e-mail: mickael.brasebin@ign.fr

J. Perret
e-mail: julien.perret@gmail.com

R. Reuillon
Institut des Systèmes Complexes Paris Ile-de-France,
113 Rue Nationale, 75013 Paris, France
e-mail: romain.reuillon@iscpif.fr

such plans is determined by national laws that define which objects are concerned by a given regulation and which types of regulation can be applied on these objects. However, their designing phase is a difficult task. Papamichael and Protzen (1993) defines it “*as an activity aimed at producing a plan which is expected to lead to a situation with specific intended properties and without side- or after effects*”. Thus, a good plan design requires a systematic assessment on a whole territory. As the knowledge of such plans is expressed through textual legal texts, a very first step is to offer the possibility to correctly translate such knowledge into a simulation system. Furthermore, as it regulates the behavior of various city actors, such a system has to integrate their different strategies notably to detect possible loopholes in regulations in order to avoid unwished developments. Another issue is that the designer may want to control some phenomenon linked to objects regulation but without possibility to directly limit them.

For this work, a particular plan is considered: the Right to Build. Such a plan aims to control new constructions by defining a set of functional and morphological constraints. The interest of this plan is that it limits the development of the urban fabric which is strongly linked with environmental phenomenon (such as photovoltaic electricity production or urban heat island effect) that designer tends to control. However, regulations do not allow to directly control them. For example, in French regulation, the French National Urban Code allows the limitation of 3D shapes (i.e. height, roof slopes) in Local Right to Build Plans but forbids fixing a minimal solar energy received by building facades.

As the design of such plans is a progressive process that may introduce new problematics during discussions with actors; it requires testing new properties. Thus, our proposition is based on a database of possible building configurations (based on city actors behaviors) on which the designer can test these evolutive properties. The idea of testing the properties on a database is to separate actors behaviors from designer expected properties and to limit time calculation as the production of new databases is time consuming on a whole territory. The designer may test a large variety of properties without assessing new databases.

The aim of this paper is to propose a system that assists the design of plans by the exploration of potential configurations allowed by possible regulations. The idea is to inverse the design of regulation and to determine it according to a set of expected properties. Firstly, we present in this paper a review of works related to building generation and aided design about Right to Build assessment (Sect. 2). In our work, we consider two levels:

- A first level is the production of a possible building configurations database that represents Right to Build according to actors behavior and according to scenarios of regulations (Sect. 3);
- A second level is the determination of regulation scenarios that match with designers expected properties. We also discuss about possible uses and explorations of generated configurations (Sect. 4).

2 Related Work

In order to produce building configurations, our system needs a building generator that integrates Right to Build regulation.

Building generation: Building generation is a technique used in several domains including architecture, geosimulation, computer graphics and urban planning. Thus, numerous systems are designed with specificities according to its domain. Vanegas distinguishes two types of generators, not totally incompatible: geometric simulator (for example Parish and Müller 2001; Müller et al. 2006) and behavior based simulator. Only the second one takes into account or imitates human processes that produce buildings. This kind of simulator is widely used in territorial studies and traduces human behaviors through utility functions. Thus, optimization methods are generally used for this kind of simulators to optimize the utility function: Multi-Agent Systems (Ruas et al. 2011) or meta-heuristics like evolutionary algorithms (Frazer 1995) or simulated annealing (Bao et al. 2013) combined with geometric generative methods like primitive instancing (Perret et al. 2010; Kämpf et al. 2010) or shape grammars (Talton et al. 2011).

Generation with urban regulation: Among these generators, a set of propositions is focused on the integration of Right to Build regulation in order to assess constructability. It is assessed by producing buildable hulls from geometric constraints (El Makchouni 1987; Murata 2004; Brasebin et al. 2011); offering the possibility to explore a predefined set of parametric buildings respecting rules (Coors et al. 2009); generating buildings (Turkienicz et al. 2008; Brasebin 2014) or proposing extensions to existing buildings (Laurini and Vico 1999).

Design with building generation tools: As it is possible to generate rapidly lots of buildings with such tools, methods have been designed to support decision making with building generation. For example, (Kämpf et al. 2010) propose a multi-objective genetic algorithm that tries to determine the height and the roof shape from a set of building footprints in order to optimize both built volume and solar energy received by building surface. The designer can explore the Pareto front in order to choose a solution that provides the best compromise. Vanegas (2013) proposes to determine parameters from building grammar generation tool in order to reach environmental objectives (natural light, built density or visibility to landmark). In (Talton et al. 2011), an original solution is described to design the skyline. These authors provide a method that generates buildings according to a grammar in order to match with an objective shape seen from a view point.

If these methods are interesting to support decisions; they give one solution for an optimal set of properties and do not investigate the varieties of optimal configurations. Studying this variety is important as city actors do not always act rationally (i.e. in our problem produce optimal configuration) and may create sub-optimal solution that can cause undesirable effects. In this paper, we try to propose a solution that allows studying these sub-optimal configurations.

3 Proposition

Our proposed system is described in Fig. 1. The main idea of this system is to explore on a studied geographic zone (Sect. 3.1) a space of possible regulations (Sect. 3.2) for which adapted building configurations according to its input parametrization are generated (Sect. 3.3). In order to guide the propositions, a utility function determines which solution are good enough to be kept and some variety criterions are introduced in order to keep solutions variety (Sect. 3.4). The sampler proposes configurations to the classifier during a certain duration, these solutions are kept according to their variety criterions and the utility function value (Sect. 3.5). The processing of this exploration (Sect. 3.6) tool produces as final result a database that includes building configurations that optimize a utility function according to variety criterions.

3.1 Geographic Environment

The geographic environment delimits the studied zone. It contains a set of objects described in a model that extends existing standards (CityGML Gröger and Plümer 2012, COVADIS 2012, INSPIRE 2009). The full model is presented in (Brasebin 2014) and can be summarized in Fig. 2. The geographic environment contains notably a set of parcels on which the sampler can independently generate building configurations. The model also integrates existing buildings at different levels of detail (LOD1 or LOD2) that can influence constructability due to regulation (i.e. distance between buildings, maximal floor area ratio, etc.). The different integrated objects, their properties and their relationships can be used to define regulation that can be applied on sampled configurations.

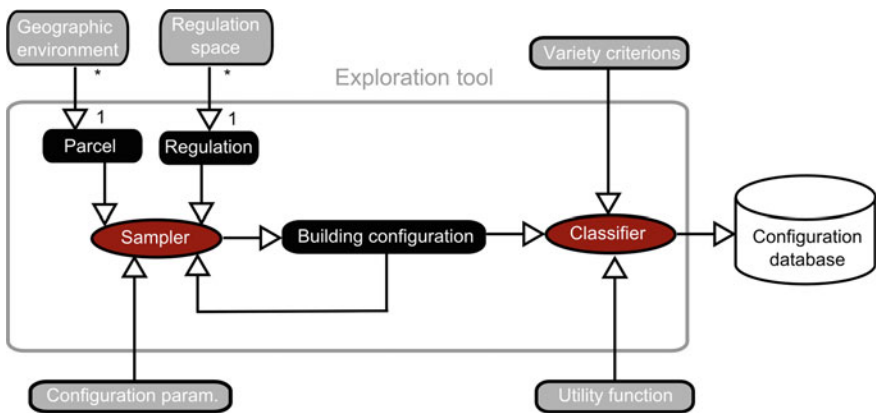


Fig. 1 Global schema of our proposition to produce a database of building configuration according to a regulation exploration space

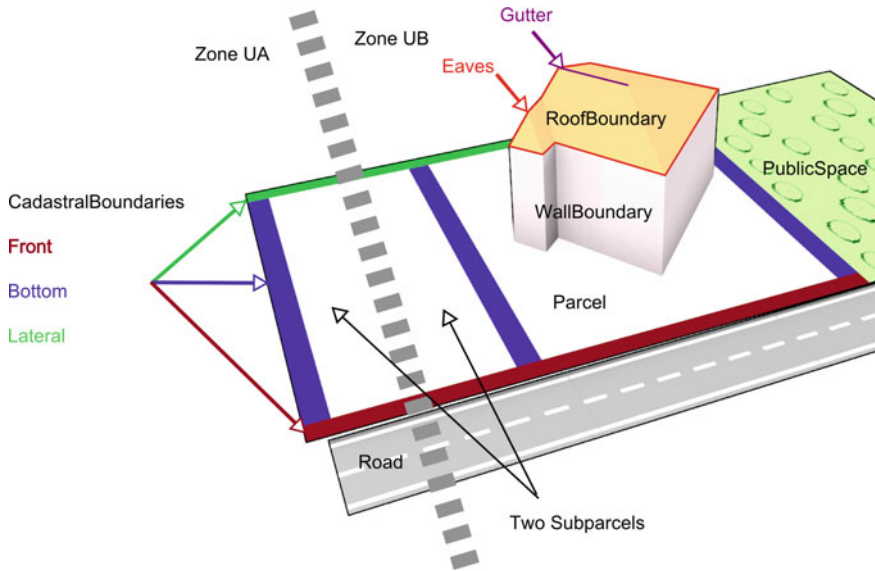


Fig. 2 Geographic environment to support rules definition

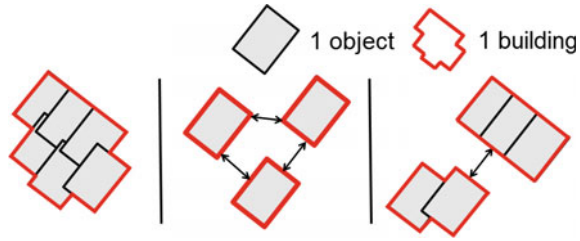
3.2 Regulation Space Exploration

In order to test different regulation scenarios, the designer has to define the space of possible regulations \mathfrak{R} . Thus, we consider a regulation as composed of a set of constraints: $r = \{c_i\}$ with $\{0 \leq i \leq n\}$. This single regulation is a parameter of the sampler in order to constraint generated building configurations. Each *const* is a Boolean function with parameters that indicate if a configuration *c* respects the constraint: $const(p, e, c, \{param_i\}) \in \mathbb{B}$ with $\{param_i\} \in \mathbb{R}^n$. For each $param_i$ an exploration space is defined. For example, in Right to Build regulation, a constraint can be a building height limitation, the parameter is the height value and the search space a set of values $\{10\text{ m}, 15\text{ m}\}$. Furthermore, the designer may test different constraint alternatives for a rule. Thus, for each $const_i$, the designer can define a set of $const_{i,j_i}$ that can be alternatively effective to form a regulation. With this notation, we can write that $\mathfrak{R} = \{const_{i,j_i}(p, e, c, \{param_{i,j_i}\})\}$. Then, the exploration task consists in simulating each $r \in \mathfrak{R}$. *c* is a building configuration as defined in the next section.

3.3 Building Configuration Sampler

In order to sample, we use a RJMCMC (Reversible-Jump Markov Chain Monte Carlo) sampler as described in (He et al. 2014; Brasebin 2014). Indeed, a RJMCMC sampler allows us to simulate building configurations of varying dimensions (Green

Fig. 3 The different types of building configuration



1995) (we do not have to set the number of buildings in advance for instance). It takes in inputs a parcel p in a geographic environment e and a regulation r formed by a set of rules. This sampler allows the generation of building configuration formed by a set of n objects. n is automatically determined by the system. In our experiments, used objects are boxes b described by a set of parameters $b = (x, y, l, w, h, \theta) \in \mathbb{R}^6$: position of its center (x, y) , length (l), width (w) and orientation (θ).¹ Parameterization of the sampler concerns the space sampling of the boxes, notably the minimum and maximum dimensions (width, height, depth) of boxes. Thus, we introduce a sampling function as: $sampling(p, r, e) = c \in (\mathbb{R}^6)^n$.

Furthermore, the sampler offers the possibility to generate different categories of building configurations (represented in Fig. 3), it can be composed by:

- n configuration of 1 box, for example to simulate individual buildings;
- 1 configuration of n boxes, for more complex buildings;
- or a mix of other types m configuration of n boxes.

The interest of this sampler is that generated configurations are relatively free and does not require initial knowledge as they are only composed of boxes. This allows the proposition of greater variety of configurations than in systems based on predefined construction processes. Nevertheless, unlikely combinations of building footprints might be generated. In this case, one can avoid such configurations by changing the parameter space (the dimensions of building footprints for instance) or by adding ad hoc constraints of the configurations.

3.4 Utility Function

The utility functions $\mu(c, e) \in \mathbb{R}^n$ aims to define the effectiveness of a configuration and to compare it to other ones in order to determine which one to keep (Michalewicz 1994). Thus, c is better than c' if $\mu(c, e) > \mu(c', e)$. This function has to embed the characteristics of the ideal solution and is the only link to control proposed configurations. In the context of building generation, the utility function can traduce an expected builder strategy (i.e. volume optimization in order to benefit from Right to

¹But other parametric objects can be used instead.

Build). It can also be used to produce configurations that incite undesirable behaviors in order to detect possible loopholes in a tested regulation (i.e. maximization of shadow projection on neighbor parcels in housing estates).

3.5 Building Configuration Classifier and Solutions Variety

In order to explore the variety of configurations proposed by the sampler, we propose to use a method to calibrate multi-dimensional models (Reuillon et al. 2015). The global idea of the method is to define a n -dimensional function $h(c, e) \in \mathbb{R}^n$ with $\{h_i(c, e)\}_{0 \leq i \leq n}$ that assesses configuration diversity. For instance, in our problem h_i can represent the number of boxes in a configuration the built ratio on considered parcel or other morphological indicators. Thus, it is possible to classify a configuration in a \mathbb{R}^n dimension space. For this classification task, each dimension is discretized according to possible h_i value ranges. In the case of continuous morphological indicators, the appropriate number of buckets has to be determined on an individual basis. During the processing of the exploration tool, an evaluation of $h(c, e)$ is processed and the configuration is classified in a n -dimension cell according to $\{h_i(c, e)\}_{0 \leq i \leq n}$ value. For each cell a configuration is stored and replaced by better configuration (in terms of utility function μ) when met. Figure 4 illustrates the process in a 2-dimension space.

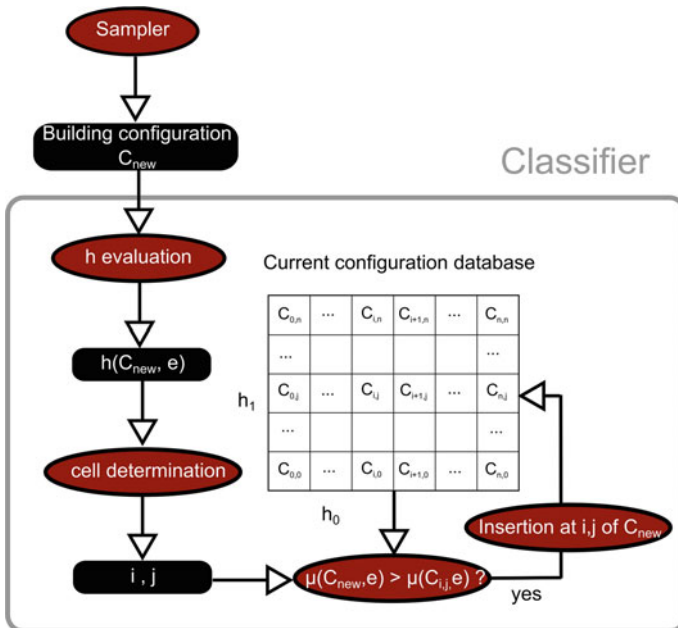


Fig. 4 Classification steps applied on a configuration with a 2 dimension variety function

3.6 Execution of the Exploration Tool

As the exploration tool is ran for one regulation and one parcel,² it is possible to distribute the execution of the whole system. Thus, for each pair $(r \in \mathfrak{R}, p)$ a partial configuration database $d_{r,p}$ can be produced. The production of such database can be modeled as an optimization process whose aim is to optimize the sum of all utility functions $\sum_{c \in d_{r,p}} \mu(c, e)$ and we propose to solve it with a simulated annealing algorithm. End condition is reached when there is no improvement during a sufficient number of iterations. It depends on the size of the search space. Methods to efficiently configure the optimization function are provided in Salamon (2002). The final database d is the union of all partial databases.

4 Uses of Generated Configurations and Exploration

In the previous section, we present a method to generate possible building configurations in order to produce a database. We discuss here the different possibilities to exploit such database.

4.1 Direct Extraction of Building Configurations

A very first result is the possibility to extract configurations for a set of parcels (some examples are presented in Fig. 5). At first intuition, we consider two approaches to extract such information:

- **Naive query:** a configuration per parcel is extracted according to a relevant partial database;
- **Best configuration query:** in order to get best configuration, this method extracts from each partial database configuration with the best utility function.

If these solutions are useful to help the designer in choosing scenarios of interest, they do not take into account the variety of generated configurations. Indeed, exploring a significant number of configurations can be quite time consuming.

4.2 Inverse Design

The aim of inverse design is to determine relevant objects from a set of properties. In the context of Right to Build regulation, the idea is to find the right regulations in order to preserve or optimize this set of properties (Fig. 6). As regulation design

²Or a urban block if simulations take into account new buildings from neighbor parcels.

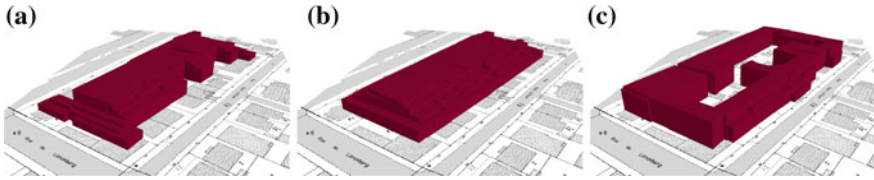


Fig. 5 Several generated building configurations on same parcels: with prospect constraint and 0.5 as maximal built ratio (a); with prospect constraint (b) and minimal distance to road and with distance to bottom separative limits and to road (c)

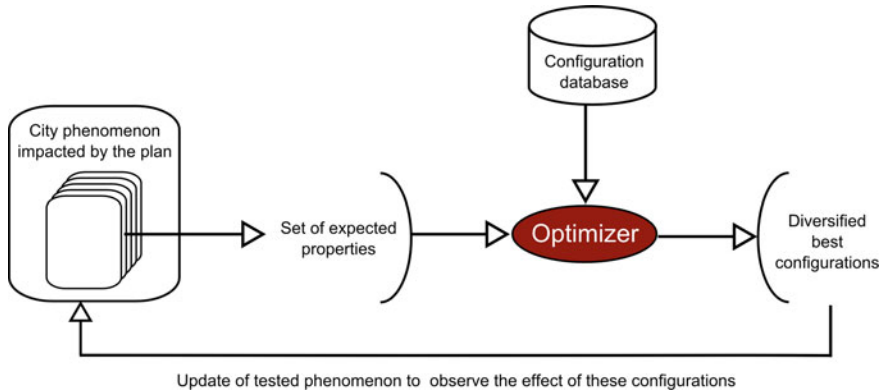


Fig. 6 Use of building configurations database to support inverse design

concerns various actors with different domains of interest (i.e. solar energy development, public park preservation, etc.), several sets of properties have to be tested in order to find a compromise between these different issues. For each issue, the corresponding set of properties is optimized in order to find in the database the best candidates in terms of properties optimization but also in term of diversity. Thus, we suggest preserving non-optimized solutions to enrich discussions between actors and to reinject them in order to explore some new aspects to assess if they fulfill requirements for being good compromises. As the exploration task described above may be time consuming, it seems to be relevant to reuse the configuration database to explore these city aspects not taken into account through utility function.

4.3 Navigation Between Configurations from the Inverse Problem

In order to take into account the variety of configurations that provide good results for a given inverse problem, we will explore in a next step the different possibilities to visually analyze building configurations.

Two types of works retained our attention:

- **Interpolation between configuration:** Bao et al. (2013) propose a method to produce intermediate building layout between two generated configurations. This method may be interesting if we want to interpolate building configurations between two regulations or between two variety measures. The major interest is to allow a smoother navigation in order to simplify the user observation and maybe to find a compromise between two solutions.
- **Navigation between configurations:** As inverse design generates different configurations, the idea is to provide a visualization of neighbour configurations according variety function evaluation (some operational propositions can be found in Averkiou et al. 2014; Kleiman et al. 2013). For one parcel or urban block of interest, it offers the possibility to see different configurations that solve similar problems but with different morphological aspects assessed by the variety function.

5 Conclusion

We present a proposition to simplify the design of Right to Build regulation with the exploration of building configurations. The main idea is based on the production of a building configurations database that integrates solution variety. Thus, the designer can explore different aspects of these building configurations in order to rapidly test different sets of properties that represent phenomena from considered territory. A research agenda is proposed in order to query this database and to interact with its content.

In the future, we will produce such a database on a zone of interest by using two open projects:

- **Open-Mole project**³ to parallelize the different tasks of the exploration process;
- **Simplu3D**⁴ in order to sample multi-dimensional building configurations.

This database will be released on dataverse⁵ in order to offer the possibility to collaborate with urban planners to help them in regulation design or to provide high-dimensional data to computer graphics or graphical interface researchers.

Acknowledgments This work was partially funded by the FEDER e-PLU projet (www.e-PLU.fr) and the Île-de-France Région.

³Website of OpenMole project: <http://www.openmole.org/>.

⁴Website of Simplu3D project: <https://github.com/IGNF/simplu3D>.

⁵<http://dataverse.org/>.

References

- Averkiou, M., Kim, V., Zheng, Y., & Mitra, N. J. (2014). Shapesynth: Parameterizing model collections for coupled shape exploration and synthesis. *Computer Graphics Forum (Special issue of Eurographics 2014)*.
- Bao, F., Yan, D. -M., Mitra, N. J., & Wonka, P. (2013). Generating and exploring good building layouts. *ACM Transactions on Graphics*, 32(4).
- Brasebin, M. (2014). *Les données géographiques 3D pour simuler l'impact de la réglementation urbaine sur la morphologie du bâti*. Ph.D. thesis, Université Paris-Est.
- Brasebin, M., Perret, J., & Haëck, C. (2011). Towards a 3d geographic information system for the exploration of urban rules: application to the french local urban planning schemes. In *28th urban data management symposium (UDMS 2011)*.
- Coors, V., Hünlich, K., & On, G. (2009). Constraint-based generation and visualization of 3d city models. In J. Lee & S. Zlatanova (Eds.), *3D geo-information sciences* (pp. 365–378). Lecture notes in geoinformation and cartography Berlin Heidelberg: Springer.
- Covadis (2012). Standard de données covadisplan local d'urbanisme—plan d'occupation des sols plu et pos - version 2. Technical report, Commission de validation des données pour l'information spatialisée.
- El Makchouni, M. (1987). Un système graphique intelligent d'aide à la conception des plans d'occupation des sols: Sygripos. In *12th Urban Data Management Symposium*.
- Frazer, J. (1995). *An evolutionary architecture*. Architectural Association: Themes Series.
- Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4), 711–732.
- Gröger, G., & Plümer, L. (2012). Citygml—interoperable semantic 3d city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71, 12–33.
- He, S., Perret, J., Brasebin, M., & Brédif, M. (2014). A stochastic method for the generation of optimized building-layouts respecting urban regulation. In *ISPRS/IGU Joint international conference on geospatial theory, processing, modelling and applications 2014*, Advances in spatial data handling and analysis—Select Papers from the 16th IGU Spatial Data Handling Symposium.
- INSPIRE. (2009). D2.8.I.6 INSPIRE Data specification on cadastral parcels—Guidelines. Technical report.
- Kämpf, J. H., Montavon, M., Bunyesc, J., Bolliger, R., & Robinson, D. (2010). Optimisation of buildings' solar irradiation availability. *Solar Energy*, 84(4), 596–603.
- Kleiman, Y., Fish, N., Lanir, J., & Cohen-Or, D. (2013). Dynamic maps for exploring and browsing shapes. *Computer Graphics Forum*, 32(5).
- Laurini, R., & Vico, F. (1999). 3d symbolic visual simulation of building rule effects in urban master plans. In R. Shibasaki, & Z. Shi (Eds.) *The second international workshop on Urban 3D/Multi-Media Mapping (UM3'99)* (pp. 33–40).
- Michalewicz, Z. (1994). Evolutionary computation techniques for nonlinear programming problems. *International Transactions of Operational Research*, 1, 223–240.
- Müller, P., Wonka, P., Haegler, S., Ulmer, A., & Van Gool, L. (2006). Procedural modeling of buildings. *ACM Transactions on Graphics*, 25(3), 614–623.
- Murata, M. (2004). 3D-GIS application for urban planning based on 3D city model. In *24th Annual ESRI International User Conference* (pp. 9–13).
- Papamichael, K. M., & Protzen, J. P. (1993). The limits of intelligence in design. In *Focus Symposium on "Computer-Assisted Building Designs Systems", of the Fourth International Symposium on System Research, Informatics and Cybernetics*, Baden-Baden, Germany.
- Parish, Y. I. H., & Müller, P. (2001). Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques, SIGGRAPH '01* (pp. 301–308). New York, NY, USA: ACM.
- Perret, J., Curie, F., Gaffuri, J., & Ruas, A. (2010). A multi-agent system for the simulation of urban dynamics. In *10th European Conference on Complex Systems (ECCS'2010)*.

- Reuillon, R., Schmitt, C., De Aldama, R., & Mouret, J.- B. (2015). A new method to evaluate simulation models: the Calibration Profile (CP) algorithm a new method to evaluate simulation models: The Calibration Profile (CP) Algorithm. *Journal of Artificial Societies and Social Simulation*, 18(1). <http://jasss.soc.surrey.ac.uk/18/1/12.html>.
- Ruas, A., Perret, J., Curie, F., Mas, A., Puissant, A., & Skupinski, G., et al. (2011). Conception of a gis-platform to simulate urban densification based on the analysis of topographic data. In A. Ruas (Ed.), *Advances in Cartography and GIScience* (Vol. 2, pp. 413–430), volume 6 of Lecture Notes in Geoinformation and Cartography Berlin, Heidelberg: Springer.
- Salamon, P., & Sibani, P. (2002). Frost, R. *Selecting the Schedule*, 13, 89–98.
- Talton, J. O., Lou, Y., Lesser, S., Duke, J., Měch, R., & Koltun, V. (2011). Metropolis procedural modeling. *ACM Transactions on Graphics*30(2).
- Turkienicz, B., Goncalves, B., & Grazziotin, B. P. (2008). Cityzoom: A visualization tool for the assessment of planning regulations. *International Journal of Architectural Computing*, 6(1), 79–95.
- Vanegas, C. A. (2013). *Modeling the appearance and behavior of urban spaces*. Ph.D. thesis, Purdue University.

3D Marine Administration System Based on LADM

Aikaterini Athanasiou, Ioannis Pispidikis and Efi Dimopoulou

Abstract Oceans cover almost two-thirds of the Earth's surface. They are the primary regulator of the global climate and sustain a huge variety of plant and animal life. Maritime environment needs to be organized and precisely determined, in order to be sustainable. The registration of marine legal boundaries is a necessary condition for the protection of a living organism, which flows, changes, reverses itself, but is not limitless. Research has confirmed that the common pattern of people-land relationship also exists in the marine environment. Moreover, the marine Cadastre concept suggests that the complexity of interests in marine space is similarly encountered in land. The extension of Cadastre functions from land to marine space is considered reasonable. An inventory of interests that exist in the marine environment is important. At the same time, laws that are the basis of these interests need to be identified and their effect qualified and visualized. The administration of the marine space remains partial and complex, mainly deriving from political interests and strategic benefits. This could be overcome by designing a marine administration system, in accordance with the international practices. A conceptual model may be considered as the base of such system. This model should clearly depict the relevant entities of the system and the relationships between them. Modeling and standardizing systems and processes at an international level, requires the harmonization with international standards and specifically with the ISO 19152—Land Administration Domain Model (LADM), which so far remains a challenge. The aim of this paper is to present how rights, restrictions and responsibilities (RRRs) relating to marine space may be organized, in order to develop a marine administration model based on LADM, followed by the database implementation, to support effective and efficient decision making in marine governance.

A. Athanasiou (✉) · I. Pispidikis · E. Dimopoulou
School of Rural and Surveying Engineering, National Technical University of Athens, 9
Iroon Polytechniou Str, 15780 Zografou, Greece
e-mail: catherineathanasiou@gmail.com

I. Pispidikis
e-mail: pispidikisj@yahoo.gr

E. Dimopoulou
e-mail: efi@survey.ntua.gr

Keywords Marine administration system • LADM • 3D modeling • 3D cadastre

1 Introduction

The interests of a nation do not stop at the land–sea interface. They continue into the marine environment (Williamson et al. 2010). This summarizes the important issue dealt within this paper. Marine environment, acknowledging the precedence of the terrestrial borders, needs to be organized and precisely determined whereas the interests involved are complex and wide. The marine environment has always been a place for different activities and uses, involving various parties; however for most countries, despite important efforts, its coordinated management remains at stake. Space must be ordered, to be sustainable. At this level marine space needs to be properly managed, within the international framework. Hence, each government should be responsible for providing appropriate management of its marine space, whereas a number of legislative rules and restrictions result from a complex framework of local, state, national legislation and international conventions, the global environment movement and maritime practices that have been implemented.

Over the last two decades, countries with extensive coastlines and confined marine space, where they exercise sovereignty and powers, archipelagic states inclusive, have dealt with the concept of Marine Administration System (MAS). Among others Australia, Canada, the Netherlands and the United States developed systems for the administration of marine interests and the sustainable management of marine resources. Their efforts are at a development stage, looking from practices in the fields of marine Cadastre and Marine Spatial Data Infrastructure (MSDI). In addition, the international geoinformation community has increased its research efforts towards the development of MAS. The predominant subject they have dealt with, is the bulk of technical, institutional, legal and stakeholder issues that must be considered (Sutherland and Nichols 2002; Binns et al. 2003; Barry et al. 2003; Fraser et al. 2003; Binns 2004; Rajabifard et al. 2006; Fulmer 2007; Abdullah et al. 2014).

For defining a MAS, certain issues need to be addressed: what type of information should a MAS contain? What is the relationship between subject-right-object in the marine environment? What types of law define the interests? What types of interests exist in marine environment? Which is the property/tenure object? Who are the stakeholders and which are the entities with statutory duty to assign rights and control the implementation? Is the 3D registration of interests in fact required? What is the framework for representing and visualizing these interests interacting with one another, capturing the inherent 3D nature of marine interests? What is the basic reference unit and how this can be defined, delaminated and demarcated? (Ng'ang'a et al. 2004; Sutherland 2005).

The answer to the above raised questions is under discussion recently. The existence of different MASs shows the different perspectives of the various

jurisdictions. The absence of common standards is apparent. However, there are common elements that must be taken into account when defining a MAS. As key points, the majority of marine interests exist because they are implemented by the law. Furthermore, the three dimensional volumetric nature of marine environment (with rights on the surface, water column, seabed, and subsurface) requires 3D cadastral representation, where even the 4-dimensional nature of the person-space relationship needs to be registered (rights and interests within a legal framework that changes over time).

The most challenging part when considering the ideal MAS, is a functional 3D registration and visualization of the legal objects and their physical counterparts. Simultaneously, this constitutes one of the crucial requirements in the development of land cadastral systems, in order to serve multiple 3D applications such as land and property taxation and public communication (Aien et al. 2013a, b).

In the conceptual level, different land cadastral data models (2D and 3D) have been developed by jurisdictions, organizations and software developers (Aien et al. 2013a). However, LADM constitutes the most solid common ground in land administration, as an officially International Standard ISO 19152. Standardization supports the design and development of an administration system and can be considered as a requirement for the development of a National Land Information System. As Lemmen (2012) states “in spite of the available basic standards (for modeling the Unified Modeling Language: UML), exchanging structured information (eXtended Markup Language: XML) and ISO generic geo-information standards, there is still one important aspect missing: a standard and accepted base model for the land administration domain”. LADM deals with this problem by providing a conceptual description for land administration based on common grounds. The data model has four core packages: (1) parties, (2) basic administrative units, rights, responsibilities and restrictions, (3) spatial units and their spatial extension, through surveying and spatial representation subpackage (Fig. 2). It constitutes a generic domain model, which is expandable. The question arises as to whether the different land cadastral data models are adequate to model marine space.

This paper is structured as follows. In Sect. 2, the various Marine Administration concepts are examined along with their developments in the different jurisdictions in terms of web applications in the public domain. Section 3 outlines the complex issues concerning the marine environment and elaborates on the classification of marine interests, in regard to the law that also defines their spatial dimension. Section 4 briefly provides different initiatives in the domain of cadastral data modeling, and investigates, based on literature, how these structures may be extended to the marine environment. The conceptual classification of the marine entities and relationships based on LADM, are presented in Sect. 5, followed by the database implementation. Finally selected cases are visualized, using ArcScene and Google Earth.

2 Marine Administration Systems

2.1 *Basic Concepts*

The concept of marine Cadastre evolved to bring coherence in the various approaches (Williamson et al. 2010). Given its stage of development, it has many definitions, as Nichols et al. (2006) extensively described. It can be broadly defined as “an information system that records, manages and visualizes the interests, and the spatial (boundaries) and non-spatial data (descriptive information about laws, stakeholders, natural resources), related to them”.

The marine Cadastre delivers the fundamental datasets that are especially vital to marine management. The functionality of a Cadastre in supporting the MSDI is recognized after a protracted debate about how to use and adapt land-based tools to service marine needs. In modern theory, the cadastral component and the SDI are fundamental to the way marine information is developed and shared, and ultimately for competent marine administration. Most ocean and coastal management problems are of a spatial nature and therefore, the development of a marine component to national and regional SDIs is imperative to the effective management of the marine environment (Williamson et al. 2010). MSDI must be based on ‘interoperability’ (seamless databases and systems). International standards’ organizations are addressing the development of standards for both land-based and marine-based spatial data and technologies. Important standards developments relating to coastal and marine data include the S-100—Geospatial Information Registry, which provides important compatibility for data sharing in the hydrographic information community (S-100 is being based on the ISO/TC211 base standard) (Ward et al. 2009).

A multidimensional management of marine space requires additional types of information concerning the spatial extents of marine Cadastre. This includes scientific information (e.g. geology, hydrology, biology etc.) and the physical counterparts of legal registrations, in case of underwater tunnels, pipelines, utility networks, submerged archaeological cities or even underwater luxury hotels and facilities—a growing trend in tourist destinations. Such a system, is implemented by using geographical information system (GIS) tools, interactive mapping applications, marine and coastal data and metadata, in order to support decision making process in the marine environment.

2.2 *International Initiatives*

The marine Cadastre concept in Geomatics-related research and literature is growing since 1999 (Griffith-Charles and Sutherland 2014). These trends mainly focus on the development of unique cadastral systems for the sustainable management of marine resources within a range of jurisdictions. An overview of online

mapping and cadastral approaches that support integrated coastal and marine planning for United States, Australia and Canada is briefly presented below. Despite developments in MASs, seafloor may be three-dimensionally presented, but the visualization of interests remains 2-dimensional.

In the United States, the National Oceanic and Atmospheric Administration (NOAA) and the Bureau of Ocean Energy Management (BOEM) have developed a web-based integrated marine information system that provides data, tools, and technical support for ocean planning, named marine Cadastre. The base data on this site focuses on providing a legal framework for authoritative data. Over time the project included many other types of data (NOAA and BOEM 2010). Geoscience Australia created the Australian Marine Spatial Information System (AMSIS), a web-based interactive mapping that contains over 80 layers of information and provides legal information for the visualized boundaries. Issues in the use of natural rather than geometric boundaries to define jurisdictional limits were considered, as well as expanding the Australian SDI to develop and support a marine Cadastre. Custom queries are possible (Geoscience Australia 2010). In 2008 Atlantic Coastal Zone Information Steering Committee (ACZISC), supported by Geoconnections Canada, developed the Coastal and Ocean Information Network for Atlantic Canada (COINAtlantic). A search utility locates marine and coastal datasets and offers the user the option to add and display datasets in a graphic map interface. Data displayed can be queried by point-and-click methods and the system delivers text/attribute data results directly from linked spatial databases (COINAtlantic 2009).

3 Registration of Legal Objects

3.1 Legal Framework

The interests, to be registered in a MAS, concern the activities that take place and have an impact on the marine environment. Commonplace in all marine interests is that they base on the law. Thus in order to develop a MAS, the registration of laws is considered reasonable.

United Nations Convention on Law of the Sea (UNCLOS), concluded in 1982, formed the cornerstone of the legal mechanism and balanced the interests of states. One of the ways established was through the division of marine space into different maritime zones (Territorial Sea, Contiguous Zone, Exclusive Economic Zone (EEZ) and Continental Shelf). Coastal states enjoy sovereignty or sovereign rights over the maritime zones in the waters adjacent to their coasts and thus the jurisdiction to establish their laws and policies, in compliance with UNCLOS guidelines. According to Cockburn et al. (2003), UNCLOS influences a ratified nation's MAS in several ways, like breadth, depth, what rights can be included in the ocean areas and hence what spatial information is contained therein and has an effect on the evidence that can be used for boundary demarcation and delineation. UNCLOS

has created a complex three-dimensional mosaic of private and public rights (Ng'ang'a et al. 2004).

In this mosaic, a number of marine¹ policies are added (regulations and directives) provided by the European Union (EU) for Member states, as Greece. Some of the prevailing EU directives refer to: maritime spatial planning (MSP) (Directive 2008/56/EC, Regulation EU No. 1255/2011), Integrated Maritime Policy (the Blue Book) which highlight that all matters relating to Europe's oceans and seas are interlinked, and that sea-related policies must be developed in a joined-up way.

Within this framework and in accordance to the above international provisions, each member state is entitled to determine its national legislation either incorporating the international developments or retaining its exclusive right to define specific rights, restrictions and responsibilities of sovereign areas.

3.2 *Type of RRRs*

UNCLOS defines thoroughly the interests of each state in marine space and establishes its legal regime (Table 1). The type and the extent of sovereign rights depend on the different maritime zones to which they refer. It is important to note that the state sovereign rights applied on the marine space may be likened to a specific kind of freehold ownership of terrestrial space - the difference lies on the fact that the core of the right is intact. Sovereign rights involved with the marine space are exercised under specific conditions excluding the wide power and freedom of a freehold ownership. Therefore the state has the power to dispose it (in order to be exploited for a specific purpose), but the state remains the exclusive owner, according to UNCLOS. Regardless this particularity, sovereign rights involve a variety of powers and rights that should be recorded by MAS. These include:

Public Interests: Public rights refer mainly to the constitutional right of every citizen of the state having an unlimited without obstacles access statewide (terrestrial and marine space). Interrelated with this, is the right of personality.

Environmental Interests: We refer to provisions relevant to the protection and conservation of water resources, places of protected beauty and cultural heritage. These places are pre-determined by the law and the rights involved are of supreme importance and mandatory, in comparison to the following functional rights.

Functional Interests: The term functional is used in order to highlight the particularities of the marine environment and the kind of different transactions that can be found in the marine space. Progressively functional rights tend to acquire a private

¹Intergovernmental Oceanographic Commission (IOC) of UNESCO, NOAA of USA and Department for Environment Food and Rural Affairs (DEFRA) of UK use the term marine spatial planning, while EU argues that maritime spatial planning underlines the holistic cross-sectorial approach of the process.

Table 1 Interests of each state defined by UNCLOS (Sutherland 2005)

	Territorial sea (up to 12 n.m)	Contiguous zone (up to 24 n.m.)	Exclusive economic zone (up to 200 n.m.)	Continental shelf (up to outer limit of the continental slope or 200 n.m. from the baselines, not exceed 350 n.m. from the baselines or 100 n.m from the 2,500 m. isopath)
Rights and duties of other States	<ul style="list-style-type: none"> • All states enjoy the right of innocent passage 	<ul style="list-style-type: none"> • All states enjoy the right of innocent passage 	<ul style="list-style-type: none"> • All states enjoy the freedoms of navigation and overflight • All states are entitled to lay submarine cables and pipelines • Land-locked and geographically disadvantaged states have the right to participate, on an equitable basis, in the exploitation of an appropriate part of the surplus of the living resources of the EEZ of coastal States of the same subregion or region 	<ul style="list-style-type: none"> • All states are entitled to lay submarine cables and pipelines
Rights of the Coastal State	<ul style="list-style-type: none"> • Complete Sovereignty to the air space, surface, water column as well as to its bed and subsoil 	<ul style="list-style-type: none"> • The Coastal State may exercise the control necessary to prevent infringement of its customs, fiscal, immigration or sanitary laws and regulations within its territory or territorial sea 	<ul style="list-style-type: none"> • Functional Sovereign rights of the waters superjacent to the seabed and of the seabed and its subsoil for the purpose of: <ul style="list-style-type: none"> – exploring and exploiting of mineral and natural resources – production of energy – conserving and managing the natural resources, whether living or non-living • Jurisdiction with regard to: <ul style="list-style-type: none"> – the establishment and use of artificial islands, installations and structures – marine scientific research – the protection and preservation of the marine environment 	<ul style="list-style-type: none"> • Functional Sovereign rights of the seabed and subsoil for the purpose of: <ul style="list-style-type: none"> – exploring it and exploiting its natural resources (consist of the mineral and other non-living resources together with living organisms belonging to sedentary species)

nature, associated with individual stakeholders that coexist with the state rights. In a wide sense, this term sets the limits of rights, which involve mainly the different ways of use, management and appropriation. In other words, in the marine environment the rights are limited in terms of space, duration and most importantly the extent, the content that refers only to the different kind of uses and management. The stakeholders are not owners but only beneficial “users”. Functional rights are granted either by leasing contracts or through licensing. It has to be said that the power of granting remains national and no freehold ownership is involved.

3.3 RRRs Requiring 3D Registration and Representation

Few marine activities take place on the “surface” of the water. Nearly every marine activity takes place in a volume of water (Ng’ang’a et al. 2004). Marine environment has an inherent volumetric nature involving the exercise of rights to the surface, water column, seabed, and subsoil. Regarding the spatial reference of marine interests, three kinds of RRRs can be distinguished:

- Exclusive use that their physical presence exclude any other rights from being present in that space so the visualization of this indicates the restriction boundaries for any other interest.
- Time based, as some RRRs may be in effect only during certain periods of time. So the coexistence of two different rights, like mooring and fishing is allowed in different times.
- Without geographic location; opportunities may be tied to simply holding a license or being owner or operator of a licensed fishing vessel. (Griffith-Charles and Sutherland 2014)

Table 2 classifies interests, taking into account their spatial dimension, time and physical presence.

3.4 Basic Reference Unit

Modern cadastral systems consider land unit (cadastral parcel—to which unique property interests are attached) as the basic reference unit for collecting, storing and disseminating information. Respectively, in a marine cadastral system a basic unit is necessary to be defined, regardless the complexity of precise descriptive and spatial definition of marine parcel, as Ng’ang’a et al. (2004) stated. Particularly, having in mind that the coexistence of marine activities in time and space is frequent, individual ownership of a parcel cannot be considered as the norm. In addition, a variety of activities can occur in the same longitude and latitude, but in different depth. The inherent volumetric 3D nature of marine space is apparent and

Table 2 Interests in marine space and their spatial extend

Interests	Spatial extent	Specializations	Presence				Representation boundaries of legal objects	
			Permanent		Non-Physical	Temporal	2D	3D
			Physical					
Sovereignty and jurisdictional rights	Shoreline				•			
	Internal waters				•			
	Territorial Sea				•			
	Contiguous zone (if exists)				•			
	EEZ (if exists)				•			
	Continental Shelf				•			
	Marine protected areas	Ecologically sensitive areas, preservation of sea birds and marine fauna and flora				•		•
Environmental rights	Natura 2000				•			•
	Archeological sites			•				•
	Heritage areas			•				•
	Shipwrecks			•				•
	Exploring and exploitation of minerals				•			•
Functional powers (state and private rights through grants, leases, licenses or concessions)	Areas of seismic surveys						•	
	Exploring and exploitation of natural resources	Fishing (commercial and recreational)				•		
		Aquaculture						•
	Utility networks	Oil and gas pipelines			•			•
		Telecommunications cables						•
		Electricity cables				•		•

Table 2 (continued)

Interests	Spatial extent	Specializations	Presence			Representation boundaries of legal objects	
			Permanent	Non-Physical	Temporal	2D	3D
						Physical	
	Sand mining areas						
	Marine energy parks		•			•	
	Underwater resorts	Hotels	•				•
		Facilities	•				•
	Coastal construction areas	Harbours	•				•
		Shipyards	•				•
		Mooring areas			•		
	Organized swimming areas						
	Areas for watersports			•			•
	Submarine touristic tours				•		
	Drilling platforms		•				
	Artificial islands		•				
	Artificial platforms		•				
	Spatial Planning areas		•				
	Special rights	Military exercise areas			•		•
		Areas of scientific research					•
Public rights	Navigation	Innocent passage					
	Swimming areas			•			•

consequently its appropriate presentation is needed. The delineation of marine boundaries presents certain legal, technical and scientific problems. The basic issue involved is the dynamic nature of the marine environment that renders the implementation and the physical demarcation of legally defined marine boundaries difficult to materialize (Ng'ang'a et al. 2004). Difficulty in delimitation of the marine boundaries increases because of the fact that these often refer to physical features (e.g., high water, the shoreline). Land water interface is ambulatory and most boundaries and limits follow the motions of that traditionally interface.

A succinct descriptive definition of marine parcel could be: “A confined space having common specifications for its internal, mainly used as reference to locate a phenomenon. A marine parcel facilitates the distinction between contiguous territories and provides information concerning this phenomenon through appropriate codification” (Arvanitis 2013).

Spatially the basic reference unit, could be defined as: a multidimensional marine parcel or a series of (special purpose) volumetric marine parcels (Ng'ang'a et al. 2004) or as sea surface objects, water volume objects, seabed objects, and sub seabed objects (Rahman et al. 2012) or as a single piece of marine space deriving from the determined and standard division of the maritime surface using a grid of specific dimensions and subdivisions if needed. It is specified by geodetic coordinates of the surrounding boundaries. This method is already in use for defining the blocks in the domain of minerals exploitation. The combination of these methods is feasible (Fig. 1).

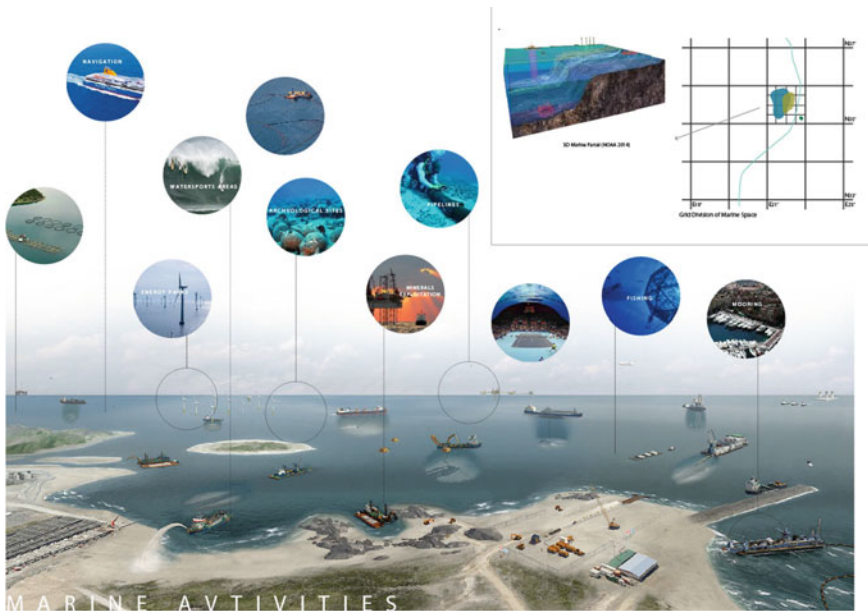


Fig. 1 Marine activities and marine parcel

In LADM the more generic term of Basic Administrative Unit (BAUnit) is introduced, in order to allow the association of a unique and homogenous right to a combination of spatial units, with unique identifiers. Respectively, in the marine environment it could be a unit with common specifications for its internal, which spatially can be described by the UNCLOS zone it belongs, the physical layer and the responsible port authority or by the coordinates of the grid it belongs (according to the reference system of the country).

4 Modeling Approaches

The development of a MAS can be termed as a “combinatory play”. It has several parts (as every system): a data model for the information itself, the data format to support the data model, a database to manage data, and visualization tools for communicating, exploring and representing the information (Shojaei et al. 2013). This paper focuses on the data modeling aspects of a marine administration system and more specifically examines the implementation of LADM to marine space.

Data modeling development is a repetitious and cyclical process used to create a perfect model of the real world, which aligns the seemingly random jumble of stuff into entities with relationships. The first step in the data modeling process is to define the overall scope and content of the model. It starts from mapping the concepts and their relationships of the real world to a conceptual model, through the UML and Entity-Relationship (ER) model. The ER model is a way of graphically representing the logical relationships of entities (or objects) in order to create a database. The main components of ER models are entities (things) and the relationships that can exist among them. The conceptual data model is then translated into a logical data model, which records in detail the structures of the data that can be implemented in a database. The last step in data modeling involves transferring the logical data model to a physical data model that organizes the data into tables (Aien et al. 2013). The abstraction is increased as one goes from human-orientation to computer-orientation. The physical design of the database specifies the physical configuration of the database on the storage media. This includes detailed specification of data elements, data types, indexing options and other parameters residing in the database management system (DBMS) data dictionary.

Starting point of data modeling in land administration can be considered the visualization of the triple Object–Right–Subject, as introduced by Henssen (1995) and that pattern embodies the most solid common ground in land administration until today (Lemmen 2012).

Different initiatives in land cadastral data modeling have been developed (Fig. 2), since the administration requirements differ among the various jurisdictions. The endeavor for the development of 3D Cadastre is increased.

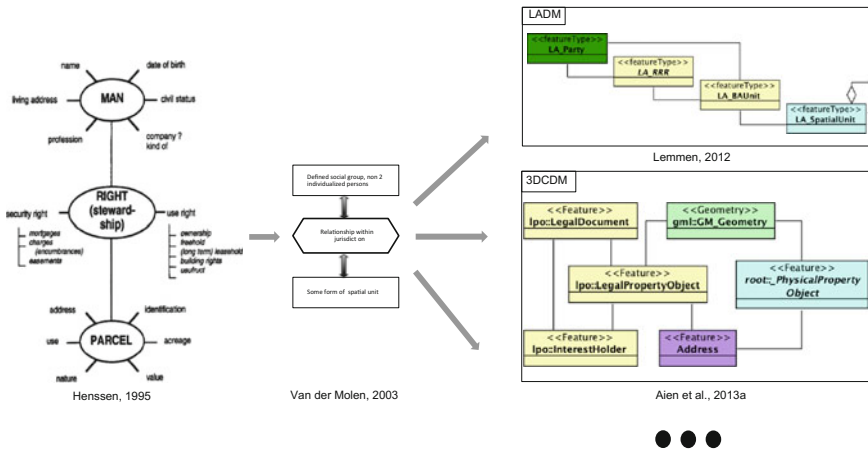


Fig. 2 Cadastral data models

4.1 3D Data Models for Land Administration

From the 3D perspective, LADM supports both 2D (LA_BoundaryFaceString) and 3D objects (LA_BoundaryFace) and it makes distinction between legal and physical objects by introducing external classes for BuildingUnit and UtilityNetwork. It covers only the legal space and the physical counterparts are not directly created in LADM. At the semantic level, legal entities are not enriched by classifying data in relationship to each other (Aien et al. 2013a). Furthermore, LADM through VersionedObject class provides the attributes beginLifespanVersion and endLifespanVersion, allowing the recreation of a dataset at a previous point in time leading to a 4D visualization of the Cadastre (Griffith-Charles and Sutherland 2014).

Another initiative, the 3D cadastral data model (3DCDM) aims to achieve a conceptual framework for 3D Cadastres. It was developed to support integration of legal and physical information that are required for 3D cadastral applications. The 3DCDM utilizes the concepts and terminology of LADM, but the primary focus is on geometrical 3D spatial units and the combination of legal and physical objects together in the data model. In addition 3DCDM only supports spatial units that are surveyed in a define coordinate system (Aien et al. 2013a).

Current deliberations have focused on the incorporation of 3D legal and physical objects. The integration of data on legal spaces and on physical features is critical to occur also in marine environment, since the coincidence of boundaries of legal and physical spaces, is very rare if ever existent. Taking as a reference the underwater tunnels, pipelines, utility networks, submerged archaeological cities or even underwater luxury hotels and facilities.

Integration enables to reuse geometrical data in different domains in general and to define legal spaces based on physical constructions in specific (Rönsdorf et al. 2014).

The modeling of 3D physical objects can be achieved by using 3D city models such as CAD files CityGML (Groger et al. 2012). The adjustment of these models to marine space and the integration with marine cadastral model, need to be further investigated.

4.2 Extension to Marine Environment

In marine environment, despite the fact that a number of jurisdictions have shown interest in the development of MAS and the academic community has dealt with the marine Cadastre concept focusing on varying technical, institutional, legal and stakeholder issues, there is a lack of literature to deal with marine data models in terms of data objects and the relationships among them (Griffith-Charles and Sutherland 2014).

The question arises as to whether the LADM is adequate to model required data and their relationships, relevant to marine spaces. And if yes how can LADM be used as a basis for MAS development.

Ng’ang’a et al. (2004) describe a marine property rights data model, “which provides a standard way to capture the laws that facilitate the allocation, delimitation, registration, valuation and adjudication of marine property rights; the interests that are allocated; the resources that the interests refer to; and their 3D spatial extent” (Fig. 3). The authors argue that there exists a marine parcel object as base for data collection, storage, and retrieval on marine interests. Lemmen (2012) based on the above proposal, states “With some imagination the laws (formal or informal) can be seen as parties; in fact the laws allow people to have interests in “marine objects”. The interests are RRRs and the MarineObject corresponds with the SpatialUnit in LADM Version C. So it is expected that LADM can be used to

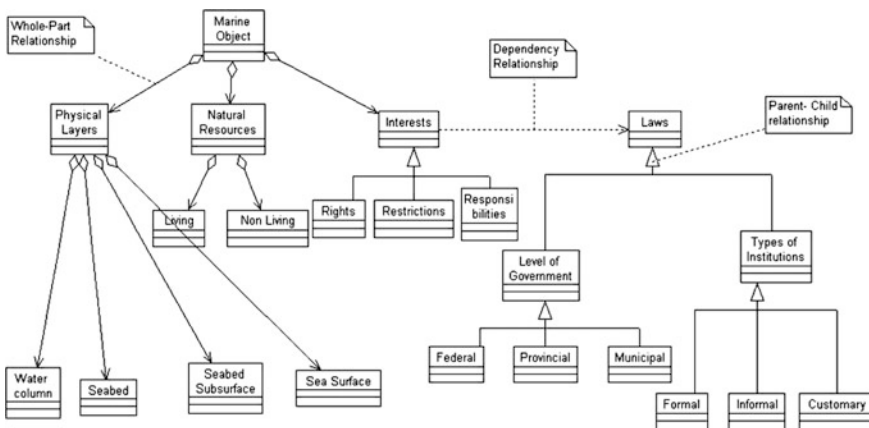


Fig. 3 Marine property data model (Ng’ang’a et al. 2004)

marine space.” The application of the LADM to the marine environment highlights the differences that can arise in the application of the concept to different jurisdictions. The main issues are in the decisions regarding the visualization of the basic administrative units and any derivative spatial units, and in the linking or interfacing of the land based Cadastre with its own existing standards with newly defined standards in the marine space (Griffith-Charles and Sutherland 2014). Duncan and Rahman (2013) advocate the integration of marine blocks with land volumes.

5 Applying the LADM to Marine Environment

5.1 Conceptual Description of Proposed Model

This paper proposes a marine cadastral data model, based on the LADM structure, which allows the registration of:

- the marine interests (RRRs) that are allocated,
- the legislation that recognize these interests,
- the different stakeholders that hold rights,
- the 3D spatial extend of interests,
- the natural resources that the interests refer to.

In the marine space the common relationships’ pattern “people—land” is valid. The basic difference from land is that the property object in this case, is the resources, which can be spatially defined. The proposed data marine model contains various adjustments, compared to LADM. The model’s design is presented in this section. Every class in the model, except of sources, inherits the attributes from VersionedObject, which leads to a 4D visualization of marine Cadastre.

Non-Spatial

The non spatial part of the model consists of the Party and administrative package (Fig. 4). In marine environment the party can be a stakeholder or most likely a group of individuals or companies. The party can also be the state. In the proposed model, the additional attribute of the level of responsible governmental body is introduced. It specifies who has the supervisory control to exercise a RRR according to the relative law and its registration is compulsory.

Rights are divided to private and state rights. The MA_StateRight refers to the sovereign and administrative interests. In the MA_PrivateRight, the rights, which are granted by the State to natural or non-natural entities are recorded. The model proposes the registration of administration sources and laws in different classes. The fact that all different rights find their base in some kind of transacting document is represented by the association between MA_RRR and MA_AdministrativeSource and this transacting document is recorded in the latter class. However in marine

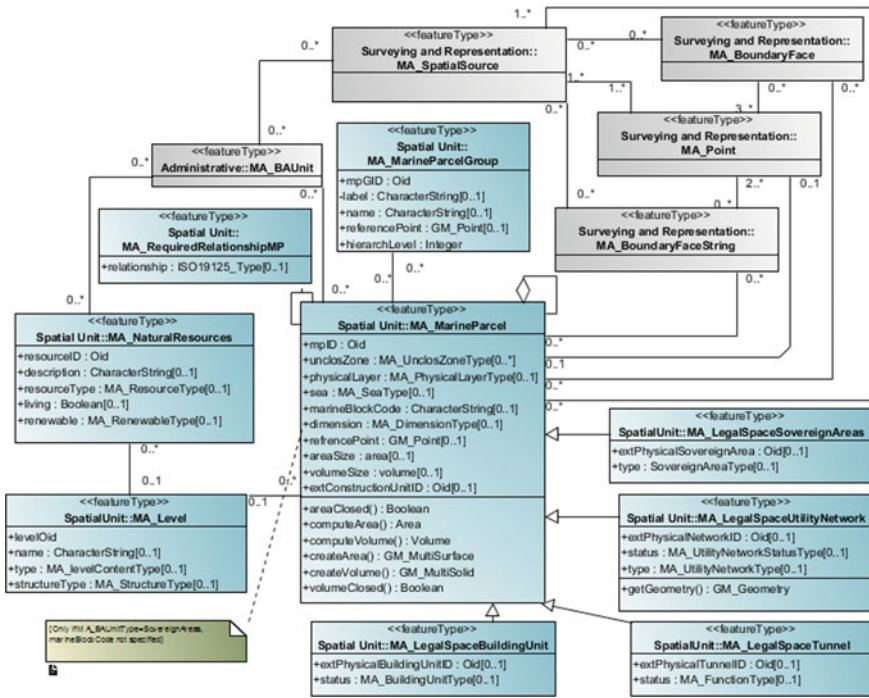


Fig. 5 Spatial package of proposed model

External Classes

The LADM provides stereotype classes for external datasets. The classes of extPrivateEntities, extState, extLevelofResponsibleGovernment are associated to class MA_Party. The class extArchive is for the external registration of sources, according to LADM and is associated with the MA_SpatialSource, MA_AdministrativeSource and MA_Law. Additionally, the classes extPhysicalSpaceBuildingUnit and extPhysicalSpaceUtilityNetwork, extPhysicalSpaceTunnel are associated with the respective specializations of MA_MarineParcel. Finally the class extPhysicalSpaceConstructionUnit is associated with MA_MarineParcel in case the legally defined object contains constructions (their physical registration is outside the scope of this paper).

5.2 From Conceptual to Technical

Having structured the UML data model, the ER model was designed following limitations and capabilities of the PostgreSQL, by making use of PgModeler (PostgreSQL Database Modeler). The automatic physical design of the database

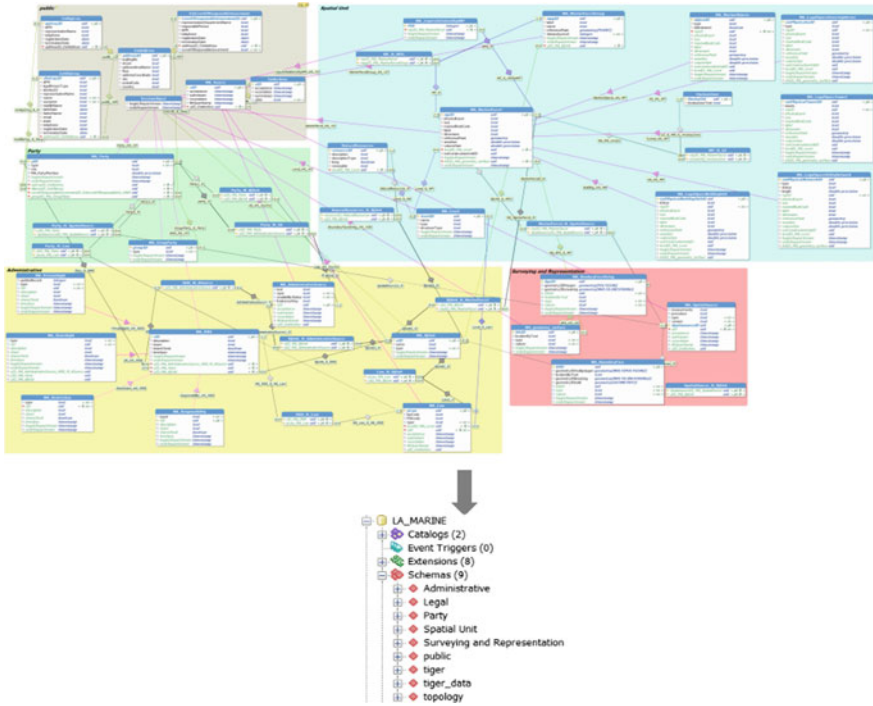


Fig. 6 Entity Relationship model and physical design

was achieved through PgModeler software (Fig. 6). The schema was implemented and populated using PostgreSQL, combined with PostGIS, which supports spatial functions. PostGIS adds support for geographic objects allowing spatial queries to be run in SQL. Table 3 illustrates the assignment between the UML and the PostgreSQL. The following step was the registration and visualization of spatial data in a 3D GIS environment and in the second case in Google Earth. The

Table 3 Assignment between the UML and the PostgreSQL

UML (LA marine)	PostgreSQL/PostGIS
Package	Schema
Codelist	Datatype
Oid	Oid
Fraction	Double
Relationship M-N	Create an intermediate table to reduce cardinality to 1-N
Class	Tables
BoundaryFaceString	Geometry (MULTILINESTRING)
BoundaryFace	Geometry (MULTIPOLYGON)
Constraints (OCL)—Multiplicity	Constraint-functions

comparison and the validation of the methodologies are outside the scope of this paper. Firstly, a study area was chosen and the digital terrain model was generated manually, using open data.

5.3 Case Studies

This section presents examples of practical usage of the proposed data model for marine space. The dataset registered in the database, are shown in Fig. 7, in ArcScene environment. We have to point out that the data are not real, however they are in accordance with the restrictions set by Greek legislation.

Case Study 1: A gas pipeline traversing a space defined for minerals exploration and exploitation

In first analyzed instance presents the registration of a leased unit for hydrocarbons exploration and exploitation, and in the same space the existence of a pipeline. A spatial relationship should be verified between the pipeline and the underlying marine block (Fig. 9). The structure of relationships between the entities at the instance level is presented in Fig. 8.

Case Study 2: Navigation line within aquaculture area

The specific case concerns granted areas for the purpose of aquaculture and the construction of Wind-energy Park. Across this area navigation lines are provisioned. A 3D visualization of the confined places will precede (through querying)

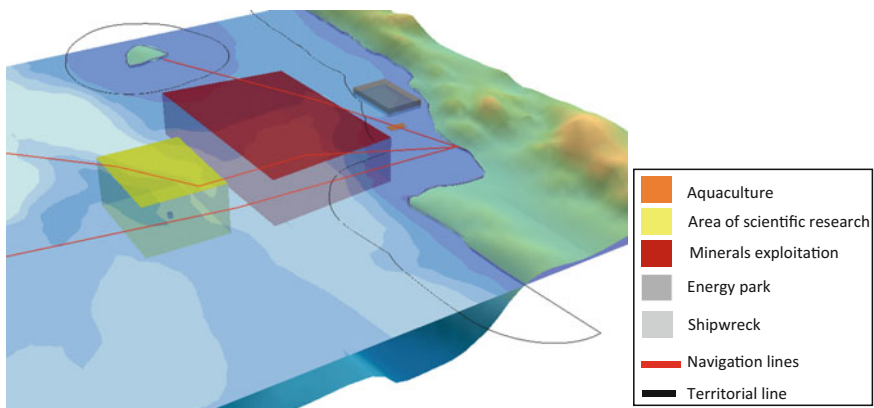


Fig. 7 The 3D representation of all legal objects registered in the database

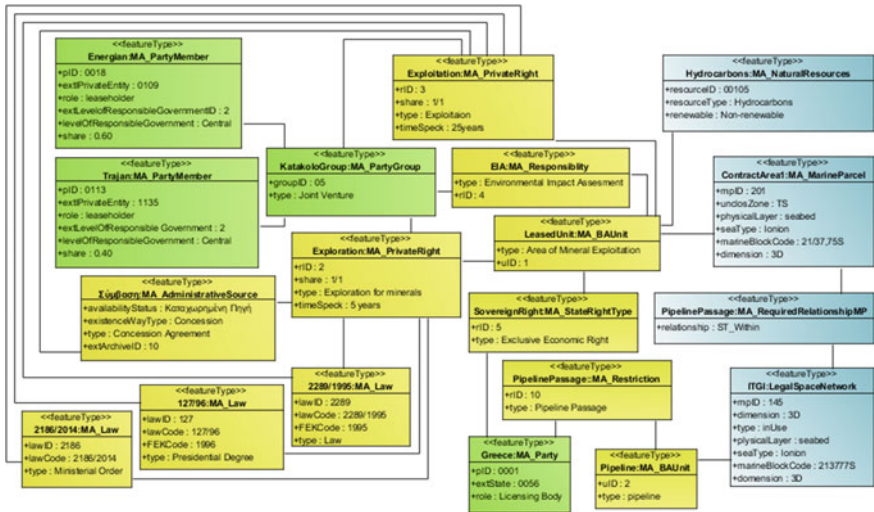


Fig. 8 The structure of relationships between objects at the instance level, according to the proposed model

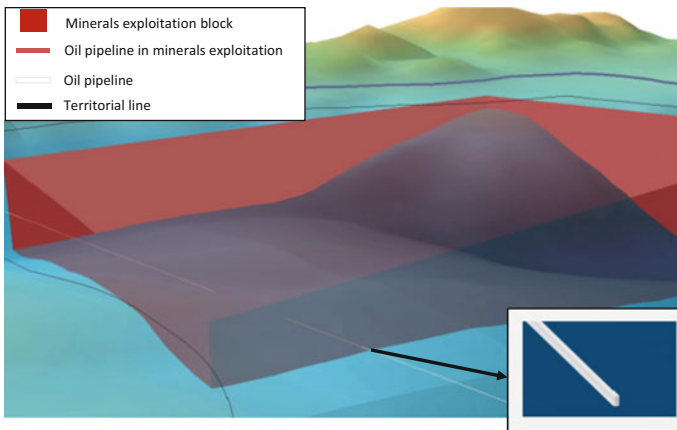


Fig. 9 The proposed 3D registration of marine rights in the system

and subsequently the navigational lines will be amended in cases where they coincide with other uses and rights for a specific period. In this way, the role of MAS as a decision making tool is identified. The visualization of spatial data was implemented in Fig. 10 in a 3D GIS environment and in Fig. 11 in Google Earth.

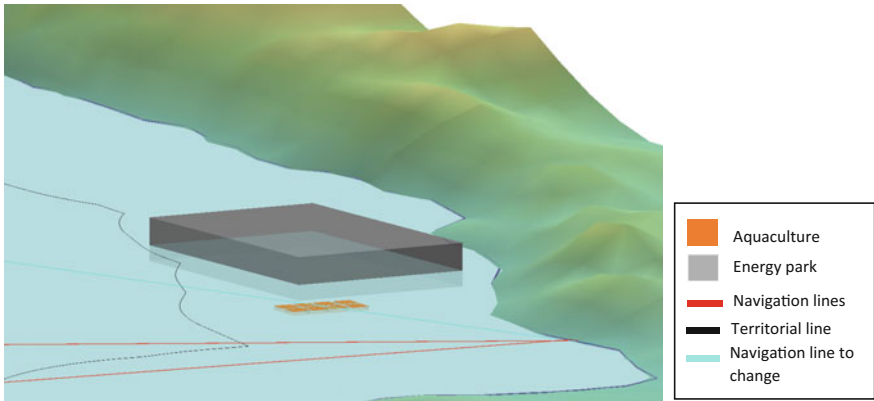


Fig. 10 The navigation line, which intersects with the marine parcel of aquaculture, in ArcScene



Fig. 11 The navigation line, which intersects with the marine parcel of aquaculture, in Google Earth

6 Conclusions

The model described in the present paper serves as a proposal for the management of the legal objects of the marine space, including the possibility of a 3D data record. The establishment of such a managing tool could contribute to a sensible use and optimization of the marine space and its resources. The design of the model in accordance with the international standard LADM offers a common language for all parties involved, interoperability of data and simplification of the procedures. It

is important to note that a broad range of standards is required to improve the accessibility and use of marine data.

The use of a unique code of identification for each marine parcel, as developed by the attributes of the class MA_MarineParcel is considered necessary for the establishment of a single management system. The use of WGS84 UTM allows for a smooth transitioning between the land Cadastre and marine Cadastre and constitutes a standard coordinate system for the Earth, being also the reference used by the GPS.

The integration of physical and legal objects by virtue of international standard models (LADM, CityGML) needs to be subjected to further research. It is proposed to associate the cadastral marine parcel unit with the module of Water Body in CityGML, the LegalSpaceBuildings with the module Buildings, and the MA_LegalSpaceTunnel with the module Tunnel.

References

- Abdullah, A., Omar, A. H., Chan, K. L., Mat Arof, X., Jamil, H., & Teng, C. H. (2014). *The development of marine cadastre conceptual model for Malaysia*. Kuala Lumpur, Malaysia: FIG Congress.
- Aien, A., Kalantari, M., Rajabifard, A., Williamson, I., & Bennett, R. (2013a). Utilising data modelling to understand the structure of 3D cadastral. *Journal of Spatial Science*, 58(2), 215–234.
- Aien, A., Kalantari, M., Rajabifard, A., Williamson, I., & Wallace, j. (2013b). Towards integration of 3D legal and physical objects in cadastral data models. *Land Use Policy*, 35, 140–154.
- Arvanitis, A. (2013). *Development of an integrated geographical information system for the marine space*. Athens: Hellenic Cadastre.
- Barry, M., Elema, I. & van der Molen, P. (2003). Ocean governance in the Netherlands North Sea. In *New Professional Tasks. Marine Cadastres and Coastal Management, FIG Working Week 2003*, Paris, France.
- Binns, A. (2004). *Defining a marine cadastre: Legal and institutional aspects*. M.Sc thesis, The University of Melbourne, Australia.
- Binns, A., Rajabifard, A., Collier, P.A. & Williamson, I.P. (2003). Issues in defining the concept of a marine cadastre for Australia. In *FIG and University of New Brunswick Meeting on Marine Cadastre Issues*. University of New Brunswick, Canada.
- Cockburn, S., Nichols, S. & Monahan, D. (2003). UNCLOS' potential influence on a marine cadastre: depth, breadth, and sovereign rights. In *Proceedings of the Advisory Board on the Law of the Sea to the International Hydrographic Organization (ABLOS) Conference "Addressing Difficult Issues in UNCLOS"*. Presented at the International Hydrographic Bureau, Monaco, October 2003.
- COINAtlantic (2009). COINAtlantic Overview. <http://coinatlantic.ca/index.php/coinatlantic>.
- Duncan, E. E. & Rahman, A. (2013). A multipurpose cadastral framework for developing countries-concepts. *Electronic Journal On Information Systems In Developing Countries*, 58(4), 1–16.
- Fraser, R., Todd, P. J. & Collier, P. A. (2003). Issues in the development of a marine cadastre. In *Addressing Difficult Issues in UNCLOS: 2003 ABLOS Conference*, Monaco.
- Fulmer, J. (2007). The multipurpose marine Cadastre web map. In *Proceedings of the 2007 ESRI Surveying and Engineering GIS Summit*, San Diego.
- Geoscience Australia (2010). <http://www.ga.gov.au/scientific-topics/marine/jurisdiction/amsis>.

- Griffith-Charles, C., & Sutherland, M. D. (2014). Governance in 3D, LADM compliant marine cadastres. In *4th international workshop on 3D cadastres*, Dubai, United Arab Emirates.
- Gröger, G., Kolbe, T. H., Nagel, C. & Häfele, K.-H. (2012). *OpenGIS City Geography Markup Language (CityGML) Encoding Standard—Version: 2. 0*. Open Geospatial Consortium Inc.
- Henssen, J. L. G. (1995). Basic principles of the main cadastral systems in the world. In *Proceedings of the One Day Seminar held during the Annual Meeting of Commission 7, Cadastre and Rural Land Management, of the FIG*, Delft, The Netherlands.
- Lemmen, C. H. J. (2012). A domain model for land administration. Ph.D. thesis, Technical University of Delft.
- Ng'ang'a, S. M., Sutherland, M., Cockburn, S., & Nichols, S. (2004). Toward a 3D marine Cadastre in support of good ocean governance: A review of the technical framework requirements. *Computers Environment and Urban Systems Journal*, 28, 443–470.
- Nichols, S., Ng'ang'a, S. M., Sutherland, M. D. & Cockburn, S. (2006). *Marine cadastre concept. chapter 10 in canada's offshore: jurisdiction, rights and management* (3rd ed.). Trafford Publishing, Victoria, Canada.
- NOAA & BOEM. (2010). <http://marinecadastre.gov/>.
- Rahman, A., van Oosterom, P., Hua, T. H., Sharkawi, K. H., Duncan, E. E. (2012). 3D modelling for multipurpose cadastre. In *3rd International Workshop On 3d Cadastres: Developments And Practices*, Shenzhen, China.
- Rajabifard, A., Williamson, I., & Binns, A. (2006). Marine administration research activities within Asia and pacific region—towards a seamless land—sea interface. *FIG, Administering Marine Spaces: International Issues*, 36, 21–36.
- Rönsdorf, C., Wilson, D., & Stoter, S. (2014). Integration of land administration domain model with CityGML for 3D cadastre. In *4th International Workshop on 3D Cadastres*, Dubai, United Arab Emirates.
- Shojaei, D., Kalantari, M., Bishop, I. D., Rajabifard, A., & Aien, A. (2013). Visualization requirements for 3D cadastral systems. *Computers, Environment and Urban Systems Journal*, 41, 39–54.
- Sutherland, M. D. (2005). *Marine Boundaries and Good Governance of Marine Spaces*, Ph.D. thesis, Department of Geodesy and Geomatics Engineering, University of New Brunswick, Fredericton, Canada.
- Sutherland, M. D. & Nichols, S. (2002). Marine boundary delimitation for ocean governance. In *Proceedings of FIG Working Week 2002*, Washington, DC, USA.
- Ward, R., Alexander, L. & Greenslade, B. (2009). IHO S-100—the new IHO hydrographic geospatial standard for marine data and information. In *International Hydrographic Review* (vol. 5, No. 1). Monaco.
- Williamson, I. P., Enemark, S., Wallace, J., & Rajabifard, A. (2010). *Land Administration for Sustainable Development*. Redlands, California: Published by ESRI Press Academic. 2010.

Assessing the Suitability of Using Google Glass in Designing 3D Geographic Information for Navigation

Kelvin Wong and Claire Ellul

Abstract No longer are we bound by traditional 2D physical representations; there is a steady shift towards three-dimensional (3D) data. Existing research recognises landmarks to be important navigationally but specific requirements for geometric and semantic attributes in 3D have not been identified. This study assesses the suitability of using Google Glass in real-world experiments investigating the saliency of environmental objects which facilitate pedestrian navigation. From the experiment carried out with fourteen participants, initial results show geometric and semantic detail for navigation are most pertinent between 1.65–7.5 m for buildings. Visual characteristics such as colour, shape and texture are more relevant than function and use.

Keywords Navigation • Google glass • Landmarks • User-centred design

1 Introduction

Navigation is an implicit requirement of our daily lives. In recent years, the way we traverse the world has been strongly impacted by the emergence of mobile navigational technologies, altering our perception of space and specifically, our navigational strategies. No longer are routes meticulously planned in advance using paper maps and local knowledge but are calculated on-the-fly with considerations

K. Wong (✉)

Department of Computer Science, University College London,
London WC1E 6BT, UK
e-mail: kelvin.wong.11@ucl.ac.uk

C. Ellul

Department of Civil, Environmental and Geomatic Engineering,
University College London, London WC1E 6BT, UK
e-mail: c.ellul@ucl.ac.uk

© Springer International Publishing AG 2017

A. Abdul-Rahman (ed.), *Advances in 3D Geoinformation*,
Lecture Notes in Geoinformation and Cartography,
DOI 10.1007/978-3-319-25691-7_23

of current traffic conditions, road closures and other live updates. For many, asking for directions has become an archaic practice superseded by a near instantaneous response from a smartphone application. The rise in the aforementioned technologies has led to the decline in the use of traditional representations of the world around us, specifically the two-dimensional paper map such as the London A–Z. Without the boundaries of physical representations, there is a steady shift towards utilising three-dimensional data whereby the additional dimension allows for a more representative and less abstract version of the real world.

While 2D geographic information systems (GIS) have facilitated navigation for over a decade in various forms, from automotive navigation systems to Google Maps, the use of 3D GIS is in its infancy. Significant developments in 3D data acquisition techniques, visualisation, image-processing and computing power have facilitated the creation of 3D models around the globe and the ability to handle and work with three-dimensional data (Jazayeri et al. 2014). Although the specifications for 2D data are thoroughly and explicitly defined, the requirements for 3D data is ambiguous and lacks standardization. Different 3D GIS applications demand different levels of detail geometrically and semantically, at an intrinsic and extrinsic level. There have been attempts to define the initial requirements for 3D navigation with 3D GIS, but much remains to be done before full implementation arises (May et al. 2003; Musliman et al. 2006).

This study aims to address the above issues by exploring landmark saliency at a finer, intrinsic level through a real-world navigation experiment. It will assess the suitability of Google Glass in the design of 3D geographic information. The paper will outline the design of the user-centric experiment and the subsequent results from its first iteration. It will conclude with possible directions for future work.

2 Background

Navigation strategies vary from person to person depending on factors such as age, sex, cultural background and familiarity with environment (Saucier et al. 2002; Xia et al. 2009; Moffat et al. 2006; Iaria et al. 2009; Hölscher et al. 2006). These factors affect one's experience of the world and therefore their own internal representation of the space they are navigating. While disruptive technologies have altered navigation strategies, the basic process of pedestrian navigation has remained unchanged and can be divided into four categories: (1) orientation; (2) route decision; (3) route monitoring and; (4) destination recognition (Downs and Stea 1974). At a higher level, pedestrian navigation is based on vision and consists primarily of physical cues but it also relies on inexact, error-prone and distorted cognitive maps (Golledge 1995).

When considering pedestrian navigation, objects within an environment have differing salience—as a largely visual exercise, it is known that landmarks which stand out relative to its neighbours provide the most dominant navigational cue, followed

by distance information and street names (May et al. 2003). Landmarks are difficult to define as they differ depending on individual spatial cognition abilities—age, culture and sex affect the ability to perceive, memorise, utilise and convey properties of spatial environments (Richter and Winter 2014). Further, human navigation strategies vary with exploration environments, conditions and navigation tools. Easily recognisable and distinct landmarks within environments act as points of reference and also as a beacon in relation to the destination. It informs one of its orientation as well as providing a reference frame for encoding spatial information (Chan et al. 2012). Although existing studies have found landmarks to be the most salient feature in an urban landscape for navigation (May et al. 2003), the specific salient attributes of these landmarks, both geometric and semantic, have not yet been identified.

A complex and ever-evolving phenomena, the study of navigation has a long history in a diverse number of fields ranging from psychology to geography to computer science. Navigation experiments, however, remain predominantly in virtual and simulated environments (Goldin and Thorndyke 1982; Satalich 1995). There is a lack of real world experiments examining human navigation behaviour, perhaps due to the lack of control and higher costs incurred. Direct observation is rare as it is difficult to interpret what is recorded and what is to be analysed (Peponis et al. 1990). Though experiments and studies conducted within a virtual environment have been found to be nearly as good as the real experience in navigation, there is a need for navigation research in real world environments with respect to three-dimensional navigational information and spatial interactions (Witmer et al. 1996). The emergence of wearable technologies may offer assistance in collecting data from real-world experiments which are easier to interpret and understand.

3 Methodology

In order to capture the true dynamism of real navigation, fourteen subjects were asked to navigate in the real-world rather than in a computer-simulated environment. A novel approach using a pair of Google Glass (Fig. 1) was implemented to record the gaze and movement of the participants. A minimalist and light optical head-mounted display, Google Glass was selected as it allowed for unobtrusive and natural tracking. Retailing at \$1,500 USD, it provided a comparatively cheap and accessible option for gaze tracking but does not include true eye tracking (Google 2014). Systems with true eye tracking, however, are large, cumbersome and can cost between \$11,400–\$14,900 USD (Applied Science Laboratories 2015; Tobii 2015).



Fig. 1 Google Glass device

3.1 Apparatus

For this experiment, each participant were fitted with a pair of Google Glass, an optical head-mounted display. First released in the United States in 2013, the prototype device was available in the United Kingdom for beta testing between 23rd June 2014 and 19th January 2015 as part of the Explorer Program.

In addition to the Google Glass, each participant was provided with a notepad, pen and a basic map of the surrounding area with the route. The base map used was Ordnance Survey Street View in black and white (Fig. 2).

3.2 Study Area

The experiment was carried out around University College London (UCL) and the Bloomsbury area (OSGB 1936 min 529469.6, 182025.1, max 182616.6, 529813.3). The area covers 0.2 km² and is predominantly residential and office space.

The route selected for the experiment (1.52 km) was designed to test the navigation strategies of the participants and passes through an area without landmarks bearing major branding or obvious iconic buildings. The complex and diverse route

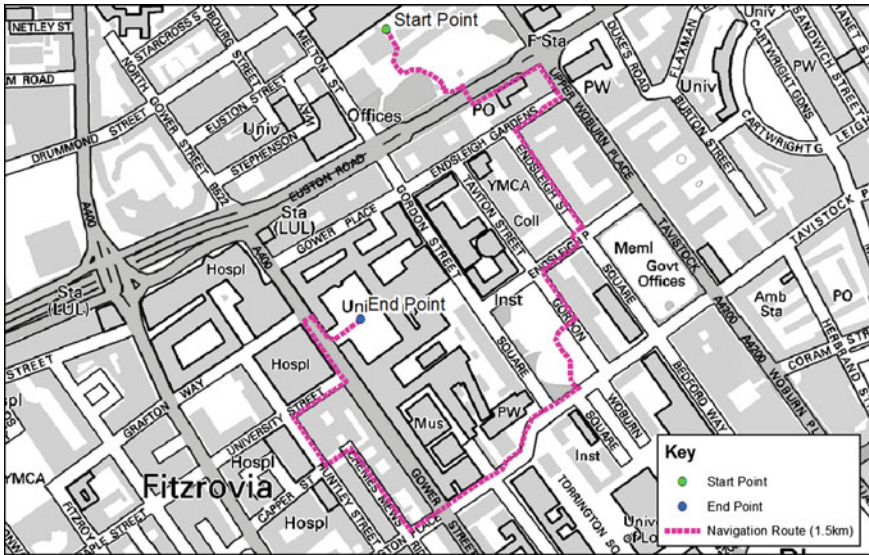


Fig. 2 Study area with navigation route

imposes a significant navigation task demand on participants and promoted participants to describe landmarks without obvious branding or logos. From a safety perspective, the route was also designed to avoid the crossing of multiple major roads.

3.3 Task

Participants were instructed to follow a specified route on the map provided. The selected route was designed to test the navigation strategies of the participants and passed through an area without major branding or obvious iconic buildings. By choosing a complex and diverse route, a significant navigation task demand was imposed on the participants and promoted the description of geometric and semantic features of navigational landmarks. Participants were told that the task was not to navigate the route most successfully or the fastest but were instead asked to produce a set of written instructions of the route for an unfamiliar traveller who had not been to the area before. This was to encourage active thinking and to engage with the process of navigation. The participants were then fitted with Google Glass, presented with the route map and were allowed to navigate freely (Fig. 3). As the Glass device does not have true eye-tracking, the participants were asked to look purposefully with complete head movements in order to mitigate this issue.

Fig. 3 Participant wearing Google Glass and navigating



3.4 Google Glass Application

A Google Glass application was specifically designed for this experiment. Developed in Android Glass Development Kit Preview 4.4.2, the application runs on Google Glass XE 19.1 and was paired with a smartphone (iPhone 5S) in order to obtain locational information via GPS. The user does not directly interact with Google Glass, but rather it passively records a first-person video as well as tracking their gaze vector (orientation and pitch), location (latitude and longitude) and elapsed time. The video was recorded concurrently while the gaze vectors and location were logged every 0.125 s. Figure 4 shows the various screens of the application during the experiment.

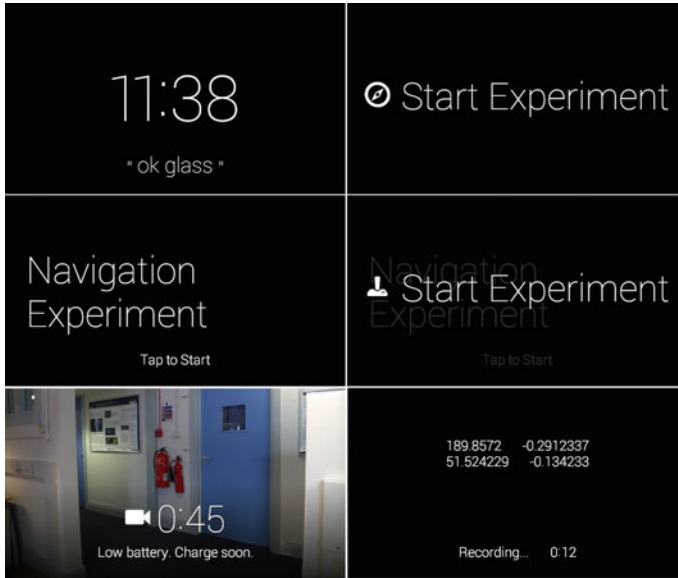


Fig. 4 Screen shots of Google Glass application flow

4 Results and Discussion

The experiment collected two sets of data: (1) qualitative written instructions and; (2) quantitative gaze vector tracking. An example of the written instructions results can be found below (Fig. 5). Previous navigation studies using text analysis were con-

“Walk away from the busy entrance of the station, towards the main road Euston Road. When you reach it, cross over the pedestrian crossing, and turn left. Carry on until you come to a corner, where youll see the restaurant Prezzo immediately on your right and the Fire Station on the left (on the other side of the road). Turn right, and walk down the road until you come to the first right where a set of Boris bikes is parked. Take this road, and walk down until you come to a place called Bentham House on your left. Take this left turning and continue down until you reach a park ahead, on the left. At this point, turn right. Youll reach another park, and this time turn off left and walk alongside it until you reach a gate. Turn into the park, and follow the path as it forks around to the left. You will reach a main road. Turn left, carry on past a large cathedral, three red telephone boxes, and two main roads. When you come to a small lane on your right, turn down it and carry on to the very end, where you will find a small gap on your left. Go through until you reach another road, and turn right and right again. There should be a large red brick hospital on your left and a Zoology department on your right. At the end of the road, turn left and walk down until you can see the Euston Square Tube station sign. Take the first Pelican crossing and turn right, walking until you come to two small buildings on your left. Go through them and walk straight up to the huge white building.”

Fig. 5 Extract of written navigation instructions from a participant

Table 1 Summary table of words used more than 10 times, with common English words and directional terms omitted

Word	Count
Road	98
Street	67
Endsleigh	41
Building	36
Place	29
Square	27
Garden	27
Park	25
Euston	22
Gower	21
Mew	18
UCL	18
University	16
Gordon	15
Light	14
Sign	13
Entrance	12
Main	12
Path	11
Hit	11
Centre	10
Upper	10
Small	10
Corner	10

strained predominantly in virtual environments or pre-recorded video (Partala et al. 2010; Miller and carlson 2011). In this study, written instructions were produced after the participant was free to navigate a specified route. The assumption here was that the written instructions would reflect the most pertinent and salient environmental cues within a participants cognitive map. The instructions from all the participants were collated and common English words and directional terms were removed (Table 1). This allowed for the identification of landmarks and reference points used in the navigation instructions. The results showed that road names and landmarks feature heavily. This is consistent with terms used in navigation instructions in existing literature (May et al. 2003). It is key to note here the use of OS StreetView as a base for the route map may have led to a distorted high usage of road names and road signs. It can be surmised that where buildings are visually homogenous and there is a lack of obvious landmarks, street names become more important especially for egocentric navigation strategies using personal directional instructions.

Further, the preceding and following description of the world building were extracted as a proxy to assess the intrinsic salient properties of urban landmarks (Table 2). Each term was categorised into two groups: visual and semantic (Notheg-

Table 2 Summary table of words preceding and following “building”

Description	Count	Description type
Red	7	Visual
Glass	5	Visual
Columned	3	Visual
Large	3	Visual
Brick	3	Visual
Big	3	Visual
Classical style	2	Visual
Grey	2	Visual
Stone	2	Visual
Huge	2	Visual
UCL Hospital	2	Semantic
Pointy	2	Visual
Black	1	Visual
Quaker’s	1	Semantic
Impressive	1	Visual
Church-like	1	Visual
Dark	1	Visual
Single-storied	1	Visual
UCL	1	Semantic
Cube	1	Visual
Cream	1	Visual
Rubin	1	Visual
University of London	1	Semantic
Unusual	1	Visual
Slatted	1	Visual
UCL Women’s	1	Semantic
Green	1	Visual
Fancy	1	Visual
Old	1	Visual
Giant	1	Visual
Grand	1	Visual
UCL Engineering	1	Semantic
Small	1	Visual
White	1	Visual
Grant Museum of Zoology	1	Semantic

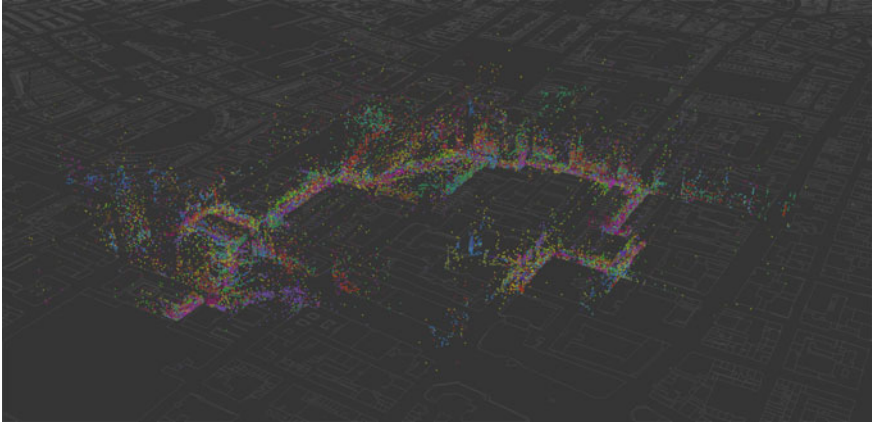


Fig. 6 Gaze positions of all participants mapped in 3D

ger et al. 2004). Visual descriptions include façade shape, colour, size and visibility while semantic properties include function, use or meaning. Although Nothegger et al. (2004) identifies structural salience as a third category of landmark attributes, it is not investigated in this study as it is considered extrinsic. From Table 2, it can be seen that visual, rather than semantic, characteristics and cues such as colour, shape, texture and architectural style are most relevant when navigating at street level. In this study, a gaze vector is defined as a geodesic line which most accurately represents the shortest distance between the participant location and the gaze position. The participant is assumed to be looking at the same orientation as the Google Glass and at object(s) in the centre of their view. The data collected from the participants were corrected for GPS error using individually digitised routes from the video. The median adjustment of 5.773 m is consistent with the average median error of 8 m for iPhones integrated positioning technologies (Zandbergen 2009). Each data point was also corrected to have an average viewing height of 1.65 m and the gaze vectors were then mapped using the orientation, pitch and locational data collected from the Google Glass. An assumption was made that the gaze position is calculated to be the first building or built structure from OS MasterMap that the gaze vector intersects within 200 m (Fig. 6). This was necessary as the device lacked true eye-tracking. Acknowledging that general conclusions cannot be drawn from a small sample of fourteen participants, initial exploratory statistics showed that gaze height was predominantly between 1.65–7.5 m, with navigators focusing around eye-level to the first two stories of a building. Roof characteristics, though useful for completeness and shape, are largely irrelevant in pedestrian navigation as they could not be seen. The ideal subsequent analysis for this study would have been to intersect gaze position points with a true 3D model, thereby identifying the exact features viewed while navigating every 0.125 s e.g. the window on the ground floor or a street sign at an intersection. This was not possible, however, as a 3D building city model for the area with a sufficient geometric and semantic detail was not available. For this analysis

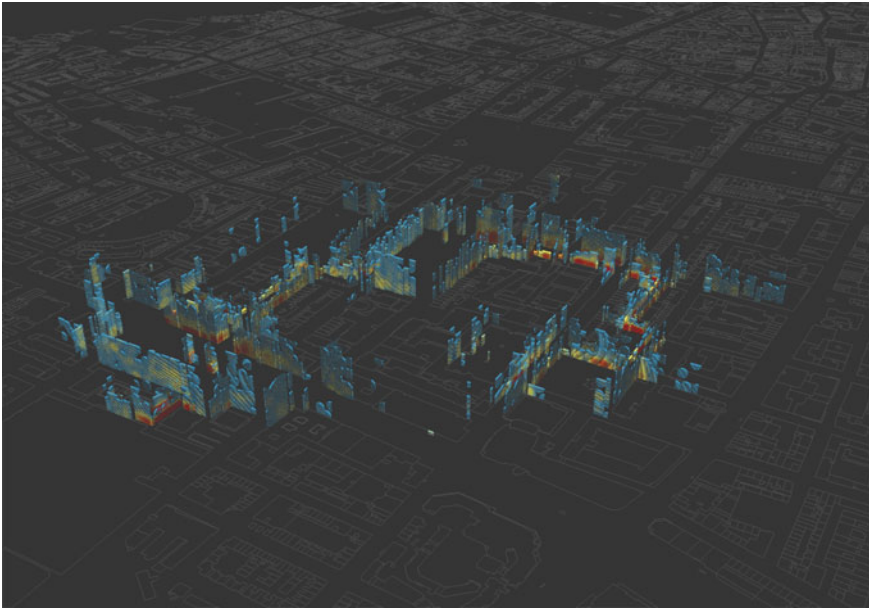


Fig. 7 Gaze positions point density function of all participants mapped in 3D

to be feasible, a building model at CityGML's LoD3 or higher (Kolbe et al. 2005) is required. The minimum requirement would be the inclusion of detailed wall surfaces including openings such as doors and windows. In this instance, details of roof structures and the roof itself, however, are not required as this application focuses on pedestrian navigation at ground height. This limitation of the analysis stems not from the proposed methodology or data but rather the inherent deficit in suitable 3D GIS tools and datasets. An alternative exploratory analysis was carried out using 3D heat maps. Every gaze position was passed through a point density function to create a heat map for each building façade using the Gram-Schmidt orthogonalization process [see Wong (2012) for more detail] (Fig. 7). A similar clustering approach was adopted in a web-based experiment identifying the features in a number of urban scenes which could be used in forming navigational instructions (Bartie et al. 2014). The 3D heat maps were overlaid against buildings in Google Earth as well as over textured buildings (Figs. 8 and 9). The results showed that ground floor front-facing building façades were most examined followed by road signs. Buildings at decision points such as junctions were also used as landmarks.

The study and its subsequent analysis clearly demonstrates a severe shortcoming within the current state of GIS—here 2D is insufficient in analysing the data collected yet, true 3D data of adequate quality both geometrically and semantically is unavailable. Further, there is a lack of GI systems able to perform geospatial queries in 3D which are readily available in 2D such as buffering, intersection and topological relations.

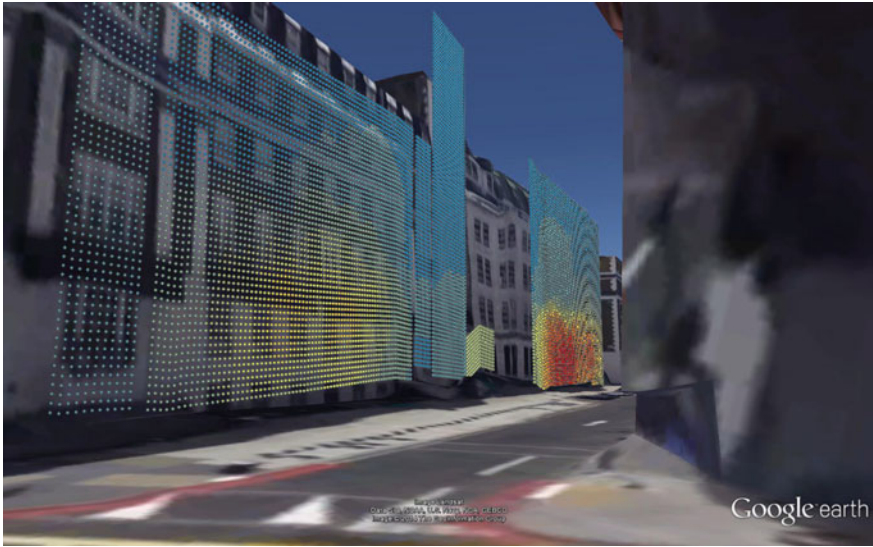


Fig. 8 Gaze position heat map for Endsleigh Gardens in Google Earth

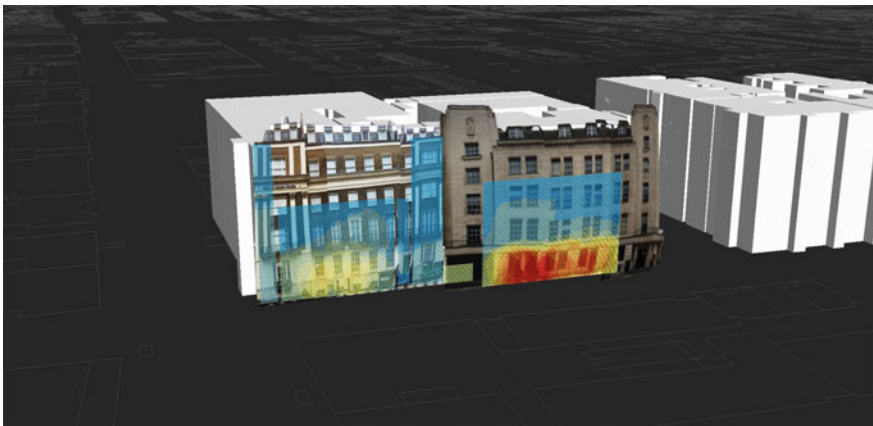


Fig. 9 Gaze position heat map for Endsleigh Gardens with building textures

Since the study was carried out, the open beta Google Glass Explorer program ended on 19th January 2015 (Google 2015). The methodology implemented in this study, however, is not device dependent and the techniques of capturing saliency of landmarks attributes using a user-centric approach is universally applicable. Alternative wearable technologies such as SMI Eye Tracking Glasses 2 Wireless (SensoMotoric Instruments 2015), Tobii Glasses 2 Eye tracker (Tobii 2015) and Mobile Eye-XG Eye Tracking Glasses (Applied Science Laboratories 2015) offer similar, if not enhanced mobile gaze tracking functionality albeit at the sacrifice of less natural

tracking through more obtrusive designs and at a higher cost. The implementation of true eye tracking does have its advantages, however, as it could provide finer and more precise results than offered by Google Glass. The techniques developed in this preliminary study offer an insight into the opportunities user centric design contributes to the wider understanding of the requirements of 3D GIS and identifying end-user needs.

5 Conclusion

This study has provided a novel methodology in gathering requirements for a 3D navigation dataset. It outlines the first steps in an iterative study whereby the above recommendations would be implemented in developing a true 3D GIS dataset for navigation. Further work with participants from different age ranges and cultural backgrounds would be desirable to capture a representative sample of navigation strategies. In addition, true eye tracking and the availability of a 3D city model with full geometric and semantic attributes would enable the realisation of the full potential of the experiment. While the study demonstrates the possible value of 3D, it also shows the inherent deficiencies in the wider 3D field. In all, 3D is a multifaceted and ill-defined problem and it is unclear whether the benefits of the extra dimension outweighs its complexity. This study shows 3D is beneficial in the application of pedestrian navigation but argues that existing technologies are incapable of delivering the envisaged true 3D navigation system. Where 2D maps works well on existing smartphone technologies, 3D navigation may require other enabling technologies such as a form of heads-up display or an ambient device. Regardless the direction 3D takes, what is key is an user-centric design approach will ensure that the resulting outcome is effective, efficient, and enjoyable to use.

Acknowledgments This project was funded and supported by the Engineering and Physical Sciences Research Council (EPSRC) and Ordnance Survey.

References

- Applied Science Laboratories. (2015). Mobile eye-XG eye tracking glasses. <http://www.asleyetracking.com/Site/Products/MobileEyeXGGlasses/tabid/70/Default.aspx>
- Bartie, P., Mackaness, W., Petrenz, P., & Dickinson, A. (2014). Clustering landmark image annotations based on tag location and content. In *Proceedings of RefNet Workshop on Psychological and Computational Models of Reference Comprehension and Production*, 31st August 2014
- Chan, E., Baumann, O., Bellgrove, M. A., & Mattingley, J. B. (2012). From objects to landmarks: the function of visual location information in spatial navigation. *Frontiers in Psychology*, 3.
- Downs, R. M., & Stea, D. (1974). *Image and environment: Cognitive mapping and spatial behavior*. Transaction Publishers.
- Goldin, S. E., & Thorndyke, P. W. (1982). Simulating navigation for spatial knowledge acquisition. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 24, 457–471.

- Golledge, R. G. (1995). *Path selection and route preference in human navigation: A progress report*. Springer.
- Google. (2014). Google Glass UK. "Google Glass Explorer" Program. <https://www.google.co.uk/intl/en/glass/start/>. Retrieved August 17, 2014.
- Google. (2015). Google Glass UK. <https://www.google.co.uk/intl/en/glass/start/>. Retrieved August 18, 2015.
- Hölscher, C., Meilinger, T., Vrachliotis, G., Brsamle, M., & Knauf, M. (2006). Up the down staircase: Wayfinding strategies in multi-level buildings. *Journal of Environmental Psychology*, 26, 284–299.
- Iaria, G., Palermo, L., Committeri, G., & Barton, J. J. S. (2009). Age differences in the formation and use of cognitive maps. *Behavioural Brain Research*, 196, 187–191.
- Jazayeri, I., Rajabifard, A., & Kalantari, M. (2014). A geometric and semantic evaluation of 3D data sourcing methods for land and property information. *Land Use Policy*, 36, 219–230.
- Kolbe, T. H., Gröger, G., Plümer, L. (2005). CityGML—interoperable access to 3D city models. In *Proceedings of the International Symposium on Geo-information for Disaster Management*, 21st–23rd March 2005, Delft, Netherlands.
- May, A. J., Ross, T., Bayer, S. H., & Tarkiainen, M. J. (2003). Pedestrian navigation aids: Information requirements and design implications. *Personal Ubiquitous Computing*, 7, 331–338.
- Miller, J., & Carlson, L. (2011). Selecting landmarks in novel environments. *Psychonomic Bulletin and Review*, 18, 184–191.
- Moffat, S. D., Elkins, W., & Resnick, S. M. (2006). Age differences in the neural systems supporting human allocentric spatial navigation. *Neurobiology of aging*, 27, 965–972.
- Musliman, I. A., Rahman, A. A., & Coors, V. (2006). 3D navigation for 3D-GIS—initial requirements. In *Proceedings from 3DGeoInfo 2006*.
- Nothegger, C., Winter, S., & Raubal, M. (2004). Selection of salient features for route directions. *Spatial Cognition and Computation*, 4, 113–136.
- Partala, T., Nurminen, A., Vainio, T., Laaksonen, J., Laine, M., & Vninen, J. (2010). Salience of visual cues in 3D city maps. In *Proceedings of the 24th BCS Interaction Specialist Group Conference* (pp. 428–432).
- Peponis, J., Zimring, C., & Choi, Y. K. (1990). Finding the building in wayfinding. *Environment and Behavior*, 22, 555–590.
- Richter, K.-F., & Winter, S. (2014). *Landmarks: GIScience for intelligent services*. Springer International Publishing.
- Satalich, G. A. (1995). Navigation and wayfinding in virtual reality: Finding the proper tools and cues to enhance navigational awareness. University of Washington.
- Saucier, D. M., Green, S. M., Leason, J., Macfadden, A., Bell, S., & Elias, L. J. (2002). Are sex differences in navigation caused by sexually dimorphic strategies or by differences in the ability to use the strategies? *Behavioural Neuroscience*, 116, 403.
- Sensomotoric Instruments. (2015). SMI Eye Tracking Glasses 2 Wireless. <http://eyetracking-glasses.com/>
- Tobii. (2015). Mobile eye tracking—Tobii Glasses 2. <http://www.tobii.com/en/eye-tracking-research/global/products/>
- Witmer, B. G., Bailey, J. H., Knerr, B. W., & Parsons, K. C. (1996). Virtual spaces and real world places: transfer of route knowledge. *International Journal of Human-Computer Studies*, 45, 413–428.
- Wong, K. (2012). 3D geographic information and solar panel positioning. MSc Dissertation (unpublished), University College London.
- Xia, J., Packer, D., & Dong, C. (2009). Individual differences and tourist wayfinding behaviours. In *18th World IMACS/MODSIM Congress* (1272–1278).
- Zandbergen, P. A. (2009). Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning. *Transactions in GIS*, 13, 5–25.

Review and Assessment of Current Cadastral Data Models for 3D Cadastral Applications

Ali Aien, Abbas Rajabifard, Mohsen Kalantari and Ian Williamson

Abstract Three-dimensional (3D) cadastres are often described as the 3D digital representation of real property rights, restrictions, and responsibilities (legal objects). They can also contain physical counterparts (physical objects) of legal objects such as buildings and utility networks, on, above or under the surface. Implementation of 3D cadastres requires many elements such as existing 3D property registration laws, appropriate 3D data acquisition methods, 3D spatial database management systems, and functional 3D visualisation platforms. In addition, an appropriate 3D cadastral data model can also play a key role to ensure successful development of the 3D cadastre. Many jurisdictions have defined their own cadastral data models. However, none of them can fully support the requirements of 3D cadastres. This paper aims to explore the theories and concepts of the most common existing cadastral data models and investigate how they manage 3D legal and physical data. The result of this research can be used by cadastral data modellers to improve existing or develop new cadastral data models to support the requirements of 3D cadastres.

Keywords 3D cadastre • 3D cadastral data modelling • RRR • Legal Property Object • LADM

1 Introduction

Management of stratified land rights, restrictions and responsibilities (RRRs) is one of the most important challenges in the current land administration systems which are equipped with cadastres that are only able to maintain 2D spatial information.

Current cadastral systems are two-dimensional (2D) and land parcel based, that is, geometric and descriptive information is based on 2D land parcels, even if the properties have three dimensions. They cannot effectively represent the reality.

A. Aien (✉) · A. Rajabifard · M. Kalantari · I. Williamson
Department of Infrastructure Engineering, University of Melbourne, Melbourne, Australia
e-mail: aiena@unimelb.edu.au; ali_aien@yahoo.com

© Springer International Publishing AG 2017
A. Abdul-Rahman (ed.), *Advances in 3D Geoinformation*,
Lecture Notes in Geoinformation and Cartography,
DOI 10.1007/978-3-319-25691-7_24

Current 2D cadastral systems are not able to manage and represent land ownership RRRs in a 3D context (Kalantari et al. 2008).

3D cadastres would overcome these problems. 3D cadastre should be capable of storing, manipulating, querying, analysis, updating, and supporting the visualisation of stratified land RRRs. There is not yet such a system in the world (Oosterom 2010; Godard 2004; Navratil 2009; Stoter and Oosterom 2006; Aien et al. 2013a).

There are several reasons why 3D cadastres have not been successfully implemented: legal, institutional, and technical aspects all play a role (Stoter 2004). Specific reasons include:

- Lack of legal support and mandate to register 3D properties (legal and institutional issues),
- Lack of specified guidelines and standards for surveyors to capture required 3D data, and a lack of data formats to integrate and exchange 3D data (legal, institutional and technical issues),
- Lack of available technologies for storing, manipulating, and visualising 3D objects (technical issues) and,
- Most importantly, lack of a comprehensive data model is one of the main obstacles to advancing implementation.

3D cadastral data model supports 3D cadastre's users to understand the structure or behaviour of the system and has a template that guides them to construct and implement the 3D cadaster (Aien 2013).

Many jurisdictions, organisations and software developers have built their own cadastral data models. The variation between these data models is the result of different attitudes towards cadastres (Aien et al. 2011). Six cadastral data models have been reviewed; however, only three (ArcGIS Parcel Data Model, ePlan, and LADM) have been discussed in detail in this paper (see Table 3 for the complete review). The aim of this review is to explore the advantages and deficiencies of these models in terms of supporting the requirements of 3D cadastre. Six cadastral data models are:

- The core cadastral data model (Henssen 1995)
- FGDC Cadastral Data Content Standard for the National Spatial Data Infrastructure (FGDC 1996)
- ArcGIS Parcel Data Model (Meyer 2004)
- The Legal Property Object Model (Kalantari et al. 2008)
- ePlan (ePlan 2010)
- ISO 19152, Land Administration Domain Model (LADM) (ISO19152 2012).

These data models are the most popular cadastral and land administration data models that were found in the literature and background research. They were assessed and compared based on selected criteria. The aim was to analyse how they manage stratified land ownership RRRs; meet 3D cadastre's requirements; and what data modelling techniques they use to support 3D data.

The remainder of the paper presents the assessment of data models in detail. Section 2 describes the selected criteria for assessment. Sections 3, 4, and 5 describe the above-mentioned data models respectively. Research conclusions and the summary of the models are presented in Sect. 6.

2 Assessment Criteria of Cadastral Data Models

The criteria were selected in a way to be able to assess the data models from different aspects and provide information on how they manage stratified RRRs. Table 1 summaries the criteria used for comparing the above-mentioned cadastral data models.

This assessment enables an understanding to be gained of important cadastral data models, their advantages and disadvantages for improving the existing or developing a new data model. Each data model is reviewed in the following sections.

Table 1 Criteria for comparing current cadastral data models

Criteria	Description
(a) Core objects	What are the core objects of the data model? (person, right, spatial unit, parcel)
(b) Basic spatial unit	What are the basic spatial units of the data model? (2D parcel, 3D parcel)
(c) Other forms of spatial units	Does the data model have other forms of spatial units? (text-based, point-based)
(d) Reference documents	What are the data sources? (survey plans, architectural plans, titles, deeds)
(e) Applications	For what applications can the data model be used? (registration, taxation, valuation, planning, etc.)
(f) Inclusion of other types of interests	Whether or not other types of interests are considered in the data model? (utility network right, biota right, mineral right)
(g) Temporal aspects	Whether or not temporal aspects of interests are considered? (time the right is created or terminated)
(h) Management and representation of stratified RRRs	How does the data model render stratified RRRs? (projection on the ground level, 3D primitives)
(i) Semantic-level	At what level does the data model support semantics? (class-level, attribute-level, geometry level)
(j) Physical objects	How does the data model support the physical counterparts of legal objects? (internally [in the model], externally [external databases])

3 ArcGIS Parcel Data Model

The purpose of the ArcGIS Parcel Data Model is to describe parcel information to support local government and private sector decision making (Fig. 1). Parcel managers and GIS professionals can use the model as a starting point for defining parcel information in the GIS environment and plan for migration strategies from current data designs to the new object environment (Meyer 2004).

(a) What are the core objects of the data model?

Core objects of the data model are: *Ownership*, *Encumbrances*, and *Separated Rights*.

- *Ownership Parcel*: a parcel is a unit of real property with rights and interests.
- *Encumbrances*: limitations on the rights and use of the land.
- *Separated Rights*: rights and interests in land ownership that can be disconnected from the primary or fee simple surface ownership.

(b) What are the basic spatial units of the data model?

Parcel is the only basic spatial unit of the data model.

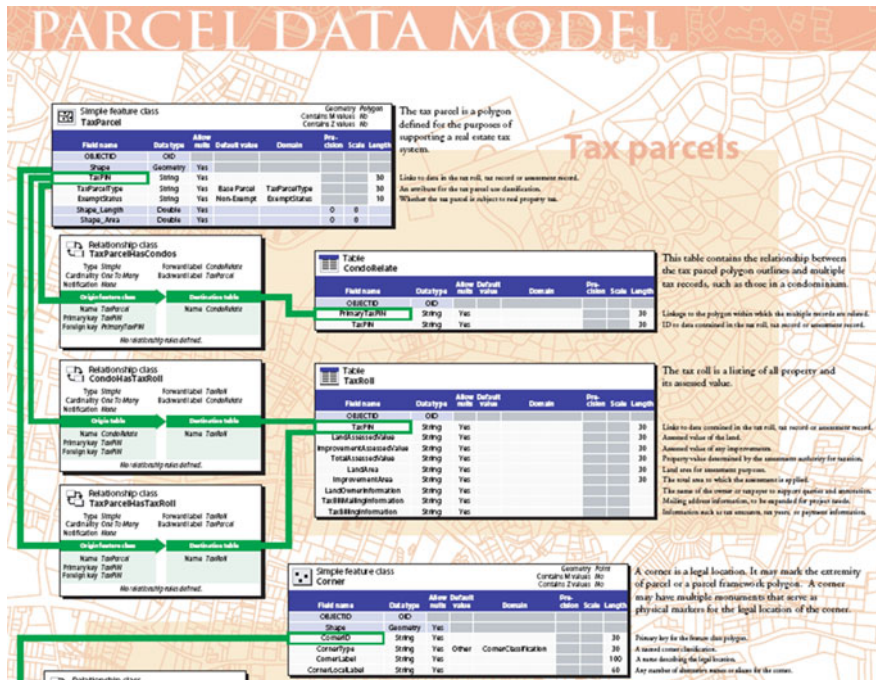


Fig. 1 Part of the ArcGIS Parcel Data Model (Meyer 2004)

(c) Does the data model have other forms of spatial units?

No other forms of spatial units are considered in the data model.

(d) What are the data sources (reference [legal] documents)?

All legal and authoritative documents such as deeds, survey plans, mortgages and lease contracts are used as a source of information.

(e) For what applications can the data model be used?

The data model can be used for development of parcel level management and can support parcel level functionality in the GIS environments.

(f) Does the data model have other types of interests (rights)?

Types of rights are mineral rights, oil rights, grazing right, and fishing rights.

(g) Does the data model consider temporal aspects of interests?

Temporal information (transaction date and time) is recorded in the data model.

(h) How does data model render stratified RRRs?

Many jurisdictions have condominiums or other structures that can form common interest areas and three-dimensional surfaces with different owners on different levels of the structures. A condominium is a separate system of ownership of individual units in a multiple-unit building. The units or buildings in the condominium are part of the ownership parcels with a vertical aspect and are called vertical parcels in the ArcGIS Parcel Data Model (Meyer 2004).

Figure 2 illustrates a vertical parcel that is a condominium building with condominium unit F that is on three separate floors. Unit F is connected by common elements, such as stairways and elevators. In the ArcGIS Parcel Data Model there are several ways to model or represent vertical parcels:

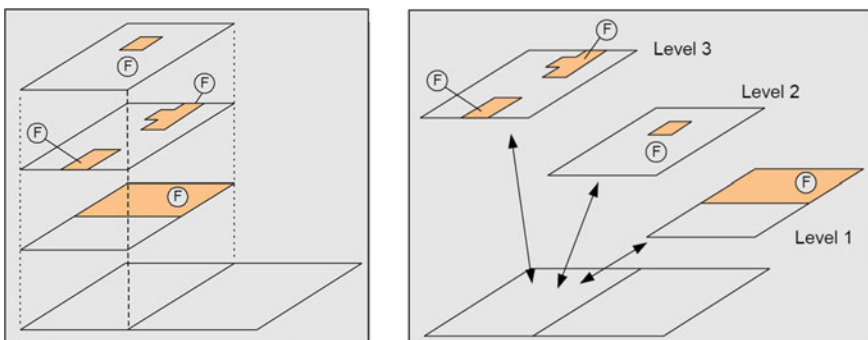


Fig. 2 Vertical parcels (*Left*) and a method of representing vertical points (*Right*) (Meyer et al. 2001)

- single outline polygon pointing to multiple parcel records,
- single outline polygon pointing to another series of polygons that represent the levels or floor, and
- single outline polygon that points to a three dimensional model of the building.

Stratified RRRs are projected on the polygons and they are not rendered with 3D primitives. Three-dimensional model of the building is not supported in the model.

(i) At what level does the data model support semantics?

Semantics are used to define the model's classes, attributes and parcel geometry.

(j) How does the data model support the physical counterparts of legal objects?

Land parcel is the representation of both legal and physical objects.

Overall, the ArcGIS Parcel Data Model is a general and flexible data model that helps users to manage land parcels using GIS technology. Vertical parcels (condominium) are supported in the data model; however, 3D primitives are not used to represent the vertical parcels.

4 ePLAN

The ePlan model was developed to model geometrical and textual information of Australian survey plans under the direction of the Intergovernmental Committee on Surveying and Mapping (ICSM). The ePlan model accommodates all of the survey geometry and administrative and titling data required to process a plan of subdivision from its initial preparation by the surveyor through to its lodgement with council for certification and subsequent registration (Kalantari et al. 2009).

The ePlan model has been classified into a number of packages (Fig. 3). They are Document, Surveyor, Survey, Parcel, Address, Geometry, Point, and Observation. 2D land parcels form the basis of ePlan. All administrative information is collected based on the land parcels, which are defined in the parcel package of the model (Kalantari et al. 2009).

(a) What are the core objects of the data model?

Core objects of the data model are: *Parcel*, *Document*, *Survey*, *Surveyor*, *Observation*, *Address*, *Point*, and *Geometry*.

- *Parcel*: The parcel element provides a basic unit to describe a spatial area.
- *Document*: Any other legal documents that define rights or ownership of land attached to the parcel.
- *Survey*: The survey element contains the survey components of the ePlan such administrative information, observation elements, and observation setup points.
- *Observation*: Observation elements.
- *Address*: Street address information for the parcel.

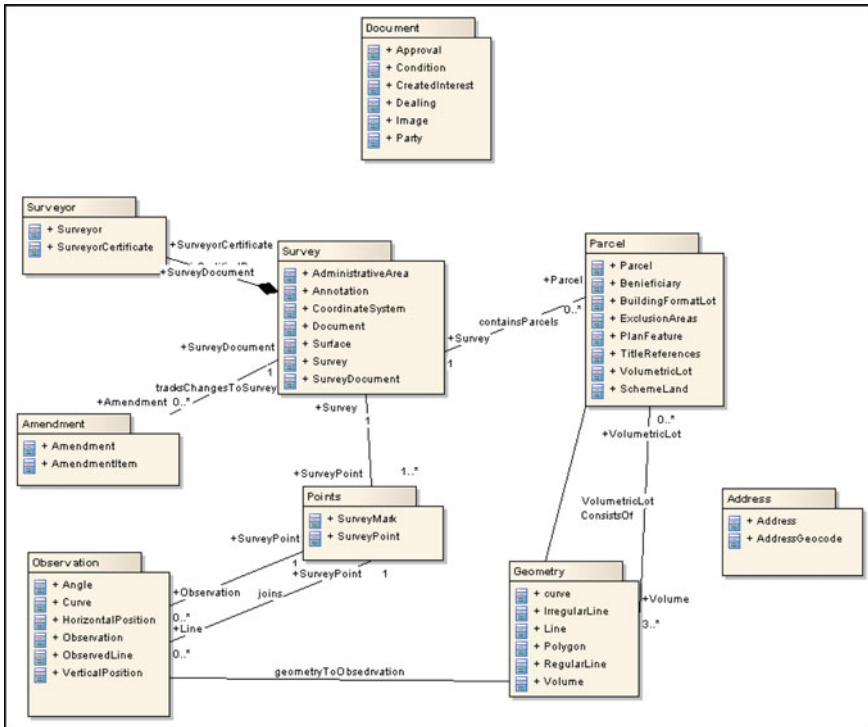


Fig. 3 ePlan model packages (ePlan 2010)

- *Surveyor*: information about the surveyor who participated in the survey.
- *Point*: Various administrative points such as boundary points, traverse points, reference marks, and permanent survey marks.
- *Geometry*: Consists of *Curve*, *IrregularLine*, *Line*, *Polygon*, *RegularLine*, and *Volume* to represent geometry of the parcel.

(b) What are the basic spatial units of the data model?

Parcel is the only basic spatial unit of the data model. *BuildingFormatLot* and *VolumetricLot* are used to describe 3D legal objects.

(c) Does the data model have other forms of spatial units?

No other forms of spatial units are considered in the data model.

(d) What are the data sources (reference [legal] documents)?

Title, Approval, Dealing and any other legal documents that defines rights or ownership of land attached to the parcel are used as sources of information.

(e) For what applications can the data model be used?

They can be used to: eliminate the current reliance on hardcopy or PDF plans, improve the quality of plan data and associated documents, improve plan examination processing, reduce requisitions, and enhance the accuracy of the Digital Cadastral Database.

(f) Does the data model have other types of interests (rights)?

Primary parcels are based level parcels that form the continuous cadastral fabric. They consist of lots, roads, reserves, common property, crown parcels and staged lots. In ePlan, they are captured using the parcel element (ePlan 2010).

Secondary interests in cadastral survey plans provide benefits and/or pose restrictions on primary cadastral parcels. These include easements, restrictions and depth limitations (ePlan 2010).

(g) Does the data model consider temporal aspects of interests?

Temporal information such as transaction date, surveying date, and administrative date is recorded in the data model.

(h) How does data model render stratified RRRs?

Each Australian state and territory has modified the ePlan’s national protocol specification (ePlan 2010) to cater for its requirements. For example, the Victorian ePlan version does not support volumetric lots and ePlan’s 3D elements, such as *VolumetricLot* (3D parcel) and *VolumeGeom* (3D primitive), have been excluded from the Victorian ePlan model (ePlanVictoria 2010). In Victoria, stratified interests are represented as 2D parcels.

Queensland, in contrast, supports 3D parcels and defines the properties of a 3D coordinate geometry collection (ePlanQueensland 2010). Building Format Plan and Volumetric Format Plan are two types of 3D parcels (Fig. 4).

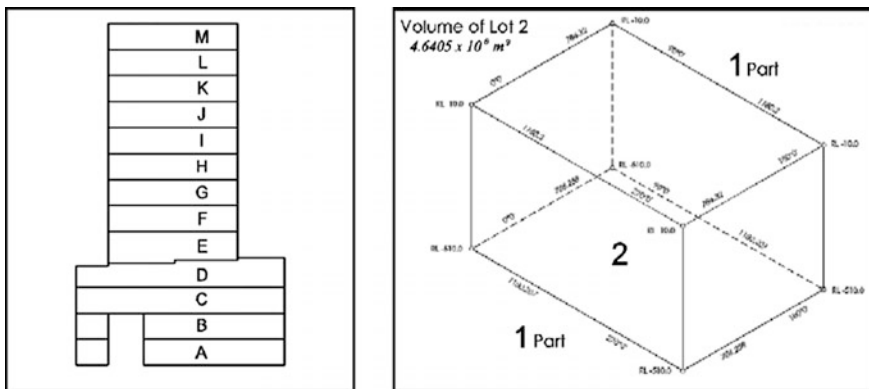


Fig. 4 Building format plan (Left) and volumetric format plan (Right) (DERM 2011)

In Queensland, if the format of the parcel is Building or Volumetric, it is mandatory to use building level number (*BuildingLevelNo*) and volumetric geometry (*VolumetricGeom*) to represent stratified interests (ePlanQueensland 2010).

(i) At what level does the data model support semantics?

Semantics are used to define the model's classes and attributes. However, the data model does not define the geometrical description of parcel.

(j) How does the data model support the physical counterparts of legal objects?

This ePlan does not support physical counterparts of legal object (parcel).

Overall, the current ePlan model has been exercised rigorously over the last several years in Australian jurisdictions. It has been designed to support 3D surveys, which include Volumetric and Strata (Building) surveys. These types of surveys can be prepared with the current protocol but have not been fully exercised (Cumerford 2010).

This ePlan is serving very well for 2D cadastre; however, having only *VolumetricLot* and *BuildingFormatLot* as attributes of ePlan's Parcel class (Fig. 4) to support Volumetric and Strata (Building) surveys, they are not enough to support the requirements of 3D cadastre (Aien 2012), which will be described in the next chapters.

5 LADM (ISO 19152)

The Land Administration Domain Model (LADM) is now the ISO standard and most recognisable data model in land administration discourse. This data model has been under development since the early 2000s. The LADM provides a conceptual description for a land administration system. The model aims to provide an extensible basis for the development and refinement of efficient and effective land administration systems and to enable involved parties, both within one country and between different countries, to communicate, based on the shared vocabulary implied by the model (ISO19152 2012).

(a) What are the core objects of the data model?

LADM has four basic classes (Fig. 5). They are *LA_Party*, *LA_RRR*, *LA_BAUnit*, and *LA_SpatialUnit* (ISO19152 2012).

- *LA_Party*: A person or organisation that plays a role in a rights transaction.
- *LA_RRR*: *Right* (action, activity or class of actions that a system participant may perform on or using an associated resource), *Restriction* (formal or informal obligation to refrain from doing something), *Responsibility* (formal or informal obligation to do something).
- *LA_BAUnit*: A Basic Administrative Unit is an administrative entity, subject to registration (by law), or recordation consisting of zero or more spatial units

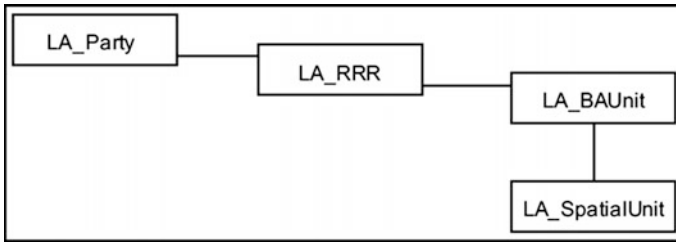


Fig. 5 Basic classes of LADM (ISO19152 2012)

against which (one or more) unique and homogeneous rights, responsibilities, or restrictions are associated to the whole entity, as included in a land administration system.

- *LA_SpatialUnit*: A single area (or multiple areas) of land and/or water, or a single volume (or multiple volumes) of space. Spatial units are structured in a way to support the creation and management of basic administrative units.

(b) What are the basic spatial units of the data model?

LA_SpatialUnit is the basic spatial unit of the data model. LADM’s code lists for Spatial Unit Package represents all types of spatial units supported by LADM (Fig. 6).

(c) Does the data model have other forms of spatial units?

Yes, LADM’s spatial component (*LA_SpatialUnit*) supports many ranges of spatial units such as *Sketch-based*, *Text-based*, *Point-based*, *Line-based*, *Polygon-based*, and *Topology-based* units. These spatial units are applicable in different land administration systems.

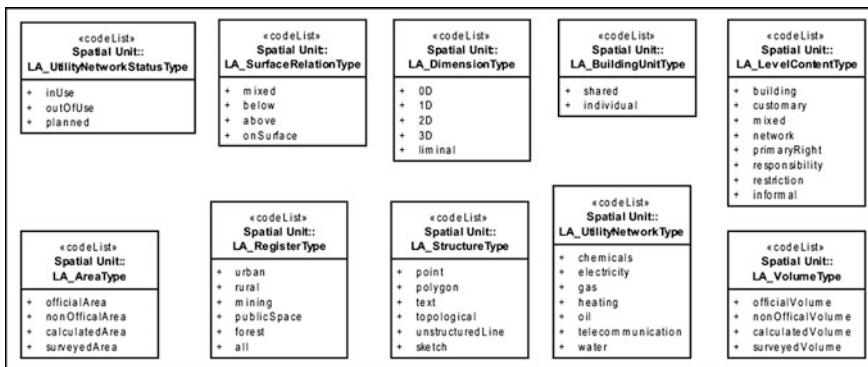


Fig. 6 Code lists for spatial unit package (ISO19152 2012)

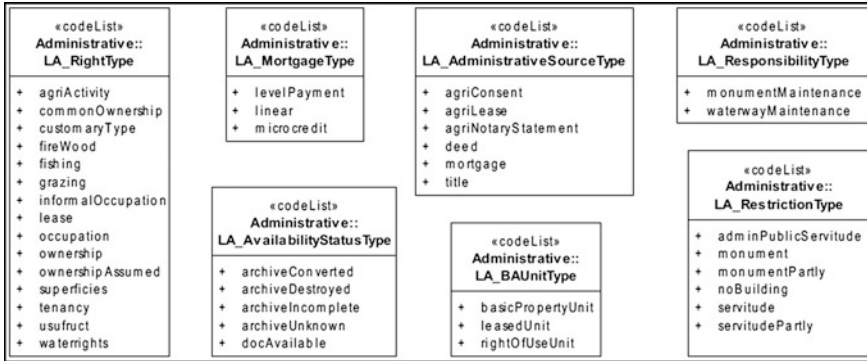


Fig. 7 Code lists for administrative package (ISO19152 2012)

(d) What are the data sources (reference [legal] documents)?

The data sources are all documents providing legal and/or administrative facts on which the land administration objects such as rights, restrictions, responsibility, basic administrative units, parties, or spatial units are based on. Deeds, titles, mortgages, agreements are examples of administrative or legal documents.

(e) For what applications can the data model be used?

The data model can be used for a number of land administration applications.

(f) Does the data model have other types of interests (rights)?

LADM’s code lists for the administrative package that represents all types of interests supported by LADM (Fig. 7).

(g) Does the data model consider temporal aspects of interests?

Yes. LADM covers history and dynamic aspects. *Class VersionedObject* is introduced in the LADM to manage and maintain historical data in the database. History requires, that inserted and superseded data, are given a time-stamp. In this way, the contents of the database can be reconstructed, as they were at any historical moment. Most of LADM’s classes are subclasses of *Class VersionedObject*.

(h) How does data model render stratified RRRs?

LADM has a solution to represent stratified interests and 3D parcels using Class *LA_BoundaryFace*. However, the LADM does not use solid geometry (*GM_Solid*) to represent 3D parcels. Pouliot et al. (2011) also suggests how solid representation can increase LADM’s 3D functionalities. Solid geometry facilitates 3D representation, volumetric calculation, and 3D spatial analysis.

LADM uses Class *LA_BoundaryFaceString* to represent 2D *LA_SpatialUnit* such as land parcels, and Class *LA_BoundaryFace* to represent 3D *LA_SpatialUnit*

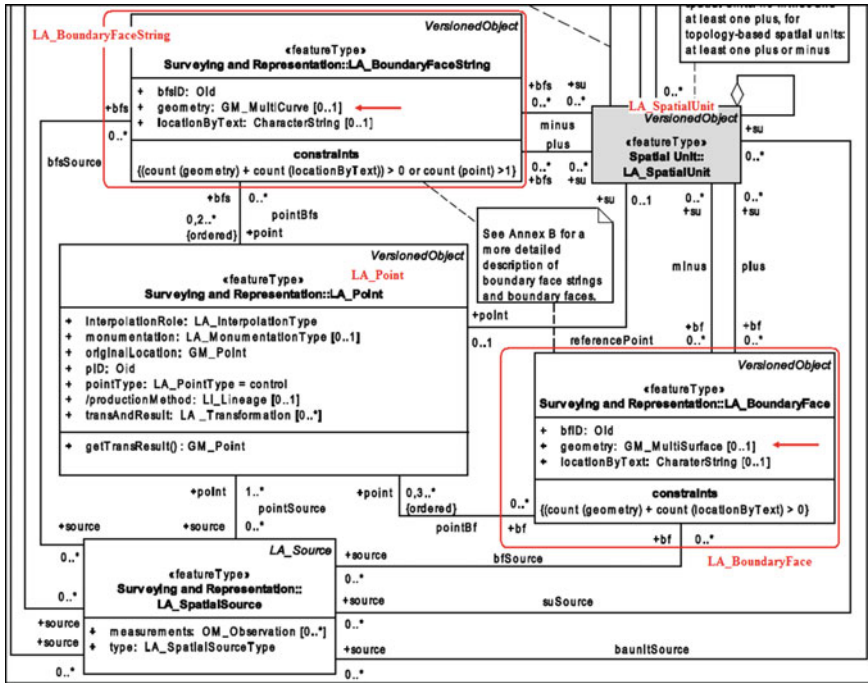


Fig. 8 Surveying and representation subpackage of LADM (ISO19152 2012)

such as 3D parcels (apartment units). These classes are defined by associating to Class *LA_Point* and Class *LA_SpatialSource* (survey documents).

LADM also utilises ISO 19107’s *GM_MultiCurve* and *GM_MultiSurface* geometry objects to represent Classes *LA_BoundaryFaceString* and *LA_BoundaryFace* respectively. They are included as attributes of those classes (Fig. 8).

According to ISO 19107, *GM_MultiSurface* is an aggregate class containing instances of *GM_OrientableSurface*. *GM_OrientableSurface* consists of a surface and an orientation inherited from *GM_OrientablePrimitive*, which is a subclass of *GM_Primitive*. Figure 9 shows that *GM_MultiSurface* only returns accumulated area and perimeter of all *GM_Surfaces* contained in the *GM_MultiSurface* and does not return volume (ISO19107 2005).

In contrast, using *GM_Solid* and *GM_CompositeSolid* in 3D cadastral data models to construct and represent 3D objects would facilitate 3D spatial analytical methods (3D buffering, overlap, intersect, etc.), interoperability for 3D data (integration, data discovery), 3D computation, and 3D visualisation (Zlatanova and Lee 2008). Therefore, LADM’s *LA_BoundaryFace* cannot return volume of the 3D objects according to ISO 19107 standard. Table 2 summarises LADM’s representation objects, their geometry attributes and corresponding objects in ISO 19107.

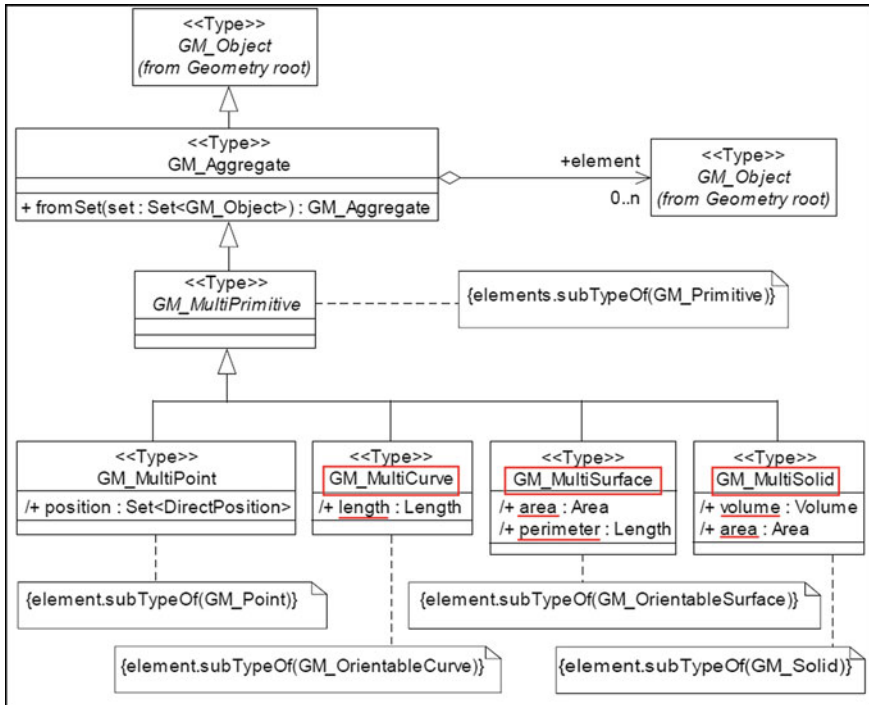


Fig. 9 GM_MultiSurface returns accumulated area of all GM_Surfaces contained in the GM_MultiSurface (ISO19107 2005)

Table 2 LADM’s representation objects and their attributes

LADM’s representation objects	Corresponding legal objects	LADM’s geometry attributes	Corresponding attributes in ISO 19107
LA_BoundaryFaceString	2D parcel	GM_MultiCurve [0..1]	Returns: length
LA_BoundaryFace	3D parcel	GM_MultiSurface [0..1]	Returns: area and perimeter . No volumetric attributes

(i) At what level does the data model support semantics?

Semantics are used to define the model’s classes, attributes. However, LADM’s 3D geometry representation class (LA_BoundaryFace) does not fully support semantics.

LA_BoundaryFace (semantic object), which is used to represent LA_SpatialUnit (semantic object) or its two specialisations LA_LegalSpaceBuildingUnit and LA_LegalSpaceUtilityNetwork, is not semantically enriched (Fig. 10). Semantically

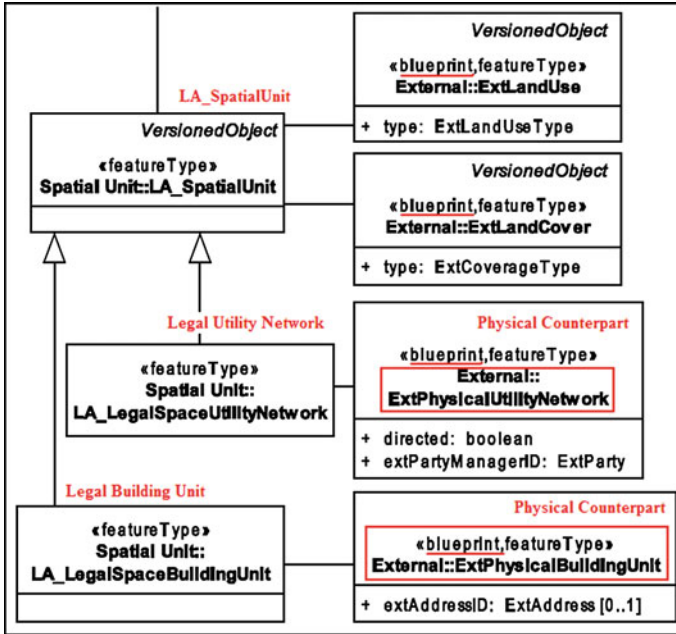


Fig. 10 External classes such as *ExtPhysicalUtilityNetwork* and *ExtPhysicalUtilityNetwork* in relation to LADM’s *LA_SpatialUnit* class (ISO19152 2012)

enriched data means enriching the content/context of the data by categorising or classifying data in relationship to each other. In a level of semantics, for example, a 3D object can be defined as a residential unit or a car park using appropriate names and attributes in a 3D virtual city models. In more detailed and enriched level, various parts/structures of the unit or car park including its walls, floors and ceilings can be identified.

A *LA_LegalSpaceBuildingUnit* that is represented by *LA_BoundaryFace* is used to describe the extent or part of an administrative entity (*LA_BAUnit*) that is associated with one or more unique and homogenous right (*LA_Right*) such as an ownership right. In the case of apartment units, ownership rights are defined based on physical structures and bounded by walls, roofs, floors and ceilings. It must also define whether the surface boundaries are interior, median, or exterior boundaries. *LA_BoundaryFace* does not provide such semantics and it represents merely the bound shape of a spatial unit using surfaces regardless to their definitions.

(j) How does the data model support the physical counterparts of legal objects?

LADM makes a distinction between legal and physical objects by introducing external classes. However, it is not possible to explicitly model a physical object in LADM. Using <<blueprint >> stereotype in external classes (Fig. 10,

highlighted in red colours) indicates that the classes are outside the scope of the LADM (ISO19152 2012), which means that they are not directly created in the LADM.

As such, LADM's legal objects are connected using object identifiers to the physical objects, which are in the external databases. For example, *ExtPhysicalUtilityNetwork* is a class for the external registration of mapping data of utility networks and is associated to Class *LA_LegalSpaceUtilityNetwork* (Fig. 10).

Also, *ExtPhysicalBuildingUnit* is a class for the external registration of mapping data of building units. *ExtPhysicalBuildingUnit* is associated to class *LA_LegalSpaceBuildingUnit* (Fig. 10).

In this model, the legal and physical objects are kept in different sources and are manipulated by different data providers. Diversity in data providers would create inconsistency in the integration of datasets, including institutional, technical, social, legal, and policy heterogeneity (Mohammadi et al. 2010). As a result, legal models may not match with their corresponding physical models in this data models.

Overall, LADM can be used as a basis for land administration system developments. It enables communication between land administration parties, both within one country and between different countries, based on the shared vocabulary implied by the model. However, implementation of LADM is not obvious (Pouliot et al. 2011). It supports 3D representation of interest. But, semantics are not fully supported by the data model in terms of geometry representation.

6 Summary of the Models and Conclusions

In this paper, six popular and important cadastral data models were reviewed and assessed. Actually, a few more data models such as DM.01 (Stuedler 2005), Harmonised Data Model (ICSM 2008), South Korean Cadastral Data Model (Lee and Koh 2007; Park et al. 2010) were investigated; however, only six of them were chosen because they were well documented and accessible to the researcher.

The aim of this paper was to understand how existing cadastral data models manage stratified land rights, restrictions, and responsibilities and their physical counterparts. Ten criteria were developed for the assessment purpose. They helped to explore each data model in detail. The results were summarised in Table 3.

Overall, it is concluded that although cadastral data models vary among jurisdictions, they rely on the basic building block of a 2D land-parcel. This trend is working satisfactorily for areas where no layered and stratified land rights exist. However, for 3D developments above and below ground, such as apartments, multi-story buildings, tunnels, and utilities, a 2D land parcel is no longer the most effective building block of cadastres; it is replaced by the 3D parcel. The 3D parcel is a volume of space on, above, or below the ground that defines and represents a particular array of rights, restrictions, or responsibilities (Aien et al. 2015).

Table 3 Specifications of the data models

Specification	Core cadastral Data Model	FGDC	ArcGIS parcel Data Model	Legal property Object Model	ePlan	LADM
Core objects	Person, right, parcel	Agent, Right and Interest, Parcel	Ownership, Encumbrances, Separated Rights	Person, Legal Property Object	Parcel, Document, Survey, Surveyor, Observation, Address, Point, Geometry	LA_Party, LA_RRR, LA_BAUnit, LA_SpatialUnit
Basic spatial unit	2D parcel	2D parcel	2D parcel	Legal Property Object	2D Parcel	LA_Spatial Parcel
Other forms of spatial units	N/A	N/A	Vertical parcels	Every object will be treated as a separate Legal Property Object	BuildingFormatLot, VolumetricLot	Sketch-based, Text-based, Point-based, Line-based
Reference documents	Transactions evidences, mortgages	Deed, aerial, photograph, agreement, mortgage, satellite image, survey notes	Deeds, survey plans, mortgages, lease contracts	All legal Document	Title, approval, dealing, any other legal documents	All documents providing legal and/or administrative facts
Applications	Valuation, taxation, security of land tenures, spatial or physical planning	Model cadastral information, Basis for automating the legal elements	Development of parcel level management, Support parcel level functionality in the GIS environments	Facilitates the land administration system to be more extensible and scalable in terms of new legislations and land-related laws	Eliminate the current reliance on hardcopy or PDF plans, Improve the quality of plan data, Improve plan examination processing	Land administration applications
Inclusion of other types of interests	N/A	Mineral rights, oil rights, grazing rights, fishing rights, development rights, floodplains	Mineral rights, oil rights, grazing rights, fishing rights	Biota rights, Mineral rights	N/A	Utility Network

(continued)

Table 3 (continued)

Specification	Core cadastral Data Model	FGDC	ArcGIS parcel Data Model	Legal property Object Model	ePlan	LADM
Temporal aspects	Yes	Yes	Yes	Yes	Yes	Yes
Management & representation of stratified RRR	2D parcel	2D Parcel	2D Parcel	2D Parcel	2D Parcel, VolumeGeom	L_A_BoundaryFaceString (2D Parcel), L_A_BoundaryFace (3D Parcel)
Semantic-level	Class and attribute description	Class and attribute description	Class and attribute description	Class and attribute description	Class and attribute description	Class and attribute description
Physical objects	N/A	N/A	N/A	N/A	N/A	External databases

Most of the cadastral data models such as the Core Cadastral Data Model, FGDC, and ArcGIS Parcel Data Model have been developed based on 2D land-parcels. ePlan and LADM support 3D parcels to model 3D RRRs. However, it has not been fully exercised in ePlan (Cumerford 2010). And, LADM does not use solid geometry (GM_Solid) to represent 3D parcels.

None of cadastral data models' geometrical representation is semantically enriched. Semantic enrichment reduces the ambiguities for geographic integration and geometrical inconsistencies (Kolbe 2009).

3D cadastres can be used by different customers within multiple applications, provided a common information model could extend over the different users and applications. A semantically enriched 3D cadastral data model would then enable collaboration in heterogeneous environments.

The cadastral data models do not integrate physical counterparts with 3D legal objects (3D parcels) (Aien et al. 2013b). The required level of detail of physical information is dependent on the application. For example, land registries may require a simplified overview of the physical models (walls, ceilings, and roofs), while very detailed information (windows, doors, stairs, and pipes between walls) may be required in property management. Using LADM's external classes would not allow the users to define the level of detail of required information for a specific application.

Acknowledgments The authors wish to acknowledge the support of the Australian Research Council Linkage Project (LP110200178) on Land and Property information in 3D, its industry partners and the members of the Centre for Spatial Data Infrastructures and Land Administration in the Department of Infrastructure Engineering, The University of Melbourne, in the preparation of this paper and the associated research. However, the views expressed in the paper are those of the authors and do not necessarily reflect those of these groups.

References

- Aien, A. (2012). 3D cadastral data model: A foundation for developing a national land information infrastructure. In A. Rajabifard, I. Williamson, & M. Kalantari (Eds.), *A national infrastructure for managing land information—research snapshot* (pp. 116–121). Melbourne: The University of Melbourne.
- Aien A. (2013). 3D cadastral data modelling. PhD Dissertation, Department of Infrastructure Engineering, The University of Melbourne.
- Aien, A., Rajabifard, A., Kalantari, M., Williamson, I.P. (2011). Aspects of 3D cadastre—A case study in Victoria. In: FIG Working Week 2011, Marrakech, Morocco. FIG.
- Aien, A., Kalantari, M., Rajabifard, A., Williamson, I., & Bennett, R. (2013a). Utilising data modelling to understand the structure of 3D cadastres. *Journal of Spatial Science*, 58(2), 215–234.
- Aien, A., Kalantari, M., Rajabifard, A., williamson, I., & Wallace, J. (2013b). Towards integration of 3D legal and physical objects in cadastral data models. *Land Use Policy*, 35, 140–154.
- Aien, A., Rajabifard, A., Kalantari, M., & Shojaei, D. (2015). Integrating legal and physical dimensions of urban environments. *ISPRS International Journal of Geo-Information*, 4(3), 1442–1479.

- Cumerford, N. (2010). The ICSM ePlan protocol, its development, evolution and implementation. In: FIG Congress 2010, Sydney, Australia, April 11–16, 2010. FIG.
- DERM. (2011). *Registrar of titles directions for the preparation of plans*. Brisbane: Department of Environment and Resource Management.
- ePlan (2010) ePlan Model version 1.0. ePlan. ICSM, Sydney, Australia.
- ePlanQueensland. (2010). ePlan Protocol-Queensland LandXML Mapping- Version 0.1. Spatial Information- Spatial and Scientific Systems. Department of Environment and Resource Management.
- ePlanVictoria. (2010). ePlan Victoria-CIF Specification version 1.0. Electronic Plans of Subdivision. Victorian Government Department of Sustainability and Environment, Melbourne, Victoria, Australia.
- FGDC. (1996). Cadastral Data Content Standard for the National Spatial Data Infrastructure-Version 1.1. NSDI, Standards Working Group, Standards Reference Model.
- Godard, J.P. (2004). Urban underground space and benefits of going underground. Paper presented at the World Tunnel Congress 2004 and 30th ITA General Assembly, Singapore.
- Henssen, J. (1995). Basic principles of the main cadastral systems in the world. In: Seminar Modern Cadastres and Cadastral Innovations, Delft, The Netherlands. In *Proceedings of the One Day Seminar held during the Annual Meeting of Commission 7, Cadastre and Rural Land Management* (pp. 5–12). The International Federation of Surveyors (FIG).
- ICSM. (2008). ICSM Harmonised Data Model, Version 2. ICSM.
- ISO19107. (2005). Geographic Information-Spatial Schema. Standards Australia. AS/NZS ISO.
- ISO19152. (2012). ISO 19152. International Standard, Geographic Information—Land administration domain model (LADM). Geneva, Switzerland: ISO.
- Kalantari, M., Rajabifard, A., Wallace, J., & Williamson, I. P. (2008). Spatially referenced legal property objects. *Land Use Policy*, 25(2), 173–183.
- Kalantari, M., Lester, C., Boyle, D.R., Coupar, N. (2009). Towards eLand Administration—electronic plans of subdivision in Victoria. In: Baldock, B. O. P., Bruce, D., Burdett, M., Corcoran, P. (eds) *Proceedings of the Surveying & Spatial Sciences Institute Biennial International Conference, Adelaide 2009* (pp. 155–162). Surveying & Spatial Sciences Institute.
- Kolbe, T. H. (2009). Representing and exchanging 3D city models with CityGML. In J. Lee & S. Zlatanova (Eds.), *3D geo-information sciences*. Berlin: Springer.
- Lee, J., & Koh, J. H. (2007). A conceptual data model for a 3D cadastre in Korea. *Korean Journal of Geomatics*, 25(6–1), 1–10.
- Mohammadi, H., Rajabifard, A., & Williamson, I. P. (2010). Development of an interoperable tool to facilitate spatial data integration in the context of SDI. *International Journal of Geographical Information Science*, 24(4), 487–505.
- Navratil, G. (2009). Cadastral boundaries: benefits of complexity.
- Nv, Meyer. (2004). *GIS and land records: the ArcGIS parcel data model*. Redlands, CA, U.S.: ESRI Press.
- Nv, Meyer, Oppmann, S., Grise, S., & Hewitt, W. (2001). *ArcGIS parcel data model version 1*. Redlands, CA: Esri.
- Oosterom, P. (2010). FIG joint commission 3 and 7 working group on 3D-Cadastres—Work plan 2010–2014. <http://www.gdmc.nl/3DCadastres/>.
- Park, S., Lee, J., Li, H. –S. (2010). Data model for 3D cadastre in Korea. *Paper presented at the 2010 Second International Conference on Advanced geographic information Systems, Applications, and Services*, St. Maarten, The Netherlands.
- Pouliot, J., Vasseur, M., Boubehrezh, A. (2011). Spatial representation of condominium/co-ownership: comparison of Quebec and French cadastral system based on LADM specifications. In: P. V. Oosterom, E. Fendel, J. E. Stoter, A. Streilein (Eds) *2nd International Workshop on 3D Cadastres Proceeding*, Delft, The Netherlands (pp 271–290). FIG.
- Stuedler, D. (2005). Swiss cadastral core data model- experiences of the last 15 years. *Computers, environment and urban systems*, 30(5), 600–613.

- Stoter, J. E. (2004). *3D Cadastre*. Delft: Delft University of Technology.
- Stoter, J. E, Oosterom, P. v. (2006). 3D cadastre in an international context: legal, organizational, and technological aspects. Taylor & Francis.
- Zlatanova, S., & Lee, J. (2008). A 3D data model and topological analyses for emergency response in urban areas. In J. Lee & S. Zlatanova (Eds.), *Geospatial information technology for emergency response*. Francis: Taylor.

The Hierarchical Three-Dimensional (3D) Dynamic Water Infiltration on Multi-layers of Soil According to Voronoi Sequence Nodes Based on the Three-Dimensional Triangular Irregular Network (3D TIN)

Siti Nurbaidzuri Reli, Izham Mohamad Yusoff, Habibah Lateh and Uznir Ujang

Abstract Understanding soil water infiltration movement has been birthed from extensive interest and concern in the last few decades. The arrangement of particles (i.e. structures and sizes) and the interaction between both the soil and soil water have a profound effect on the soil water infiltration. The challenging task in the soil fluid modelling is the indeterminate spatial extent that has no specific boundaries and the fact that it is difficult to sense. Plenty of investigations and studies have been conducted to measure the water movement. However, less focus has been given on the movement of the dynamic soil water infiltration. This paper will focus on modelling the three-dimensional (3D) soil water infiltrations that flow downward due to the gravitational factor and gradient pressure. The 3D hierarchical soil water infiltration model proposes the integration of techniques which includes the Tree-map to isolate the depth of the soil that acts as a route of the soil water flow from the surface of the terrain to the subsurface flow. Moreover, the 3D Gosper curve is used to represent the soil water flow pattern that is based on the law of gravity and Horton equation, which control the flow of the soil water in the model. The curves that consist of a series of nodes adopt the Three-Dimensional Triangular Irregular Network (3D TIN) which creates a network of flow direction that allows the water to pass through the nodes according to a predetermined sequence. The study area has an average of 8.5 mm total rain and -5 m water level. The soil is divided into a few layers to represent the flow of the soil water according to the sequence of nodes. The soil depth (40, 80, 120, 160 and 200 cm) isolation in the

S.N. Reli (✉) · I.M. Yusoff · H. Lateh
School of Distance Education, Universiti Sains Malaysia (USM),
11700 Minden, Pulau-Pinang, Malaysia
e-mail: baidzuri88@gmail.com

U. Ujang
Department of Geoinformatics, Faculty of Geoinformation Science & Engineering,
Universiti Teknologi Malaysia (UTM), 81310 Skudai, Johor, Malaysia

form of Voronoi-shaped polygon nodes allows the soil water to flow down where the depth is chosen based on the soil wetting range of the subsurface soil.

Keywords Soil water • Infiltration • Three-Dimensional (3D) • Fluid modelling

1 Introduction

The study on the streamflow has been gaining much attention nowadays as the knowledge is essential in studies related to hydrology, geology, environment and hazard management. The research on streamflow includes the studies that focus on the flow of the stream, stream flow direction, stream management and also the process that generates stream. The process that generates stream can be divided into the Infiltration Excess Overland Flow (IEOF), Saturation Excess Overland Flow (SEOF), Shallow Subsurface Flow (SSF), Direct Precipitation onto Stream Surface (DPOSS), percolation, evapotranspiration and ground water (GW). Infiltration happens in most of the processes of stream generation and it is important in hydrological studies where the process involves several mathematical equations and formulas.

Among the earliest studies on water movement is by Barnes and Allison (1988) who traced the water movement in the saturated zones using stable isotopes of hydrogen and oxygen. In 1941, Hursh and Brater came out with the studies on the role of the subsurface storm flow where they showed the stream hydrograph response to storm rainfall that was composed of two main components namely the channel precipitation and subsurface stormflow (Hursh and Brater 1941). The study on streamflow was continued by Hoover and Hursh (1943) and they showed that the soil depth, topography, and hydrologic characteristics were associated with different elevations that were influenced by peak discharge. Dunne and Black (1970) on the other hand, pointed out that the most important transition between the flow systems seems to be at the soil surface, when water is released from the extreme damping effect of the subsurface flow. There are several methods used to detect the movement of water such as by using stable-isotope and chloride tracers (Newman et al. 1997), hydrogen and oxygen isotopes, chloride, and chlorine-36 (Liu et al. 1995) and gravimetric sampling (Nielsen 1964).

The use of the three-dimension (3D) in hydrology has received increasing attention in the present studies but only for the visualization purpose. However, it is very difficult to see the focus of study that uses the temporal data to create a 3D model used to represent the soil water infiltration process that happens underneath the soil. The previous studies on the soil water movement and 3D model are mostly carried out separately. Thus, this study will propose for the advancement using both the graft and mathematical representation models of the soil water infiltration that used the 3D model to show the dynamic movement of the soil water through a static pattern of sequence that represents spaces in soil. The nature of the soil water which is in the liquid state, has resulted in the decision on using the 3D dynamic fluid flow

model according to the voronoi sequence grid nodes hierarchically based on the multi-layers of the soil. The soil water that flows within the soil particles requires further research, particularly on the suitability of these data models to cater for the acquired 3D soil water movement. This research is not only focusing on the visualization but it also translates and converts the temporal data of rainfall and dynamic data of soil water flow into the 3D model. The natural process of the soil water flow requires the 3D model because the soil water not only moves downward but also horizontally in any direction which depends on the soil water potential. Static objects can be represented and analyzed in 2 Dimension (2D). However, the dynamic nature of the soil water is not suitable to be represented in 2D because the soil water flow constantly needs the depth information (z). In addition, the geometric representation of soil which involves volume of water requires 3D representation. The method used in this study is explained in Sect. 3.

The reviews on the role of the GIS in soil water infiltration studies, the advancement of 2D into 3D soil water movement and the practical relationship between the intended 3D modelling and the dynamic water infiltration that show the contribution of the 3D GIS to the studies related to soil water movement are discussed in Sect. 2. Meanwhile, the methodology part (Sect. 3) will discuss and explain the relevance of the chosen methods for the development of the 3D model. The tree-map is used to provide a hierarchy to the depth of the soil and tree algorithm shows how the isolation is done. The 3D Gosper curve is chosen to represent the soil water flow pattern that best suits the direction of the soil water, and the infiltration and volume of soil water depend on the Horton equation.

2 Literature Review

This section describes the related issues and information on the role of the GIS in soil water infiltration studies which show the needs to execute the 3D modelling for soil water infiltration movement with the help of the advancement of 2D, into 3D.

2.1 *The Role of GIS in Soil Water Infiltration Studies*

Soil-related studies using GIS technologies are mostly dedicated to water balanced (Alemaw and Chaoka 2003; Ju et al. 2010; Pimenta 2000; Portoghese et al. 2005), real time flood prediction (Al-Sabhan et al. 2003), characterized maximum infiltration rate (Brito et al. 2006) and distributed rainfall-runoff model (Zollweg et al. 1996; Jain et al. 2004). One of the examples of the GIS and soil water studies is done by Tan et al. (2008) that implements a 3S-based hydro-geological model for conducting a study regarding an effective assessment of a regional rainfall-induced landslide in order to investigate shallow landslide in the watershed. In this research, the 3S refers to the Geographic Information System (GIS), the Global Positioning

System (GPS) and the Remote Sensing (RS) framework that is essential in establishing physical, mechanical, geological and hydraulic properties. The TRIGRS model helps in analysing the response of transient pore-pressure during a rainfall event that also estimates the result for landslide susceptibility while the Kriging interpolation method is used to analyse the rainfall intensity distribution. The 3S-based hydrological model provides an effective assessment of the regional rainfall-induced landslide where every system offers great contributions to the studies.

With the advent of the increasing GIS technique and computing power, a robust innovative physical-based hydrologic modelling has become important in contemporary hydrology to evaluate the impact of human intervention or possible climatic change on basin hydrology and water resources, but most of the designs have been found to be less suitable for real time applications and are usually not well integrated with spatial datasets for example the GIS (Alemaw and Chaoka 2003; Al-Sabhan et al. 2003).

Although the awareness on the importance of GIS technologies is rising but the studies that concentrate on soil water infiltration are less emphasized, even though it is very important in the hydrological process. Infiltration is an important soil feature that controls the runoff, leaching and crop water availability. Besides, soil water infiltration also becomes one of the triggering factors to hazards like soil erosion, flood and landslide. As soil water infiltration contributes to surface terrain changes, it is relevant to adapt to the use of the GIS in soil water studies. Attention towards this research has been scarce, even in the scope of knowledge that the soil water infiltration leads to many hazards such as flood and landslide.

Aspects of the real world objects can be described and represented in a computer by the data model comprising of a set of constructions or rules. The geographic space representation should be first identified in order to visualize natural phenomena. The GIS can best represent either geographic objects or surfaces as data through mathematical construction. The evolution of the data model proves the advance and proliferation in computer technology and the competitive nature of the GIS. Vector, raster and TIN data models are examples of data models with different types of the geographical representation used in the GIS. Simple spatial features are represented as a collection of point, line and polygon in the vector data model. The vector data model can be processed, accessed and interpreted by the computer by organizing geometric objects and their spatial relationships into digital data files. However, in the raster data model, geography is represented as cell matrixes which store numeric values where this model is a better option to represent continuous phenomena that use a regular grid in order to cover the space. The TIN data model on the other hand, represents geography as a set of contiguous non-overlapping triangles.

Since the soil water infiltrations occur underneath the soil and the flow direction is unpredictable, the dynamic flow requires a better representation and analysis which is in the form of 3D models. Apart from the use of the GIS techniques, the indeterminate spatial extend of soil water infiltration necessitates the 3D to show its dynamic movement based on the precipitation data. The advancement of 2D into 3D soil water movement is to be discussed next.

2.2 *The Advancement of 2D into 3D Soil Water Movement*

According to MacEachren and Kraak (1997), the concept of visualization is a key issue in the changing nature and using of maps in science as a consequence of the growing need of geo-referenced data and rapidly evolving technologies that can provide this information in more innovative ways. This technology evolution also important and give huge contribution in the field of hydrology. As stated by Fuhrmann (2000) the spread of modern information and communication technologies within the last three decades has led to an increased collection, availability and use of spatial and temporal digital hydrological data. Therefore the key issue for both public authorities and scientists is the needs and essential for visualization and regionalization of hydrological data and particularly temporal and spatial aspect of hydrological modelling.

This section reviews some of the studies that involve 3D in hydrological related studies. In 2002, Droque et al. applying GIS by using ArcView as a 3D visualization tools and Hydrological Recursive Model (HRM), a conceptual rainfall-runoff model that represent downstream variation of daily streamflow where 3D spatio-temporal cartography of mean annual high raw and specific discharges are illustrated. Spatial analyst in ArcView was used to determined hydromorphometric data, land cover data and geological data for every catchment. The obtained regional parameter set in this study was used for model transposition and production of realistic streamflow mapping in combination with the 3D module of ArcView.

The linear spatialization and 3D mapping of the hydrograph is an important issues in hydrological analysis because it allow visualization and a temporal survey of streamflow in any point of the drainage network and is helpful from a pedagogical point of view for river management, as well as for the identification of runoff generating areas and critical influences in a drainage basin (Droque et al. 2002). As the demand from GIS applications in the 3D environment increases, the basic form (e.g. single z-value for an x and y location) of data representation are no longer adequate (Raper and Kelk 1991). Stream and other flowing soft geo-object can be simulated in 3D by using various technique such as the combination of GIS flow element (FE) and GIS soft voxel (SV) which were applied in simulating soil erosion and overland flow (Shen et al. 2006) and using 3D Volumetric Soft Geo-object (VSG) data model to visualize areas and overland flow volume generated from IEOF process dynamically, driven by Green-Ampt infiltration equation (Izham et al. 2010). Spatial data representations may be describe as surface-based (grid, shape model, boundary representation (b-rep and facet model) and volume-based (3D array, octree, constructive solid geometry (CSG) and 3D TIN) (Li 1994) where surface based of an object was represented by surface primitive while volume-based describe as object's interior that describe by solid information.

2.3 Practical Relationship Between Intended 3D Modelling and Dynamic Water Infiltration

The research on soil water usually involves mathematical model, graph representation or 2D soil moisture profile. As the study involve soil water infiltration focus more on 2D model, this study shows the importance and relevance of the advancement into 3D model that able to show the movement of fluid according to appropriate pattern as a result of gravity forces. The previous studies of integrated hydrology and 3D GIS includes 3D model for stream water, flood, over land flow and IEOF. This research attempt to emphasize the 3D modelling of liquid movement under the soil surface that use soil water and subsurface flow as the representation of the liquid movement in the model. Apart from showing the dynamic movement of soil water and subsurface flow, this model create a geometric representation of soil that consist of pores that provides the path for the water flow.

Since soil water infiltrates dynamically with uncertain direction, and the movement is influence by volume of precipitation and soil characteristics, the best way to show the dynamic movement of soil water is through 3D modelling. Besides, the movement of water into the soil is difficult to describe with text or represent solely by equation. However, the equation can be used to control the movement of liquid that represent the soil water in the 3D model. This study shows the relationships between mathematical equation (Horton), soil water infiltration and soil water direction movement (tree-map and 3D Gosper curve) that used to create the 3D model which applied 3D TIN for representing the sequence of soil water flow direction. Precipitations, rate of infiltration, rate of water accumulate in soil and volumes of water that become subsurface flow are the dynamic data that modeled in form of fluid flow in the model. The triggering factor of this model is precipitation. The soil water infiltration does not stop although the precipitation stops, but always moving with the times due to the present of gravity force that continue to move the water. Thus, the used of 3D model in this research can shows the movement of soil water that receive less attention in most hydrological model.

3 Methodology

This section explains the relevance of the method chosen based on previous related studies that highlight the ways to divide layers in the soil. Appropriate algorithm, curve and equation used for the 3D soil water infiltration modelling are also included.

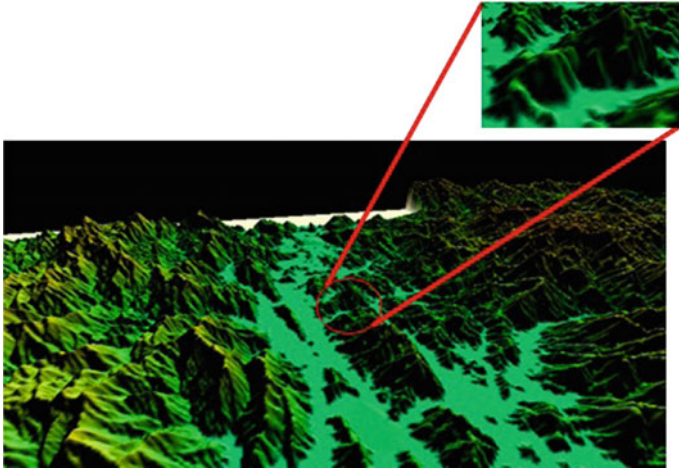


Fig. 1 Study area of East West Gerik-Jeli highway

3.1 Determination of Soil Water Infiltration Using Digital Terrain Model (DTM)

According to Brito et al. (2006), the area with sandy soil types, vegetated land used and the non-existent slope has high infiltration while the area with the steep slope of greater than 25 %, clay soil, and an impervious land use shows poor infiltration. The location of the study area is at N 05, 33.113 E 101, 21.253 at km 70.52, East West Gerik-Jeli Highway, the cut-slope sides of the highway heading to Kelantan, Malaysia (Fig. 1). The surface terrain is computed by the ASCII format data to 2.5 Dimension of surface representation from Global Mapper 14. The study area consists of clay soil and is covered by grass and trees and has -5 m water level and 8.5 mm total rain. The soil water infiltration determination is based on the soil types, soil surface (DTM), slope percentage and temporal precipitation.

3.2 Isolation Layer of Indeterminate Spatial Extend of Soil Water Movement Using 3D Gosper Curve

The distribution of water in soil is one of the indeterminate spatial extends which is difficult to sense and which has no specific boundary. Other examples of the indeterminate spatial extend are smoke, soil type and temperature. However, this paper will discuss on the soil water movement that proves to be important in the streamflow generating process and hazard management. A Gosper curve that originally generates shapes to facilitate visualization and navigation of hierarchy (Auber et al. 2013) is converted into a 3D Gosper curve that is able to show the

sequence of the water flow infiltration that moves into the soil with the voronoi-shaped polygon nodes sequence. The algorithm created is able to isolate areas according to the depth of the corresponding nodes that represent the wetting area of soil after precipitation. In other words, this paper focuses on the isolation of soil layer in the form of 3D vertically where the soil water moves downward through the soil particles due to gravity force. The tree-map used and the linear arrangement of leave are modified according to the appropriateness of the model. The soil water distribution pattern, flow arrangement and the explanation the output are discussed in Sect. 4. Since the soil water involves a dynamic movement of fluid underneath the soil surface, the water flow in the model that is influenced by the temporal data of precipitation is stored and managed using a dynamic data structure which allows the growth of data.

3.3 Tree-Map and Algorithm Used to Isolate the Soil Layer

Humans tend to give different responses to the descriptions delivered through different methods. The ability to identify the spatial configuration of element in the form of picture has been tested and approved. Besides, humans also have the ability to notice the relationship between those elements easily. Visual ability enables people to get the content of the picture faster than those represented in the form of text (Kamada and Kawai 1991).

Hierarchical information usually presented by a method called the Tree-Maps, is expected to be able to produce similar visualization techniques achieved in other areas (Johnson and Shneiderman 1991) where the hierarchical information represented in the form of a rectangular 2D is displayed in a space-filling manner. The Tree-Map allows interactive control that can display both content and structural information. Every point of the Tree-map corresponds to a node in the hierarchy. The sibling nodes will never overlap due to the node's bounding box that completely encloses the bounding boxes belonging to its children. The methods of the Tree-maps include the slice and dice, Squarified treemap, Strip layout, Quantum treemap, Mixed treemap, Voronoi treemaps and Rectangle Hillbert treemap (Auber et al. 2013).

Large hierarchies contribute to the problem of navigation and visualization. The algorithm designed preserves the region containment based on the hierarchy. GosperMap creates boundaries as contour lines that mark the region with a similar altitude that has the same level of hierarchy. The method that is usually used to visualize hierarchies is the tree representations that slice information displaying a different hierarchy by using a single image, and the solution is based on the 2D space-filling curves (SFC).

Every curve has its own advantages and disadvantages. Researchers need to consider the needs of their research in order to select the most appropriate curve. The study by Asano et al. (1997) combines several recursive and general SFCs namely the Recursive SFC (RSFC) that improve the number of seek operations.

The SFC revolves around the ways on mapping multi-dimensional space in 1 Dimensional (1D) space where it functions as the route that passes through every cell available in a cell element to ensure that every cell is visited once (Mokbel, 2003). There are many kinds of SFC available that suggest different ways of mapping to the 1D space such as the z-order, Hillbert's curve, Gray code and Snake code (Asano et al. 1997). The shape of the curve is shown in Fig. 2.

A non-standard database system considers the importance of the query performance and the use of SFC for a multidimensional data structure is to map multi-dimensional points to one dimension which produces two points in the data structure. Nevertheless, the research by Uznir et al. (2013) presents an opponent data technique of the 3D Hilbert curve used to represent the 3D City Model. These new techniques that extend Hilbert SFC into a higher dimension prove on the improvement in the data retrieval time.

This study has proposed the use of the Tree-map for dividing soils into several stages of depth in order to represent the flow of the soil water infiltration. The

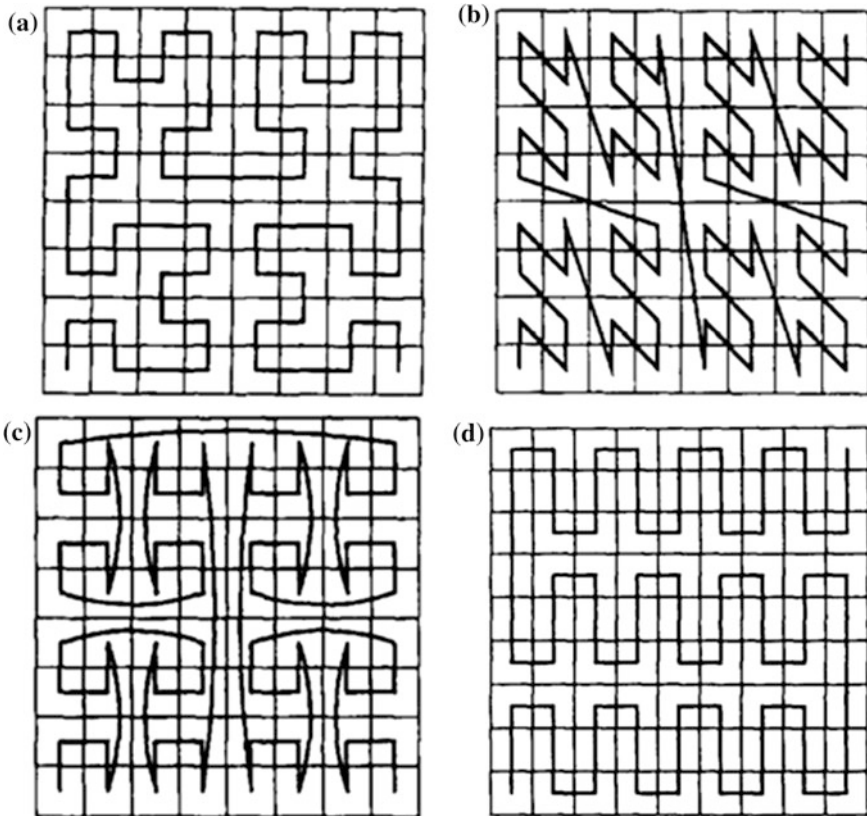
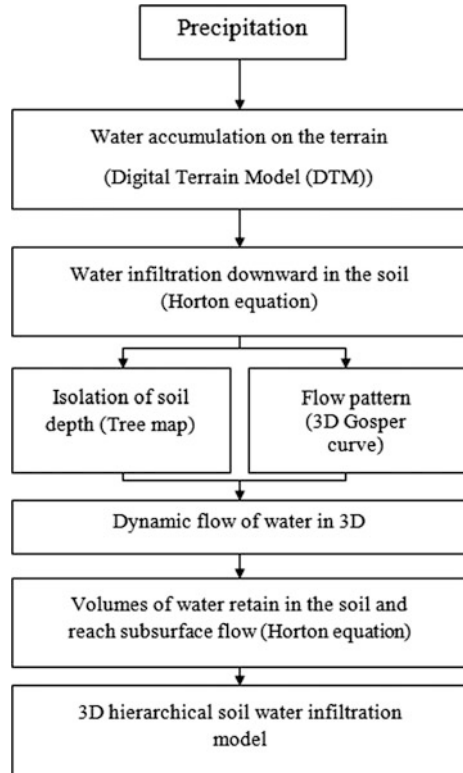


Fig. 2 Categories of space-filling curve. **a** Hillbert's curve. **b** z-curve. **c** Grey code curve. **d** Snake curve. Image from (Asano et al. 1997)

Fig. 3 The flow of methodology



Tree-map as explained previously is able to isolate information with boundaries. Tree-map capabilities in doing this isolation give an inspiration to the application of these techniques for the 3D soil depth segregation. The 3D Hillbert SFC curve by Uznir et al. (2013) proves that the curve can be upgraded into 3D. However, these studies proposed that the use of the Gosper curve can meet the requirement of the studies instead of using the Hillbert curve where the Gosper curve represents the hierarchical depth of the soil in the form of 3D. The results of using the Tree-map, Gospers curve, Horton equation and Tulips are described in Sect. 4. The overview of the methodology is summarized in the chart below (Fig. 3).

4 Result and Analysis

The Tree-map is used to represent every depth of the soil from the surface of the terrain to the subsurface flow suitable with its capabilities to divide the plane, from the top of the tree to the bottom. The overview of the algorithm is shown in Fig. 4a. The green node represents the parent, other colours represent the child and the blue numbered nodes are the linear arrangement on the leaves.

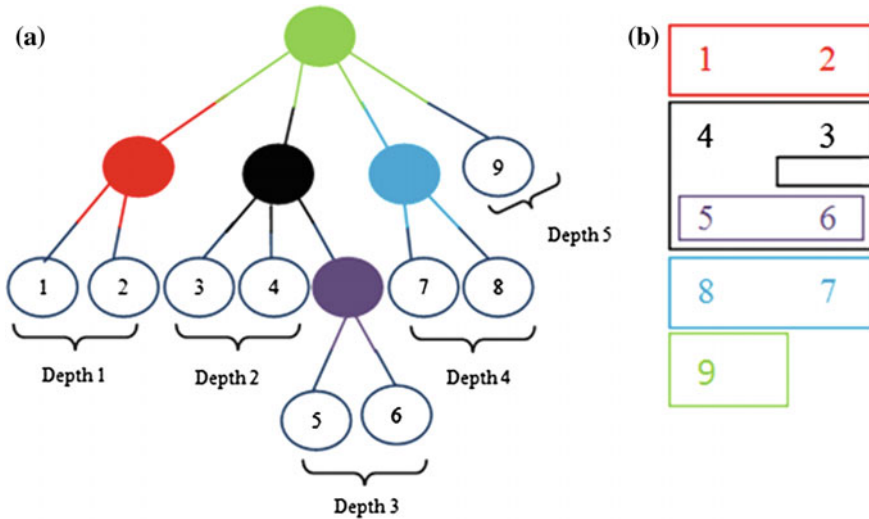


Fig. 4 a The overview of the algorithm with soil depth isolation in tree-map. b Leaves arrangement

The depth of the soil created from the Tree-map is represented by numbers that belong to every parent and child as shown in Fig. 4a. The isolation of the soil depth is represented well in the Tree-map. The arrangement of leaf is represented in Fig. 4b where the pattern is applied from the Gosper curve. The colour of the number and boxes represents the colour of the nodes before. There is open source software that can be used to show the interesting visualization representation of Tree-map, namely Tulip.

Tulip is capable of visualizing a framework of information for analyzing and visualizing relational data that support the design of the interactive information visualization. The model as mentioned before, is divided into five layers with the respective depths of 40, 80, 120, 160 and 200 cm where the depth of the water flow is ascertained by the volume of precipitation. The isolation of depth is in the form of Voronoi-shaped polygon nodes that allow the soil water to flow down where the depth is chosen based on the soil wetting range of the subsurface soil. For clayish soil, Dingman (2002) stated that at steady water input rate, with 0.50 Φ porosity the water infiltrated to 10 cm at 8.3 h and 33 cm after 83 h where the infiltration increased with time. Thus, the wetting range from 0 cm to 200 cm is suitable for modelling soil water infiltration for the subsurface flow since the soil water starts to fulfill the soil composition, which started at the depth of 125 cm onwards (Juma 1999). The precipitation that falls on dry soil will wet the layer of the soil from the present water content to the lower layers. In order to visualize the prototype of the depth information of the soil water flow, Tulip is used to show some of the interactive graphic visualizations of the soil water according to a predetermined depth (Fig. 5).

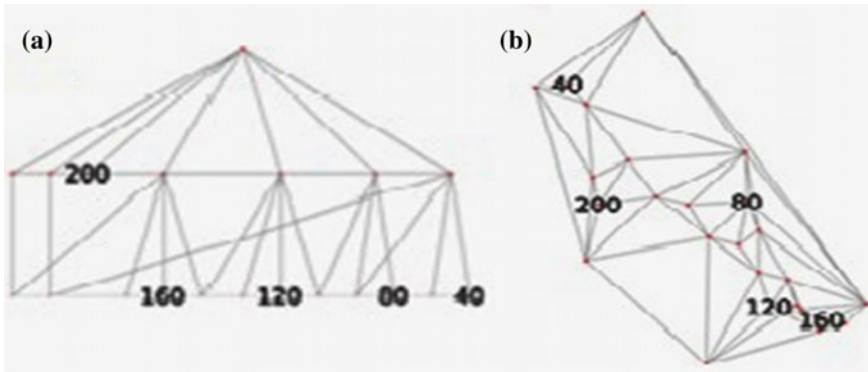


Fig. 5 Interactive information visualization by Tulips. **a** Hierarchy tree. **b** Delaunay triangle

As mentioned before, the Gosper Curve (Fig. 6a) is chosen to represent the pattern of the soil water flow. As mentioned by Asano et al. (1997), this curve divides the plane into hexagons where the obtuse angles lead to smoother boundaries and the curve has the locality of property, with $c = \sqrt{6.34}$. However, the linear arrangement that represents the pattern of the soil water flow (Fig. 6b) has changed slightly according to the Tree-map created before where the curve does not make any repetition upward to comply with the soil water that moves downward due to gravity. In addition, the curve is not represented in a planar form but vertically down to meet the requirement of the 3D modelling.

Figure 7a shows the position of nodes along the Gosper curve, where the image includes the soil surface, soil water flow direction, subsurface flow and depth which are represented by each node. Based on the arrangement of nodes, the flow pattern can be seen in Fig. 7b. Water from the precipitation that reaches node 1 is infiltrated into the soil and then flows to the next nodes, depending on the rate of precipitation. Low precipitation rate may result in the absorption of water not reaching 200 m and accumulating as subsurface flow in consequence of which there is water remaining

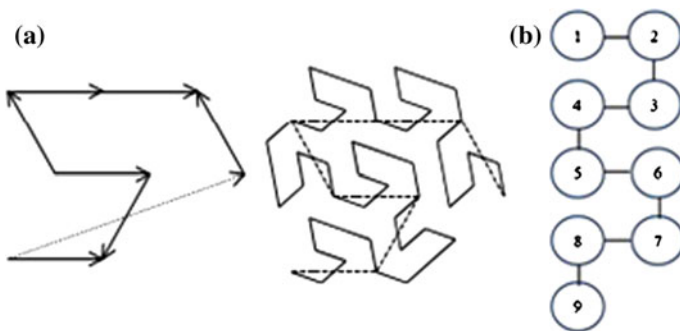


Fig. 6 **a** Gosper curve. **b** Linear arrangement of the leaves

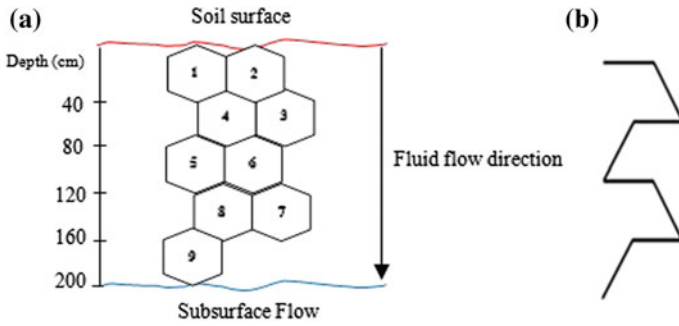


Fig. 7 a Position of nodes along Gosper's curve. b Soil water pattern

in the soil before succumbing to the downward trend. The rate of precipitation will not be the same with the volume of water that reaches the subsurface flow due to the soil characteristics that retain some of the water that flows through it. The number of nodes that are occupied by soil water is determined by calculating the soil wetting depth. In order to calculate the depth of wetting, it is necessary to calculate the storage capacity on a mass as well as on a volume basis (Juma 1999) shown as follows.

Storage capacity on a gravimetric basis (Θ_m)

$$= \text{Field capacity} - \text{present water content}$$

Storage capacity on volumetric basis (Θ)

$$= \Theta_m \times (\text{Bulk density of soil} / \text{density of water})$$

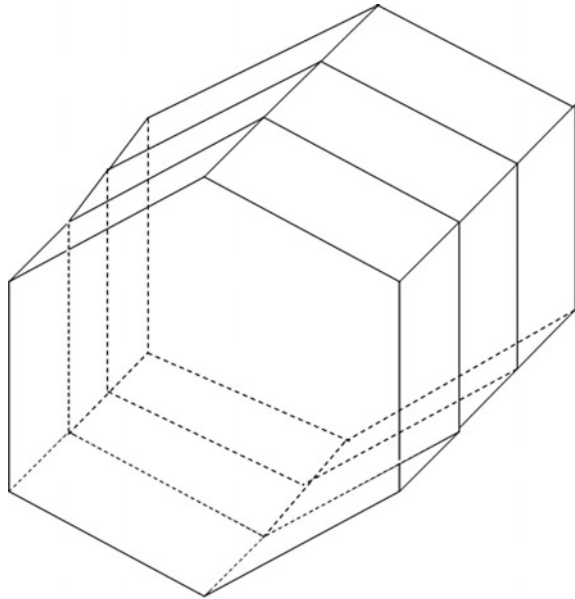
Depth of wetting

$$= \text{Volume of rain} / \text{volume of water stored per cm of soil}$$

Each and every Voronoi-shaped node illustrated in Fig. 8 is known as mask ($3 \times 3 \times 3$). The water infiltrated in the nodes does not enter and flow through the empty nodes but it flows through a series of TIN which produces the 3D TIN. However, this study uses the 3D TIN application in Voronoi polygon instead of using square.

The principle of the 3D Triangular Irregular Network (3D TIN) is adopted in order to create the 3D flow sequence of soil water in each 3D Gosper node. The mask measuring $3 \times 3 \times 3$ is numbered and it has 16 voxel elements (Fig. 9). The mask is divided into 4 sections of triangle, where each triangle is divided into three non-overlapping tetrahedra. Thus, every 3D Gosper node consists of 12 tetrahedra (Fig. 9a–d) that represent the flow sequence and the path of the soil water movement. As mentioned previously, there are 9 nodes in the 3D Gosper curve and the number of the tetrahedra for this model is 109. This non-overlapping tetrahedra

Fig. 8 A mask for each Voronoi-shape polygon nodes



form a 3D TIN that creates a network of flow directions that allows the water to pass through the nodes according to a predetermined sequence. This 3D TIN structure is designed for subsurface soil water flow.

In contrast of the use of the Gosper curve that lays out the hierarchical data and which is visualized in the 2D map, this study adopts the use of the Gosper curve to isolate layers of soil horizontally that forms a 3D pattern of dynamic soil water flow. In this model, the subsurface soil is represented by nodes that are arranged in sequence, from number 1 to 9. The nodes which are in a Voronoi-shaped polygon use the Voronoi tessellation to create a convex polygon (Auber et al. 2013). For the modelling part, the Voronoi-shaped polygon is used to retain water underneath nodes 1, 3, 5 and 7 from flowing down to the subsurface flow as a representation of water retained in the soil. The water starts its infiltration process when it reaches every first node of the 3D Gosper's curve and the water flow according to the sequential number of the nodes. The remaining water that stops moving is collected to calculate the overall volume of the subsurface flow. The rate of the infiltration of water in the porous media of soil is calculated using the Horton equation (Horton 1933):

$$f_t = f_c + (f_0 - f_c)e^{-kt}$$

Where:

- f_t infiltration rate at time t
- f_0 initial infiltration rate or maximum infiltration rate
- f_c constant or equilibrium infiltration rate after the soil has been saturated or minimum infiltration rate
- k decay constant specific to the soil

5 Concluding Remarks

This study aims to enliven the 3D research that has been burgeoning and the focus of the study is on the 3D modelling of soil water infiltration in porous media of soil. Environmental-related studies have started to show an improvement on the use of the GIS for the analyzing part. As we know, the process that happens naturally requires appropriate modelling methods as some of the processes sometimes lead to hazards that are life-threatening, damaging property and disrupting ecosystems. A proper and accurate modelling is needed and the most appropriate model that can handle the dynamic and temporal data is the 3D GIS geospatial modelling. The integration of the 3D GIS with a mathematical method can produce a better model for environmental studies. The tree-map is used to show the hierarchical depth of the soil modified from its original use that groups together similar information from a high number of data. Besides, by adopting the idea of the Gosper curve, the 3D pattern of the soil water movement was introduced to represent the flow route of the soil water infiltration. The application of 3D TIN in Voronoi-shaped Gosper nodes explains the route and sequence direction of the soil water flow in the model. The non-overlapping tetrahedral that forms a 3D TIN helps in representing the flow in multilayered soils. TIN that is used to connect three neighbouring (2D TINs) adopted a similar principle for the 3D TIN where in this study, the 3D TIN has been constructed to provide a sequence of soil water flow in the Voronoi-shaped polygon of 3D Gosper curve. 3D TIN served to give the coordinate for each node of the tetrahedral that represents the location of infiltration occurred. This model is beneficial for those who need to know the soil water content, soil water distribution, soil water movement and subsurface water that are required in their research. The 3D GIS is needed to facilitate the dynamic flow of water that does not simply flow linearly down the soil but also towards an unpredictable direction. The information of the water content at certain depth of the soil can also be predicted using this model that often changes over time. The spatial data model is needed to define the content and structure of the data that can be used to carry out the task and operations for the purpose of identifying soil water infiltration.

Acknowledgments The authors would like to thank UTM Research University Grant, Vote J.130000.2427.02G77 and MOHE-JICA Grant No. 203/PJAUH/6711279 for supporting this work.

References

- Alemaw, B. F., & Chaoka, T. R. (2003). A continental scale water balance model: A GIS-approach for Southern Africa. *Physics and Chemistry of the Earth*, 28(20–27), 957–966. <http://doi.org/10.1016/j.pce.2003.08.040>.
- Al-Sabhan, W., Mulligan, M., & Blackburn, G. a. (2003). A real-time hydrological model for flood prediction using GIS and the WWW. *Computers, Environment and Urban Systems*, 27(1), 9–32. doi:10.1016/S0198-9715(01)00010-2.

- Asano, T., Ranjanb, D., Roosc-, T., Welzld, E., & Widmayer, P. (1997). Space-filling curves and their use in the design of geometric data structures, 3975(96).
- Auber, D., Huet, C., Lambert, A., Renoust, B., Sallaberry, A., & Saulnier, A. (2013). GosperMap: using a gosper curve for laying out hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 19(11), 1820–1832. <http://doi.org/10.1109/TVCG.2013.91>.
- Barnes, C. J., & Allison, G. B. (1988). Tracing of water movement in the unsaturated zone using stable isotopes of hydrogen and oxygen. *Journal of Hydrology*, 100(1–3), 143–176. [http://doi.org/10.1016/0022-1694\(88\)90184-9](http://doi.org/10.1016/0022-1694(88)90184-9)
- Brito, M. G., Costa, C. N., Almeida, J. a., Vendas, D., & Verdial, P. H. (2006). Characterization of maximum infiltration areas using GIS tools. *Engineering Geology*, 85(1-2), 14–18. doi:10.1016/j.enggeo.2005.09.022.
- Dingman, S. L. (2002). Water in soils: Infiltration and redistribution. In *Physical hydrology* (pp. 222–242).
- Drogue, G., Humbert, J., Deraisime, J., Mahr, N., & Freslon, N. (2002). A statistical-topographic model using an omnidirectional parameterization of the relief for mapping orographic rainfall. *International Journal of Climatology*, 22(5), 599–613. <http://doi.org/10.1002/joc.671>
- Dunne, T., & Black, R. D. (1970). Partial area contributions to storm runoff in a small New England watershed. *Water Resources Research*, 6(5), 1296–1311. <http://doi.org/10.1029/WR006i005p01296>
- Fuhrmann, S. (2000). Designing a visualization system for hydrological data. *Computers and Geosciences*, 26(1), 11–19. [http://doi.org/10.1016/S0098-3004\(99\)00040-0](http://doi.org/10.1016/S0098-3004(99)00040-0)
- Horton, R. E. (1933). The role of infiltration in the hydrologic cycle. *Transactions of the American Geophysical Union*, 445–460. <http://doi.org/10.1029/TR014i001p00446>
- Hursh, C., & Brater, E. (1941). Separating storm-hydrographs from small drainage-areas into surface-and subsurface-flow. *Papers, Hydrology*, 863–871. Retrieved from <http://cwt33.ecology.uga.edu/publications/806.pdf>
- Hoover, M. D., & Hursh, C. R. (1943). Influence of topography and soil-depth on runoff from forest land. *Transactions of 1943 of the American Geophysical Union*, 1. <http://doi.org/10.1017/CBO9781107415324.004>
- Izham, M. Y., Uznir, U., & Rahman, A. A. (2010). Developments in 3D geo-information sciences. *Applied sciences*, vol. 219.
- Jain, M. K., Kothyari, U. C., & Ranga Raju, K. G. (2004). A GIS based distributed rainfall-runoff model. *Journal of Hydrology*, 299(1-2), 107–135. doi:10.1016/j.jhydrol.2004.04.024.
- Johnson, B., & Shneiderman, B. (1991). Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Proceeding Visualization '91*. <http://doi.org/10.1109/VISUAL.1991.175815>.
- Ju, W., Gao, P., Wang, J., Zhou, Y., & Zhang, X. (2010). Combining an ecological model with remote sensing and GIS techniques to monitor soil water content of croplands with a monsoon climate. *Agricultural Water Management*, 97(8), 1221–1231. doi:10.1016/j.agwat.2009.12.007.
- Juma, N. G. (1999). Introduction to soil science and soil resources. *The Pedosphere and Its Dynamics: A System Approach to Soil Science*, 1.
- Kamada, T., & Kawai, S. (1991). A general framework for visualizing abstract objects and relations. *ACM Transactions on Graphics*, 10(1), 1–39. doi:10.1145/99902.99903.
- Liu, B., Phillips, F., Hoines, S., Campbell, A. R., & Sharma, P. (1995). Water movement in desert soil traced by hydrogen and oxygen isotopes, chloride, and chlorine-36, southern Arizona. *Journal of Hydrology*, 168(1-4), 91–110. doi:10.1016/0022-1694(94)02646-S.
- MacEachren & Kraak (1997). The map reader: Theories of mapping practice and cartographic representation. *Computers & Geosciences*, 23(4), 335–343. doi:10.1016/S0098-3004(97)00018-6
- Mokbel, M. F., Aref, W. G., & Kamel, I. (2003). Analysis of multi-dimensional space-filling curves. *Geoinformatica*, 7(3), 179–209. doi:10.1023/A:1025196714293.
- Newman, B. D., Campbell, A. R., & Wilcox, B. P. (1997). Tracer-based studies of soil water movement in semi-arid forests of New Mexico, 196, 251–270.

- Nielsen, D. R. (1964). Water movement through panoche clay loam soil. *Journal of Agricultural Science*, 35(17), 491–503.
- Pimenta, M. T. (2000). Water balances using GIS. *Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere*, 25(7-8), 695–698. doi:10.1016/S1464-1909(00)00087-3.
- Portoghese, I., Uricchio, V., & Vurro, M. (2005). A GIS tool for hydrogeological water balance evaluation on a regional scale in semi-arid environments. *Computers and Geosciences*, 31(1), 15–27. doi:10.1016/j.cageo.2004.09.001.
- Raper J. F. & Kelk, B. (1991). Three-dimensional GIS. In D. J. Maguire, M. F. Goodchild & D. V. Thind (Eds.), *Geographic information systems: Principles and applications* (Vol. 1, pp. 299–316). Longman: London.
- Shen, D. Y., Takara, K., Tachikawa, Y., & Liu, Y. L. (2006). 3D simulation of soft geo-objects. *International Journal of Geographical Information Science*, 20(3), 261–271. <http://doi.org/10.1080/13658810500287149>
- Tan, C. H., Ku, C. Y., Chi, S. Y., Chen, Y. H., Fei, L. Y., Lee, J. F., et al. (2008). Assessment of regional rainfall-induced landslides using 3S-based hydro-geological model. In *Landslides and engineered slopes: From the past to the future* (Vols. 1, 2). Chinese Inst Soil Mech & Geotech Engr; Chinese Nat. <http://doi.org/10.1201/9780203885284-c225>
- Uznir, U., Anton, F., Suhaibah, A., Rahman, A. A., & Mioc, D. (2013). Improving 3D spatial queries search: Newfangled technique of space filling curves in 3D city modeling. *ISPRS Annals Photogrammetry, Remote Sensing and Spatial Information Sciences*, II, 319–327.
- Zollweg, J. A., Gburek, W. J., & Steenhuis, T. S. (1996). SMoRMod—A GIS-integrated rainfall-runoff model. *Transactions of the Asae*, 39(4), 1299–1307.

A Data Model for the Interactive Construction and Correction of 3D Building Geometry Based on Planar Half-Spaces

Martin Kada, Andreas Wichmann, Nina Manzke
and Yevgeniya Filippovska

Abstract 3D city models of large areas can only be efficiently (re-)constructed using automatic approaches. But since there is always a certain number of buildings where the automation fails, there is a need for interactive construction and correction tools. These tools should ideally use the reconstruction results as input, so that the amount of manual labor is minimized. However, automatic 3D building reconstruction approaches make use of different solid modeling techniques that are not all suitable for interactive modeling purposes. One such representation is half-space modeling that exhibits several advantages for the automatic (re-)construction of 3D building models (from segmented point clouds). Because planar half-spaces are infinite entities that are usually represented as mathematical inequality equations, it is difficult to design an interactive modeling system that allows their direct manipulation. In this paper, we propose an interactive modeling concept specifically for 3D building geometry based on a half-space kernel. Following from it, a special-purpose object-oriented data model is developed that hides the kernel under a layer of parameterized primitives and boundary representation (B-rep) that give semantic meaning to building elements and is thus better comprehensible to human users.

M. Kada (✉) · A. Wichmann · Y. Filippovska
Institute of Geodesy and Geoinformation Science, Technische Universität Berlin,
Berlin, Germany
e-mail: martin.kada@tu-berlin.de

A. Wichmann
e-mail: andreas.wichmann@tu-berlin.de

Y. Filippovska
e-mail: yevgeniya.filippovska@tu-berlin.de

N. Manzke
Institute for Geoinformatics and Remote Sensing, University of Osnabrück,
Osnabrück, Germany
e-mail: nina.manzke@uni-osnabrueck.de

1 Introduction

In the last few years, the use of 3D city models has increased in the everyday life of people. For example, virtual models became recently popular for the visualization of 3D maps and virtual globes in car and pedestrian navigation systems. There is also a great demand for city models in mapping offices and in the real estate sector. Furthermore, they are required for applications such as noise and solar potential analysis, flood and pollutant simulations, and location planning of mobile antennas. In general, 3D city models include buildings, digital images of the urban area, and often roads, vegetation, and street furniture. For the reconstruction of 3D building models, several automatic approaches have been developed over the last two decades. An overview is, e.g., given in (Haala and Kada 2010).

The quality of such automatically reconstructed buildings is sufficient for most visualization purposes. However, for critical analyses and simulations, which require more accurate models, fully automatically reconstructed models might not be sufficient and need to be manually corrected. This process is usually very time consuming especially for large towns and cities, so that efficient modeling tools are needed. Therefore, we present in this paper an intuitive and interactive 3D modeling framework, which helps to avoid typical errors in the final building models and which supports the user during the construction process.

For the modeling of buildings, there already exist several solid modeling techniques. Popular methods are, e.g., boundary representation (B-rep), cell decomposition, and binary space partitioning tree (BSP-tree). However, they are not very intuitive to use for interactive modeling purposes, because in order to consider certain dependencies the underlying data structure has to be changed several times during the modeling process. Additionally, the operations which can be applied to the solid do not ensure the topological correctness of a model (e.g. its closeness) and the retention of geometric constraints (e.g. parallelism, rectangularity). Especially for a real-time application this can be implemented only with difficulty.

Modeling approaches belonging to the so called family of constructive modeling (Mäntylä 1988) better fulfill these requirements. For example, a widely used modeling technique in computer aided design (CAD) is constructive solid geometry (CSG). In CSG, a solid is built from a small number of predefined simple primitives that can be modified by affine transformations and combined with Boolean set operations. Using CSG automatically solves a number of issues. Amongst others, the Boolean set operations guarantee topological correctness of the resulting solids if the used primitives are already topologically correct. Additionally, CSG implicitly preserves in the resulting solid the geometric constraints of the primitives. If required, a CSG solid can be automatically and unambiguously converted to a B-rep, e.g. by the topology preserving Euler operators (Baumgart 1975).

In our modeling framework we use half-space modeling, which is a simplified form of CSG modeling. Each half-space is represented by a façade or a roof face that can be interactively modified by a user. The additional effort, which is accompanied by the use of half-space modeling, is in our framework hidden from

the user and is nevertheless suitable for the interactive 3D modeling in real-time. This concept allows therefore on the one hand a time efficient construction of new building models. On the other hand, existing models, which are obtained by an automatic reconstruction approach from an airborne captured point cloud as described, e.g., in (Kada and Wichmann 2013), can be corrected manually in a post-processing step.

2 Related Work

For the representation of 3D solids several modeling techniques have been developed (Hoffmann and Vaněček 1991). All these representations have in common that they organize in different ways the same geometric and topological data in form of a data structure. Generally, these data structures are based on the boundary, on the spatial subdivision, or on the operations of volumetric primitives that construct the solid (Mortenson 1985). According to Mäntylä (1988), there are three major types of data model schemes: boundary representation (B-rep) models, decomposition models, and constructive models.

A solid in B-rep is composed of its topological and geometrical information. It is represented as a collection of connected surfaces that defines the boundary between the solid and its complementary exterior according to the Jordan-Brouwer theorem. For an efficient access to adjacent and incident elements special data structures like Winged-Edge (Baumgart 1974), Half-Edge (Weiler 1985), Quad-Edge (Guibas and Stolfi 1985) have been introduced. In general, B-rep can represent a wide class of objects, but they are usually complex and additional effort is needed to guarantee the topological correctness of a solid and to maintain its geometric constraints. Therefore, this representation is usually more suitable for visualization rather than for interactive modeling purposes. The B-rep for the automatic reconstruction of buildings is, e.g., used in (Ameri and Fritsch 2000).

In a decomposition model, a collection of adjoining, nonintersecting simple primitives comprises the solid. One of its most general form is the cell decomposition model that defines a solid by a set of primitive cells. Due to its approximate nature, it is rarely used for visualization purposes but more frequently in analyses, e.g. by applying the finite element method. Some examples for the reconstruction of buildings based on cell decomposition and the BSP-tree are given in (Kada and McKinley 2009; Sohn et al. 2008).

Constructive models represent a solid as a combination of simple geometric shapes. One of its most common form is constructive solid geometry (CSG). It describes a solid as a hierarchical Boolean combination of primitives in terms of a tree. Thereby, primitives are stored in the leaf nodes and regularized Boolean set operations or rigid motions in the internal nodes (Mäntylä 1988). The evaluation of the tree always results in valid solids (as long as the primitives are also valid). Examples of CSG models in the reconstruction process of 3D buildings are given in (Brenner 2004; Kada and Wichmann 2013; Xiong et al. 2015).

The described data models are also used in interactive modeling systems. For example, several interactive mesh editing techniques based on B-rep models have been developed (Sorkine 2005). These techniques are generally suitable for the intuitive modification of freeform surfaces, but are not practically applicable to regularized polyhedral building models which have to preserve many geometric constraints. Another shape-modeling technique based on the concept of Lagrangian surface flow is used in (Duan et al. 2005). A real-time interactive visual editing of shape grammars for procedural architecture is shown in (Lipp et al. 2008). An interactive 3D solid modeling system for generating 3D models of architectural structures and urban scenes from unordered sets of photographs is given in (Sinha et al. 2008). Other well-known modeling systems (e.g. Cinema 4D, 3ds Max, AutoCAD, SketchUp, etc.) maintain several representation schemes of a solid and support the efficient conversion between them. They are, however, designed for modeling general shapes and not specifically designed to efficiently construct 3D building models.

3 3D Modeling Using Half-Spaces

In this section, the fundamentals of half-space modeling are summarized. For a detailed description, see, e.g., (Mäntylä 1988) or (Foley et al. 1990).

Half-space modeling is a form of solid modeling and belongs to the family of constructive modeling. It defines a solid as a combination of half-spaces obtained by Boolean set operations. A half-space H of \mathbb{R}^3 is a point set of the form $H = \{(x, y, z) \mid g(x, y, z) \leq 0\}$ where $g: \mathbb{R}^3 \rightarrow \mathbb{R}$ is its characteristic function. The subset of points $S_H = \{(x, y, z) \mid g(x, y, z) = 0\}$ is called the surface of the half-space H . The complementary half-space H^C to H is the half-space $H^C = \{(x, y, z) \mid g(x, y, z) \geq 0\} = \{(x, y, z) \mid (-g)(x, y, z) \leq 0\}$. Obviously, the intersection of half-space H and its complementary half-space H^C is the surface S_H . In our modeling framework only half-spaces with a linear characteristic function g are considered. These convex half-spaces are called planar half-spaces and their points, called interior points, satisfy the inequality

$$g(x, y, z) = Ax + By + Cz + D \leq 0 \quad (1)$$

where (A, B, C) is the normal unit vector of the plane $\{(x, y, z) \mid Ax + By + Cz + D = 0\}$ with the distance $|D|$ to the origin.

Any polyhedral convex solid can be constructed by applying the Boolean set operation intersection to a set of planar half-spaces. Then, the solid points are just the interior points common to all of these half-spaces and each planar surface of the solid lies in the surface of a half-space. For example, a cuboid building can be represented as the intersection of six planar half-spaces. For the construction of a saddleback roof building only the half-space that limits the height of the flat roof has to be replaced by two sloped half-spaces as shown in Fig. 1.

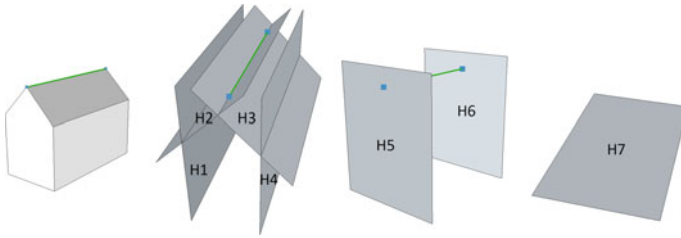


Fig. 1 A 3D building model with a saddleback roof defined by seven planar half-spaces

However, the construction of a non-convex solid is more difficult, because it cannot be modeled with the Boolean set operation intersection only. It therefore needs to be composed from convex components. This can be done since any non-convex solid can be decomposed into convex components. Then, the Boolean set operation union is applied to the set of convex components in order to construct the final non-convex solid. Therefore, half-space modeling is not restricted to certain geometric shape types.

Half-space modeling can conveniently be used in the automatic 3D building reconstruction from 3D point clouds, as buildings are commonly approximated by polyhedrons and there is a one-to-one correspondence between the segmented planar areas, the half-spaces, and the faces of the (convex) building blocks used to construct such models (see e.g. Kada and Wichmann 2013). Due to its mathematical formulation, the generated shapes are guaranteed to be topologically correct, although some care has to be taken to construct finite solids. There is also not as much interdependence in the definition of the half-space parameters as compared to the elements of a B-rep. For example, if a sloped half-space of a hip roof is changed, it has no effect on the other half-spaces, whereas changing the geometry of a face in B-rep results in further changes related to adjacent faces and incident edges and vertices. Also due to the use of Boolean set operations, geometric shapes can be generated by several loosely coupled components, e.g. by separating the model construction of the main building roof shape, the endings, the connections, and the outlines as described in Sect. 5. Such a flexibility is more complicated to implement using B-rep, because the faces, edges, and vertices have to be tediously calculated considering all shape combinations and parameter values.

4 Interactive Modeling Concept

We defined and formulated an interactive modeling concept that serves as a basis for the development of a data model that is suitable for our intended purposes. See Fig. 2 for a depiction of our concept.

The utilization of planar half-spaces in the (automatic) (re-)construction of 3D building models has several advantages. However, these come with the

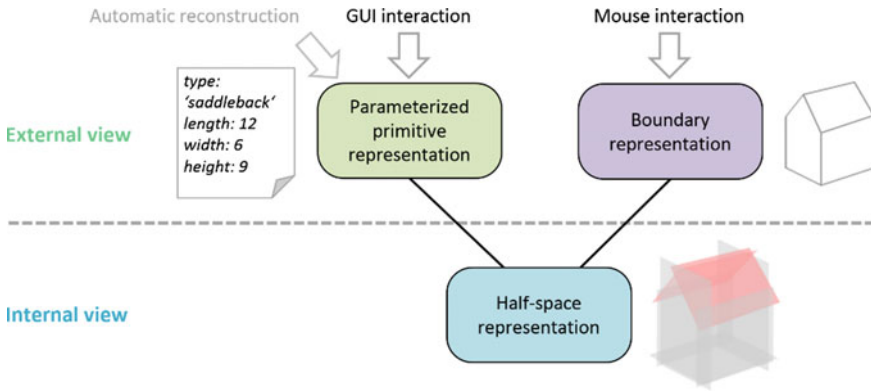


Fig. 2 The relation of the geometric representations within our interactive modeling concept

shortcoming that planar half-spaces are infinite entities represented as mathematical inequality equations. It is therefore very difficult to design an interactive modeling environment with a natural look and feel that allows their direct manipulation. So, the half-space entities as well as the entire half-space modeling functionality are in our concept encapsulated inside a kernel module and thus completely hidden from the user. This module is, on the one hand, as generic as possible to allow the construction of all common building shapes in existence and, on the other hand, it reflects our targeted interaction patterns and workflows to simplify and support the whole 3D building model construction process. Because of its invisibility towards the user, the kernel module can be regarded as the internal view of our data model.

On top of the kernel module, two further modules are specified that serve as the interface to the automatic 3D building reconstruction and to the user; thereby constituting the external view. They use different geometric representations, each one being best suited for their specific task, namely parameterized primitive representation and B-rep.

In the parameterized primitive representation module, the 3D building model construction process is presented to the reconstruction application and to the interactive user via a library of building blocks that feature common roof shapes. These primitives can be semantically parameterized, geometrically molded (up to a certain extent), freely placed in a building specific coordinate system, and combined with other primitives to form all possible 3D building shapes. In our context, we consider semantic parameters as being descriptive with regard to the building roof geometry and to be in most instances specific to certain building roof types (e.g. ridge length, eaves height, hip slope, etc.). Users can directly set and change these semantic parameters with common controls (buttons, spinners, list boxes, etc.) within a graphical user interface. From the primitive type information and the primitive parameter values, a unique half-space representation can be computed. Because a conversion is not practically feasible in the other direction without providing further semantic information, the parameter representation is internally

our basic representation for manipulations that is also used to store and exchange 3D building models via files.

Neither the parameterized primitives nor the half-space representation allows a graphical output that is needed for an interactive system. Real-time 3D visualization and rendering systems work (in their core) only with points, lines, and (triangulated) polygons. They cannot make use of models given as a set of parameters or mathematical half-space equations. For this reason, a B-rep is further needed that specifies a building model and its components as a collection of vertices, edges, and faces. The B-rep is not derived and constructed from the primitive parameters themselves, but by a conversion from the half-space representation to exploit the above mentioned advantages of half-space modeling. Besides rendering purposes, the graphical primitives also provide the means for human-machine interactions as they can be selected and moved in a 3D view with the help of a computer mouse. The modeling system then translates the user interactions to the internal parameterized primitive representation. In order to be able to identify the relevant parameters, the boundary elements are semantically linked to those half-spaces that they have been generated from. A face is associated with one half-space, an edge with two half-spaces, and a vertex with three or more half-spaces respectively. Most half-spaces have a meaning and their combination (in terms of intersections) form specific roof shape elements. For example, the intersection of the two sloped half-spaces of a saddleback roof form a special edge in the B-rep: the ridge line. If the ridge line is lowered or elevated, the system evaluates that it effects the ridge height of the primitive.

Automatic model-based 3D building reconstruction approaches as well as manual editors using a parameterized primitive representation usually offer a library of 3D primitives with rectangular footprints and standard roof shapes (saddleback, hip, mansard, etc.). Simple shaped buildings with no roof superstructures (dormers, chimneys, etc.) can then be constructed by a single primitive. The shape of most buildings are, however, more complex and need to be constructed by combining several primitives. Besides placing primitives and choosing roof shape parameters, the user also needs some means to determine the individual outline of each primitive.

Our proposed concept for an interactive 3D building model construction workflow consists of the following four steps that are repeated for each building primitive (cp. Fig. 3):

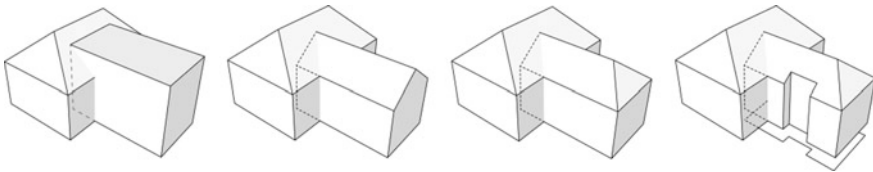


Fig. 3 The four steps of placing and defining the geometry of a parameterized building primitive in the described interactive 3D building modeling concept

1. A basic parameterized building primitive with cuboid shape is added to the 3D building model and selected as the current primitive. The position, orientation (with regard to the horizontal plane), and the extent (length, width, and height) of the primitive are set, preferably via mouse interactions.
2. The basic roof shape of the currently selected primitive is chosen from a list of available roof types (flat, shed, saddleback, hip, mansard, tip roof, etc.). Then the roof type specific parameters are set. For example, the position and the height of the ridge line of a saddleback roof as well as the two heights of the eaves lines.
3. For primitives of type ridge roof, the shape of the front and the back side is determined by choosing an ending type and setting ending specific parameters. Examples for common endings are gable, hip, broken hip, mansard hip, and Dutch gable. It should also be possible to interconnect primitives of this roof type with specific ending objects to form L- and T-connections.
4. For each primitive, a 2D polygon can be digitized that determines the detailed outline (and thereby the facades) of the primitive by cropping away those parts of the primitive that lie outside this polygon.

The user repeats these four steps until the constructed primitives reproduce the building according to the user's requirements. An example is given in Fig. 3.

5 Data Model for 3D Buildings

In this section, we describe our data model that combines half-space modeling and parameterized primitives to represent the 3D geometry of buildings. It is modeled in an object-oriented way and defines a hierarchy of classes (see Fig. 4). The half-space kernel serves as its foundation and describes the general structure of 3D building models. For a user oriented description of common roof shapes, a number of specialized classes for parameterized roof shapes, as well as endings and connections that are specific for ridge based roofs are further derived.

5.1 *Half-Space Kernel*

In our data model, every 3D building model consists of a collection of building components. These components are solids with flat side faces and common roof shapes. They are combined to form the basic shape of a building and also to model roof superstructures like dormers and chimneys. In terms of half-space modeling, a 3D building model is the Boolean union of n components:

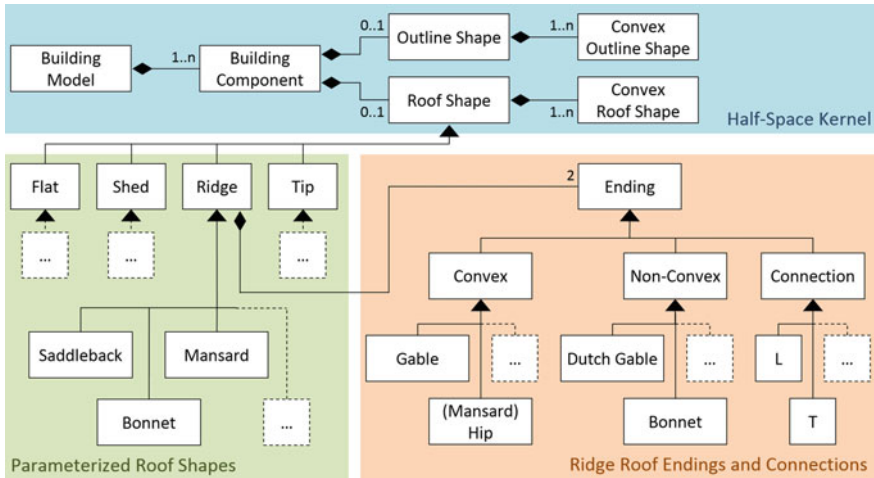


Fig. 4 The proposed data model for 3D building models

$$Building = \bigcup_{i=1..n} Component_i \tag{2}$$

Initially, each component has a cuboid shape with parameterized extent (length, width, height). They are each defined in their local coordinate system and have position and orientation parameters that allow their transformation into the superior coordinate system of the building. The use of local coordinate systems greatly simplifies the definition of the components' half-space parameters and their interactive placement and rotation. The half-space representation of a cuboid shape is then defined as the intersection of six axis-aligned planar half-spaces:

$$Cuboid_i = \bigcap_{j=1..6} H_{ij} \tag{3}$$

As components with cuboid shape are not sufficient to construct realistic building models, it is necessary to be able to further develop their shape with regard to their roof and outline. So a component may have a roof shape object and an outline shape object associated with it. The overall shape of a component is then defined as the Boolean intersection of the cuboid, the roof, and the outline shape:

$$Component_i = Cuboid_i \cap RoofShape_i \cap OutlineShape_i \tag{4}$$

Because the roof shape and the outline shape as well as the components cuboid shape should all be independently defined of one another, they solely consist of only those half-spaces that are actually required to shape out their specific part of the component. Therefore, roof shapes consist of mainly sloped and horizontal half-spaces, whereas outline shapes consist only of vertical half-spaces. Both shapes are not bounded in all directions, which makes them semi-infinite solids. If no

half-spaces are provided, roof shapes and outline shapes are regarded as infinite sets. Only by their intersection with the cuboid, a finite solid component is generated.

As mentioned above, non-convex shapes are constructed as a Boolean union of convex shapes. So each roof shape and each outline shape is defined as a collection of convex roof shapes and a collection of convex outline shapes respectively:

$$RoofShape_i = \bigcup_{k=1\dots n} ConvexRoofShape_{ik} \tag{5}$$

$$OutlineShape_i = \bigcup_{l=1\dots n} ConvexOutlineShape_{il} \tag{6}$$

Convex roof shapes and convex outline shapes are, again, described by the intersection of half-spaces, analogous to the definition of the cuboid shape. A saddleback roof shape, e.g., is then defined by one convex roof shape that contains two sloped half-spaces. A bonnet roof shape, which has a non-convex shape, is defined by two overlaid convex roof shapes that each contains two sloped half-spaces (see e.g. Fig. 5).

The outline shape can be regarded to only effect the component in 2D. It crops away surplus regions of a component that are outside the outline shape like a cookie cutter. In fact, during the interactive model construction the outline is inputted by the user as a 2D polygon. However, since the outline shape is defined for a 3D model, its half-space representation must also be given in 3D. It consists of strictly vertical half-spaces and is therefore not bounded in the upward and downward direction. Non-convex outlines are given as the union of convex outline shapes. An example for an L-shaped outline is given in Fig. 6.

In order to maintain meaningful parameters, roof shapes should completely fill out the footprints of cuboids. Otherwise, e.g. when the slope of a half-space from a saddleback roof is too steep and portions of the rectangular footprint of the

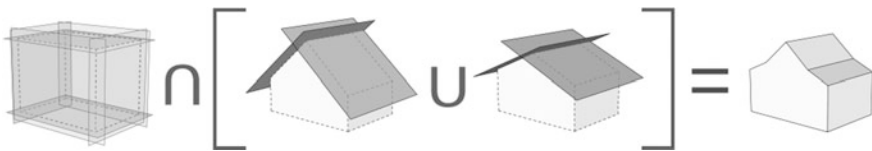


Fig. 5 The construction of a building component with bonnet roof shape using half-spaces

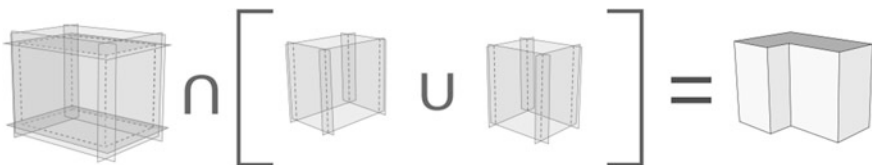


Fig. 6 The construction of an L-shaped building outline using half-spaces

component gets clipped away, then the shortened geometry of the component does not fit its semantic extent parameters anymore.

5.2 *Parameterized Roof Shapes*

As described in the interactive modeling concept, the parameterized roof shape module hides the half-space representation from the user. So from the roof shape class of the half-space kernel module, four base classes are derived that assign every roof shape to the types flat, shed, ridge, and tip roof. From these base classes further specialization classes are derived. Their main purpose is to semantically restrict the types of roof shapes that can be constructed and to introduce meaningful roof shape parameters. As the extent parameters of a component define the limiting geometry of each specialized roof shape, they are incorporated in the parameters of the derived specialization classes. For example, the height parameter of the component represents the maximum height of each roof shape and is identical to the ridge or tip height of a saddleback roof or tower roof respectively. From the roof shape parameters, the roof shape objects are constructed and the half-spaces are derived according to the half-space kernel.

The ridge roof shape is of particularly interest here, because building components with this roof type are often interconnected and their modeling is therefore interdependent. From the ridge roof shape class, all specialization classes for roof shapes that feature a ridge are derived. Ridge roof shapes may have two ending objects associated with them. These are used to further define the two sides of the roof shape where the ridge line ends or where two (or more) components with ridge roof shapes interconnect. This is explained in more detail in the next subsection.

Other specialized roof shape classes with flat, shed, and tip roof type can be defined accordingly. Building components with specialized roof shapes can not only be used as basic building blocks for the 3D building shape itself, but also for the construction of dormers, chimneys, towers and other roof elements.

5.3 *Ridge Roof Endings and Connections*

Each object of type ridge roof shape has two ridge roof ending objects associated with it that allow to further form the front and back side of this particular roof shape class. Without ending objects, ridge roof shapes are only delimited by the two vertical half-spaces from the basic cuboid component; resulting in two gables. The ending concept allows to support further ridge roof ending shapes that are semantically distinguishable and that provide their own geometric embodiment. The two ending objects E_1 and E_2 define, on the one hand, two sets of half-spaces E_1H and E_2H that form convex shapes and that can therefore be directly intersected with the roof shape. In order to construct non-convex ending shapes for both

endings, the ending objects may, on the other hand, also define two sets of convex solids E_1S and E_2S that must be combined with the roof shape using the Boolean union operation. This results in an extended definition of the roof shape formula:

$$RoofShape_i = \left(\left(\bigcup_{k=1..n} ConvexRoofShape_{ik} \right) \cap E_1H \cap E_2H \right) \cup E_1S \cup E_2S \quad (7)$$

Besides the base ending class, the data model defines three intermediate classes to distinguish between convex, non-convex, and connection endings. Further specialized ending classes are derived from these classes according to their geometric characteristic and purpose.

Ending objects of type convex are rather simple to define and to implement because they only specify a set of half-spaces that form a convex shape, but not a convex solid. Prominent examples of convex endings are gable, hip, and mansard hip. Because the geometry of a gable ending is already fully defined by the cuboid component in combination with a ridge roof shape, the gable ending class defines no further geometry and can be regarded as the standard ridge roof ending. The hip ending class defines one sloped half-space that is intersected with the ridge roof shape of the component. Fig. 7 depicts a building component with a saddleback roof shape that is associated with a hip ending. The advantage of half-space modeling becomes apparent in this example, because the definition of the hip ending is independent of the component's roof shape. The hip ending object could be associated, e.g., with a barrel or bonnet roof to have the same effect as on the saddleback roof. Other convex ending classes are defined analogously and may provide further half-spaces to form more complex ending shapes.

The intersection of ridge roof shapes with convex ending shapes is not expressive enough to define all kinds of standard roof shape endings. For example, non-convex endings like the Dutch gable roof cannot be constructed in this way. But the intersection with a vertical half-space, e.g., creates the necessary space within the cuboid building component that is required to construct the non-convex part of this particular roof (cp. Fig. 8). The missing front part is then constructed as a convex solid that is adjoined to the roof shape object by a Boolean union operation. Non-convex roof endings can define any number of convex parts to form any roof ending shapes in existence.

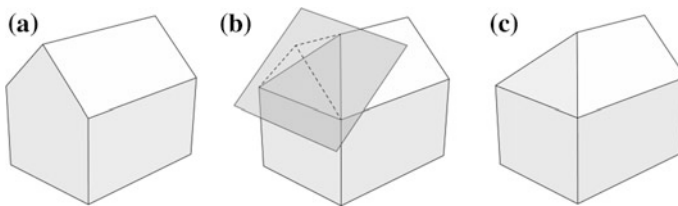


Fig. 7 A building component with a saddleback roof shape (a) clipped by the half-space of a hip ending (b) resulting in a building component with an asymmetric hip roof shape (c)

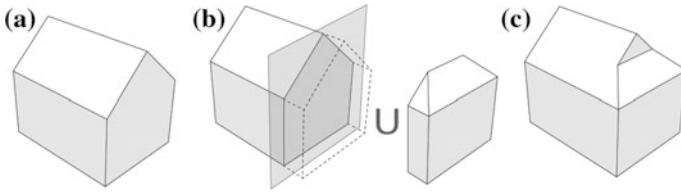


Fig. 8 A building component with a saddleback roof shape (a) clipped by the vertical half-space of a Dutch gable ending makes room for the front solid (that is attached to the component by a Boolean union operation) (b) resulting in a building component with an asymmetric Dutch gable roof shape (c)

For complex building shapes, it is often not sufficient to model building components independently from one another. The components rather need to be interconnected to form common shapes like L-, T- and X-connections. Endings of the derived type connection are a convenient way to accomplish this for components with a ridge roof shape.

Connection objects are special in regard to how they provide their half-spaces, because they reuse some of the half-spaces of one component for the refinement of the shape of the other component and vice versa. So the connection objects facilitate an information exchange in the form of half-spaces to simplify the model construction process. For example, to join two components with saddleback roof shapes in an L-connection, the connection adds the sloped half-space from the right component to the half-spaces of the left component to clip the overhanging saddleback roof and vice versa (see Fig. 9). It also exchanges the façade half-spaces in the same way to clip overhanging component parts in cases where the components form an acute angle.

The advantage of reusing half-spaces is not just the reduction of their overall number, but also that one component is automatically adapted if the other component changes. For example, if the ridge height of one component changes, then only the half-space parameters of the same component need to be updated; the half-space parameters of the other component remain unaffected.

Although we have only introduced two connection types (see Fig. 4), further types like the X-connection might be useful. Some shapes like the U-connection, however, are not necessary as they can be modeled by several L-connections.

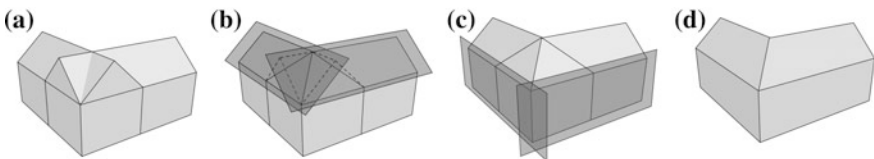


Fig. 9 Two building components with saddleback roof shape (a) that are clipped by the sloped roof shape half-spaces of the other component (b) as well as by the façade half-spaces (c) resulting in an L-shaped building model with ridge roof shape (d)

6 Implementation and Discussion

We have implemented the described data model and the prototype of an interactive system for constructing and correcting 3D building models. It is written in C++, uses Qt and OpenGL for the GUI and the 3D rendering part as well as the commercial CAD kernel 3D ACIS Modeler from Spatial for the Boolean evaluation of half-spaces and their conversion to a B-rep.

In an interactive 3D modeling system, the real-time behavior is always of major concern. As the data model uses three representations for one and the same geometric entity, there is a continuous conversion process happening in the application's background, especially during the actual user interactions. Because the user typically modifies the geometry of only one building component at the time, the evaluation of its half-space representation and its conversion to B-rep is fast enough for real-time rendering purposes. In our implementation, an instantaneous feedback could still be observed on a standard desktop computer system even during the simultaneous interaction with about ten to fifteen components. To reduce the number of geometric computations, the rendering system could also be utilized, e.g. to perform the affine transformations during the interactive placement and rotation of component groups.

We have also implemented a number of validity checks for parameter values in our interactive environment that are executed in the background during user interactions to prohibit the construction of invalid building components. These are necessary to avoid, e.g., degenerated shapes.

Practical experiments with the systems showed that it satisfies its intended purpose and that the half-space modeling kernel could successfully be hidden from the user without restricting any functionality.

7 Conclusion and Future Work

In this paper, we have presented an interactive modeling concept for the construction and correction of 3D building models in cases where automatic methods fail. Here, it is assumed that the automatic reconstruction approaches generate solids using half-space modeling, which is not suited for manual editing. Therefore, a data model has been proposed with a half-space kernel that is hidden by a layer of parameterized primitives and B-rep. We have shown how half-space modeling can be used to construct the 3D building geometry by loosely coupled components like roof shapes, endings, and connections.

We conducted experiments to restrict the movement of B-rep elements in the 3D view. Hard coding restrictions into the application is, however, tedious, time-consuming, and not very flexible with respect to adding further roof and ending shapes. Therefore an automatic generation of restrictions for faces, edges,

and vertices of the B-rep is envisioned that results from the combination of basic restrictions on half-spaces. For example, the restrictions of an edge could be derived from basic restrictions of the two half-spaces it was generated from.

Acknowledgments The research work presented in this paper is supported by the German Research Foundation (Deutsche Forschungsgemeinschaft) under grand number KA 4027/1-1.

References

- Ameri, B., Fritsch, D. (2000). Automatic 3D building reconstruction using plane-roof structures. In *Proceedings Of The American Society Of Photogrammetry And Remote Sensing Conference*, Washington, D.C.
- Baumgart, B. G. (1974). *Geometric Modeling for Computer Vision*, Ph.D. thesis, Stanford University.
- Baumgart, B. G. (1975). A polyhedron representation for computer vision. In *Proceedings of the AFIPS conference*, (vol. 44, pp. 589–596).
- Brenner, C. (2004). Modelling 3D objects using weak CSG primitives. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Science*, 35(3/B3), 1085–1090.
- Duan, Y., Hua, J., & Qin, H. (2005). Interactive shape modeling using lagrangian surface flow. *The Visual Computer*, 21(5), 279–288.
- Foley, J., van Dam, A., Feiner, S., & Hughes, J. (1990). *Computer Graphics: Principles and Practice* (2nd ed.). Reading, Massachusetts: Addison-Wesley.
- Guibas, L., & Stolfi, J. (1985). Primitives for the manipulation of general subdivisions and the computation of voronoi diagrams. *Transactions on Graphics*, 4(2), 74–123.
- Haala, N., & Kada, M. (2010). An update on automatic 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6), 570–580.
- Hoffmann, C. M., & Vaněček, G. (1991). Fundamental techniques for geometric and solid modeling. In C. T. Leondes (Ed.), *Advances in Control and Dynamics* (pp. 101–165). : Academic Press.
- Kada, M., & McKinley, L. (2009). 3D building reconstruction from lidar based on a cell decomposition approach. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Science*, 38(3/W4), 47–52.
- Kada, M., & Wichmann, A. (2013). Feature-driven 3D building modeling using planar halfspaces. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3, 189–196.
- Lipp, M., Wonka, P., & Wimmer, M. (2008). Interactive visual editing of grammars for procedural architecture. *ACM Transactions on Graphics*, 27(3), 1–10.
- Mäntylä, M. (1988). *An Introduction to Solid Modeling*. Rockville, Maryland: Computer Science Press.
- Mortenson, M. (1985). *Geometric Modeling*. New York: John Wiley & Sons.
- Sinha Sudipta, N., Steedly, D., Szeliski, R., Agrawala, M., & Pollefeys, M. (2008). Interactive 3D architectural modeling from unordered photo collections. *ACM Transactions on Graphics*, 27(5), 1–10.
- Sohn, G., Huang, X., & Tao, V. (2008). Using a binary space partitioning tree for reconstructing polyhedral building models from airborne LIDAR data. *Photogrammetric Engineering and Remote Sensing*, 74(11), 1425–1438.
- Sorkine, O. (2005). Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (pp. 175–184).

- Weiler, K. (1985). Edge-based data structures for solid modeling. *Curved-Surface Environments, IEEE Computer Graphics & Applications*, 5(1), 21–40.
- Xiong, B., Jancosek, M., Oude Elberink, S., & Vosselman, G. (2015). flexible building primitives for 3D building modeling. *ISPRS Journal of Photogrammetry and Remote Sensing*, 101, 275–290.

The Potential of the 3D Dual Half-Edge (DHE) Data Structure for Integrated 2D-Space and Scale Modelling: A Review

Hairi Karim, Alias Abdul Rahman, Pawel Boguslawski,
Martijn Meijers and Peter van Oosterom

Abstract Scaling factor is one of the most crucial aspect in 2D and 3D models especially in computer graphics, CAD, GIS, and games. Different user or/and application need different scale models during various stages of the use of data, including visualization and interaction. There are some arisen issues on 3D data model especially to meet GIS requirements while minimize the redundancy of the datasets. In GIS modelling, various data structures and data models have been proposed to support variety of applications and dimensionalities, but only a few in scale dimension. Some of them have succeeded in modelling scale such as in Space-Scale Cube (SSC) model. The recently implemented Dual Half-Edge (DHE) data structure within the PostgreSQL database is suitable for any valid 3D spatial model; not yet being explored for other dimensional such as scale environment. Using the same vario-scale approach, the DHE data model is also capable to implement a variable Level of Detail (LoD) representation such as SSC model. Some advantages of the DHE are described in this paper such as the dynamic property (valid updates based on Euler operations) and topology approach in

H. Karim (✉) · A. Abdul Rahman (✉)

3D GIS Research Laboratory, Faculty of Geoinformation and Real Estate,
Department of Geoinformatic, Universiti Teknologi Malaysia Johor Bahru,
Johor Bahru, Johor, Malaysia
e-mail: wnhairigis@gmail.com

A. Abdul Rahman
e-mail: alias@utm.my

P. Boguslawski (✉)

FET - Architecture and the Built Environment, University of the West of England,
Bristol, UK
e-mail: pawel.boguslawski@uwe.ac.uk

M. Meijers (✉) · P. van Oosterom (✉)

Department of GIS Technology, OTB Research Institute for the Building Environment,
Delft University of Technology, Jaffalaan 9, 2628 BX Delft, The Netherlands
e-mail: B.M.Meijers@tudelft.nl

P. van Oosterom
e-mail: P.J.M.vanOosterom@tudelft.nl

comparison with other existing data structures. The last section of this paper describes capability of the DHE data structure to provide a better platform for GIS integrated space-scale data model.

Keywords Scale dimension • Data structures • Spatial models • Level of details

1 Introduction

Traditionally, spatial data is usually presented in two-dimensional (2D) environment such as a printed map. But, with current technology and new research knowledge, three-dimensional (3D) Geographic Information System (GIS) is becoming a norm in modelling software.

As far as 3D modelling is concerned, commercial and open source software offers functionalities to model and represent a model in 3D environment. For instance, the latest version of Blender, Google Sketchup, FreeCAD, Paraview, Transmagic and others provide a lot of modelling and visualization functions, especially for animations, games and building models with a focus on geometry and graphic visualization. As a matter of fact, these general purpose 3D modelling techniques are not really suitable for 3D GIS spatial modelling since it does not fulfil some basic data validation requirements especially on topology and thematic semantic information. For example, a 3D solid must be completely closed, otherwise it is not possible to compute its volume (and many other operations). Thus, many basic and complex functions or analysis available in 2D GIS can be hardly implemented in the 3D GIS environment.

There are three basic requirements that need to be preserved in order to provide basic and complex analysis in n -dimensional ($n > 1$) spatial model representation. They are geometry, topology and thematic semantic integration within the same model. A particular data model should be able to provide these three key components in order to widen analysis functions in the 3D spatial model.

However, these user or application-oriented 3D spatial data models are only suitable for and satisfy a specific demand as reported by Liangchen et al. (2008). In order to fulfill these demands, many available data models had been designed and implemented. A data model can be defined as a method or a logical or mathematical way of visualizing and representing data (informational needs) in an information system. From GIS point of view, a spatial data model is needed to simplify sketch the real world and a pursue disperse model in geographical way of representation within a computer (Liangchen et al. 2008).

There are some drawbacks highlighted by Liangchen et al. (2008) from this application-oriented data model: (1) very hard to meet different situation demand and multiple coupling area requirements; (2) lack of mathematical completeness, redundancy and lack of topological relationships.

In general, 3D data models can be divided into four main groups: 3D geometric models, 3D topological and graph models, 3D city models and 3D CAD models (Lee and Zlatanova 2008; Boguslawski 2011). The spatial models are used in

accordance with specific application requirements, for example using Constructive Solid Geometry (CSG) or Boundary Representation (B-Rep) in 3D environment of CAD system (Boguslawski 2011).

A spatial data model can be represented by different data structures (Ledoux 2006). For instance, tetrahedral meshes in 3D or triangular meshes in 2D can be represented with radial-edges, half-edges or other data structures (Boguslawski 2011). A data structure can be defined as a specialized format or approach for organizing and storing digital geographic dataset in any data model. It refers to the problem on how to systematically encapsulate the geometry, topology and thematic semantic information of GIS dataset.

The capability of the selected data structure to support the required geometrical or/and topological analysis will increase the performance of the data model and quality of the data. For example, Tse and Gold (2004) combine the quad-edge data structure and the boundary representation to implement the extended TIN for supporting holes and caves.

The next section explains GIS data structures with special focuses on: topological connections and type of data model (kinetic and dynamic). Section 3 describes the scale dimension in GIS and discussion on scale implementation approaches. Section 4 introduces the Dual Half-Edge (DHE) as one of the potential data structures in scale dimension representation. A summary and future plan for the 3D-scale DHE data model implementation is provided in the last section of this paper.

2 GIS Data Structure

Nowadays, GIS is an important tool to help professionals in geography, science and engineering fields, especially for decision support system. Early GIS systems focused on the automating map making process and they provided simple analysis of two dimensional (2D) spatial data. Modern systems incorporate the capability of software, hardware and knowledge to perform GIS complex spatial analysis as well as the visualization in 3D spatial model. However, GIS is still relatively poor as far as the 3D is concerned; with most systems barely support 3D storage, analysis (functionalities) and visualization.

2.1 Existing 3D Data Structure

The CAD systems allow the model construction with a set of individual polyhedra, which are not topologically connected. Such connections are calculated from geometry each time they are required (Lee and Zlatanova 2008). Vertices, edges, faces and the relationships among them are defined within the solid boundary of B-Rep. Euler Operators are used to modify the geometry in CAD systems. There are various data structures used in B-Reps for solid objects (3D, some example in Fig. 1) such as:

1. Winged-edge of (Weiler 1988)
2. Half-edge of (Mäntylä 1988)
3. Quad-edge (Guibas and Stolfi 1985)
4. Doubly-Connected Edge List (DCEL)
5. Radial Edge (Weiler 1988)

2.2 Kinetic and Dynamic Data Model

According to Banks et al. (2009), there are two types of models; physical and mathematical models. As for GIS implementation, researchers such as Boguslawski (2011) prefer to use mathematical model since it can be easily analysed on a computer and effectively compared with a physical model. Physical models are still used in other disciplines: e.g. ‘maquettes’ in architecture and urban spatial planning or a scaled down version of a coastal area or river in lab basin for water management research. Further classifications of the mathematical (digital) models are: static or dynamic, discrete or continuous, deterministic or stochastic model (Banks et al. 2009; Boguslawski 2011).

For describing changes of the model, Gold (2005) divided data structures into two categories: dynamic and kinetic. A dynamic data structure refers to the capability to adapt any changes of the model after the batch construction phase. Gold (2005) defined a “dynamic” or updateable locally data structure if it offers the local insertion, deletion, movement and navigation in the model. This in contrast with some other models that are based on global criterion’s during an update (which is practically not feasible for large data sets, with frequent changes).

On the other hand, the kinetic data model refers to static properties of the data structure such as lack of support for local modifications and non-dynamic user-end applications. It is also pre described as a static viewpoint or a set of objects and relationships, inability to make changes or interactive queries and perspective mood such as colour kontras (Gold et al. 2004). For example, kinetic Voronoi diagram is

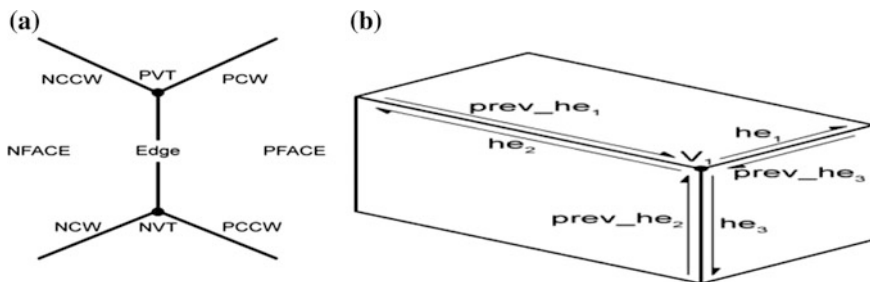


Fig. 1 Example of B-Rep data structure, **a** Winged-edge, **b** half-edge (Boguslawski 2011)

used in ship navigation (Gold et al. 2004) and Free-Lagrange for fluid flow simulation (Mostafavi and Gold 2004).

3 Scale Dimension

The research trend on scale dimension or multi-scale dataset in 3D models is observed not only in GIS field but also in fields such as 3D visualization of multi-scale geological models (Jones et al. 2009), Scale-Space Theory in computer vision (Lindeberg 1994), and cross scale dynamic and social ecology (Vervoort et al. 2012). Due to limitation of up scaling in geology, new multi-scale methods that incorporate fine-scale information into a coarse-scale equations has started to gain popularity (Aarnes et al. 2007).

In GIS, a multidimensional model may consist of one, two, three or more spatial dimensions that represent spatial objects (Gold 2005). It is important to distinguish the dimension of the embedding space and the dimension of the used primitives; e.g. a 0D point feature may be represented in 3D space. However, there are many approaches in defining higher dimensions in the model. The 2D geometry, that represents spatial dimensions, is accompanied by the third dimension to represent other non-spatial dimensions such as scale or time (Oosterom 2005). Gold (2005) suggest that 3D model may consist of a primitive (2D) map and unconnected data of points or objects. As far as the scale is concerned, the unconnected data can be understood as the reflection of viewpoint or observation of spatial objects (Zhou and Jones 2003).

Some researches attempted to add non-spatial dimension to the existing 3D spatial dimension (Worboys 1994; Raper 2000; Peuquet 2001; Ohori et al. 2013). Peuquet (2002) and Worboys (1994) focused on the time dimension, while Oosterom (2005; Li 1994) on the scale dimension. The implementation of the scale dimension faces many problems mostly due to limitation of the available data structures and models for three and higher dimension. Only a few models such as Multi-Scale Line tree (Jones and Abraham 1986), Arc-tree (Günther 1988), and Binary Line Generalization (BLG) (Oosterom 1990) integrate the geometric and scale aspects in one representation.

Since different applications and users need a specific fineness (number of details) of data representation, Sester (2007b) suggested that different representations or different Level of Details (LoDs) of the same reality have to be made available (with or without explicit relationships between corresponding features at the adjacent LoDs). Thus, there is a need to combine all level of detail into a single container called scale dimension.

Scaling dimension in GIS has gained some popularity in recent years due to the demand from users and applications. GIS research community is now moving forward to integrate a highly formal definition of geo-data (Oosterom and Stoter 2010)

and thus focuses on designing the most efficient framework and implementation for scale and temporal dimension. Important functionality of these space-scale models is efficient zooming and progressive data transfer between server and client (in environment that avoids redundancy and inconsistency as much as possible). Currently, scale dimension can be categorized into two main approaches: multi-scale and vario-scale.

3.1 *Multi-scale Approach*

In urban modelling, a multi-scale concept is an important issue because different applications require different abstractions of building models (Sester 2007b) as well as city objects (represented as 3D), which can be visualized and analysed in multiple scales (El-Mekawy 2010). Typical frameworks for multi-scale integration are either storing individual LoD dataset in separate databases or using generalization techniques. Most of the national mapping agencies use the 2D digital data in different groups or databases such as 1:20,000 and 1:50,000 topographic maps, which can be used by a single application (e.g. web-mapping portal).

Multi-scale GIS spatial datasets may be maintained in two ways: separately maintained different databases with predefined scale-steps (Fig. 2) or/and maintaining only the most detailed data which can be automatically generalized into small scale data on the fly transition (Oosterom and Stoter 2012), such in Fig. 3.

The main drawback of the first maintenance method is data redundancy. Storage of the same dataset in different scale ratio (pre-defined datasets) and different databases will introduce data redundancy in terms of geometry and attributes. Thus, it will consume a lot of storage capacity as compared to use the second method; a single dataset (most detailed) storage and undergo any generalization process for coarse details. Furthermore, it also will cause a lot of upcoming problems such as difficulty in updating process (need more works to update certain area of the data for every pre-defined scale), introducing errors if it is not handled properly and problems with the geometry and topology consistency.

The main drawback of the second approach, on the fly generalization, is that this is a very hard and computation intensive. It may not be feasible to use this approach with sufficient cartographic quality in an interactive setting especially with many users of the service.

While in 3D spatial modelling, a particular generalization technique (involve aggregation, simplification and others) or a set of pre-defined LoDs (e.g. CityGML); will be selected to represent as the scale dimension. The main drawback of these techniques is the 3D objects are not topologically connected or connected with a very minimal connectivity.

A 3D spatial multi-scale concept, such as CityGML, introduces two kinds of possible transitions: generalization and progressive transmission as shown in Fig. 4. Multi-scale generalization can be described as a process of transmigration of a spatial object from a detailed level to the less detailed level within the same

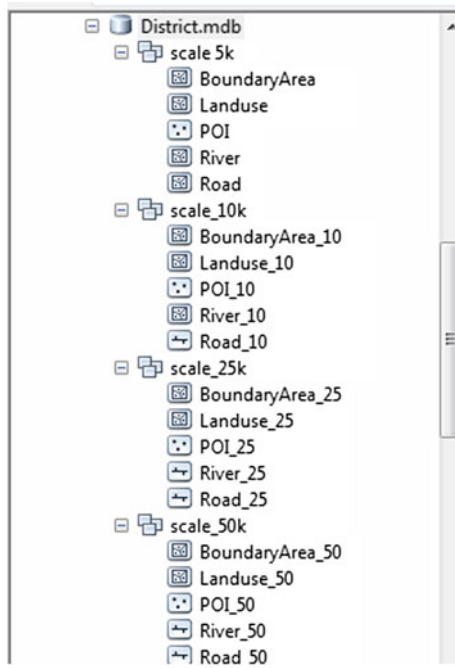


Fig. 2 2D redundancy problem in the multi-scale approach, which stores the same object in separate LOD datasets

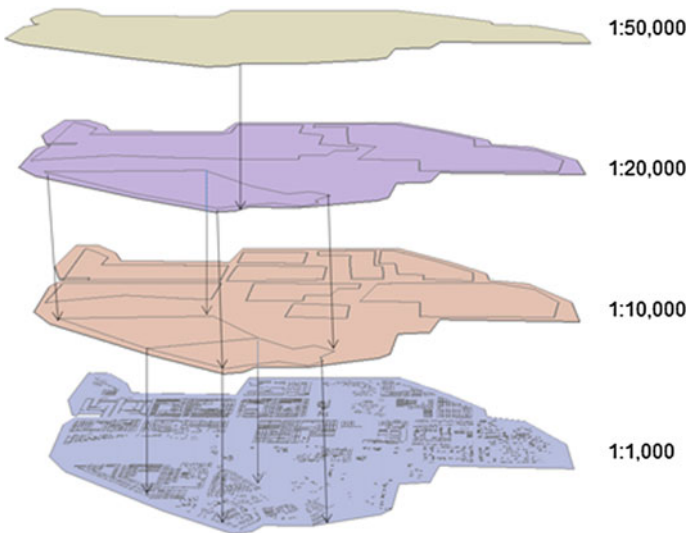


Fig. 3 Illustration of a typical multi-scale representation using generalization technique



Fig. 4 Multi-scale concept in CityGML (Sester 2007a)

application or model. While, a progressive transmission is the inverse process for generalization; the use of generalization chain (history; maximum and minimum elementary generalization operations) to revert the previous generalization levels. Many multi-scale generalization approaches has been implemented for 3D city models. Most of the researchers attempted different generalization methods in order to produce less-detailed LoDs by hiding less important details and reducing the data storage volume for efficient 3D analysis and visualization (Baig et al. 2011). It should be noted that these LoD representations are independent (no explicit relationships between features at different LoDs).

3.2 Vario-Scale Approach

In variable-scale approach, the basis is formed by a data structure in which each area of the map can be represented by a topological face (Meijers 2011). In principle, any topological data structure, such as Winged-Edge can be used. The true vario-scale structure, Space-Scale Cube (SSC), is the extension from the topological Generalized Area Partitioning (tGAP), which uses the hierarchical tree structure. The vario-LoD is defined as an additional dimension (third dimension) as for a series of maps at a range of scales; horizontal slice planes can be used. Similar to tGAP terminology, the third dimension is represented by the concept of “importance” (are diver for scale, LoD content) (Meijers 2011).

The importance of the objects is highly depending on the feature classification and the size of the object. For instance, a small city is more important than a large area of forest; a large residential area is more important than a small water body. The new importance value is calculated after the generalization (merging) from two predecessor objects as illustrated in Figs. 5 and 6.

The tGAP model helps to overcome the data redundancy problem since it was constructed using the most detailed dataset to the most coarse one. It also preserves the topology of the horizontal plane and provides the LoDs within the scale axis.

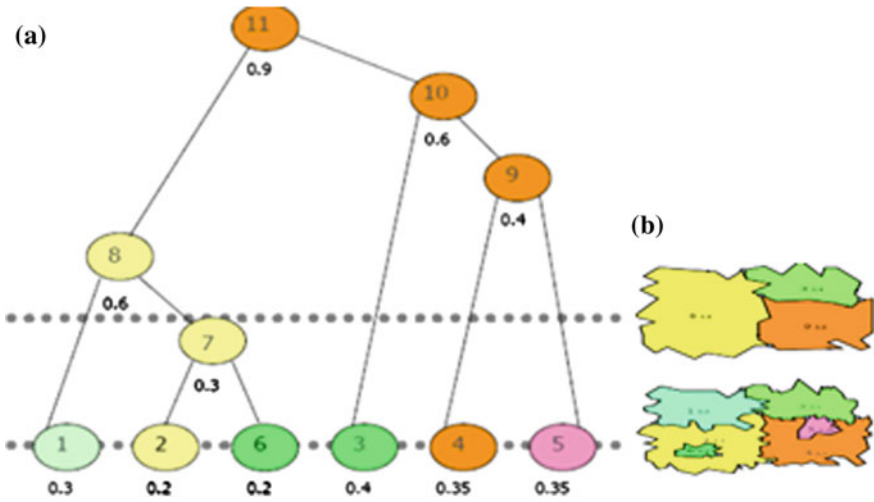


Fig. 5 Structure of variable-scale model, **a** tGAP hierarchical tree structure for the thematic semantic information and importance value, **b** respective slice planes (Meijers 2011)

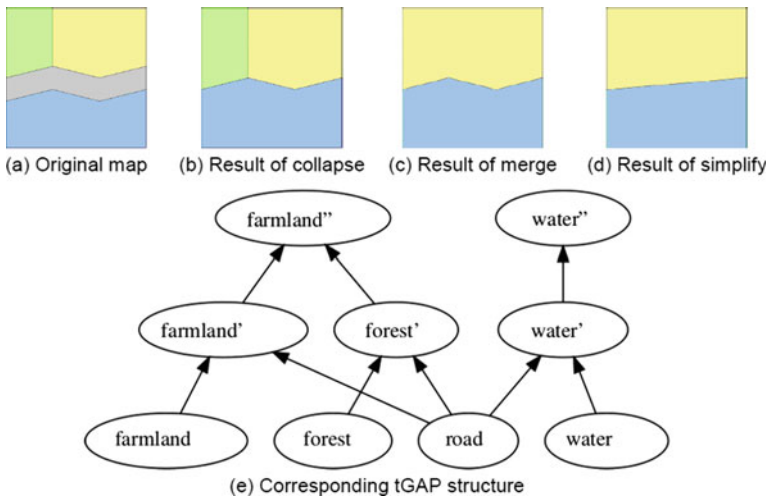


Fig. 6 The process of generalization and hierarchical tGAP structure (Meijers 2011)

The original tGAP structure (Oosterom 2005) capable to link the objects at different levels of detail.

The 3D, integrated 2D space and scale, tGAP model is represented in the Space-Scale Cube (SSC), which can be selected as a data model for the tGAP structure. While for the 2D dataset, it can be obtained by slicing the SSC model in horizontal plane. In the SSC model (see Fig. 7), the topology of the vario-scale

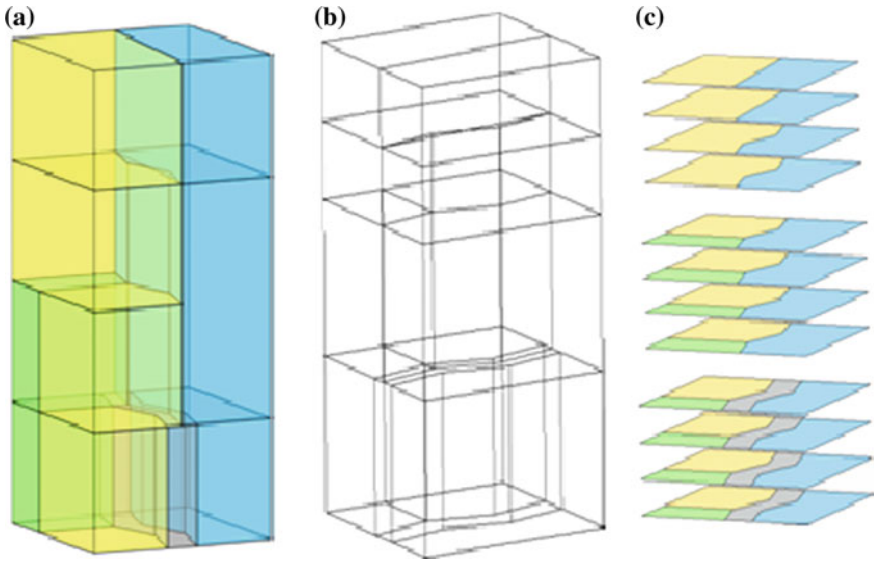


Fig. 7 Classic tGAP model, **a** classic SSC, **b** wireframe model, **c** slicing planes for several LoDs (Meijers 2011)

model in the vertical axis highly depends on the hierarchical tGAP structure in a database. The tGAP structure in vario-scale approach can be used in a web-based environment and/or in a desktop environment. For analyzing the content of a tGAP dataset ('data debugging'), 3D views can be generated; e.g. via ParaView software (Suba et al. 2013).

4 The Potential of DHE Data Structure in Scale Modelling

For the higher dimension modelling, there are several data structures available, which are able to represent models in four or more dimensions such as polytopal meshes (Sohanpanah 1989) and decompositions of polytopes (Bulbul et al. 2009). However, despite the fact that they are able to maintain various topological relationships, none of them provides a navigable network for the efficient navigation within the model.

Ohori et al. (2013) had identified two candidates of data structures that are able to model higher dimensions; Generalized Map (G-Maps) and the DHE. The DHE is a spatial 3D GIS data structure related to the radial-edge, facet-edge and half-edge data structures (Boguslawski 2011) and is based on the Augmented Quad Edge (AQE) data structure (Ledoux 2006; Ledoux and Gold 2007). In this paper we will further explore the DHE for realizing the vario-scale SSC.

4.1 Geometry, Topology and Thematic Semantic Integration

The DHE uses two structures: the dual and the primal graphs. The dual structure of the DHE model is associated with the topology, while the primal represents the geometry of any object in 3D model. The presentation of the entity (vertex, edge, face, and volume) in the primal space with its dual space conforms to the rules of the 3D Poincaré duality. Thus, a cell in the primal is presented as a single dual vertex; a face as a dual edge, an edge as a face and a vertex will be presented as a cell. Figure 8 illustrates the concept of 3D Poincaré duality used in DHE data structure.

The DHE is also capable of handling the thematic attributes-semantic integration of information. For instance, a room is presented by a dual vertex, while the attributes describing primal geometry (e.g. a room name) can be assigned to the dual vertex of the primal cell. The thematic attribute semantic information is embedded in the DHE data structure (see Fig. 8) to support GIS spatial data requirements.

Navigation in DHE data structure is possible because all of elements of the model are connected using pointers. A half-edge (HE) is the concept used to represent edges in the model, where all edges are split into two directed halves. The DHE has five navigation operators: around a symmetrical edge, shared vertex, face and edge and adjacent edges as shown in Fig. 9 respectively. The navigation is

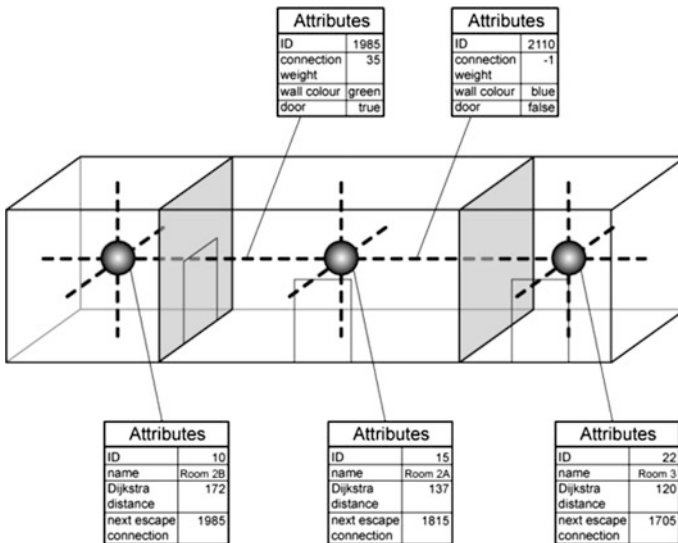


Fig. 8 Integration of Poincaré duality (vertex and dotted edge) with the primal space to support both navigation and the geometry of a 3D cell complex model (Boguslawski 2011)

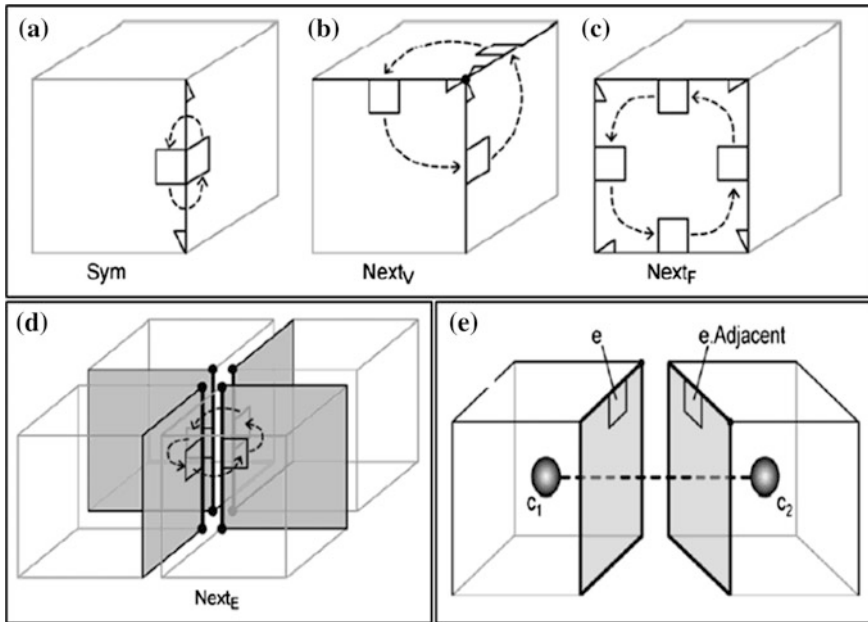


Fig. 9 Navigation of the dual half-edge (Boguslawski 2011)

performed from half-edge to half-edge. There are four basic operators available for navigation: Sym, NextV, NextF and Dual and compound operators: NextE and Adjacent based on the basic ones (see Fig. 9).

4.2 Concept of the DHE Vario-Scale Data Model

The implementation of the true vario-scale of tGAP structure is based on the SSC model (can be used with clients in desktop-based, web-based or mobile environments), using DHE with scale implementation may be categorized as a 3D model. The DHE data structure is capable to support the geometry, topology and attribute with additional scale dimension as replacing the Z (height) axis by adding some conversion processes (Fig. 11a, b) from original 2D scale dataset. The relationships between the representations at the previous and the next level as directly available in the variable scale DHE will be exploited to minimize the redundancy in scaling datasets. The concept of DHE vario-scale model is illustrated in Fig. 10.

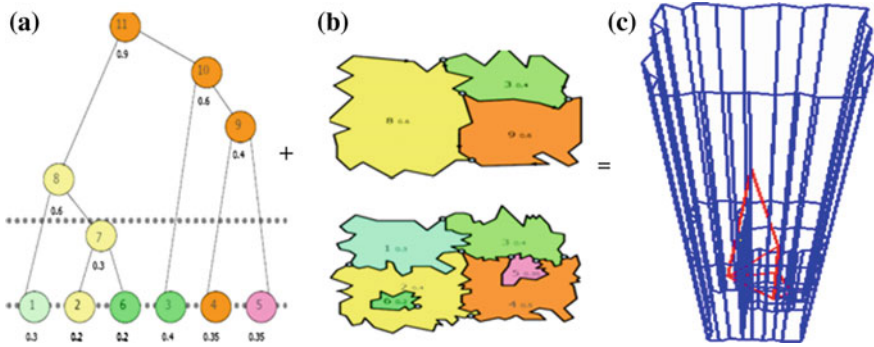


Fig. 10 Vario-scale data model, **a** hierarchical tree, **b** horizontal navigation, **c** idea of the DHE vario-scale data model. **a** and **b** are taken from (Meijers 2011)

To have an access query across LoDs, it is necessary to link object in one LoD to its respective object in the previous or next LoD (Paul et al. 2013). The DHE dual edge, which associated with the primal face (a loop of XY coordinates), can be used for hierarchical navigation and query purposes as well as storing the attribute and thematic semantic information (see Fig. 11c). While, the horizontal faces representing different LoDs can be illustrated in Fig. 11d.

Based on the PostgreSQL, foundation of the 3D DHE extension has been added to this DBMS. This is motivated by the fact that geographic data sets are often quite voluminous and do not fit in main memory for efficient use. Especially large scale base maps are large and these are also the types of maps that are used as basis for a range of scales (LoDs): either resulting in a multi-scale or vario-scale representation. Also, these geographic datasets are used by many persons and also maintained by multiple persons. These aspects, data size and multiple users, require a DBMS approach.

However the implementation of 3D DHE database using PostgreSQL and navigation operators will not be discussed in this paper. Goudarzi et al. (2015) discussed the implementation of 3D DHE database for a big dataset.

5 Discussion and Summary

In this paper a discussion on the spatial dimension in GIS modelling with some basic requirements and the use of data models/structures is presented. The existing 3D data models and structures in the context of capability of local modification are also discussed. The paper elaborates in detail on implementation approaches for representation of the scale-dimension as the third dimension: multi-scale and vario-scale models. The DHE is suitable for the vario-scale approach because of the

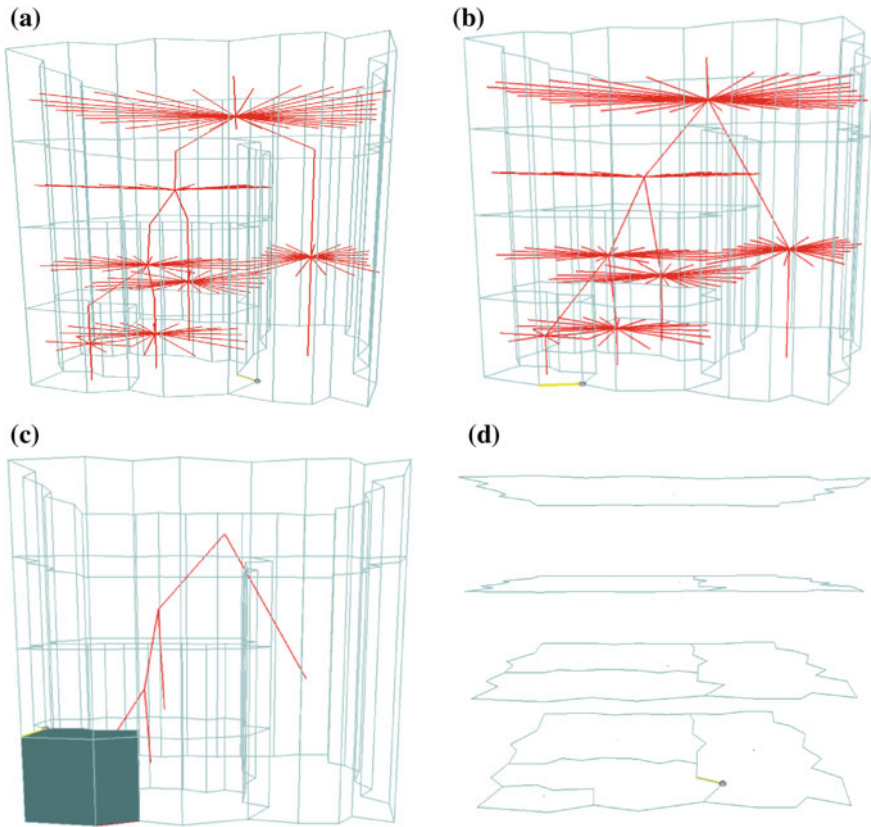


Fig. 11 Visualization of DHE-scale model dataset, **a** Original 3D scale DHE model (extrusion from single 2D dataset); the primal geometry in *light blue* and the dual in *red colour*, **b** intermediate process (merging by face), **c** Ideal DHE vario-scale model includes hierarchical navigation (the dual), attribute or semantic storing information in the primal or dual node, **d** 2D scale maps from slicing technique of SSC model

minimizing the redundancy of the dataset and preserving topological connections in each LoD.

The advantages of the data structure are its ability to perform local modifications and navigation using well defined operators (e.g.: Euler operators). It can be categorized as a consistent data model for scale modelling since it simultaneously updates both geometry and topology of the model via local modification.

Further implementation of the 3D (integrated 2D space and scale) DHE data model may produce better topological scale analysis such as zoom in and out for particular LoD using hierarchical query. Thus, a further study on this matter will be carried out to extend the capability and idea of the DHE data structure for scale dimension.

In future research we also intend to explore the 4D DHE for representing the integrated 3D space and scale model in order to realize true vario-scale 3D models. This in contrast to current 3D models, such as CityGML, which offer a discrete number of LoD with ‘unconnected’ and potentially inconsistent data.

The proposed research attempts to provide a new database-based service for web-based or desktop-based clients, exploiting the 3D (integrated 2D space and scale) data model which is suited for users and applications which require topological element in each level. It provides hierarchical and horizontal navigation, attribute storage and local modifications of the model with simultaneous and automatic update of the topology with the change of the geometry.

Acknowledgments The authors of this paper would like to thank to MyBrain15 (MyPhD), a program under Ministry of Higher Education (Malaysia) for the sponsorship.

Secondly, thanks to TU Delft (Department of OTB), The Netherlands for accepting the author to have three months research internship.

Finally, this research was also supported by: (i) the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by the Ministry of Economic Affairs (project code 11185); (ii) the European Location Framework (ELF) project, EC ICT PSP Grant Agreement No. 325140.

References

- Aarnes, J., Kippe, V., Lie, K., & Rustad, A. B. (2007). Modelling of multiscale structures in flow simulations for petroleum reservoirs. *Geometric modelling, numerical simulation and optimization* (pp. 307–360). Springer.
- Baig, S. U., Hassan, M. I., & Rahman, A. A. (2011). *Automatic generalization of 3D building models—A review*. Paper presented at the 10th International Symposium & Exhibition on Geoinformation (ISG 2011), Shah Alam Convention Centre (SACC), Malaysia.
- Banks, J., Carson, J., Nelson, B. L., & Nicol, D. (2009). *Discrete-event system simulation*.
- Boguslawski, P. (2011). *Modelling and Analysing 3D building interiors with the dual half-edge data structure*. (Ph.D.), University of Glamorgan, UK.
- Bulbul, R., Karimipour, F., & Frank, A. U. (2009). *A simplex based dimension independent approach for convex decomposition of nonconvex polytopes*. Paper presented at the Proceedings of Geocomputation.
- El-Mekawy, M. (2010). *Integrating BIM and GIS for 3D city modelling (the case of IFC and CityGML)*. Royal Institute of Technology (KTH), Stockholm, Sweden.
- Gold, C. M. (2005). *Data structures for dynamic and multidimensional GIS*. Paper presented at the 4th ISPRS Workshop on Dynamic and Multi-dimensional GIS.
- Gold, C. M., Chau, M., Dziesko, M., & Goralski, R. (2004). *3D geographic visualization: The marine GIS*. Berlin: Springer.
- Goudarzi, M., Asghari, M., Boguslawski, P., & Rahman, A. A. (in press, 2015). *Dual half edge data structure in database for big data in GIS*. Paper presented at the 3D GeoInfo 2015, Kuala Lumpur, Malaysia.
- Guibas, L., & Stolfi, J. (1985). Primitives for the manipulation of three-dimensional subdivisions. *Algorithmica*, 4(3), 32.
- Günther, O. (1988). *Efficient structure for geometric data management*. Berlin: Springer.
- Jones, C. B., & Abraham, I. M. (1986). Design considerations for a scale dependent cartographic database, pp. 384–398.

- Jones, R. R., McCaffrey, K. J. W., Clegg, P., Wilson, R. W., Holliman, N. S., Holdsworth, R. E., et al. (2009). Integration of regional to outcrop digital data: 3D visualisation of multi-scale geological models. *Computers & Geosciences*, 35(1), 4–18.
- Ledoux, H. (2006). *Modelling three-dimensional fields in geoscience with the voronoi diagram and its dual certificate of research*. (Ph.D.), University of Glamorgan.
- Ledoux, H., & Gold, C. M. (2007). Simultaneous storage of primal and dual three-dimensional subdivisions. *Computers, Environment and Urban Systems*, 31(4), 393–408. doi:10.1016/j.compenvurbsys.2006.03.003.
- Lee, J., & Zlatanova, S. (2008). *A 3D data model and topological analyses for emergency response in urban areas*. Taylor & Francis.
- Li, Z. (1994). Reality in time-scale systems and cartographic representation. *Cartographic Journal*, 31(1), 50–55.
- Liangchen, Z., Guonian, L., Yehua, S., Hangbo, X., & Haixia, W. (2008). A 3D GIS spatial data model based on cell complex. In *The international archives of the photogrammetry, remote sensing and spatial information sciences* (Vol. XXXVII).
- Lindeberg, T. (1994). Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2), 225–270.
- Mäntylä, M. (1988). *An introduction to solid modeling*. New York, USA: Computer Science Press.
- Meijers, M. (2011). *Variable-scale geo-information*. (Ph.D.), Technische Universiteit Delft, Netherland.
- Mostafavi, M., & Gold, C. M. (2004). A global spatial data structure for marine simulation. *International Journal of Geographical Information Science*, 18, 211–227.
- Ohuri, K. A., Boguslawski, P., & Ledoux, H. (2013). *Representing the dual of objects in a four-dimensional GIS*. Paper presented at the International Workshop on Geoinformation Advances.
- Paul, N., Bradley, P. E., & Breunig, M. (2013). Integrating space, time, version and scale using alexandrov topologies. In *International Symposium on Spatial and Temporal Databases SSTD 2013*.
- Peuquet, D. J. (2001). Making space for time: Issues in space-time data representation. *Geoinformatica*, 5, 11–32.
- Peuquet, D. J. (2002). *Representations of space and time*. Guilford Press, New York.
- Raper, J. (2000). *Multidimensional geographic information science*. London: Taylor & Francis.
- Sester, M. (2007b, 03 Sep–7 Sep 2007). 3D visualization and generalization. In *Photogrammetric week 07, week 07* (pp. 285–295).
- Sester, M. (Producer). (2007a, Apr 2013). 3D visualization and generalization. Institute of cartography and geoinformatics. Lecture note.
- Sohanpanah, C. (1989). Extension of a boundary representation technique for the description of n dimensional polytopes. *Computational Graphics*, 13(1), 17–23.
- Suba, R., Meijers, M., & van Oosterom, P. (2013). *2D vario-scale representations based on real 3D structure*. Paper presented at the 16th ICA Generalisation Workshop, Dresden, Germany, 2013. http://www.gdmc.nl/publications/2013/2D_vario-scale_representations_3D_structure.pdf.
- Tse, T.O.C., & Gold, C.M. (2004). TIN meets CAD: extending the TIN concept in GIS. *Future Generation Computer Systems*, 20(7): 1171–1184.
- van Oosterom, P. (1990). *Reactive data structure for geographic information systems*. (Ph.D.), Leiden University.
- van Oosterom, P. (2005). Variable-scale topological data structures suitable for progressive data transfer: The GAP-face Tree and GAP-edge Forest. *Cartography and Geographic Information Science*, 32, 331–346.
- van Oosterom, P., & Stoter, J. (2010). *5D data modelling: Full integration of 2D/3D space, time and scale dimensions*. Paper presented at the 6th International Conference on Geographic Information Science, Berlin, Heidelberg.
- van Oosterom, P., & Stoter, J. (2012). Principle of 5D modelling.

- Vervoort, J. M., Rutting, L., Kok, K., Hermans, F. L. P., Veldkamp, T., Bregt, A. K., et al. (2012). Exploring dimensions, scales, and cross-scale dynamics from the perspectives of change agents in social–ecological systems. *Ecology and Social*, 17(4), 24.
- Weiler, K. (1988). The radial edge structure: A topological representation for nonmanifold boundary modeling. Paper presented at the In Geometric Modeling for CAD Applications. AmsterdamL: Elsevier.
- Worboys, M. F. (1994). A unified model for spatial and temporal information. *The Computer Journal*, 37(1), 26–34.
- Zhou, S., & Jones, C. B. (2003). Multi-scale spatial database and map generalisation.

Towards Integrating BIM and GIS—An End-to-End Example from Point Cloud to Analysis

Claire Ellul, Gareth Boyes, Charles Thomson and Dietmar Backes

Abstract Building Information Modelling (BIM) is becoming increasingly important within the UK, not least because of a UK Government directive that mandates Level 2 BIM for companies tendering for Government work, with the aim of reducing the cost of construction of public assets by 20–30 %. While this is aimed at new construction, it can be foreseen that a wider introduction of BIM could also result in savings during large refurbishment projects, which form a significant part of construction work in the UK, as well as during the occupancy phase of the building. However, unlike new projects, where the model for the BIM can be obtained from CAD drawings, deriving a detailed BIM for pre-existing structures requires some form of scan-to-BIM operation using laser scanning. To contribute to sustainability in construction, an underlying driver for BIM, the BIM must also be integrated with other data sources. Therefore, once the scan is complete, the resulting point cloud must be converted into geometry objects and geo-referenced for integration with Geographical data such as air quality or noise information. This paper presents an end-to-end example of this process, focusing in particular on the challenges of integrating BIM and GIS into one framework, and highlighting preliminary steps to be carried out during BIM creation in order to enable this to take place.

Keywords BIM · 3D GIS · Integration · Environmental data · Spatial databases

1 Introduction

A sustainable built environment should minimise whole-life carbon and material costs through efficient use of resources (energy, waste, water), contribute to the physical and mental health of its users, enhance productivity and be adaptable for future uses (Sustainable Development Commission 2011). Building Information Models

C. Ellul (✉) · G. Boyes · C. Thomson · D. Backes
Department of Civil, Environmental and Civil Engineering, University
College London, London WC1E 6BT, UK
e-mail: c.ellul@ucl.ac.uk

© Springer International Publishing AG 2017
A. Abdul-Rahman (ed.), *Advances in 3D Geoinformation*,
Lecture Notes in Geoinformation and Cartography,
DOI 10.1007/978-3-319-25691-7_28

(BIM) aim to improve sustainability by giving architects and engineers access to a wealth of detail and digital information about a projects construction and maintenance through its life-cycle, providing an improved, collaborative, efficient way of working. They are of great interest to major Civil Engineering and other construction projects in the United Kingdom (UK), not least because of the UK Governments requirement for “fully collaborative 3D [three dimensional] BIM (with all project information and data being electronic) as a minimum by 2016” (Cabinet Office 2011).

While much focus has been given to questions relating to the definition of BIM (see Sect. 2.1), and to the creation of BIM for new projects, to date, less research has been targeted at creating BIM for existing buildings (either for facilities management or as part of a retro-fit process). Additionally, a major challenge that currently prevents BIM from achieving its full potential for improving sustainability (and reducing cost) in construction is the lack of integration of data held within BIM with other sources of data about the wider context of the built environment for example details about the urban landscape, traffic congestion, air quality, noise, flooding, population demographics, maintenance and staffing schedules, infrastructure such as water pipes, drains, fibre-optic cables. These types of information are commonly held in a Geographical Information System (GIS) and integration with BIM could answer questions such as “Where is the nearest supplier of concrete (GIS), when and how much should I order (BIM)?”, “How do I route plasterboard to my site (GIS) and where do I need to store it on site for maximum convenience (BIM)?”, “Is the current capacity of the drainage system (GIS) sufficient for this new construction (BIM)?”, “What is the flood risk in this underground station (GIS) and what internal wiring will be damaged in different flooding scenarios (BIM)?”

Integration between BIM and GIS is required if BIM is to meet its goal to support a sustainable built environment. Indeed, “BIM is about infrastructure, and therefore GIS is critical to its delivery” (Kemp 2011) and predicted savings of \$15.8 billion per year could be made from integrated processes (Kemp 2011). However, there is currently no clear path to interoperability or integration between GIS and BIM. This is in part a technical challenge (e.g. incompatible data formats) but is also a conceptual challenge as the gap between two approaches to modelling the world must be bridged.

A number of case studies (Sebastian et al. 2013; Irizarry et al. 2013; Zhang et al. 2009) illustrate the potential of the integration of these two sources of information and (El Meouche et al. 2013) have reviewed software based approaches to this problem. However, to date it is difficult to find a case study relating to the creation of a BIM of an existing building and its subsequent integration with GIS data. A first example of this end to end process is presented in this paper. We describe the creation of a BIM for the Chadwick Building (home of the Department of Civil, Environmental and Geomatic Engineering—CEGE—at University College London) and the subsequent integration of this BIM with noise and air quality data collected via GPS-enabled sensor devices as part of the EveryAware Citizen Science research project (Loreto 2012). The result shows the potential of such integration for an investigation into noise and air quality distribution around the building, and importantly also

highlights some integration challenges that could be overcome by providing appropriate BIM capture guidelines early on.

The remainder of this paper is structured as follows—Sect. 2 provides background information on BIM and GIS integration. Section 3 describes the data used in the project, and Sect. 4 presents the process used to integrate the disparate data sources. Section 5 provides a discussion of some of the issues encountered, leading to recommendations for BIM capture in future projects.

2 Background

2.1 Building Information Modelling

Building Information Modelling (BIM) is a comprehensive process that enables parties in the Architecture Engineering and Construction industries to collaborate on a project. The UK BIM Task Group defines BIM as value creating collaboration through the entire life-cycle of an asset, underpinned by the creation, collation and exchange of shared 3D models and the intelligent, structured data attached to them (Building Information Modelling Task Group 2013). In general, the aim of BIM is to improve the performance of infrastructure, reduce waste, increase resource efficiency, reduce risk, increase resilience and increase integration (Kemp 2011).

The acronym BIM has acquired different meanings in different contexts. In one sense it is the corporate attitude needed to ensure that collaboration succeeds and the management of information and the complex relationships between social and technical resources that represent the complexity, collaboration, and interrelationships (Jernigan 2008). The second sense of the acronym BIM refers to the server network and software tools that store, construct and visualise the 3D model. The 3D model is then the final sense of the BIM acronym, and the focus of this research. This model is built up from parametric objects representing building components in object-orientated software. These parametric objects may be geometric or non-geometric and are attributed with functional, semantic or topologic information (Volk et al. 2014).

The development of BIM can be graphically represented by the BIM wedge in Fig. 1 with the different maturity levels being summarised in Table 1 (Building Information Modelling Task Group 2013). The UK Government has mandated that all government construction contracts must incorporate fully collaborative Level 2 BIM (with all project and asset information, documentation and 3D data being held electronically) as a minimum by 2016 (Cabinet Office, 2011). In the UK, 39 % of 1350 participants at the National Building Specification survey between December 2012 and February 2013 responded that they were using BIM and 71 % agreed that BIM represents the future of project information (Zeiss 2013).

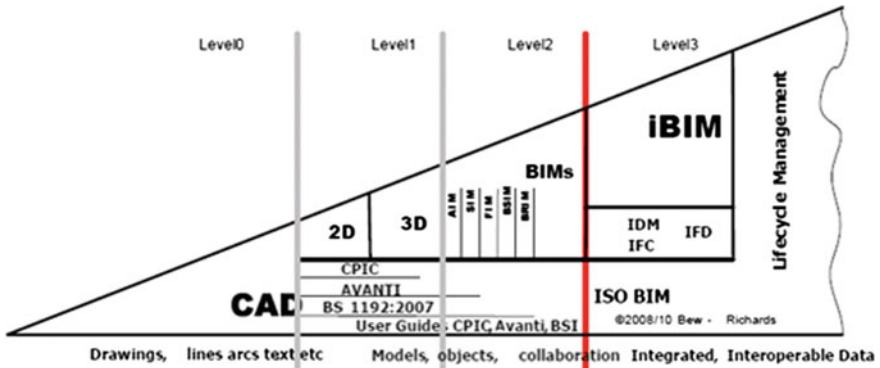


Fig. 1 Levels of building information modelling, (Building Information Modelling Task Group 2013)

Table 1 BIM levels

Maturity level	Description
Level 0	Unmanaged 2D CAD
	Paper or electronic paper
Level 1	Managed 2D/3D CAD
	Collaboration tools provide a common data environment
Level 2	Proprietary BIM (pBIM)
	Managed 3D BIM data held in BIM software (e.g. Revit)
	Integration on the basis of proprietary interfaces
Level 3	Integrated BIM (iBIM)
	3D BIM in IFC format enabled by web services
	Managed through a collaborative model server

2.2 Comparing BIM and GIS

BIM has been stated to be “a modelling technology that combines the design and visualisation capabilities of CAD with the rich parametric object and attribute modelling of GIS” and unlike Computer Aided Design (CAD) entities are meaningful (Casey and Vankadara 2010)—i.e. walls and windows are modelled as such. This broad definition, at first glance, is similar to that for a GIS, which models both attributes and geometry. Both can model the built environment in 3D and both can model both indoor and outdoor features within this environment. Additionally, both BIM and GIS data can be managed in a Database Management System (although for BIM this is usually just done for storage purposes rather than offering direct query capabilities). They both provide efficient methods for the documenting, editing, managing and visualising spatial and non-spatial information, and both can represent the world ‘as is’ and also model historic data and future planning and modelling

outcomes, and model data at varying scales. BIM has been stated to be “a modelling technology that combines the design and visualisation capabilities of CAD with the rich parametric object and attribute modelling of GIS” and unlike Computer Aided Design (CAD) entities are meaningful (Casey and Vankadara 2010)—i.e. walls and windows are modelled as such. This broad definition, at first glance, is similar to that for a GIS, which models both attributes and geometry. As can be seen by comparing the descriptions of IFC (in Sect. 2.4) and CityGML in Kolbe et al. (2005) Level of Detail 4, both can model the built environment in 3D and both can model both indoor and outdoor features within this environment. Additionally, both BIM and GIS data can be managed in a Database Management System (although for BIM this is usually just done for storage purposes rather than offering direct query capabilities). They both provide efficient methods for the documenting, editing, managing and visualising spatial and non-spatial information, and both can represent the world ‘as is’ and also model historic data and future planning and modelling outcomes, and model data at varying scales.

However, a number of key differences exist which present a challenge to interoperability, in part due to their different origins. Interoperability is defined as “the ability of two or more information systems to share data, information or processing capabilities” Worboys and Duckham (2004). Bishr (1998) identifies three types of interoperability:

1. **Semantic**—looks at the interaction of two or more systems without a pre-defined or agreed upon interface, assuming that the meaning of the objects is embedded in the objects themselves—a fact can have more than one description, for example a road can also be called a street
2. **Schematic** —the same object in the real world is represented using different concepts in the database i.e. it is an object in one database and a property in another—for example the publisher of a book can be a property of the book, but also a separate entity
3. **Syntactic** —the databases use different paradigms e.g. relational or object-oriented models or rasters versus vectors

While initially it may appear that issues preventing BIM and GIS integration relate primarily to syntactic interoperability, and in particular to data exchange formats, schematic issues and semantic issues are of great relevance, in particular given the diverse origins of the two domains. BIM modelling developed within the Computer Aided Design (CAD) world as a specialised process for designing infrastructure and buildings for construction. GIS on the other hand developed from the need to record and manage information already in existence, and models not only the built environment but also demographics, transportation, natural resources, disease spread and many other areas. It addresses the who, what when why how and also what exists at a particular location, what has changed over time, what spatial patterns exist buffering, proximity analysis, spatial modelling tools (Casey and Vankadara 2010) and is particularly strong when integrating spatial and non-spatial data.

Authors including (Pu and Zlatanova 2006; Kemp 2011; Casey and Vankadara 2010; Cowen 1988; Zhang et al. 2009 and Zlatanova and Prospero 2006) list the following key differences:

For BIM the visual representation of an engineering project provides the primary source of information with semantic information, such as construction material, represented by patterns and hatching. Building elements modelled include railings, plates, ramps, slabs, stairs and walls. Extensive, detailed, 3D models including wall thicknesses and other structural details can be visualised. BIM models are stand-alone (single project) with no link to their real-world location. Additionally, for pre-existing buildings or infrastructure, geometry is often generated by laser-scanning, which can cause issues with completeness.

In contrast, for GIS, very rich semantic information (attributes) can be stored alongside geometric representations of features as spatial data. In many cases, such data is stored in a spatially-enabled relational database facilitating interoperability with other data via queries. Information is located within both local and global spatial reference frameworks, making it easy to interrogate the data based on location (e.g. what buildings are within 100 m of this point?, what is under this road?). However, detailed 3D geometry is not well modelled e.g. a wall is represented as a line, no matter its thickness or construction material.

2.3 Previous Attempts at BIM/GIS Integration

A number of authors describe work towards BIM and GIS integration, with particular focus on transforming Industry Foundation Class data (IFC) to CityGML (Kolbe et al. 2005). While much of this is uni-directional (e.g. El-Mekawy et al. 2011; Hijazi et al. 2011; Tobiáš 2015), bi-directional integration has also been attempted, with (El-Mekawy et al. 2011) presenting an intermediate 'neutral' model that acts as a staging point between the two. Semantic interoperability (Casey and Vankadara 2010; El-Mekawy et al. 2012), is a core focus of these efforts, which create a number of domain ontologies to identify the important objects and classes on either side of the integration.

Implementing this theoretical approach at a more practical level, (El Meouche et al. 2013) examine syntactic interoperability, providing a review of potential software packages for data interchange between the various systems. Application-based case studies include (Sebastian et al. 2013) who describe the process of migrating IFC data into CityGML Level 4 to support noise and disturbance modelling in the construction and maintenance of bridges. An application domain extension (ADE) to CityGML is developed for this purpose, and two case studies (in Spain and The Netherlands) are presented. Integration permits a detailed focus on worker and resident safety, noise and traffic disruptions. A second example transforms utility network information (inside a building) from IFC to CityGML, again creating an ADE for the purpose. A 1:1 mapping between IFC and ADE is achieved in the majority

of the cases (Hijazi et al. 2011). Additional research into utilising BIM and GIS for flood damage investigation is presented by Amirebrahimi et al. (2015).

2.4 Industry Foundation Classes

The IFC (industry foundation classes) exchange format was developed by the International Alliance for Interoperability (now known as building SMART) and provides a formalized representation of typical building components e.g. wall, door, and their attributes e.g. type, function, geometric description, relationships. It also supports topological information (“connected to”) and abstract concepts such as schedules, activities and construction costs (Casey and Vankadara 2010). Unlike CAD, objects are grouped into logical entities (classes, with properties such as name, materials, relationships, constraints)—i.e. a wall is an object with attributes rather than just a collection of lines (much like standard GIS modelling). Product information is grouped according to construction trade—such as HVAC, building controls, electrical, plumbing, fire protection, structural elements (Casey and Vankadara 2010). Full details of the IFC standard can be found on the Building Smart website (Building Smart 2015a).

3 Data

3.1 Preparing the BIM

The data being used in this project is a BIM of University College London’s Chadwick Building (Fig. 2). The full process of creating the BIM is described in Backes et al. (2012, 2014).

The Chadwick is a Grade I listed building and as such is subject to stringent policies and planning procedures regarding building works and the model was constructed via terrestrial laser scanning. Terrestrial laser scanners generally make use of a beam of laser light that is either pulsed or phase modulated and by calculating the time delay between the out and return in the former or the phase shift in the latter. Each scan creates a volume of 3D point measurements called a pointcloud that can also be coloured if imagery is captured as part of the capture process. For the Chadwick BIM, multiple scans were required to capture areas within the building by moving the instrument to mitigate shadows from obstacles in the line-of-sight of the scanner. To bring the pointclouds from different setups together, a registration process is required. This identified common features in the scenes and made use of strategically placed targets to align the scans.

The scanning project was carried out over the course of ten weeks with the help of four engineering undergraduates and the majority of the scans were captured to

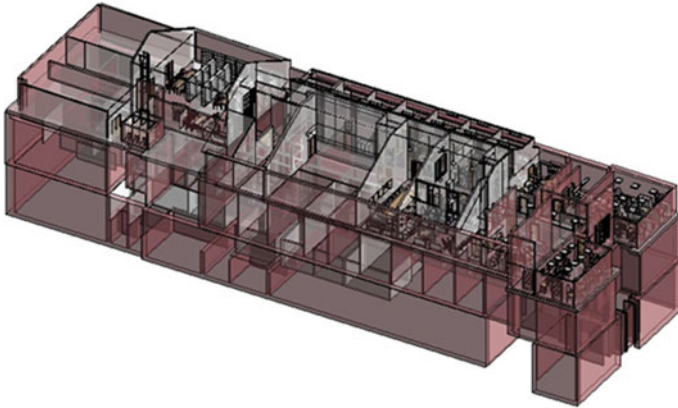


Fig. 2 Chadwick BIM

provide at least 8 mm sampling density (Backes et al. 2014). A typical scan contained about 27 million points and about 250 MB in size (Backes et al. 2014). The resulting point clouds were stitched together and were used as a guide to create the geometric elements of the model, within Autodesk's Revit Software. For the first floor test area (which is the focus of the integration study described in this paper), the resulting geometry was created with a relatively low level of detail—i.e. detail such as sockets and so forth was omitted and only walls, floors and ceilings modelled. Walls were assumed to be the blank space between scans of adjacent rooms and the scans were supplemented with an internal survey of the building. Figure 2 shows the resulting BIM.

3.2 Capturing the Noise and Air Quality Data

The noise and air quality data forming part of this study were captured as part of a separate project evaluating the potential of low-cost sensors for crowd sourcing of such data. The EveryAware project (Loreto 2012) designed and developed an App—Widenoise—to enable mobile phone users to measure sound levels, annotate them and then upload these to a central map. Each noise measurement is 30 s in duration, and the measuring process is manually triggered by the user. An air-quality sensor box that can be coupled to a mobile phone App via blue tooth was developed to allow continual monitoring of the environment (i.e. no user intervention required). Details of Widenoise can be found in Loreto (2013) and of the AirProbe system in Chap. 1 in Loreto (2014). Noise data was captured over a 2-day period at various locations around the Chadwick Building, with air quality data captured by leaving the sensor boxes in fixed locations for 24 h intervals. In both cases, given that the

sensors were designed for outdoor use, the data was tagged to ensure that its indoor capture location could be identified during post-processing. Within the Chadwick Building, a total of 420 noise points and 4161 air quality points were captured.

4 Integrating the Data

4.1 *Geo-Locating the BIM*

Figure 3 shows the first floor of the Chadwick building, with the ceilings removed. As noted above, the BIM coordinate system maintains a local reference and the geometry must therefore be geo-located prior to integration—in this case, into British National Grid. This can be performed within AutoDesk Revit (Autodesk 2015a) by identifying a number of coordinate points at the base of the building by using 1:1250 topographic mapping provided by the UK National Mapping Agency, the Ordnance Survey. Within Revit, the “Specify Coordinate at a Point” option can be used to transform the points. It is also important to change the units within Revit—as noted in Section Table 2 one of the key differences between BIM and GIS is the scale. Units can be changed from millimeters to meters via the “Project Units” option in Revit.

Fig. 3 CEGB BIM—first floor

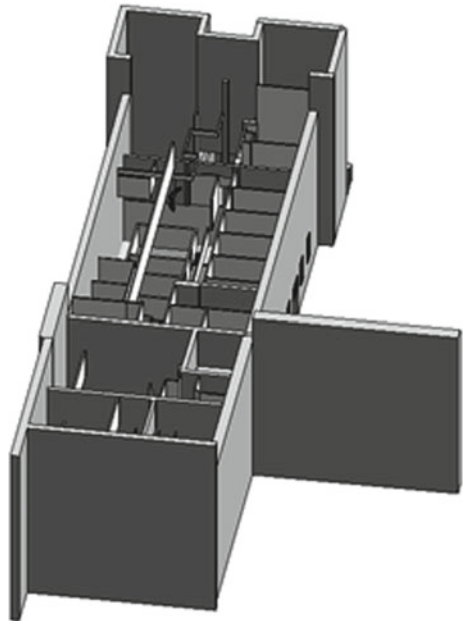


Table 2 BIM versus GIS

BIM	GIS
Describes buildings and facilities very detailed, mm measurement units	Describes buildings, entire sites, regions or countries, meter measurement units
Initiated during procurement phase of facility lifecycle, focussed on built environment	Wide-ranging focus
Used to organise information to specific contractual deliverables	Used to organise multiple types of information, and to integrate spatial and non-spatial information
Very sophisticated 3D geometry can be modelled—B-Rep, NURBS, Splines and CSG	Less sophisticated geometrically, mostly 2D, B-Rep for 3D
Focus on geometry, primarily on construction materials	Focus on generic geometry, attributes and sophisticated spatial analysis
No ability to model networks	Models networks and connectivity
Basic database integration	Sophisticated database integration
Local coordinate systems generally used	Regional, National or Global Coordinate Systems

4.2 *Converting the BIM to GIS*

Once the data is appropriately positioned and units set, the Autodesk Revit suite (Autodesk 2015a) offers the option to export data into the neutral IFC format described in Sect. 2.4. To minimise export time the “Export Base Quantities” box should be unchecked and the “Split Walls and Columns by Story” checked. The IFC file can then be imported into a GIS using the SafeSoft FME suite (Safe Software (n.d)) which permits a direct translation of an IFC file into a spatial database. Given it’s more advanced 3D spatial query functionality, Oracle Spatial (Oracle 2013) was used in this case. The resulting data is automatically divided into a number of layers that correspond to the IFC Elements provided by the schema, such as IFCWall, IFCSlab_solid, IFC_Stair, IFCWallStandardCase. A summary of the key resulting layers is given in Table 3.

A total of 65 layers and tables were created by the conversion process, with 47 of these having no associated geometry. These are used to store additional information such as the relationships between objects, additional descriptors for objects (e.g. door panel or lining properties) and project related information (ownership and organisation details).

4.3 *Geo-Locating the Noise and Air Quality Data*

As can be expected, the use of GPS to capture positional information for the noise and air quality data proved problematic indoors. However, the EveryAware Apps

Table 3 GIS layers from the Chadwick BIM

Number of objects	Layer description
118	IFCBuildingElementProxy used to model monitors and notice boards
1	IFCColumn used to model a supporting column
16	IFCCovering ceiling elements
52	IFCDoor
282	IFCFurnishingElements for chairs, wall units, desks and shelving
78	IFCOpeningElements
15	IFCPlates used to model internal glass walls
2	IFCSlabs
1	IFCStair
16	IFCWall
309	IFCWallStandardCase
33	IFCWindow

provided for each measurement device also permitted each sample to be tagged. A map was created sub-dividing the internal space on the first floor of the Chadwick building into grids and each measurement tagged to allow geo-referencing. All data was imported into QGIS and edited to translate it into the correct location manually.

Figure 4 shows the resulting integrated data, with additional points outside the building being related to other EveryAware sensing activities carried out on the

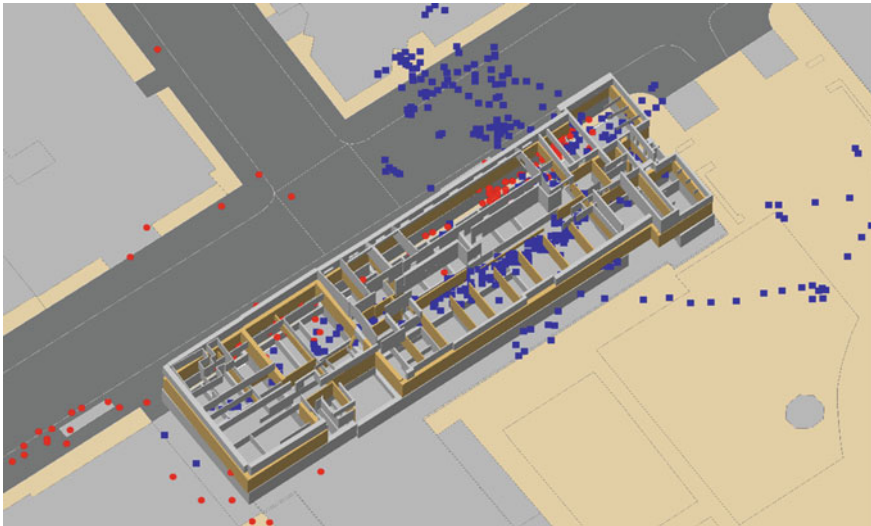


Fig. 4 Integrated data (noise data in *blue*, air quality data in *red*). (Topographic mapping data Ordnance Survey Crown Copyright)

street. The data was also assigned a height value to ensure that it was geo-located to the correct floor in the building.

4.4 Querying the Integrated Data

The original aim of the study was to use SQL to query the integrated dataset to find out, for example, the average, minimum and maximum noise level in each tested room in the building at different times of day and days during the week. However, although this was theoretically possible, two issues initially prevented this final level of querying. Firstly, the export process to convert the BIM into an appropriate format for use in GIS (via IFC) did not create enclosed spaces on which topological queries (contains) could be applied to identify the noise and air quality points within a specific room. Although a number of these elements could have been constructed by combining wall, floor, door, window and ceiling geometry these were not tagged appropriately in the BIM. Figure 5 highlights the different tagging of similar objects within the BIM—internal walls are tagged as ‘WallStandardCase’ (in grey) and ‘Wall’ (in burgundy).

Secondly, parallel research calibrating the noise and air quality sensing devices, carried out within the EveryAware (Loreto 2012) project, highlighted issues with the quality of the noise and air quality data which meant that while it was possible to determine whether noise was at a low, medium or high level it would not be possible to extract accurate values from the data. Figure 6 highlights the calibration results

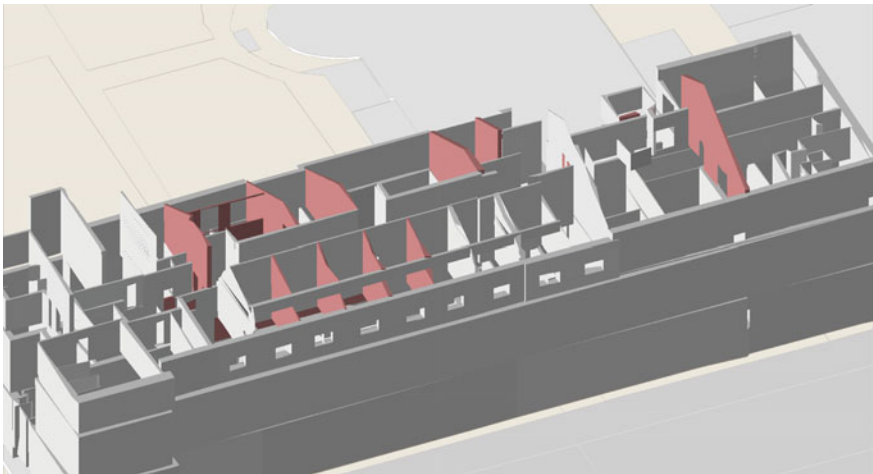


Fig. 5 BIM data in GIS, highlighting internal walls tagged differently in the BIM—burgundy shows IFCWall and grey shows IFCWallStandardCase (Topographic mapping data Ordnance Survey Crown Copyright)

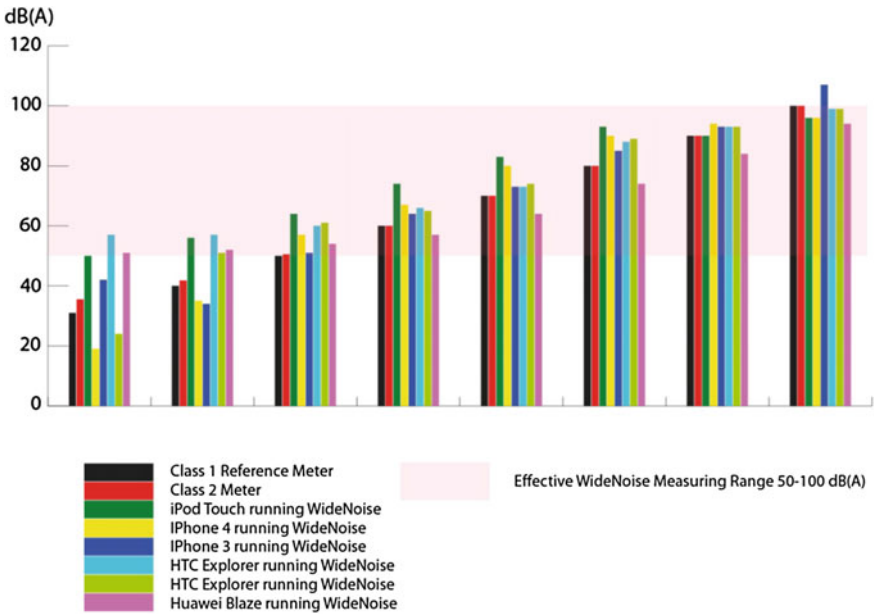


Fig. 6 Results of calibration activities in an anechoic chamber, noise sensing on mobile phones

for the noise dataset, showing differences even between phones from the same manufacturer. Further details of the calibration process for noise can be found in Sect. 4 in Loreto (2010) and for air quality in Sect. 2 in Loreto (2010).

4.5 Utilising IFC Spaces for Room Geometry

An alternative approach to reconstructing a room from its component geometry (walls, doors, floors and so forth) is to make use of the IFCSpace and the IFCRelSpaceBoundary elements. An IFCSpace “represents an area or volume bounded actually or theoretically. Spaces are areas or volumes that provide for certain functions within a building” and is hierarchical in nature (Building Smart 2015b). The IFCRelSpaceBoundary represents contiguous space, whether physical (based on wall centres) or virtual and permits adjacency relationships between spaces to be defined. It is geometrically described as a collection of planar polygon faces, with each face relating to a particular building element that is stored as a boundary attribute, and in this represents Boundary Representation geometry commonly used to model 3D GIS. A second iteration of the export process from Revit to IFC was therefore carried out to ensure that spaces were created—the result of this is shown in Fig. 7. This data was imported in to Oracle Spatial (version 12.1.0.2.0—Enterprise Edition) using FME. As this resulted in multi-surface polygons, Oracle’s

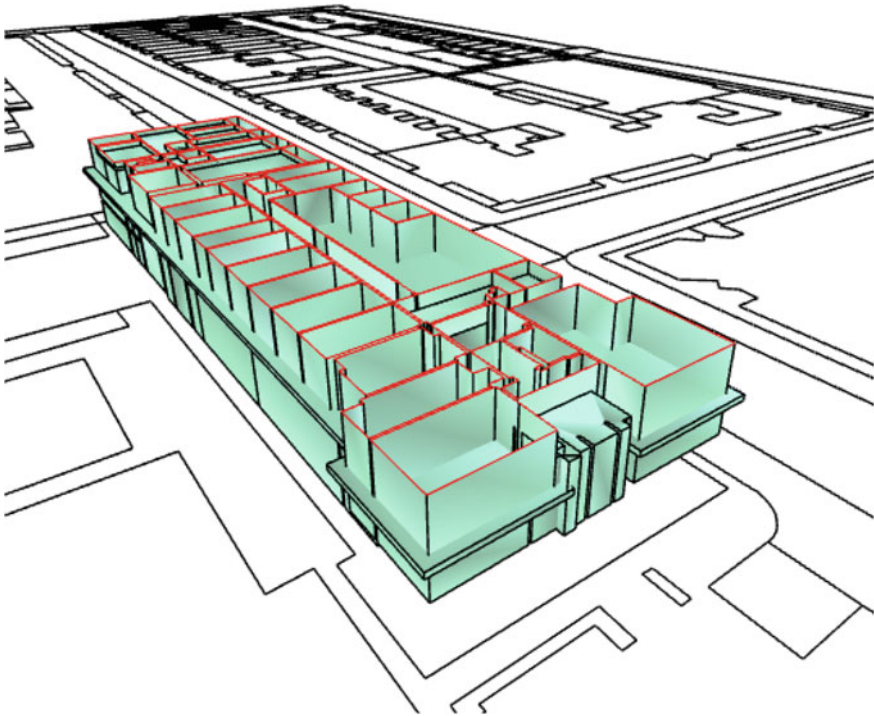


Fig. 7 Rooms in the chadwick building created using IFC spaces (Topographic mapping data Ordnance Survey Crown Copyright)

Table 4 Noise values in the Chadwick building

Room number	Number of noise points	Minimum dB	Average dB	Maximum dB
102	4	50.82	51.2	51.7
103	11	31.51	48.48	56.23
120	2	41.09	41.27	41.44
1M00A	2	51.42	54.54	57.65
1M01	2	38.76	45.85	52.95

CONVERT3007TO3008 utility was used to create the required volumes representing each room, resulting in a total of 112 rows, each representing a room in the building. To illustrate the potential of the approach on a small subset of data, Oracle SQLnqueries were then used to identify the minimum, average and maximum noise values for 5 rooms in the building—the results are given in Table 4. An example of the SQL statements used is shown here:

```
SELECT ROUND(MIN(A.DATA_AVERAGE),2) AS MIN, B.NAME
      FROM 3D_NOISE_DATA A, CHADWICK_IFCSPACE_SOLID B
```



```
WHERE SDO_INSIDE(A.GEOM, B.GEOM) = 'TRUE'  
GROUP BY B.NAME  
ORDER BY MIN ASC;
```

5 Discussion and Conclusion

The results obtained demonstrate the potential of spatially integrating BIM and GIS data to investigate sustainability-related issues in a building that is currently in use. Using the techniques described, it is possible to use laser scanning to create a detailed 3D model of the building, geo-reference it and integrate tagged sensor data to obtain room-by-room measurements. As noted above, these sensor values are subject to calibration issues, and indeed in most cases only one or two points were captured per room. However, utilising the same approach and SQL queries similar to that given above, along with higher quality equipment could provide a good understanding of the spatio-temporal distribution of air quality and noise within the building.

Although not specifically required for the Chadwick Building, this information could potentially be provided in real time if a live sensor feed were available, as appropriate triggers could be written to update average values every minute in the database and a dashboard used to display results to a building manager. As the data is geo-referenced, the results obtained could also be correlated with traffic counts on the street outside the building, as well as with swipe security access data showing when the building is occupied. Information from the BIM—such as construction material of the internal walls—could be used to identify where retrofitting could ameliorate a noisy environment, or where extractor fans may improve air quality.

The processes described above also highlights the relative complexity of creating a detailed BIM for an existing building in an appropriate format for use with GIS. This complexity relates in particular to the time and effort required to perform the scans and convert them into usable geometry, but also to ensuring that the resulting BIM is structured and tagged appropriately.

Section 2.2 highlighted the different approaches to geometry modelling in BIM and GIS. However, in this case study this syntactic difference did not prove to be a problem, with Oracle Spatial able to model the converted data in 3D (further tests are required to identify whether geometric detail was lost in the conversion). The number of layers created by the conversion process did, however, confirm that gaps between the approaches to schematic modelling in BIM and GIS will need to be addressed going forward, as currently the objects of interest to GIS (e.g. rooms), represented as single objects, need to be constructed from component windows, doors, walls, floors and ceilings. Semantically, an IFCSpace is also not directly equivalent to a GIS “room”.

The IFC-spaces based approach means that resulting data is held in ‘spaghetti’ format in the GIS (in Oracle Spatial), which gives in duplication of the walls between rooms as each room is modelled as a separate solid, and results in the loss of topological information—e.g. connectivity—contained in the IFC file (see Sect. 2.4). Further

manipulation would be required to extract this information and create a topological structure (e.g. without duplication of walls) which would in turn be useful for applications such as indoor navigation. The resulting output also lacks the door and window geometry associated with any room, although these can also be separately extracted from the IFC data.

5.1 Suggested Guidelines for BIM Creation

Based on the initial work described in this paper, we make the following suggestions towards addressing technical aspects of the BIM/GIS integration challenges described in Sect. 2.2.

1. **Geo-reference the BIM from the outset.** The process described above, where the BIM was geo-referenced by identifying corresponding locations on a 2D Map, lacks the accuracy that could be obtained by geo-referencing the BIM using survey techniques on site.
2. **Ensure measurement units are set appropriately.** While construction engineering works in mm, GIS works in m.
3. **Select the appropriate level of detail for scanning.** The scans used to create the Chadwick BIM provided at least 8 mm sampling density, which was not required for the GIS analysis proposed by this project and caused issues in terms of capture time. A very high sampling density and resulting detailed scans means that the resulting geometry can be utilised for multiple purposes, with only relevant geometry being retained in each model, there is an overhead in terms of storage and geometry capture time.
4. **Ensure consistent tagging of the geometry.** On successful conversion from BIM to GIS queries to, for example, find the thickness of the glazing of individual windows in rooms identified as ‘cold’ by their occupants are possible. However, this depends on appropriate tagging of individual objects within the BIM creation process. As shown in Fig. 5, this tagging also needs to be consistent to allow algorithms for room reconstruction to be run within the GIS environment.
5. **Capture IFCSpace Objects.** The work described above highlights the benefits of utilising IFC Spaces to convert data between BIM and GIS. However, to achieve this both the spaces and the space boundaries must be correctly modelled in Revit prior to conversion into GIS format. They must also be recognised by the conversion software (IFCspace recognition was introduced in FME 2014 and in Revit 2015 spaces can also be automatically created (Autodesk 2015b) but this still requires the capture of fully enclosed spaces as part of the BIM creation process.

5.2 Further Work

The example in this paper highlighted the potential of combining data captured via a BIM process with other GIS data sources, to address real-world sustainability issues relating to air quality and noise, as well as producing a first list of guidelines for BIM creation that will facilitate integration. We fully acknowledge that this is a very preliminary list, and plan to add to it in the future, ideally with contributions from others working in this field.

Further work will involve working with a more detailed BIM, perhaps including information relating to HVAC systems, and developing an additional facilities-management case study. Correct tagging of objects within a BIM is also important, and automated tools may be required to validate consistency as the size and scope of many BIM is beyond the capabilities of manual checking.

A final, longer term goal involves more direct integration—the current approach to migration (export to IFC then import into a spatial database) means that changes to the BIM are not propagated through, which may cause issues of inconsistency. Developing similar case studies will contribute towards a more in-depth investigation into the question of which features (both geometry and attributes) should be shared between a very detailed BIM and the GIS or vice versa.

Acknowledgments The authors would like to thank UCL students Keyan Guo and Sai Ho Wong for their efforts in collecting the air quality and noise data sets used in this project, and the many students who captured the UCL BIM data.

References

- Amirebrahimi, S., Rajabifard, A., Mendis, P., & Ngo, T. (2015). A data model for integrating GIS and BIM for assessment and 3D visualisation of flood damage to building. *Locate*, 15, 10–12.
- Autodesk. (2015a). Autodesk Revit. <http://www.autodesk.co.uk/products/revit-family/overview>.
- Autodesk. (2015b). IFC Space Creation in Revit 2015. <http://tinyurl.com/nonqptb>.
- Backes, D., Ramirez, C., Thomson, C., Bordbar, N., O'Brien, B., Kooner, K., et al. (2012). The Chadwick Green BIM, University College London.
- Backes, D., Thomson, C., Malki-Epshtein, L., & Boehm, J. (2014). Chadwick GreenBIM Advancing Operational Understanding of Historical Buildings with BIM to Support Sustainable Use.
- Bishr, Y. (1998). Overcoming the semantic and other barriers to GIS interoperability. *International Journal of Geographical Information Science*, 12(4), 299–314.
- Building Information Modelling Task Group. (2013). Frequently Asked Questions. <http://www.bimtaskgroup.org/bim-faqs/>.
- Building Smart. (2015a). IFC Release Specifications. <http://www.buildingsmart-tech.org/ifc/IFC4/Add1/html/link/annex-b.htm>.
- Building Smart. (2015b). IFC Space. <http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/ifcproductextension/lexical/ifcspac.htm>.
- Cabinet Office. (2011). Government Construction Strategy. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/61152/Government-Construction-Strategy_0.pdf.

- Casey, M. J., & Vankadara, S. (2010). Semantics in CAD/GIS Integration. *CAD and GIS Integration*, 143.
- Cowen, D. J. (1988). GIS versus CAD versus DBMS: What are the differences? *Photogrammetric Engineering and Remote Sensing*, 54(11), 1551–1555.
- El-Mekawy, M., Östman, A., & Hijazi, I. (2012). An evaluation of IFC-CityGML unidirectional conversion. *International Journal of Advanced Computer Science and Applications*, 3(5), 159–171.
- El-Mekawy, M., Östman, A., & Shahzad, K. (2011). Towards interoperating cityGML and IFC building models: a unified model based approach. *Advances in 3D geo-information sciences* (pp. 73–93). Springer.
- El Meouche, R., Rezoug, M., & Hijazi, I. (2013). Integrating and managing BIM in GIS, software review. *International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences*, 2, W2.
- Hijazi, I., Ehlers, M., Zlatanova, S., Becker, T., & van Berlo, L. (2011). Initial investigations for modeling interior Utilities within 3D Geo Context: Transforming IFC-interior utility to CityGML/UtilityNetworkADE. *Advances in 3D Geo-information sciences* (pp. 95–113). Springer.
- Irizarry, J., Karan, E. P., & Jalaei, F. (2013). Integrating BIM and GIS to improve the visual monitoring of construction supply chain management. *Automation in Construction*, 31, 241–254.
- Jernigan, F. (2008). Big BIM, Little BIM. The practical approach to building information modelling. Integrated Practice done the right way!.
- Kemp, A. (2011). BIM isn't Geospatial ... Or is it?. https://communities.rics.org/gf2.ti/f/200194/6768101/pdf/-/RICS12Pres_ACK_Speech.pdf.
- Kolbe, T. H., Gröger, G., Plümer, L. (2005). CityGML: Interoperable access to 3D city models. *Geo-information for disaster management* (pp. 883–899). Springer.
- Loreto, V. (2010). D1.1: Report on: sensor selection, calibration and testing; EveryAware platform; smartphone applications. http://www.everyaware.eu/resources/deliverables/D1_1.pdf.
- Loreto, V. (2012). The EveryAware White Paper.
- Loreto, V. (2013). D3.1: Report on the EveryAware platform performance in the Pilot Studies.
- Loreto, V. (2014). D1.2: Final report on: sensor selection, calibration and testing; EveryAware platform; smartphone applications.
- Oracle (2013). Oracle Spatial and Oracle Locator. <http://www.oracle.com/uk/products/database/options/spatial/index.html>.
- Pu, S., & Zlatanova, S. (2006). Integration of GIS and CAD at DBMS level. In *Proceedings of UDMS* (vol. 6, pp. 9–61).
- Safe Software (n.d.). FME Desktop Software. http://cdn.safe.com/resources/brochures/FME_Desktop.pdf.
- Sebastian, R., Böhms, H., & Helm, P. (2013). BIM and GIS for Low-disturbance Construction. Sustainable Development Commission. (2011). Built environment.
- Tobiáš, P. (2015). An investigation into the possibilities of BIM and GIS cooperation and utilization of GIS in the BIM process. *Geoinformatics FCE CTU*, 14(1), 65–78.
- Volk, R., Stengel, J., & Schultmann, F. (2014). Building Information Modeling (BIM) for existing buildings-Literature review and future needs. *Automation in Construction*, 38, 109–127.
- Worboys, M., & Duckham, M. (2004). *GIS: A computing perspective*, CRC Press LLC.
- Zeiss, G. (2013). Building on BIM. <http://geospatialworld.net/paper/cover-stories/ArticleView.aspx?aid=30648>.
- Zhang, X., Arayici, Y., Wu, S., Abbott, C., & Ouad, G. (2009). Integrating BIM and GIS for large scale (building) asset management: a critical review. In *The Twelfth International Conference on Civil Structural and Environmental Engineering Computing*.
- Zlatanova, S., & Prosperi, D. (2006). Bridging the Worlds of CAD and GIS, *Large-scale 3D data integration: challenges and opportunities*.