



# Concepts and Challenges for 4D Point Clouds as a Foundation of Conscious, Smart City Systems

Ole Wegen<sup>(✉)</sup> , Jürgen Döllner, and Rico Richter

Hasso Plattner Institute, Prof.-Dr.-Helmert-Straße 2-3, 14482 Potsdam, Germany  
{ole.wegen,juergen.doellner,rico.richter}@hpi.uni-potsdam.de  
<https://www.hpi.de>

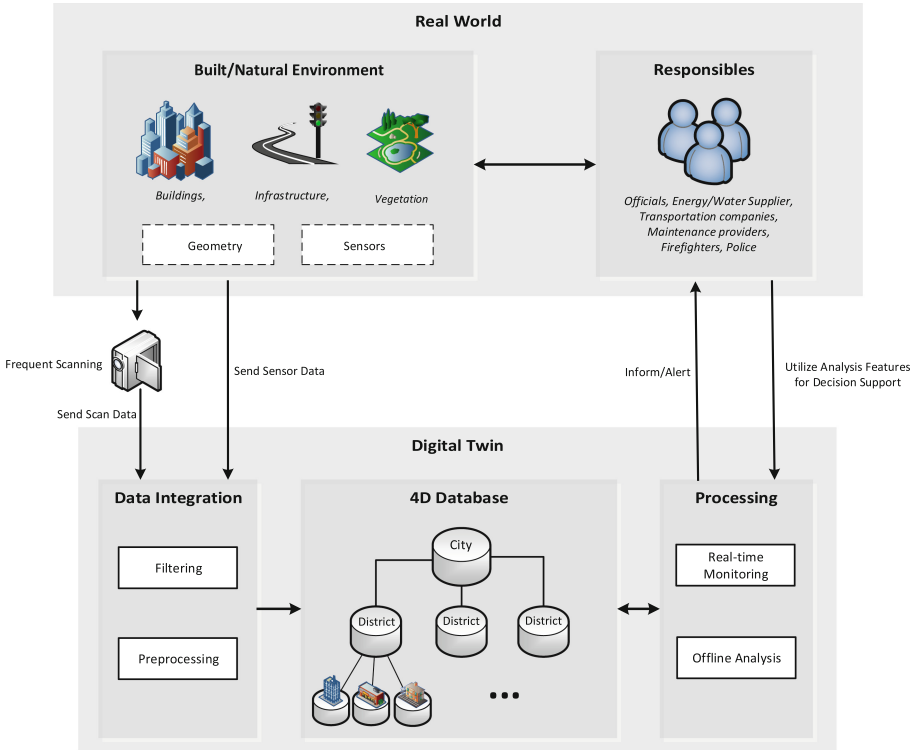
**Abstract.** Point clouds represent the as-is geometry of indoor and outdoor environments by sets of 3D points. They allow for constructing 3D models of objects, sites, cities, and landscapes and, hence, form the base data for almost any conscious, smart city system and application. For implementing such systems, we need a spatio-temporal data structure that enables efficient storage and access to 4D point clouds. In particular, the data structure should allow continuous updates, change tracking, and support for spatial and spatio-temporal analysis. This paper discusses challenges and approaches for a 4D point cloud data structure. In particular, the challenges arise from repeated scanning of environments in terms of sparsity, data redundancy, and geometric blurring of the corresponding point clouds. We outline a scheme for incremental storage of 4D point clouds via signed distance fields using a sparse, voxel-based representation. To efficiently implement analysis operations, we discuss how the data structure supports access based on both spatial and temporal criteria. In particular, we outline how machine learning-based interpretations used to classify point clouds and derive object-based information can work with the data structure.

**Keywords:** Point clouds · Spatio-temporal data · Smart city database

## 1 Introduction

The concept of smart city has been studied extensively in the last decades, but gained popularity recently due to the increase in inexpensive Internet of Things (IoT) devices. Smart cities build on the idea that “[...] cities are systems of systems, and that there are emerging opportunities to introduce digital nervous systems, intelligent responsiveness, and optimization at every level of system integration - from that of individual devices and appliances [...] to that of buildings, and ultimately to that of complete cities and urban regions” [20].

In this context, digital twins, as virtual representation of real-world objects, have been perceived as the “ultimate technological apparatus for ‘smartening’ cities” [35]. Such digital twins are tied together with their physical counterpart by means of data connections [8], enabling the monitoring, analysis, and control



**Fig. 1.** Real-world data obtained by fixed sensors or frequent scanning of the environment is used to update the spatial data base. It reflects the state of the city in a hierarchical, object-structured manner. Real-time monitoring of the current state can inform responsible parties (e.g., if the breakout of a fire is detected, an unexpected increase in energy or water usage emerges, or failure of devices occur). Additionally, analysis features can operate on top of the database (e.g., for traffic flow monitoring or vegetation analysis).

of the corresponding object in the real world. Therefore, digital twins fulfill a key role in enabling future smart cities. Figure 1 depicts the integration of a digital twin into the city environment and existing responsibility structures.

The spatial representation of the geometric realities of physical objects is an important feature of the digital twin. Brilaikis *et al.* perceive this geometric information “as a starting point for a comprehensive digital twin” [3]. For constructing such virtual, spatial representation of cities and environments, 3D point clouds can serve as the base data, as they are a common artifact of as-is environment capturing of both indoor and outdoor scenes and there exist cost-efficient methods for acquiring them [21]. In general, 3D point clouds are unorganized, unstructured, but geo-referenced sets of points. Multiple 3D point clouds that share a common geographic extent, but were captured at different points in time (i.e., spatio-temporal datasets), are denoted 4D point clouds. The

increasing degree of automation regarding the data acquisition and integration processes enables nowadays the capturing of such 4D point clouds by means of frequent environment scanning (e.g., using devices equipped with Light Detection and Ranging (LiDAR)-sensors or camera systems used to capture image data for photogrammetric point cloud creation). As 4D point clouds capture the history (i.e., changes over time) of the environment and provide the data used to derive object-related and structure-related information, they are the key component for constructing and maintaining a digital twin and operating conscious, smart city systems, services, or applications.

However, the question arises, how to store this massive, city-scale, spatio-temporal data, while at the same time enabling access for a broad range of analysis features, such as object or change detection, required for operating a smart city system. This work outlines the requirements and challenges for constructing a suitable data structure and proposes a first concept.

Section 2 reviews related work for smart cities, digital twins, and 4D point clouds. We then outline the key requirements and challenges for a spatio-temporal data structure in the context of smart cities in Sect. 3. Subsequently, we propose a scheme for storing 4D point cloud data via signed distance fields (Sect. 4) and describe the support for spatial and spatio-temporal access. Additionally, we outline how machine learning-based analysis approaches can work with the data structure. Finally, we conclude this work in Sect. 5.

## 2 Related Work

To understand the requirements for a spatio-temporal data structure in the context of smart cities, the application scenarios for smart cities, the use-cases and characteristics of digital twins, and the challenges connected to 4D point clouds are outlined in the following.

### 2.1 Smart Cities

The application scenarios for smart cities include real-time monitoring, process optimization, as well as intelligent responsiveness and control, across the whole city hierarchy from single buildings, over infrastructure networks, to whole urban areas. Key application areas of smart cities include energy, water, mobility, buildings, vegetation, and government [19] with use cases such as “enhanced street lighting controls, infrastructure monitoring, public safety and surveillance, physical security, gunshot detection, meter reading, and transportation analysis and optimization systems [...] on a city-wide scale”. Ulusoy and Mundy describe “urban growth analysis, construction site monitoring, natural resource management, surveillance and event analysis” as common applications for 4D real-world data [34]. The direct relation of these applications to the smart city concept shows the importance of 4D data in this context. Daniel and Doran stress the importance of geomatics for smart cities. They state that “location is [...] core information” and the “geographical characteristics [...] and spatial understanding capabilities participate significantly in the design and operation of Smart

City service and management infrastructures” [6]. This emphasizes the need for geometric digital twins as key components of smart city systems.

## 2.2 Digital Twins

**Table 1.** Overview of use-cases and characteristics of digital twins, according to previous work in this area.

Publication	Characteristics	Use-cases
M. Grieves, 2014 [8]	Virtual objects linked to physical objects through data connections	Comprehension through conceptualization; Comparison of as-is and should-be; Collaboration
A. El Saddik, 2018 [7]	Seamless data transmission between physical and virtual world; Unique identifier for communication; Sensors and actuators on the real-world object; AI for intelligent decisions; Virtual representation	Monitor, understand, and optimize the functions of all physical entities to improve health and well-being
Qi <i>et al.</i> , 2018 [25]	Virtualization of physical entities; Ability to guide/optimize the physical process	Virtual verification; Simulation; High-fidelity real-time monitoring; Predicting and diagnosing problems; Optimizing and improving processes
Brilakis <i>et al.</i> , 2019 [3]	Representation of current asset condition; Visualization of information and simulation results; Regular updates	Asset condition monitoring; Facility management and operation; Decision making; Sustainable development
Wan <i>et al.</i> , 2019 [35]	Progressive development; Machine learning techniques; Spatial/temporal resolution corresponding to use-case	Planning, constructing and managing national infrastructure; Optimising use of resources; Reducing service disruption; Increasing resilience; Boosting quality of life for citizens
Minerva <i>et al.</i> , 2020 [18]	Mirror the physical object; Register status changes; Store historic status changes; Simulate behaviour over time	Enabling virtual objects and smart cities

Grieves describes a concept of a digital twin for manufacturing, defining a digital twin as a three-part entity comprising the physical objects of the real world, the virtual objects in virtual space, and the data connections that links these two together [8]. Three use-cases for digital twins are presented: Comprehension is improved by eliminating inefficient mental steps through conceptualization, comparison between the as-is and the should-be state is facilitated, and collaboration becomes feasible due to a shared conceptualization. In smart manufacturing, the

digital twin is described as “the virtualization of physical entities”, guiding “the physical process to perform the optimized solution” [25]. In particular, the seamless data transmission between physical and virtual world is core feature of the digital twin concept, which El-Saddik extends to living beings with the goal to improve health and well-being [7]. In contrast, Khajavi *et al.* study the application of the digital twin concept to buildings [12]. Perceiving the developments in the area of IoT as the driving factor that enables the creation of digital twins, sensoric parts are considered as key components for operating a digital twin. For digital twin construction, geometric and structural information from a Building Information Model (BIM) is combined with a wireless sensor network and data analytics; for visualization, information extracted from the BIM or a custom 3D model is used. However, this is based on the assumption that the as-is geometry of the model still corresponds to the real-world situation; the changes over time are not considered. With respect to this, Brilakis *et al.* state that a digital twin “should be updated regularly in order to represent the current condition of the physical asset”, enabling real-time monitoring [3]. Challenges for updating, maintaining, and operating geometric digital twins include occlusion during environment capturing, effective visualization of complex information and simulation results, and finding a lightweight, scalable, stable, and exchangeable geometric representation.

Several publications describe the applications and required characteristics of digital twins. Table 1 gives an overview of these, as described in literature. Especially the tight data connection between physical and virtual object is emphasized. Regular updates of the digital twin are required to mirror the state of the real world and to enable real-time asset monitoring. Process optimization and prediction/simulation for supporting decision making are also commonly stated use-cases.

Minerva *et al.* discuss basic properties of digital twins, including [18]:

- **Reflection:** A digital twin mirrors behaviour and status of the physical object.
- **Entanglement:** A digital twin is connected to the physical object to register status changes.
- **Memorization:** A digital twin stores all the historical status changes that occurred to the physical object.
- **Predictability:** A digital twin has the ability to simulate behaviour over time.

As spatial and georeferenced digital twins operate over time, underlying spatio-temporal data structures are essential system components for reflecting and memorizing changes, integrating continuous updates, and enabling simulation and analysis.

Regarding the boundary conditions for building city-level digital twins, “the spatial/temporal resolution of the digital twin should be informed by the purpose it serves” [35]. As digital twins serve multiple purposes, they require different

spatial and temporal resolutions. To this end, spatio-temporal data structures should support Level-of-Detail (LoD).

### 2.3 4D Point Clouds

3D point clouds have become one of the most prominent geospatial data formats. Stojanovic *et al.* present a workflow for data acquisition to generate as-is BIM datasets, using regular 3D point cloud capturing to represent structural and spatial features of the environment [30]. However, “although the 3D point cloud is very practical, the huge data volume of a 3D point cloud limits its extensive applications” [15], that is, 3D point clouds always require post processing to be handled by applications or systems (e.g., compression or LoD).

The extension of 3D point clouds to the temporal domain, i.e., the capturing and reconstruction of the environment at different points in time, increases the data volume further. Thus, 4D point clouds are faced by massive storage requirements, in particular due to the high degree of redundancy leading to inefficient use of storage capacity, e.g., in scenarios where the environment changes only slightly between two points in time. Ulusoy *et al.* argue similarly that storing a 3D model in every timestamp “does not scale well in dealing with thousands of frames of data” [33] and Milani *et al.* state that dynamic point clouds are “highly inefficient in terms of storage space” [16]; This unanimous view of 4D point clouds in literature shows that storage and access issues become crucial for such data and a suitable data structure is urgently required. Additionally, visualizing such massive amounts of data is challenging and requires out-of-core approaches, as presented by Richter *et al.* for massive (city-scale) point clouds [27].

Apart from the storage issue, point clouds pose several other issues concerning noise levels (due to illumination, motion, and sensor noise), sparsity, and uneven distribution [4]. These issues can lead to inconsistent reconstruction of the same object at different timestamps and therefore impede 4D analysis, as the differentiation “between actual changes in the scene from false alarms caused by inconsistent reconstructions” is complicated [34].

To summarize, 4D point clouds as supposed to be the data basis for conscious, smart city models, are faced by a number of challenges:

- The huge overall data volume complicates efficient storage. Redundancy in the spatial and temporal domain is one part of this problem.
- The sparsity and uneven distribution of 3D points in each sample makes direct comparison of point clouds difficult, complicating analysis. Additionally, the implementation of incremental storage schemes for a more compact memory representation are also complicated.
- Inconsistencies due to error-prone acquisition and reconstruction processes as well as random noise hinder the analysis.
- The inability to control the LoD of a 4D point cloud meaningfully stands contrary to the fact that different analysis approaches require different LoD in the spatial or temporal domain. Storing multiple LoDs for point clouds

directly by removing high frequencies per level either involves a decrease in point cloud density, or an increase in memory consumption (e.g., smoothing the point cloud with different strength at each level and storing the results multiplies the required memory by the number of LoDs).

### 3 Challenges for Spatio-Temporal Data Structures

Based on the observations in Sect. 2, we now derive requirements and challenges for a 4D data structure in the context of smart cities. While some of the requirements complement each other, other requirements conflict. In practice, an appropriate trade-off between the different requirements has to be found, considering the concrete application scenario and use-cases.

#### 3.1 Requirements and Challenges for a 4D Data Structure

**Compact Memory Representation.** A 4D data structure for smart city applications should be able to represent the current state, as well as past states, of physical objects. The spatial and temporal redundancy should be exploited for enabling a compact memory representation and reducing storage requirements. Incremental storage approaches are, however, faced with the challenge of geometric fuzziness and inconsistencies in the data.

**High Spatial Scale and Sufficient Detail.** As we are considering whole buildings, roads, and even cities, a data structure should be able to handle this high spatial scale. At the same time, sufficient detail is required, for the data to be useful for representing and analyzing the real world. This requirement stands in contrast to the compact memory representation. Out-of-core streaming approaches may be required to realize digital twin construction and operation on a city-wide scale. A corresponding data structure therefore has to be able to support this.

**Fast Access.** A data structure has to provide fast access in the spatial and temporal domain for real-time visualization, continuous monitoring, and efficient spatio-temporal analysis. Especially the task of change detection is one of the core tasks in the context of digital twins for smart city applications. Fast access often conflicts with a compact memory representation.

**LoD Support.** A data structure should support LoD approaches for access, as “the spatial/temporal resolution of the digital twin should be informed by the purpose it serves. [...] not all digital twins have to aim at real-time, nor the finest spatial unit of analysis. For city and infrastructure planning, the resolution of a digital twin model should be informed by the scale/rate of change of the policy question” [35]. Additionally, the hierarchical nature of the smart city has to be considered: buildings forming a site, sites forming a district, districts forming a city. Depending on the use-case, a different hierarchy level may be required.

**Support for Offline Analysis.** Tasks such as measuring key figures or optimizing processes through simulation and prediction build on (offline) analysis methods. For 4D point clouds, a broad range of analysis methods is already available. A data structure should support interoperability with these existing and well-working analysis approaches.

**Fast Construction and Data Integration.** A data structure should be able to integrate new data that is recorded from the real world on a regular basis in a fast manner. As near real-time monitoring is one of the use-cases for digital twins in smart city applications, the speed for integration of new data should be suitable for this task. The initial construction of the data structure does not necessarily need to be very fast, but it would be a beneficial trait.

**Compression.** If data is archived or should be transmitted over a network, the size of the data needs to be as small as possible; fast access is not required in these scenarios. For these use-cases, a data structure should support compression methods for reducing the size of the data.

**Semantics.** The data structure should be able to store point-specific attributes in addition to the geometric properties (i.e., point coordinates). These attributes could result from analysis and classification operations and can be used to access subsets of the point cloud belonging to a defined surface category (e.g., ground, building, vegetation, road). Further, such semantic classification can be used to separate dynamic objects, such as pedestrians or cars, from the static scene, which is an important task for meaningful change detection.

**Graphics Processing Unit (GPU) Support.** To enable efficient analysis on the data structure and real-time visualization of the data, GPU-based approaches are necessary. These require that the data structure is manageable on the GPU and that efficient streaming methods exist for loading the currently relevant parts of the data.

### 3.2 Existing Spatio-temporal Storage Approaches

Section 2 made clear that a huge problem of 4D point clouds, which are the common artifact of as-is environment capturing, is the high memory footprint. To reduce the storage requirements, compression methods have been proposed, which exploit the redundancy in the temporal domain. Most approaches use an octree, containing the voxelized point cloud, as intermediate representation for compression.

Thanou *et al.* present a compression scheme for dynamic point clouds, based on interpreting leaf nodes of an octree as graphs and position and color attributes as signals on the graph [32]. Queiroz and Chou present a lossy compression for dynamic point clouds using block-wise motion compensation [26]. A voxelized



point cloud is split into blocks and each “block is either encoded in intra-frame mode or is replaced by a motion-compensated version of a block in the previous frame”. Milani *et al.* presented a 4D point cloud compression scheme based on a voxelization and cellular automata transforms, tailored to the statistics of the data [16]. The MPEG have launched the standardization of point cloud compression in 2017. Liu *et al.* evaluate the proposed point cloud compression approaches in extensive experiments [15]. The results show that point cloud encoding can take several seconds or even minutes (for dynamic point clouds).

These compression methods have in common that a point cloud has to be decompressed again in order to access the single points. Therefore, while point cloud compression provides a solution to the high storage requirements, other challenges, such as fast data integration or fast data access, are not solved. Additionally, most of the compression methods were tested on scenes with low spatial extent.

With respect to the challenges of high spatial scale and sufficient detail, Blaha *et al.* proposed a hierarchical scheme for reconstruction of large scale scenes (whole cities). An implicit volumetric representation is used that supports variable volumetric resolution, refining the reconstruction adaptively only near surfaces in order to save memory while maintaining sufficient detail [1].

Regarding compact memory representation with fast data integration and access, Miller *et al.* presented an approach to reconstruct and dynamically update a 3D model from images [17]. They use a hybrid representation consisting of a regular grid and a shallow octree per grid cell. This data structure stores a probabilistic, volumetric representation of the 3D model, i.e., each cell stores the probability of being a surface “to represent the ambiguity in reconstruction of surface from images”.

Based on the work of Miller *et al.*, Ulusoy and Mundy present an image-based method to update a reconstructed 3D model of a real-world object only when a change is detected in images at a later time [34]. “The resulting 4-d models allow visualization of the full history of the scene from novel viewpoints[...], as well as spatio-temporal analysis for applications such as tracking and event detection”. They use the data structure proposed in their previous work [33]: a grid of octrees for spatial decomposition, and binary trees for modeling temporal variation per cell of the volumetric model. “This representation is shown to achieve compression of 4-d data and provide efficient spatio-temporal processing”. The proposed data structure seems to fulfill many of the challenges described in Sect. 3.1. It provides a compact memory representation, fast spatio-temporal access, easy change detection, GPU support, and possibly efficient compression. Nevertheless, some challenges remain. Interoperability with existing approaches is not given and chunking the data along the temporal dimension (for out-of-core or LoD approaches) is difficult due to the nested structure. The scalability to city-level scenes and long timespans is also in question, as in their approach the tree depth is limited to enable fast GPU-based processing.

## 4 Concept for an Incremental Spatio-temporal Data Structure

In the following we describe the concept for a spatio-temporal data structure to be used for smart city applications. To provide a compact memory representation, the temporal redundancy has to be eliminated. Therefore, the basic idea is to store a base geometry  $G_0$  for a timestamp  $t_0$ , while for each subsequent timestamp  $t_i, i > 0$ , only the changes  $C_i$  compared to the previous timestamp  $t_{i-1}$  are stored (similar to Laplacian pyramids for images). As change detection is an important task in the context of smart cities, storing the changes explicitly is advantageous for monitoring and analysis.

### 4.1 The Problem of Using Point Clouds

As described in Sect. 1, 4D point clouds serve as the key component for applications in the area of smart cities. However, using a point cloud as base representation for the proposed data structure is problematic, as calculating changes is not trivial for two point clouds due to missing point-to-point correspondences. Lague *et al.* describe three basic approaches for point cloud change detection [13]: Grid-based approaches, approaches based on intermediate representations (e.g., a local plane or a mesh), or direct pointcloud-to-pointcloud comparison. Additionally, they proposed the nowadays well-known M3C2 algorithm for change detection, which is based on measuring the mean surface change along the surface normal for a number of representative core points. Nevertheless, point cloud change detection is still an active field of research. In addition to the problem of computing changes for point clouds, the question arises, how to store these changes, especially if the number of points varies between point clouds from different timestamps.

In summary, 4D point clouds, while easy to acquire, may not be the most efficient way to store spatio-temporal data. This has been underlined as well by other research in this area [17, 33]. We therefore propose, to use a voxel-based Signed Distance Function (SDF) representation that provides an interface for access similar to a point cloud and can also be converted back to a point cloud for faster processing, if necessary.

### 4.2 Voxel-Based, Signed Distance Field Representation

Point clouds are usually derived from LiDAR scans or RGB-D data. Using Truncated Signed Distance Field (TSDF) fusion, such data can also be used to reconstruct a voxel volume containing the signed distance to the surface for every voxel in vicinity to this surface [5, 10]. These signed distance fields have some desirable properties.

In contrast to point clouds, they provide an implicit, continuous surface, mitigating problems that are related to the sparsity of point clouds (e.g., rendering closed surfaces or performing spatial scaling). Further, their regularity makes them easy to process, e.g., in the context of compression. Due to their

regular nature, the distance values can be directly interpreted as signals on a 3D grid. Based on this, Jones presents a lossless compression method for distance fields [11]. Point clouds, on the other hand, have to construct intermediate data structures (e.g., octrees [15, 16] or graphs [32]). The advantage of the regularity of distance fields, e.g., for compression purposes, is also underlined by the fact that Laney *et al.* use a distance field as intermediate representation for mesh compression [14]. Additionally, the regularity of distance fields leads to the fact that they can be subtracted from each other, which is a useful property for incremental storage approaches that exploit the temporal redundancy of the data.

We therefore propose to use a TSDF volume as base representation in our data structure. However, storing the full voxel grid is memory inefficient, especially in light of the fact that a change  $C_i$  might only contains few relevant voxel cells. A sparse representation of the TSDF is therefore key to a compact storage format. Niessner *et al.* presented a spatial hashing approach for TSDFs, with a special focus on GPU-based, real-time reconstruction from depth data [22]. Their approach is well suited for sparse voxel grids that have to be updated very often. This hash-based approach seems to be suitable as a base data structure in the context of smart cities. It was developed for GPU-based processing, providing a compact memory representation through a sparse voxel structure, without the overhead of hierarchical structures. Fast access is achieved by means of a hash table, scalability in the spatial domain is ensured by out-of-core streaming.

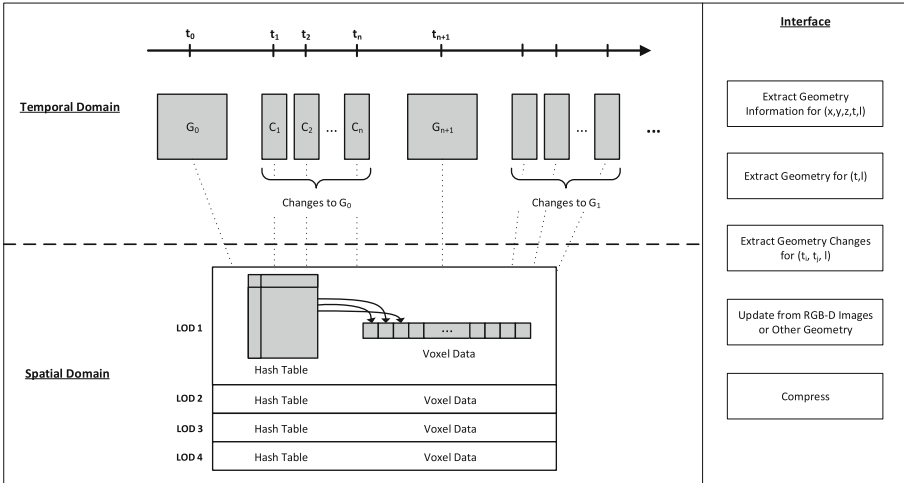
We therefore propose to use such a hash-based approach for storing TSDFs, representing the base geometry implicitly as a sparse field of distances. The changes can also be stored in the form of TSDFs, understood as distance fields in the temporal domain. Attributes, such as the color or semantic class, can also be derived from RGB-D images or point clouds and stored in the data structure for each voxel (in addition to the signed distance). Figure 2 gives an overview of our proposed data structure.

### 4.3 Concept for Data Integration

We now describe how new scan data can be integrated, using the proposed data structure. Given a base geometry  $G_0$  and changes up to this point  $C_1, \dots, C_i$ , all of them represented as TSDFs. Data recorded for timestamp  $t_{i+1}$  can be integrated in the following way:

1. **Calculate TSDF for timestamp  $t_{i+1}$ :** Calculate  $G_{i+1}$  directly from LiDAR or RGB-D data for best results. Alternatively, a TSDF can be deducted from a point cloud, e.g., using Jump Flooding [28] or neural approaches [23].
2. **Retrieve TSDF for  $t_i$ :** To retrieve TSDF  $G_i$ , accumulate the changes up to timestamp  $t_i$ :

$$G_i = G_0 + \sum_{n=1}^i C_n. \quad (1)$$



**Fig. 2.** Conceptual overview of the proposed spatio-temporal data structure. In the temporal domain, only changes to a base geometry are stored. For faster access, full geometry representations are stored in regular intervals. Each geometry or change is represented as a TSDF with different resolutions (LoDs), stored using a hashing approach for compact memory representation. The geometry can be updated using RGB-D images or other geometry representations (e.g., point clouds). This data structure enables the access of single points at any timepoint and LoD, as well as the export of the geometry at any timestamp  $t_i$ . Additionally, the changes at different LoDs can be inspected for analysis purposes. The TSDFs can be compressed easily (e.g., using wavelet transforms).

3. **Compute changes for  $t_{i+1}$  :**  $C_{i+1} = G_{i+1} - G_i$ . It has to be ensured that the spatial extent of both geometries is equal. Cases have to be handled, where voxels contain distance values in  $G_{i+1}$  but not in  $G_i$  or vice versa.

As two scans are never exactly the same, even if nothing has changed in the scene, approaches have to be developed to avoid storing “changes” that are none. A (local) threshold-based approach could be used, where the amount of change is measured and the change is only stored if it exceeds the threshold. However, it has to be considered that some things change slowly over time, e.g., plants.

To facilitate LoD approaches, the base geometry and the changes  $C_1 \dots C_n$  can also be stored in different granularity levels (TSDF resolutions). Similar to storing only the changes to the previous timestamp in the temporal domain, we store in the spatial domain for each LoD the change to the previous LoD. To obtain a representation for a specific LoD, the different levels have to be summed. The memory impact and the access times (especially for higher LoDs) for such an approach would have to be investigated.

#### 4.4 Concept for Spatio-temporal Access

Access in the spatial domain is achieved by means of a hash table, as described in [22]. Access in the temporal domain, requires accumulation of changes up to a certain timestamp. To access the distance value stored at a voxel  $v_i = (x, y, z, t_i)$ , the changes to this voxel have to be accumulated:

$$v_i = v_0 + \sum_{n=1}^i v_n. \quad (2)$$

The performance of this accumulation of changes is dependent on the access performance of the underlying spatial data structure that is used for storing the TSDFs.

A high resolution in the temporal domain could lead to decreased access performance, as possibly a lot of changes have to be accumulated. This problem can be mitigated by storing a full representation every  $x$  timestamps and calculating  $v_i$  as:

$$v_i = v_a + \sum_{n=a+1}^i v_n, \text{ with } a = \left\lfloor \frac{i}{x} \right\rfloor * x. \quad (3)$$

Alternatively, multiple changes could be squashed into one change representation after a certain valid-time for the recorded data has been exceeded.

#### 4.5 Support for Machine Learning

Machine learning is a fundamental building block for smart cities. “The advancement of data science, particularly the machine learning techniques, will complement existing theories of cities and infrastructure and jointly contribute to the essential knowledge for developing digital twins” [35]. Thus, a data structure has to support existing machine learning approaches.

Camuffo *et al.* give a comprehensive overview of deep learning approaches for point clouds in the areas of semantic scene understanding, compression, and completion. Additionally, they introduce typical data structures, acquisition approaches, and common point cloud data sets [4]. “When dealing with deep learning algorithms, point clouds are usually not the most suitable data structure to process. Thus, the input data are frequently subject to transformations that allow them to satisfy the specific needs of the architecture” [4]. Deep learning approaches either operate

- **Discretization-based:** The point cloud is transformed into a discrete structure, such as a voxel grid or an octree (e.g., SEGCloud [31], LatticeNet [29]).
- **Projection-based:** The point cloud is remapped to a simpler structure, such as multiview images, a sphere, or a cylinder (e.g., SnapNet [2], SqueezeSeg [36]).
- **Point cloud-based:** The neural net directly processes the points (e.g., PointNet [24], RandLA-Net [9]).

Using the proposed TSDF-based storage format, all of these deep learning input formats can be derived. A point cloud can be computed in a preprocessing step by retrieving the distance value for each surface voxel for a timestamp, computing the surface normals using central differences, and deriving the corresponding surface points. If no preprocessing should take place, the points can also be directly accessed without prior conversion to the full geometry. This involves, however, distance value accesses and normal calculation, every time a point is accessed. The ability to convert from TSDFs to point clouds and vice versa ensures the interoperability with existing analysis approaches. While the derived point cloud can then be converted to the other input formats, it is also possible to directly use the TSDF-based representation. As the proposed data structure is already regular and discrete it can be used directly for discretization-based approaches. However, for some deep learning approaches it might be required to convert the distances to occupancies and arrange the voxels in a hierarchical data structure. The TSDF-based data structure can also be used for projection, using ray marching methods. As a TSDF provides a continuous surface, the problem of holes, when using point clouds, is mitigated.

## 5 Conclusions

This work presented the digital twin as one of the key components for smart city applications. Such digital twin is the basis for real-time monitoring and offline analysis of the physical world. Therefore, it has to reflect the current status of the physical object, memorize the state changes, and offer functionality for simulation and prediction.

We focused on exploring the boundary conditions, requirements, and challenges of a data structure for spatio-temporal data in this context. We derived these requirements and challenges from the applications and characteristics of digital twins for use in smart city systems. Providing a high spatial scale and sufficient detail, while maintaining a compact memory representation is one of the main challenges for such a data structure. Further, fast access and LoD support for GPU-based analysis approaches is required. New data has to be integrated efficiently and existing analysis methods should be able to operate on the data structure. Compression approaches are useful for archiving and transmission of data.

These requirements should be considered, when searching for a data structure that is suited for the spatio-temporal data that we have to store and process for enabling smart cities.

While 4D point clouds are commonly used in the context of digital twins, as they are a typical artifact of environment scanning, they come with high storage requirements and deficiencies regarding LoD approaches. We therefore proposed to use a TSDF-based data structure, using an incremental storage scheme for exploiting redundancy in the temporal domain. The data structure has the following advantages:

- In comparison to storing single point clouds for each timestamp, the memory requirements are reduced by only storing changes to a base geometry.

This base geometry is represented as a TSDF using a hash-based approach, facilitating further compression.

- High spatial scale and high detail is supported by using a sparse distance field and disentangling the spatial and temporal domain, facilitating out-of-core approaches.
- Fast access is facilitated by avoiding deep hierarchical data structures.
- Interoperability with existing approaches is ensured as direct access to point data or conversion to other geometry representations is possible.
- LoD approaches for adapting to different resolution requirements are enabled by decomposing the geometry into different granularities.
- Explicitly storing the changes over time, enables efficient change detection, which is one of the main tasks for a digital twin in the context of smart cities.

The presented data structure is to be understood as a first proposal and conceptual step towards developing a base data structure for smart city applications. In the future, it has to be tested thoroughly and the performance with respect to the presented requirements, different data acquisition techniques, and different application domains has to be studied in more detail.

**Acknowledgements.** We thank the anonymous reviewers for their valuable feedback. This work was partially supported by the Federal Ministry of Education and Research (BMBF), Germany (PunctumTube, 01IS18090) and by the Federal Ministry for Digital and Transport (BMDV), Germany (TWIN4ROAD, 19F2210).

## References

1. Blaha, M., Vogel, C., Richard, A., Wegner, J.D., Pock, T., Schindler, K.: Large-scale semantic 3d reconstruction: An adaptive multi-resolution model for multi-class volumetric labeling. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3176–3184. IEEE Computer Society (2016). <https://doi.org/10.1109/CVPR.2016.346>
2. Boulch, A., Guerry, J., Saux, B.L., Audebert, N.: Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks. *Comput. Graph.* **71**, 189–198 (2018). <https://doi.org/10.1016/j.cag.2017.11.010>
3. Brilakis, I., et al.: Built environment digital twinning. Technical report, International Workshop on Built Environment Digital Twinning presented by TUM Institute for Advanced Study and Siemens AG (2019)
4. Camuffo, E., Mari, D., Milani, S.: Recent advancements in learning algorithms for point clouds: an updated overview. *Sensors* **22**(4) (2022). <https://doi.org/10.3390/s22041357>
5. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1996, pp. 303–312. Association for Computing Machinery (1996). <https://doi.org/10.1145/237170.237269>
6. Daniel, S., Doran, M.A.: Geosmartcity: geomatics contribution to the smart city. In: Proceedings of the 14th Annual International Conference on Digital Government Research, dg.o '13, pp. 65–71. Association for Computing Machinery (2013). <https://doi.org/10.1145/2479724.2479738>

7. El-Saddik, A.: Digital twins: the convergence of multimedia technologies. *IEEE Multim.* **25**(2), 87–92 (2018). <https://doi.org/10.1109/MMUL.2018.023121167>
8. Grieves, M.: Digital twin: manufacturing excellence through virtual factory replication. White paper **1**, 1–7 (2014)
9. Hu, Q., et al.: Randla-net: Efficient semantic segmentation of large-scale point clouds. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, pp. 11105–11114. Computer Vision Foundation/IEEE (2020). <https://doi.org/10.1109/CVPR42600.2020.01112>
10. Izadi, S., et al.: Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In: Pierce, J.S., Agrawala, M., Klemmer, S.R. (eds.) Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, pp. 559–568. ACM (2011). <https://doi.org/10.1145/2047196.2047270>
11. Jones, M.W.: Distance field compression. In: The 12-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2004, WSCG 2004, pp. 199–204 (2004)
12. Khajavi, S.H., Motlagh, N.H., Jaribion, A., Werner, L.C., Holmström, J.: Digital twin: vision, benefits, boundaries, and creation for buildings. *IEEE Access* **7**, 147406–147419 (2019). <https://doi.org/10.1109/ACCESS.2019.2946515>
13. Lague, D., Brodu, N., Leroux, J.: Accurate 3d comparison of complex topography with terrestrial laser scanner: Application to the rangitikei canyon (n-z). *ISPRS J. Photogramm. Remote. Sens.* **82**, 10–26 (2013). <https://doi.org/10.1016/j.isprsjprs.2013.04.009>
14. Laney, D.E., Bertram, M., Duchaineau, M.A., Max, N.L.: Multiresolution distance volumes for progressive surface compression. In: 1st International Symposium on 3D Data Processing Visualization and Transmission (3DPVT 2002), pp. 470–479. IEEE Computer Society (2002). <https://doi.org/10.1109/TDPVT.2002.1024102>
15. Liu, H., Yuan, H., Liu, Q., Hou, J., Liu, J.: A comprehensive study and comparison of core technologies for MPEG 3-d point cloud compression. *IEEE Trans. Broadcast.* **66**(3), 701–717 (2020). <https://doi.org/10.1109/TBC.2019.2957652>
16. Milani, S., Polo, E., Limuti, S.: A transform coding strategy for dynamic point clouds. *IEEE Trans. Image Process.* **29**, 8213–8225 (2020). <https://doi.org/10.1109/TIP.2020.3011811>
17. Miller, A., Jain, V., Mundy, J.L.: Real-time rendering and dynamic updating of 3-d volumetric data. GPGPU-4. Association for Computing Machinery (2011). <https://doi.org/10.1145/1964179.1964190>
18. Minerva, R., Crespi, N.: Digital twins: properties, software frameworks, and application scenarios. *IT Prof.* **23**(1), 51–55 (2021). <https://doi.org/10.1109/MITP.2020.2982896>
19. Minoli, D., Sohraby, K., Occhiogrosso, B.: Iot considerations, requirements, and architectures for smart buildings - energy optimization and next-generation building management systems. *IEEE Internet Things J.* **4**(1), 269–283 (2017). <https://doi.org/10.1109/JIOT.2017.2647881>
20. Mitchell, W.J.: Smart cities: Vision. <https://smartcities.media.mit.edu/frameset.html>. Accessed 11 May 2022
21. Nebiker, S., Bleisch, S., Christen, M.: Rich point clouds in virtual globes - a new paradigm in city modeling? *Comput. Environ. Urban Syst.* **34**(6), 508–517 (2010). <https://doi.org/10.1016/j.compenvurbsys.2010.05.002>
22. Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M.: Real-time 3d reconstruction at scale using voxel hashing. *ACM Trans. Graph.* **32**(6), 169:1–169:11 (2013). <https://doi.org/10.1145/2508363.2508374>



23. Park, J.J., Florence, P., Straub, J., Newcombe, R.A., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, pp. 165–174. Computer Vision Foundation/IEEE (2019). <https://doi.org/10.1109/CVPR.2019.00025>
24. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: deep learning on point sets for 3d classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. pp. 77–85. IEEE Computer Society (2017). <https://doi.org/10.1109/CVPR.2017.16>
25. Qi, Q., Tao, F.: Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. IEEE Access **6**, 3585–3593 (2018). <https://doi.org/10.1109/ACCESS.2018.2793265>
26. de Queiroz, R.L., Chou, P.A.: Motion-compensated compression of dynamic voxelized point clouds. IEEE Trans. Image Process. **26**(8), 3886–3895 (2017). <https://doi.org/10.1109/TIP.2017.2707807>
27. Richter, R., Discher, S., Döllner, J.: Out-of-Core Visualization of Classified 3D Point Clouds, pp. 227–242. Springer International Publishing, Cham (2015). [https://doi.org/10.1007/978-3-319-12181-9\\_14](https://doi.org/10.1007/978-3-319-12181-9_14)
28. Rong, G., Tan, T.S.: Jump flooding in gpu with applications to voronoi diagram and distance transform, vol. 2006, pp. 109–116, January 2006. <https://doi.org/10.1145/1111411.1111431>
29. Rosu, R.A., Schütt, P., Quenzel, J., Behnke, S.: Latticenet: fast point cloud segmentation using permutohedral lattices. In: Toussaint, M., Bicchi, A., Hermans, T. (eds.) Robotics: Science and Systems XVI, Virtual Event (2020). <https://doi.org/10.15607/RSS.2020.XVI.006>
30. Stojanovic, V., Trapp, M., Richter, R., Hagedorn, B., Döllner, J.: Towards the generation of digital twins for facility management based on 3d point clouds. In: 34th Annual ARCOM Conference, pp. 270–279 (2018)
31. Tchapmi, L.P., Choy, C.B., Armeni, I., Gwak, J., Savarese, S.: Segcloud: Semantic segmentation of 3d point clouds. In: 2017 International Conference on 3D Vision, 3DV 2017, pp. 537–547. IEEE Computer Society (2017). <https://doi.org/10.1109/3DV.2017.00067>
32. Thanou, D., Chou, P.A., Frossard, P.: Graph-based compression of dynamic 3d point cloud sequences. Trans. Img. Proc. **25**(4), 1765–1778 (2016). <https://doi.org/10.1109/TIP.2016.2529506>
33. Ulusoy, A.O., Biris, O., Mundy, J.L.: Dynamic probabilistic volumetric models. In: IEEE International Conference on Computer Vision, ICCV 2013, pp. 505–512. IEEE Computer Society (2013). <https://doi.org/10.1109/ICCV.2013.68>
34. Ulusoy, A.O., Mundy, J.L.: Image-based 4-d reconstruction using 3-d change detection. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8691, pp. 31–45. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10578-9\\_3](https://doi.org/10.1007/978-3-319-10578-9_3)
35. Wan, L., Nochta, T., Schooling, J.: Developing a city-level digital twin - propositions and a case study. In: International Conference on Smart Infrastructure and Construction 2019, ICSIC 2019: Driving Data-Informed Decision-Making, pp. 187–193 (01 2019). <https://doi.org/10.1680/icsic.64669.187>
36. Wu, B., Wan, A., Yue, X., Keutzer, K.: Squeezeseg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3d lidar point cloud. In: 2018 IEEE International Conference on Robotics and Automation, ICRA 2018, pp. 1887–1893. IEEE (2018). <https://doi.org/10.1109/ICRA.2018.8462926>