PFG (2023) 91:273–294
https://doi.org/10.1007/s41064-023-00243-1

ORIGINAL ARTICLE

VOX2BIM+ - A Fast and Robust Approach for Automated Indoor Point Cloud Segmentation and Building Model Generation

Jan Martens[1] · Jörg Blankenbach[1]

Received: 30 September 2022 / Accepted: 19 April 2023 / Published online: 30 May 2023

Abstract
Building Information Modeling (BIM) plays a key role in digital design and construction and promises also great potential for facility management. In practice, however, for existing buildings there are often either no digital models or existing planning data is not up-to-date enough for use as as-is models in operation. While reality-capturing methods like laser scanning have become more affordable and fast in recent years, the digital reconstruction of existing buildings from 3D point cloud data is still characterized by much manual work, thus giving partially or fully automated reconstruction methods a key role. This article presents a combination of methods that subdivide point clouds into separate building storeys and rooms, while additionally generating a BIM representation of the building's wall geometries for use in CAFM applications. The implemented storeys-wise segmentation relies on planar cuts, with candidate planes estimated from a voxelized point cloud representation before refining them using the underlying point data. Similarly, the presented room segmentation uses morphological operators on the voxelized point cloud to extract room boundaries. Unlike the aforementioned spatial segmentation methods, the presented parametric reconstruction step estimates volumetric walls. Reconstructed objects and spatial relations are modelled BIM-ready as IFC in one final step. The presented methods use voxel grids to provide relatively high speed and refine their results by using the original point cloud data for increased accuracy. Robustness has proven to be rather high, with occlusions, noise and point density variations being well-tolerated, meaning that each method can be applied to data acquired with a variety of capturing methods. All approaches work on unordered point clouds, with no additional data being required. In combination, these methods comprise a complete workflow with each singular component suitable for use in numerous scenarios.

Keywords BIM · Scan-to-BIM · Point cloud processing · Facility managment · Automated modeling

1 Introduction

Triggered by the digital transformation in all areas of society, business and administration, the introduction of Building Information Modeling (BIM) is fundamentally changing the processes in the construction industry. BIM describes 3D building geometries alongside semantic information such as technical properties and categorizations of building components. Spatial relations and hierarchies between objects are stored as well to provide additional context for exchange and maintenance (Borrmann et al. 2018). Despite being brought up mainly for digital planning, all life cycle phases benefit from end-to-end information management based on digital building models as a single source of truth, as they allow for updating and sharing a property's state in a transparent way between collaborators (Gao and Pishdad-Bozorgi 2019; Ròka-Madaràsz et al. 2016).

Building on this foundation, Computer-Aided Facility Management (CAFM; sometimes referred to as "BIM for FM") aims to provide all needed data of building assets in a computer-based information system where changes are continuously integrated, allowing facility managers to monitor and catalogue building assets, space usage and more.[1] Most CAFM processes need information related to geometry and spaces, like space management according to

✉ Jan Martens
jan.martens@gia.rwth-aachen.de

Jörg Blankenbach
blankenbach@gia.rwth-aachen.de

[1] Chair of Building Informatics & Geographic Information Systems, RWTH Aachen University, Mies-van-der-Rohe Str. 1, 52074 Aachen, Germany

[1] https://www.resonai.com/blog/what-is-cafm.

DIN 277, inventory occupancy rate, room reservations etc. (Braun et al. 2013; Nävy 2013) and therefore require up-to-date floor plan-like models with the spatial object relations and wall geometries. Typical management tasks like floor space estimation, documentation of retrofits and asset tracking are notably simplified by this paradigm, especially if models with the required room layouts, spatial relations and area footprints are regularly updated based on on-site data. BIM concepts implemented in IFC-like spatial definitions and hierarchies lend themselves notably well to solve these problems, hence they often suggested as data models for CAFM (Patacas et al. 2020). Ideally, this results in a digital twin of the building which covers the entire building life cycle starting from design to demolition (Deng et al. 2021). However, the crux of these BIM methods lies in the fact that most existing buildings have been constructed before the wide-spread use of digital models, with paper floor and design plans being preserved in best-case scenarios only and changes not being reflected in them. Consequently, such cases require the manual creation of BIM models by experts from surveying data (e.g. point clouds) captured on-site.

Like BIM, point cloud capturing devices have made large strides towards fast capturing of large data quantities (Bosché and O'Keeffe 2015). Especially terrestrial laser scanning (TLS) has become quite sophisticated, with capturing devices becoming more affordable, compact and user-friendly. Recent trends in the industry indicate that techniques like mobile laser scanning (MLS) are becoming increasingly popular (Otero et al. 2020) as they allow for faster capturing of large, complex objects than TLS. These properties make MLS an ideal match for scenarios where frequent remodelling is required, however, this comes at the price of increased noise which demands processing algorithms to be robust.

Nevertheless, the manual construction of digital models is a laborious and time-consuming process (Tang et al. 2010), making frequent re-modeling for capturing area and layout changes introduced by renovations impractical. The result is a need for highly automated modeling and analysis workflows, as they cut down on time and resources.

In response to this problem, this work deals with the automated segmentation of multi-storey point clouds into storeys and rooms and the reconstruction of BIM models including all surfaces such as floors, ceilings and walls (Scan-to-BIM). Given the aim of the underlying research project, the presented methods generate CAFM-ready models for the most relevant CAFM applications, contain relations of building storys, rooms. Due to the focus on CAFM use cases related to space management, architectural elements such as windows and doors are not the focus of our approach.

Building on a previous, purely voxel-based work for room segmentation (Martens and Blankenbach 2021) titled VOX2BIM, the novelty of the presented approach lies in the significant extension of known strategies to improve speed and robustness and the complimentary combination of methods (e.g. storey segmentation, room segmentation and wall reconstruction) to derive spatial segmentations and geometric models. The presented workflow allows for dealing with unordered point clouds, non-Manhattan layouts and multi-storey setups and unlike other works combines voxel-based and point-based processing to enable dealing with data of varying qualities, ranging from sources such as TLS, MLS and image-based 3D reconstruction. Voxel-based techniques are used to accelerate processing and to improve robustness towards noise, occlusions and density variations, while refinement steps are done in continuous space using the underlying point clouds to deliver the required CAFM-ready accuracy. The use of object-oriented data models for the reconstructed building storeys, room spaces and volumetric walls means that relations between spatial elements are being preserved, allowing for elements to be modelled and exported in the BIM-ready IFC STEP format. Additionally, point cloud segments for each storey and room are generated to provide rich spatial information for use by external processing methods.

The overall article is structured as follows: At first, an overview of related works including common strategies is discussed. Afterwards, the novel combination of algorithms explored in this paper is explained in detail. Finally, the results are presented and discussed before the conclusion and outlook are provided.

2 Related Work

Automated segmentation of internal building structures such as rooms and walls has been subject to various publications, ranging from robotics and floor plan generation to geometry reconstruction in the AECO and CAFM fields (Tang et al. 2010). With segmentation being a crucial step in the reconstruction process, even machine learning is receiving attention for object segmentation and recognition in larger workflows (Perez-Perez et al. 2021a, b). In the following, CAFM implementation strategies and related point cloud segmentation and reconstruction methods are discussed in more detail.

2.1 BIM and CAFM

CAFM has proven to have a positive effect on the maintenance of building assets and has helped track close to 60% of the yearly utility costs in case studies (Ròka-Madaràsz et al. 2016). Ideally, CAFM data is stored using a standarized format such as IFC in a common data environment (CDE) to allow for transparent inspection by the facility managers (Patacas et al. 2020). However, the lack of BIM for most

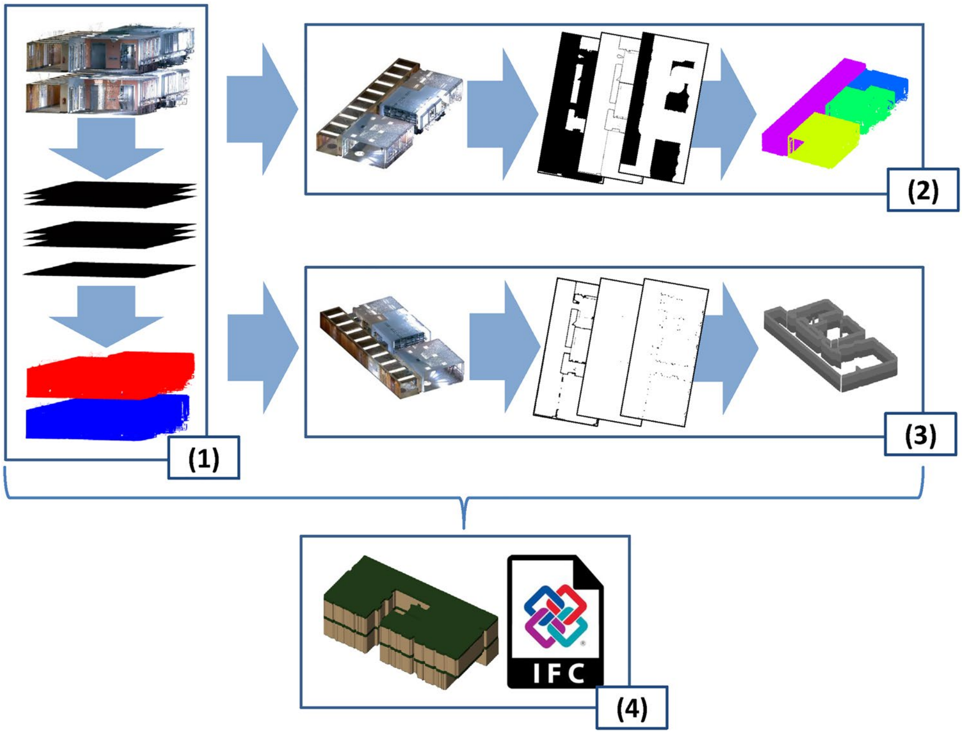


Fig. 1 Overview of the individual stages of the segmentation and reconstruction workflow. Storey Segmentation (**1**): the multi-storey point cloud is split into separate segments for each storey. Room Segmentation (**2**): For each storey segment, individual rooms are extracted. Parametric Wall Reconstruction (**3**): Wall bodies are reconstructed for the individual storey segments. IFC Model Reconstruction (**4**): The results are merged and written into a final IFC file

existing facilities, data capturing and model creation represent significant hurdles, therefore especially scenarios with a large number of facilities benefit from the creation of simplified initial models and supplementing them with additional geometric and semantic information as required by the facility managers has proven to be a pragmatic strategy (Carbonari et al. 2015; Gao and Pishdad-Bozorgi 2019). The use of IFC-inspired spatial hierarchies for buildings, storeys, rooms and assets appears to be a typical baseline for such models, as they allow for simplified management of spatial units and tracking of defective inventory objects within the facilities (Motamedi et al. 2014; Pishdad-Bozorgi et al. 2018).

Due to the required segmentation and reconstruction of spatial units and building elements being closely intertwined, most works solve both problems in tandem. In the following, these strategies are categorized into storey segmentation, room segmentation and parametric wall reconstruction methods.

2.2 Storey Segmentation

Despite being crucial in multi-storey setups, storey segmentation is rarely discussed as part of segmentation. Instead, most works merely deal with the detection of floor and ceiling planes either as a preprocessing step for removing the related points or as a way of estimating the ceiling height. A common strategy hereby lies in constructing point density histograms along an "up" axis. With point densities being significantly higher at heights running through the floor and ceiling planes, detecting them becomes quite trivial (Okorn et al. 2016; Jung et al. 2017; Wang et al. 2017). As an alternative, normals are occasionally used for the same purpose, as they give cues about the orientation of surfaces and thus allow for easy identification of floors and ceilings as vertical surfaces (Sanchez and Zakhor 2012; Shi et al. 2019). Combinations of both methods where points are filtered by their normal orientations and afterwards used for histogram construction are quite rare though (Oesau et al. 2013). Aside from reduced accuracy due to discretization, issues with the histogram-based method often originate from large vertical surfaces such as staggered ceilings or furniture objects leading to small peaks resembling floor and ceiling peaks. This is particularly problematic for storey segmentation, where peaks are usually used to identify storey boundaries and extract points located between them as storey segments (Macher et al. 2017; Oesau et al. 2013; Li et al. 2018). Oftentimes details of dealing with this issue are kept vague, however defining fixed sizes can help identify neighboring peaks which form a slab and therefore separate two storey segments (Li et al. 2018).

As seen in other works, segmentation for multi-storey setups is rarely being dealt with. Histogram-based approaches are usually used for this or similar tasks but have only experienced incremental improvements which prioritize robustness over accuracy.

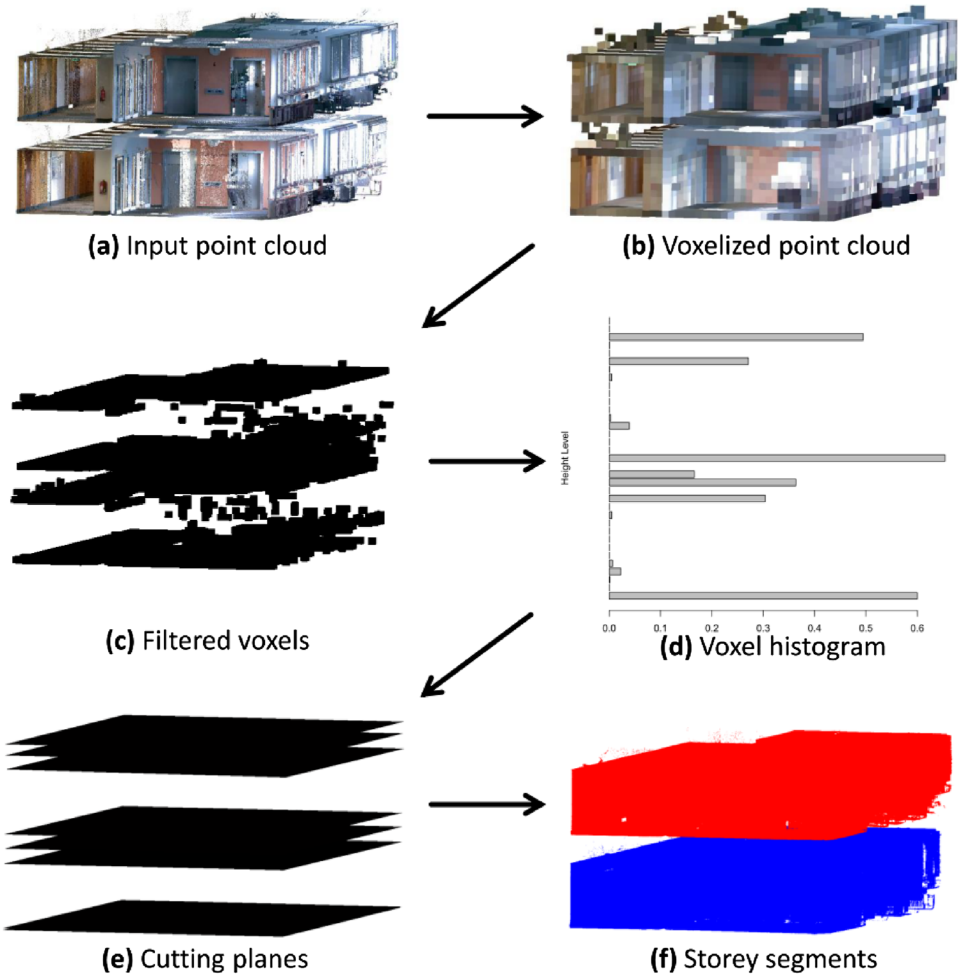


Fig. 2 Building storey segmentation process. **a** Input point cloud. **b** Voxelized point cloud using coarse resolution for illustration purposes. **c** Remaining voxels after filtering based on normal and occupancy. **d** Histogram with a relative number of filtered, occupied voxels along z-axis. Histogram peaks are used to construct cutting planes candidates. **e** Merged and refined cutting planes used for segmentation. **f** Extracted storey segments with unique colours per storey

2.3 Room Segmentation

While occasionally performed together with parametric wall reconstruction, room segmentation by itself augments point clouds with valuable spatial and semantic information. One early segmentation strategy is the use of prior knowledge about rooms having at least one scan positions. This means that planar surfaces can be assigned to them, and hence individual rooms, based on visibility (Ochmann et al. 2014). Such visibility-based approaches can be implemented with ray-casting and further generalized to label individual points in setups with more than one scan position per room (Ochmann et al. 2016; Mura et al. 2014; Wang et al. 2017). Alternatively, artificial scan positions can be generated as well (Ambruş et al. 2017). Variations using MLS trajectories rather than static TLS scan positions exist as well and assign points visible at specific trajectory positions to the room where the capturing device currently resides in. Transitions to other rooms can hereby be detected as sudden changes in the ceiling profile along the MLS trajectory (Díaz-Vilariño et al. 2017). Simplifications of this method involve a filtering step, such that only wall points are left. The remaining points transformed into a graph using a Delaunay Triangulation, which enables visibility checks and room partitioning with respect to the scan positions (Turner and Zakhor 2014).

Due to efficiency, 2D projections of the point cloud onto the XY-plane, are quite popular and can be used without scan positions. Within this group of techniques, the use of 2D region growing is quite rare, with one variation converting all points belonging to the ceiling plane to a binary image before applying region growing to it for room labelling (Macher et al. 2017). The method by which this work is built upon instead aims to estimate walls by measuring point densities in a vertical direction and then uses them as boundaries for the region growing process (Martens and Blankenbach 2021). Detecting room boundaries also represents the lynchpin of methods which rely on detecting lines in 2D point cloud projections as room boundaries (usually through Hough Transform (Hough 1962)) to subdivide the space into cell complexes. Based on graph optimization techniques, this allows for the estimation of room boundaries and the labelling of points inside them (Mura et al. 2014; Ikehata et al. 2015; Wang et al. 2017; Ambruş et al. 2017; Li et al. 2018).

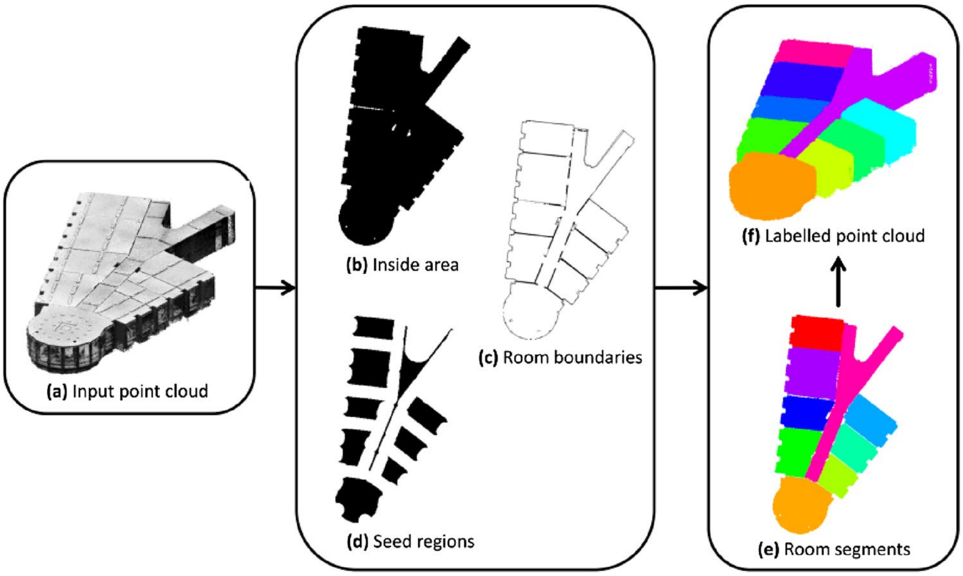


Fig. 3 Workflow stages for segmentation of individual rooms. **a** Input point cloud. **b** Inside area mask. **c** Room boundary mask. **d** Seed regions for region growing. **e** Final room segments. **f** and point cloud with labeled room segments

Keeping this in mind, existing approaches usually lack flexibility due to their reliance on scan position data or miss the opportunity of refining their results in 3D by focussing solely on 2D line detection.

2.4 Parametric Wall Reconstruction

Similar to room segmentation, other works have achieved parametric reconstruction exploiting various assumptions and types of prior data, but oftentimes focus on modelling structures as planar surfaces rather than volumetric bodies. For instance, methods requiring scan positions (e.g. Ochmann et al. 2014) oftentimes not only do room segmentations based on visibility but also reconstruct visible room boundaries as planes using common methods such RANSAC (Schnabel et al. 2007). Noteworthy examples rely on tracing rays from scan positions to individual points to differentiate between individual wall surfaces and add openings like doors and windows to them (Previtali et al. 2014, 2018). Similar strategies combining such visibility checks with RANSAC (Ochmann et al. 2014) and would later be extended by combining opposite surfaces of adjacent rooms into modelling volumetric, room-separating walls (Ochmann et al. 2016; Macher et al. 2017). Despite known scan positions providing a reliable of identifying wall openings such as doors and windows, capturing them is not always and option and dropping them as a requirement allows for improved flexiblity.

Exploiting surface orientations has proven particularly useful, as they allow for categorizations into walls and floors/ceilings for additional semantic information (Thomson and Boehm 2015). Point groups with similar normal directions extracted by means of region growing can be used to remove noise and accurately reconstruct planar models using RANSAC in manhattan-world scenarions (Sanchez and Zakhor 2012; Murali et al. 2017). Previously discussed techniques for room segmentation based on cell complex decompositions naturally lend themselves to parametric reconstructions, as their earliest step involves extracting 2D lines (Ikehata et al. 2015; Díaz-Vilariño et al. 2017; Li et al. 2018). While these 2D lines are extracted from 2D projections of the underlying points, extruding them in the vertical direction creates the final wall surfaces. The projection step can either involve discretization by creating a floor plan-like projection of point clouds where point densities in the vertical direction are mapped onto a 2D grid before estimating 2D lines with a Hough Transform (Okorn et al. 2016). Alternatively, 2D line fitting of projected points using RANSAC may also be used on individual pre-segmented vertical surface patches (Mura et al. 2014; Wang et al. 2017). Building footprints from point projections onto a 2D occupancy map (Hong et al. 2015) or pre-segmented room boundaries (Turner and Zakhor 2014; Shi et al. 2019) can be used for the same purpose.

All in all, the detection of planar or linear point segments represents an efficient solution in most cases, however commonly-used shape detection algorithm (Hough Transform and RANSAC) have issues with irregular or rounded walls and only reconstruct planar rather than volumetric objects. Furthermore, all discussed methods use histogram-based approaches to estimate ceiling heights which is trivial in single-storey scenarios, but becomes much more complex for multi-storey building where individual storey and slab heights are present.

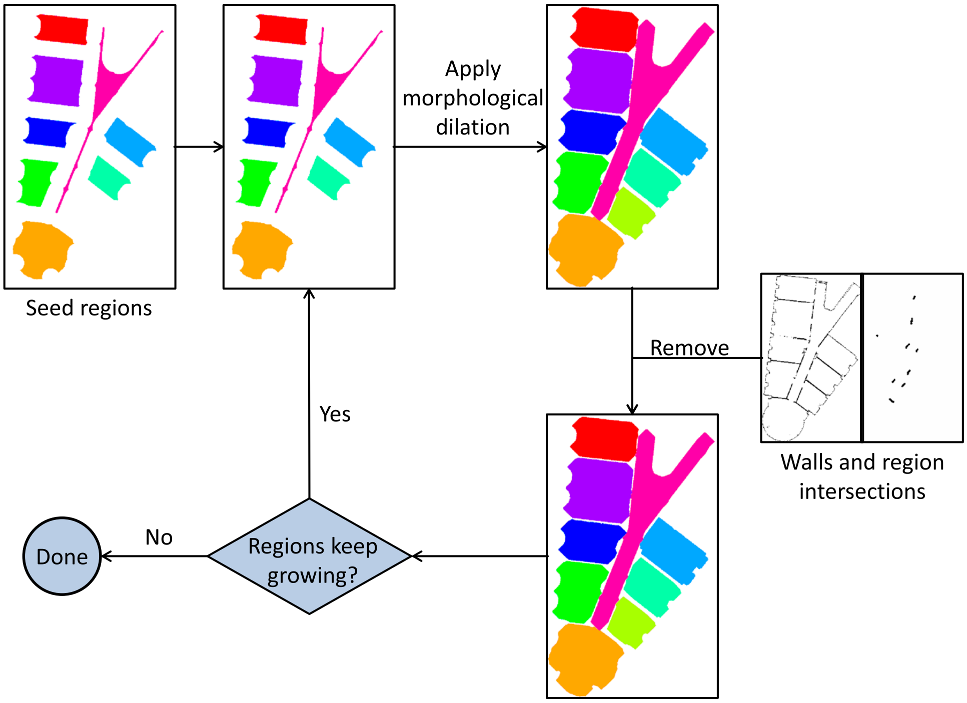


Fig. 4 Visual overview of the region growing process. A set of initial seed regions gets iteratively expanded until no growth occurs. Region overlaps and rooms boundaries are used as growth limiters. Once no more growth occurs, the process is finished

3 Methods

This section describes the implemented segmentation and parametric reconstruction process based on an input point cloud (as detailed in Fig. 1). Step **(1)** is the storey segmentation, which is crucial in multi-storey scenarios for providing the room segmentation and parametric reconstruction steps with point cloud segments that correspond to individual floors in the input point cloud. Both subsequent steps use the resulting storey segments as input and are performed independently from each other. The room segmentation step **(2)** generates non-overlapping point cloud segments for each individual room, while the parametric wall reconstruction **(3)** extracts information about the location and thickness of walls within the input point cloud. Once all steps are completed, the information generated by both methods is merged into one single IFC model **(4)**. Despite similarities with related techniques are combined in a unique way to provide semantic relations between the reconstructed objects. All operations benefit strongly from aggregating the original point cloud into voxel grids to provide higher speed than purely point-based approaches while still offering decent robustness. The lack of precision resulting from the discretization errors introduced by voxelization is compensated by refining initial results with data from the original point cloud. In contrast to previously discussed related works, the presented one only assumes at least partial visibility of floor, ceiling and wall surfaces.

3.1 Storey Segmentation

With buildings spanning through multiple storeys, the first step is concerned with extracting individual storeys as separate point cloud segments and extracting related information such as elevation, ceiling height and footprint geometry. This problem can be boiled down to finding floor and ceiling planes and extracting all points located in between them. An approach employed by other works is the construction of point histograms along different elevations of the point cloud (Oesau et al. 2013; Turner and Zakhor 2015), however, this method is notoriously sensitive to noise, scanning/registration artifacts, point density variations and clutter due to discretization and ignorance of the local context. Fine histogram resolutions or histogram borders located close to a plane provoke points to be distributed among neighboring histogram bins due to noise and lead to imprecise plane estimates. To solve these issues and achieve more accurate results, a histogram-like approach is only used during an initial phase to select candidate planes. Initially, all points are inserted into a *voxel grid*, as illustrated in Fig. 2 at stage **(b)**. Afterwards, the normal vector for each occupied voxel is estimated by means of a *principal component analysis* (PCA) as described by Rusu (2009). All voxels with normals oriented in roughly vertical direction are assumed to contain suitable points and kept for subsequent steps **(c)**. The selected voxels are organized into slices along the horizontal direction, with voxels being either marked as occupied or empty. A *majority voting* is applied

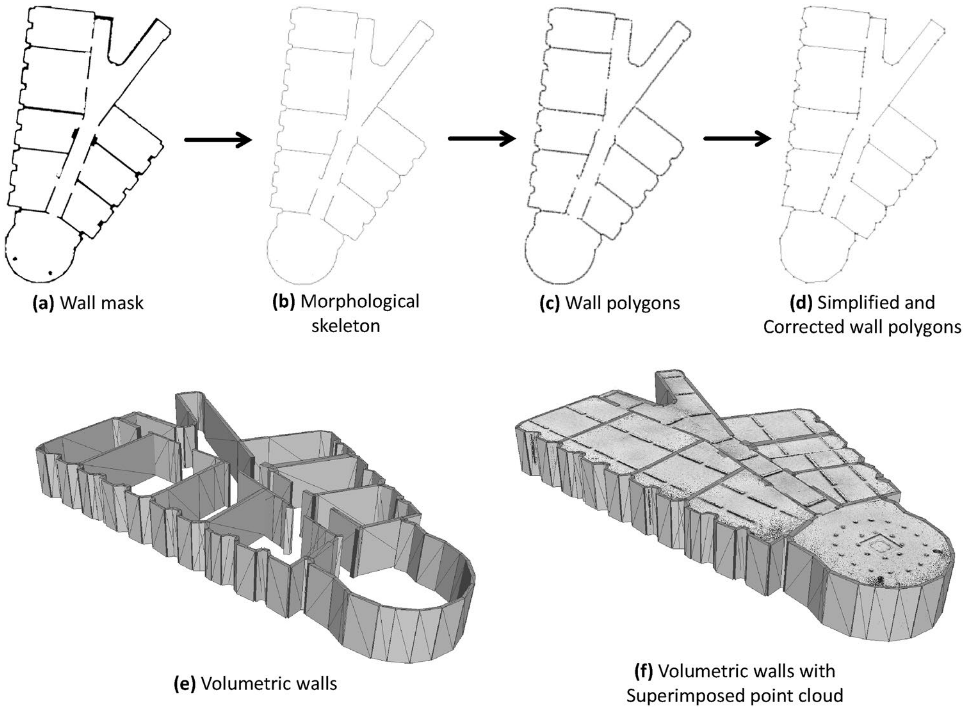

(a) Wall mask
(b) Morphological skeleton
(c) Wall polygons
(d) Simplified and Corrected wall polygons
(e) Volumetric walls
(f) Volumetric walls with Superimposed point cloud

Fig. 5 Parametric reconstruction workflow for building walls. Top row from left to right: Wall mask extracted from voxelized input **(a)**, pixel-thin morphological skeleton of wall structures **(b)**, wall polygons reconstructed from morphological skeleton **(c)**, simplified wall polygon with corrected coordinates **(d)**. Bottom row from left to right: Volumetric walls constructed from wall simplified wall polygons **(e)**, volumetric walls superimposed onto original point cloud **(f)**

to local neighborhoods in each slice to remove outliers created by artifacts and holes present in the occupied segments. Slices where the number of occupied voxels exceeds a predefined ratio are selected as candidate slices for floor and ceiling plane reconstruction **(d)**. However, additional steps are necessary for plane refinement, as noise and a slight tilt in point clouds can lead to multiple slice-candidates being clustered together. Therefore, constructing perfectly horizontal planes these slices will lead to inaccurate results where multiple planes are created at neighboring height levels. A final step, therefore, merges neighboring slices are merged and estimates the final planes from the points residing within the respective slice voxels **(e)**. Just like before, only points from voxels with an overall vertical normal orientation are used for the final PCA-based plane fitting step to exclude points located along walls and other vertical surfaces to provide reliable results. In contrast to conventional histogram-based approaches, benefits of extracting, merging and refining the fitting planes in continuous space are a greatly increased accuracy, robustness towards noise and a forgiving parameter selection where suitable grid sizes may go up to $0.2m$.

Finally, the resulting fitting planes are used to extract the points located in between floor and ceiling plane pairs as building storey segments **(f)**. Problematically, the distinction between floor planes, ceiling planes and faux ceilings is not obvious, but this issue can be solved by categorizing plane-pairs into storeys and slabs based on their height. With slab segments typically being lower in height than storeys, both can be separated using either clustering approaches or a separation into two classes based on the *between-class variance metric* akin to the automatic thresholding method by Otsu (1979). For simplicity, the presented implementation sorts all segments i by their height h_i and chooses a threshold that lies in the center of the two neighboring segment heights h_i and h_{i+1} which maximize the distance:

$$d_{max} = max(\mid h_{i+1} - h_i \mid) \tag{1}$$

Faux or staggered ceilings however can still lead to false-positive slab segments and oversegmentation and can only be identified with prior knowledge. This entails defining an expected ceiling height and merging storey segments with slab candidate segments above them if the ceiling height is considered too low. With the final storey segments determined, all points located between the plane pair of each segment are extracted and associated with an individual building storey. Due to planes always vaguely cutting through the center of all points running along floor or ceiling surfaces, a small margin may be added to the plane boundaries to gather points close to them. Each resulting point cloud segment corresponds to an individual storey and thus carries new spatial information with. Additional geometric informations such as segment height, elevation and footprint polygon are stored alongside the storey segments for use with later reconstruction steps. The small spaces located between individual storey segments are treated as slabs with points inside being treated as artifacts and excluded from further processing. Projecting the occupied voxels of each storey to a 2D grid allows for the reconstruction of a footprint polygon which, when extruded along the storey's height, forms a volumetric representation of it.

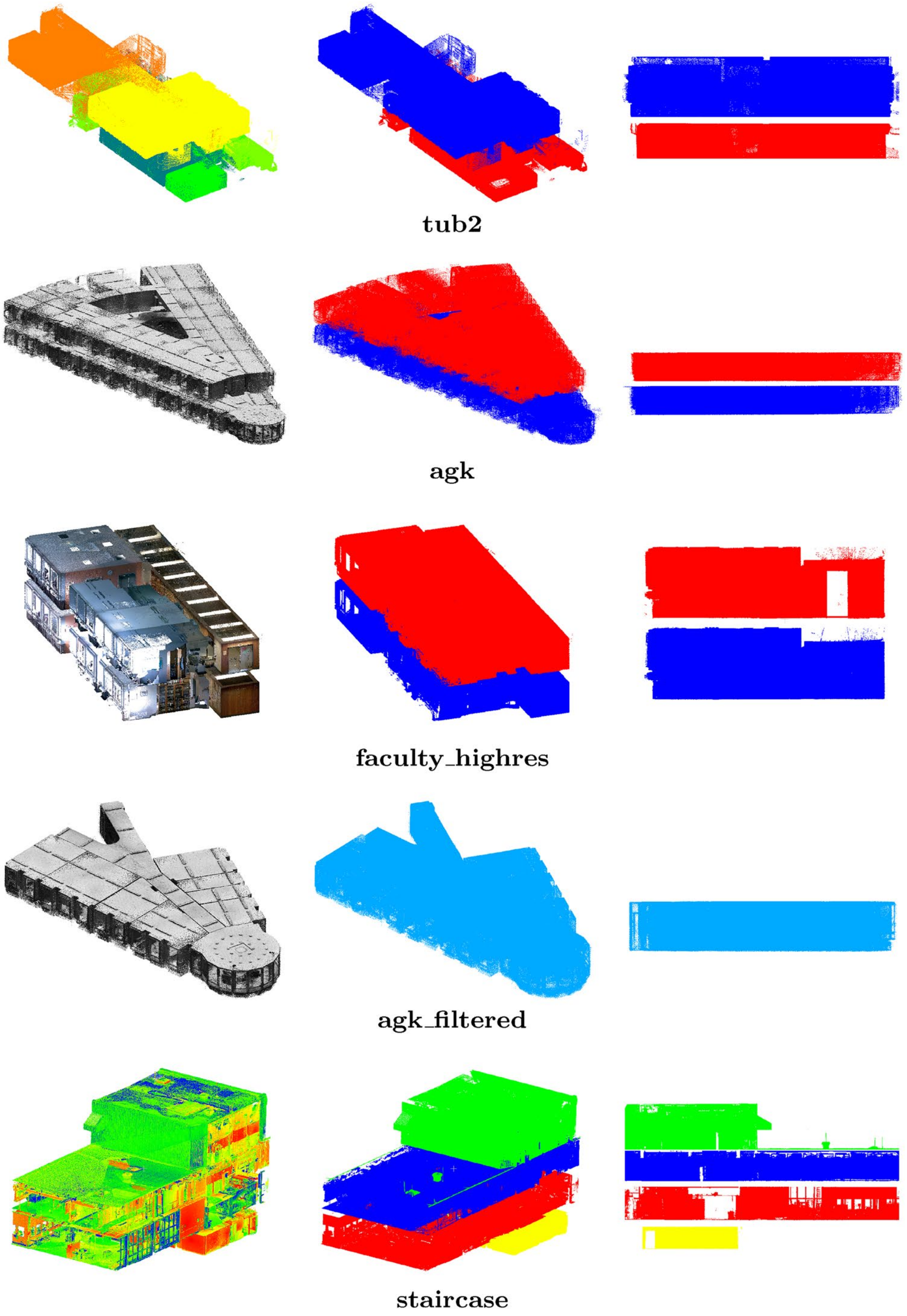
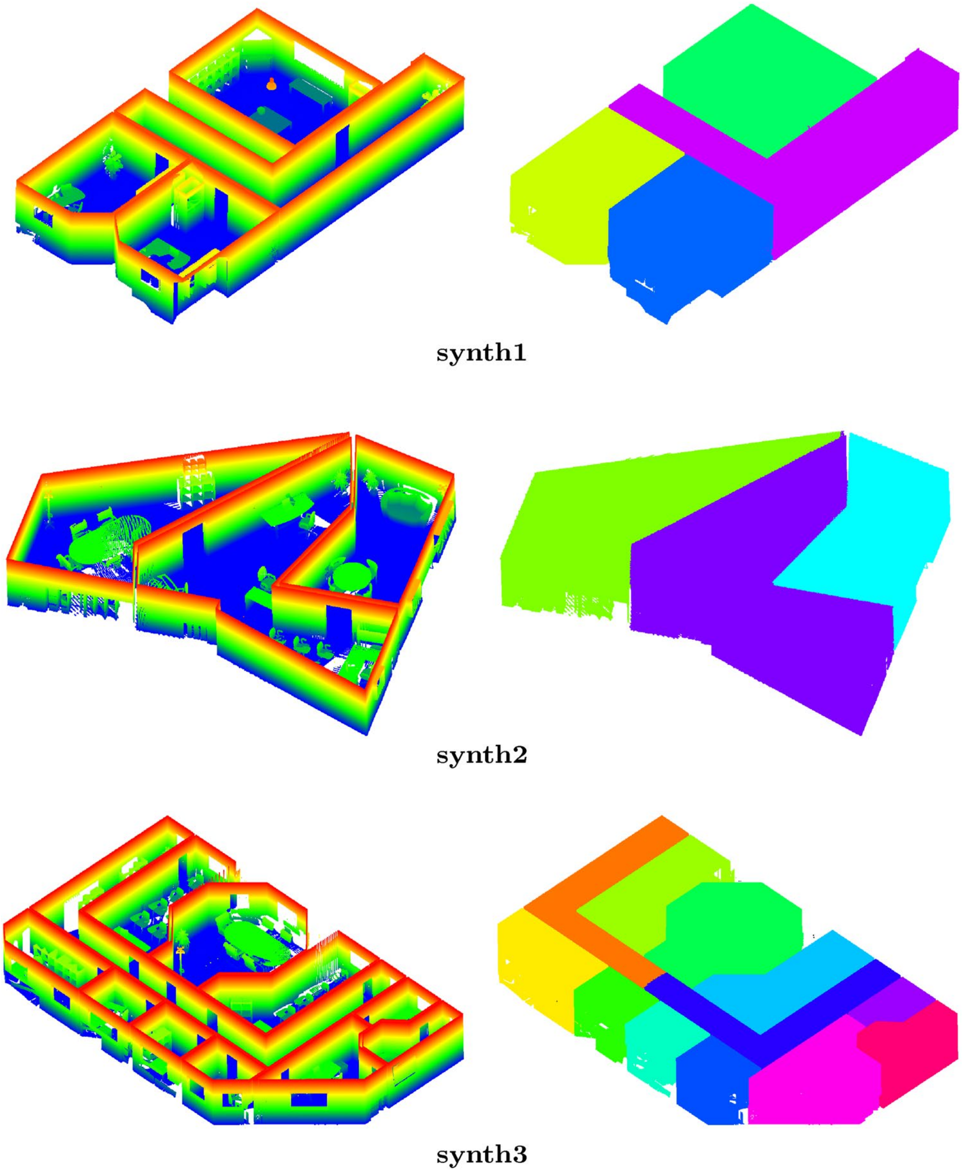


Fig. 6 Visual results of storey segmentation for each point cloud organized in rows. First column: Input point clouds. Second column: Output point clouds with individual storeys labeled. Third column: Labeled output point clouds in side view

3.2 Room Segmentation

With the storey segmentation being performed beforehand, the room segmentation algorithm assumes input point clouds to span across only one single floor. Estimating the boundaries surrounding each room represents a key step, as they form separate compartments in the point cloud. Once room boundaries are known, *morphological operators* (Matheron 1697; Young 1983) are used for extracting the final, non-overlapping room segments. Once created, the initial segments are mapped onto the original point cloud. Most operations exploit the 3D context given in the *voxel grid*, before breaking the data down into 2D floor plans. Despite the seemingly apparent loss of information when dealing with 2D floor plans, it should be noted that working with the full 3D data can promote oversegmentation due to furniture and other large planar clutter objects being indistinguishable from room boundaries. The overall process is built on the



synth1

synth2

synth3

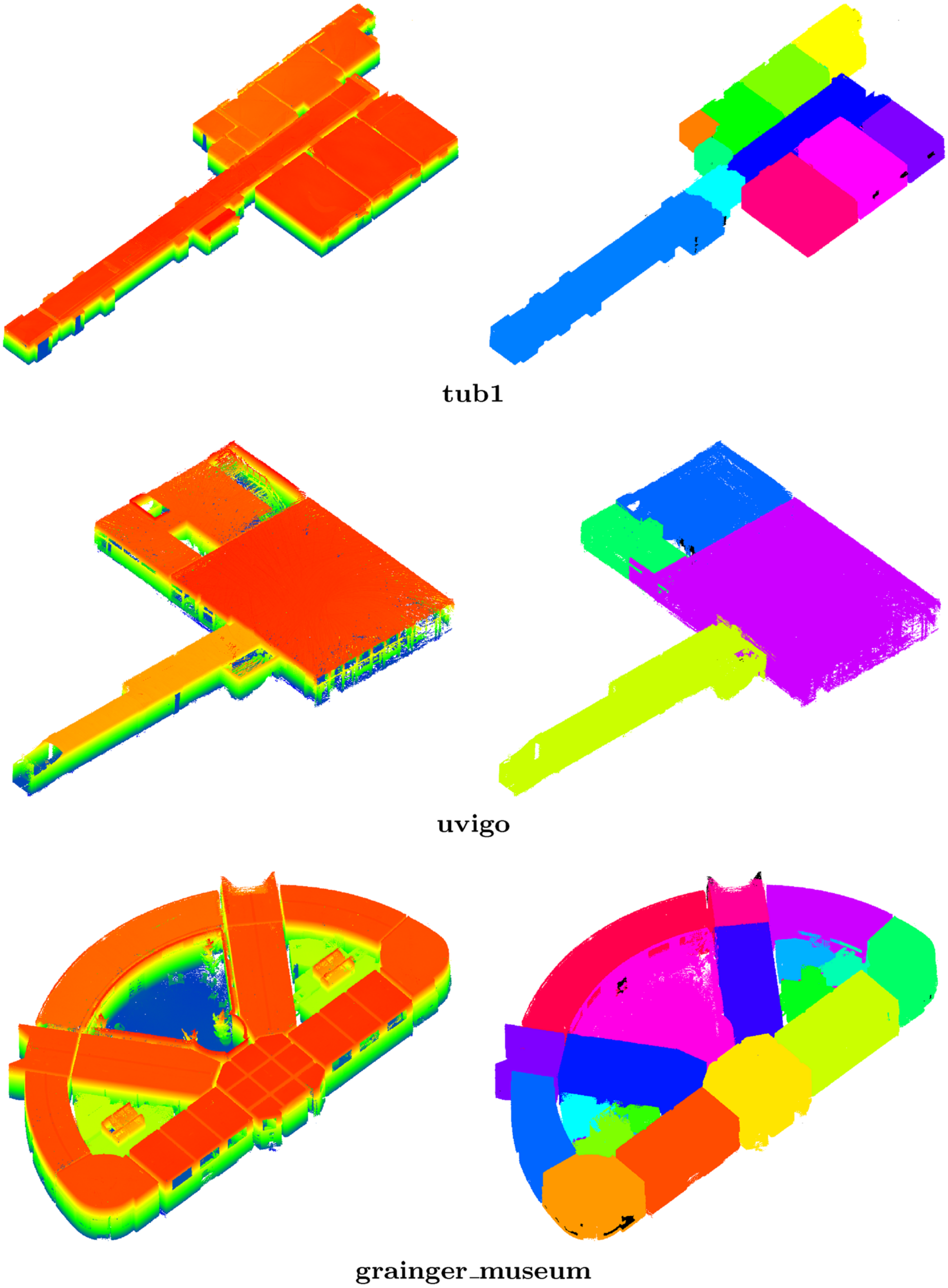
Fig. 7 Results for room segmentation for synthetic data from the UZH dataset (Mura 2016) with original point clouds in the left and labeled point clouds in the right column. Unlabeled points are marked black, ceilings have been removed for illustrative purposes. Despite the presence of non-Manhattan layouts, the separation between rooms remains clean. Only the long corridor in the bottom row represents a failure case as it has been split into two separate corridors

region-growing idea presented in Martens and Blankenbach (2021), but uses more sophisticated and robust methods for estimating room boundaries and footprints which are used as growth boundaries.

The entire process is portrayed in Fig. 3. Initially, the original point cloud is voxelized using a uniform voxel grid which forms the foundation for all subsequent operations. With the space corresponding to the inside area of the point cloud being unknown, it is assumed that each vertical stack in the voxel grid belongs to the inside area if at least one voxel in it is occupied. The information in the voxel grid is thus reduced to a *binary image* marking the inside area. The result will however suffer from scan artifacts and holes, thus requiring a *morphological closing* pass to clean it up as shown in subfig. 3 **(b)**.

Afterwards, the locations of room boundaries inside the voxel grid are determined. A first estimate is already given by the binary image denoting the inside area: the outer border of the inside area is treated as a set of exterior walls separating inside and outside space. As a side effect, borders of incomplete geometries or regions where parts of the point cloud have been occluded or cut out are treated as external walls as well.

For the voxelized point cloud the remaining boundaries are defined as densely occupied vertical stacks located between the floor and ceiling planes. Therefore floor and ceiling locations are again estimated, although this time variable ceiling heights are taken into account. This step is required to improve accuracy, as elevations are not guaranteed to be identical across one single floor in the presence

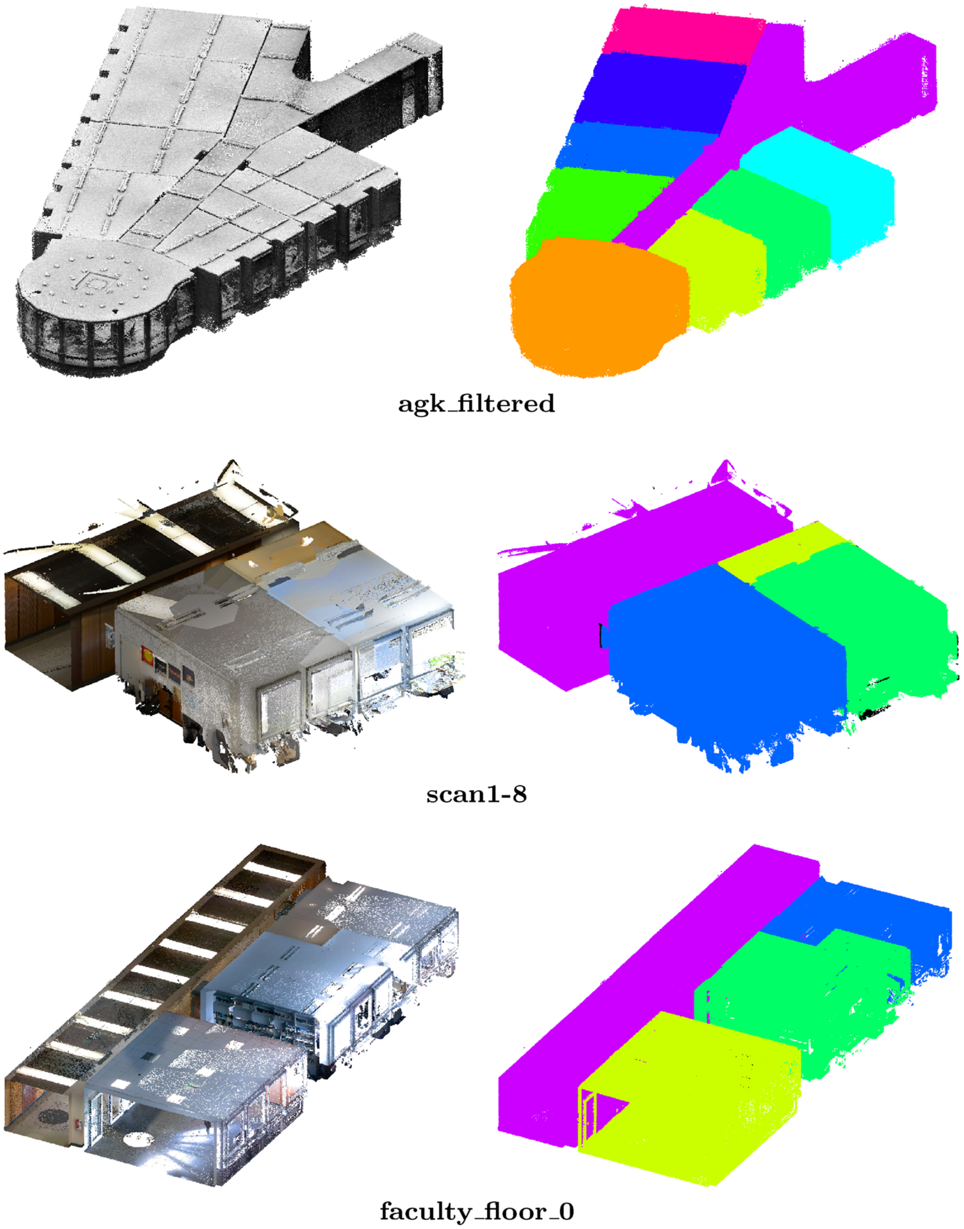

tub1

uvigo

grainger_museum

Fig. 8 Results for room segmentation for synthetic data from the UZH dataset (Khoshelham et al. 2017) with original point clouds in the left and labeled point clouds in the right column. Unlabeled points are marked black. As seen in the second row, small compartments of large rooms and halls can end up being split into multiple smaller rooms. The sample in the third row is properly segmented despite the non-Manhattan layout, but heavy clutter can promote oversegmentation, especially in narrow spaces

of false or staggered ceilings, ventilation shafts or staircase steps. The floor elevation is thus estimated by choosing the lowest occupied voxel within a vertical stack, while the ceiling elevation is determined by the highest occupied voxel within such a stack. When applied to the entire point cloud, this approach results in two masks denoting the respective floor and ceiling elevations for each pixel, with the former one being filtered using a morphological erosion and the latter one being filtered with a morphological dilation filter. This filtering step is crucial in filling holes and removing noise from both images, as some floor and ceiling sections may be missing from the point cloud. A large relative number of occupied voxels within the newly-defined floor and ceiling segment boundaries now indicates the presence of wall segments if the distance between floor and ceiling is sufficiently large. While the relative number of occupied voxels can be thresholded manually to extract the wall segments, using *Otsu's automated thresholding method* represents a convenient and viable option. This method aims to find a suitable threshold t by maximizing the *between-class variance*:


agk_filtered

scan1-8

faculty_floor_0

Fig. 9 Results for room segmentation with original point clouds in left and labeled point clouds in right column. Unlabeled points are marked black and occasionally occur at the outer edges of room segments when recessed walls are present. Such points can be labeled in a nearest-neighbor fashion during a post-processing step at the cost of additional runtime. Across different scenarios, panorama windows, irregular ceilings and non-Manhattan layouts pose no problem to the algorithm

$$sigma_b^2(t) = \omega(t) \cdot (\mu_0(t) - \mu)^2 + (1 - \omega(t)) \cdot (\mu_1(t) - \mu)^2 \tag{2}$$

The between-class variance depends on the mean values μ_0 and μ_1 of the thresholded sections, the overall mean μ, their relative probabilities $\omega(t)$ and $1 - \omega(t)$. As a result, reasonably pronounced structures are still picked up as wall among the mostly irrelevant background signals. Walls extracted using this method are oftentimes represented by very fine lines in the resulting binary mask. A *morphological opening* improves the results and deals with disconnected wall segments without introducing noise **(c)**. Despite the algorithm working well in most scenarios, it may be required to manually override the estimated floor elevation value in rare corner cases where large parts of walls are occluded by furniture objects such as shelves to get more stable results.

To extract the rooms, a *region growing* algorithm starting in seed regions located in the center of each room is used **(d)**. As a means of creating these seed regions, a *distance transform* is performed on a mask that combines both, the building's inside area and walls. The resulting mask defines the closest distance of each pixel to the walls and outside area. With this distance being maximized in the center of each room, candidate seed regions are extracted using automated thresholding. Different automated thresholding methods could be used for creating the seed regions. One example would be a threshold selection metric where the number of unconnected regions is being maximized alongside their size. Even though different threshold selection methods were investigated *Otsu's thresholding method* has proven again more suitable for this task, as it leads to relatively large but still disconnected seed regions. Regardless of the method used, resulting seed regions may still require filtering based on their area due to possible oversegmentation. Sufficiently large seed regions are then iteratively expanded using a custom region-growing implementation detailed in Fig. 4. During each iteration, the region-growing method uses morphological dilations to grow individual segments simultaneously, where regions are not allowed to cross walls or claim parts of other regions. Given the individual segment and wall masks, operations for limiting region growth can be modelled efficiently by combining masks with bit-operations. Once no more growth occurs, the resulting regions are extracted from the 2D mask and mapped back onto their corresponding points in the original point cloud. Furthermore, the 2D masks are used to create footprint polygons of each room which are extruded to form volumetric room elements. Leftover points which have not been labeled by the mapping from 2D to 3D can be assigned to the room located closest to them in an optional postprocessing step using a nearest-neighbor fashion. This postprocessing delivers visually attractive results and helps clean up missing regions. The high quality of this step results from the method operating on individual unlabeled points rather than voxels but does come at the cost of high execution times if many unlabeled points are present.

3.3 Parametric Wall Reconstruction

As an extension to the room segmentation step which relies on the estimation of room boundaries, the reconstruction of wall volumetric bodies deals with the extraction of support polygons for said walls. Once again the algorithm takes advantage of a 2D representation to benefit from accelerated execution speed.

In a series of initial steps depicted in Fig. 5, the method estimates the location of walls present in the point cloud by inserting it into a voxelgrid to speed up the process. Floor and ceiling elevations are then extracted as 2D masks from the voxelgrid, culminating in the estimation of wall boundaries from densely-occupied voxels located between the individual floor and ceiling segments. In contrast to the room segmentation, however, merely estimating potential room boundaries is insufficient, as walls have a volumetric geometry. Filling in the space between room boundaries using a morphological opening deals with this problem while simultaneously limiting the impact of noise **(a)**. To reduce the thickness of connected segments in the wall mask, the *Zhang-Suen algorithm* is used to create a morphological skeleton (Zhang and Suen 1984) as seen in Fig. 5 **(b)**. This skeleton is traversed by selecting the skeleton endpoints and junctions as starting points.

For the construction of the wall support polygon, the algorithm then jumps to neighboring pixels located within a predefined radius until a new skeleton junction or endpoint is met. The traversal is performed for each skeleton endpoint and junction to cover the entire skeleton structure **(c)**. All detected paths are simplified using the *Douglas-Peucker algorithm* (Douglas and Peucker 1973) to generate simplified polygon lines suitable for representing the wall structures **(d)**. The associated distance tolerance for line segments and their internal control points of 6.5 (in pixel space) has proven to be an ideal simplification parameter and was used for evaluation purposes as well. With possible deviations being introduced during the simplification process, polygon line endpoints are moved to the closest point in the center of the wall mask. Finally, the lines are extruded in orthogonal direction to model the individual wall thicknesses, where the thickness is determined as the distance of the wall endpoints to the closest boundary in the wall mask. These reconstructed volumetric walls are saved alongside their geometric attributes such as elevation and height for conversion to IFC.

4 Results

With the overall workflow being composed of different stages which can be used independently from one another, results and execution times for each stage are henceforth presented individually. All tests were performed on a PC running Windows 7 Enterprise with Service Pack 1, equipped with an Intel Core i7-4770 CPU running at 3.40GHz and 16GB of RAM. All programs for geometric point cloud processing were written in C++, with region growing and voxel slice filtering implemented using OpenCV (Itseez 2015) and vector operations making use of the Eigen library (Guennebaud et al. 2010). Execution times include loading the point clouds from the hard disk and writing out the results. Synthetic point clouds taken from the UZH dataset (Mura 2016) (labeled as synth1, synth2 and synth3) and real-world data from the ISPRS benchmark dataset (Khoshelham et al. 2017) (labeled as tub1, tub2, uvigo and grainger_museum) have been used alongside other real-world point clouds of varying quality. Alongside high-quality point clouds acquired with a Riegl VZ-400 TLS system, a large share of MLS point clouds captured with BLK2GO and ZEB Revo RT systems are used to analyse and underline the robustness of the methods (for details on the data, refer to Tables 1, 2 and 3). At the end of this section, IFC models created by combining all techniques are presented for multi-storey and single-storey laser scans.

4.1 Storey Segmentation

As discussed earlier, the storey segmentation represents the earliest step in the presented workflow and extracts the individual building storeys as point cloud segments located between floor and ceiling planes. In terms of speed, the segmentation process runs rather fast due to the reliance on voxels for plane candidate estimation, even when applied to larger point clouds. The overall point count and point cloud volume still affect the segmentation speed during the voxel grid buildup and point extraction phases though, as indicated by the execution times in Table 1. When it comes to parameter choice, choosing a suitable expected ceiling height has proven to show a bigger impact on accuracy of floor and ceiling plane estimation than choosing more restrictive values for the minimum area a slice should cover within the voxel grid. Increasing the voxel grid resolution may improve result accuracy in some cases at the cost of higher execution times.

Visual segmentation results are shown in Fig. 6, with original point clouds and individual storeys marked in different colours. Points located outside these cutting planes have been removed by this method, which results in the exclusion of scan artifacts and therefore aids subsequent room and wall segmentation steps as well. As illustrated, the method can therefore be used on single-storey point clouds to estimate storey heights and to remove scan artifacts and other unwanted structures not related to the building in a robust way. Overall, given a vague approximate for the expected ceiling height, potential false positive cutting plane candidates such as faux ceilings or table surfaces are correctly discarded, meaning that overall parameters can be reused for buildings of similar type. Problems will, however, arise if ceiling and floor planes of different storeys overlap within a slice, as the algorithm was not conceptualized with this case in mind. One such scenario is shown in the bottom row of Fig. 6, where the second storey is cut off prematurely, thus removing the ceiling plane of the right room.

4.2 Room Segmentation

While in the overall workflow context the room segmentation is used on point clouds resulting from the storey segmentation, it can be used on any point cloud as it requires no preconditions. The use of voxel grids once more proves beneficial as the discretization introduced by it keeps execution times low, but has a slightly more noticable effect regarding overall accuracy here.

As seen in Figs. 7, 8 and 9, the method fares very well even when room layouts follow a non-Manhattan geometry, but point clouds aligned to the coordinate system of the voxel grid (e.g. using suitable geometric methods (Martens and Blankenbach 2020)) generally produce more robust and accurate results with fewer instances of oversegmentation caused by faulty seed regions. Large panorama windows and irregular ceilings do not cause any issues. Nonetheless, elements such as recessed wall niches and large vertical furniture elements like shelves will lead to notable issues though and are commonly mistaken for walls during the early phase of the process, thus rendering room boundary estimations more complicated. With the region growing process using these structures as growth boundaries, this either leads to points not being marked or oversegmentation. Similar effects can be seen when suspended ceilings with multiple visible layers are present, as they make the estimation of floor and ceiling elevations more difficult. These effects can however be circumvented by adapting algorithm parameters. Filtering and removing small seed regions has shown effective when dealing with oversegmentation, while choosing a floor height offset during wall estimation helps reject vertical structures which would otherwise be incorrectly recognized as walls. Cases where room areas or walls are completely occluded from view can not be resolved through parameter tweaking though, the same holds true for extremely small rooms which can be mistaken for artifacts. While the structure of most long corridors like the ones at the top of Figs. 7 and 9 is quite inconsequential to the segmentation quality, a failure case is shown at the bottom of Fig. 7, where the corridor is split into two smaller ones. This effect is the result of oversegmentation during the generation of the seed regions. The same effect also leads to the oversegmentation visible in the second row of Fig. 8, where a large hall is subdivided into smaller rooms due to the presence of large, room-partitioning building elements. High amounts of large vertical clutter in narrow rooms will lead to oversegmentation as well.

Depending on the success of the room labeling, the chance of missing points always exists, for example outliers might be located too far away from any relevant region to be labelled meaningfully. The optional postprocessing step which marks the remaining points within a pre-specified radius in a nearest neighbor-fashion can be applied in such cases to improve results. This method will consider all points within a user-specified range around labeled regions and improves result quality significantly by labelling individual missing points. However, this postprocessing step comes at the cost of additional processing time and may be exceedingly time-consuming if many unlabeled points are present. Such cases manifest themselves as noticable outliers in the result Table 2, where execution times exceed $60s$. While these execution times seem quite excessive, the visual quality of the results is rather solid, as unlabeled points (labeled in black in Fig. 9) are relatively rare. In scenarios where execution times represent a major concern, optional post-processing may be disabled though. In direct comparison to other related works (Ikehata et al. 2015; Ochmann et al. 2016; Shi et al. 2019), execution times have always proven superior when no post-processing is involved. Even with post-processing enabled, execution times are only worse in rare cases where a large number of points is involved in this optional step.

4.3 Parametric Wall Reconstruction

Like the room segmentation, the parametric wall reconstruction is meant to be run after the storey segmentation step, but can also be run in a standalone manner on single-storey point clouds. Execution times are swift as seen in Table 3, given mean deviations are calculated as the orthogonal distance between each point p_i and the closest reconstructed wall surface π_j, however, the capturing method, furniture and clutter elements play a role in impacting the results. Therefore, the median was provided as a robust mean that better describes the average deviation and also corresponds to the accuracy metric introduced by Tran et al. (2019):

$$M_{Acc}(r) = Median\,(\pi_j^T p_i) \text{ if } |\pi_j^T p_i| \leq r \tag{3}$$

In addition, the maximum range for points to consider for distance estimations has been capped to $r = 0.3m$. Due to the lack of reference models for all point clouds and for the sake of consistency, geometric deviations were calculated between the reconstructed models and the input point clouds. To guarantee faithfulness of the results, all points which do not belong to walls (such as furniture and clutter objects) have been removed from the point clouds before distance estimation. For more clarity, an overview of the distribution of geometric deviations is depicted in Fig. 13 for the presented point clouds. As seen in Figs. 10, 11 and 12, furniture objects, floors and ceilings contribute most to the deviations and have therefore been removed before estimating the distances seen in Table 3 and Figs. 13. Among the remaining structures, recessed wall niches, windows and closets are common contributors to high distance deviations. Overall, the algorithm proves to be the most accurate for long wall segments. Short wall segments which oftentimes occur in cramped and complex environments on the other hand suffer from decreased accuracies. In rare cases, the method can fail to estimate wall thicknesses incorrectly which will lead to larger deviations. Additionally, wall endpoints may sometimes be slightly shifted during the simplification process, leading to subtle deviations from the reference point cloud. With these factors in mind, it should be noted that skipping the simplification process would lead to lower deviations with reconstructed models being considerably more complex and visually less appealing in return. However, for almost all evaluated point clouds the median deviations to the automatically generated model are not greater than $5cm$ on average, in 4 out of 5 cases the deviations are even smaller than $3cm$ which fully meets the requirements of CAFM applications. In terms of speed, the presented method outperforms other established works which have their execution times documented (Ikehata et al. 2015; Ochmann et al. 2016; Shi et al. 2019).

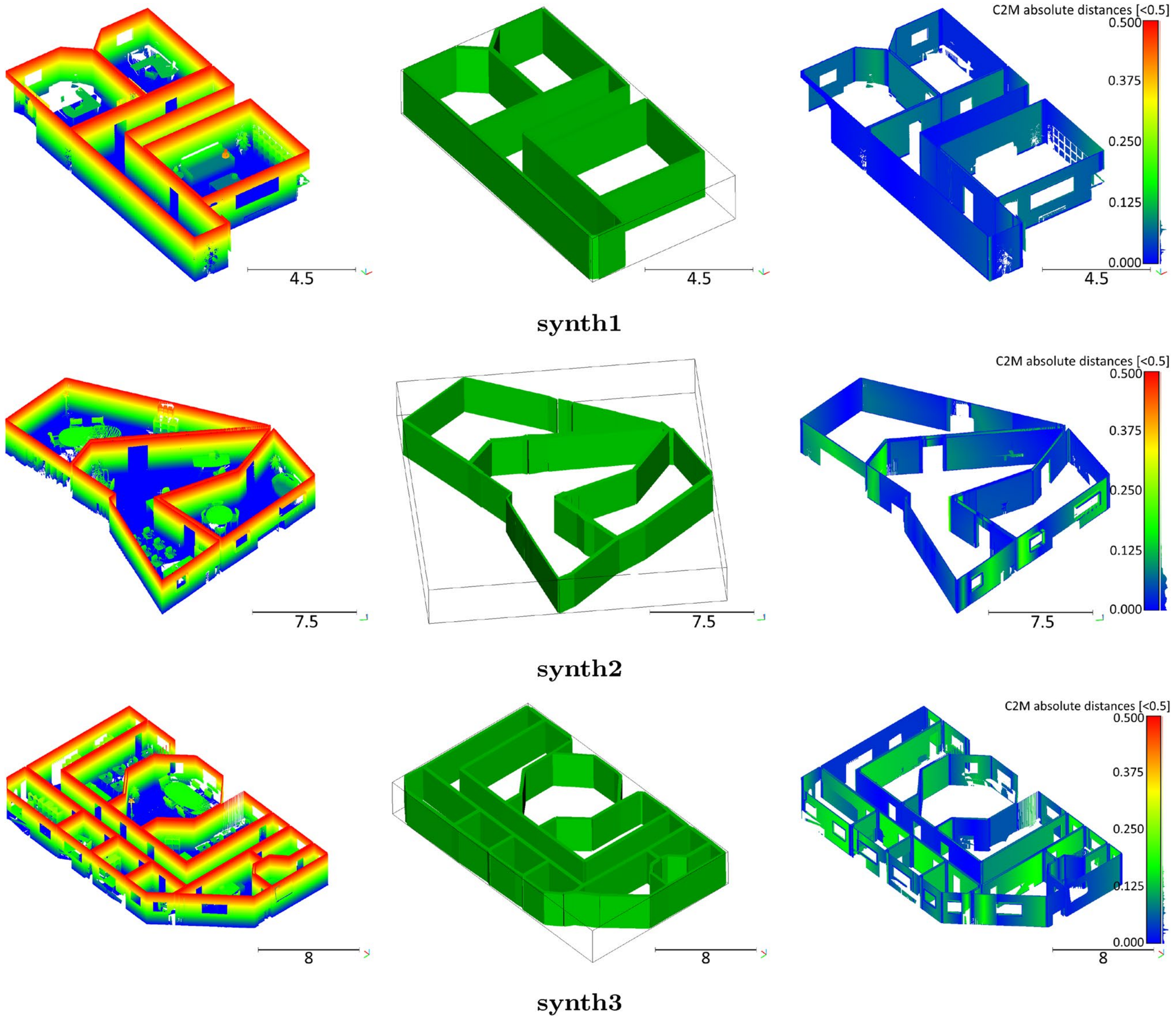
4.4 IFC Model Reconstruction

Using the results from the previous segmentation steps, building a complete IFC model representing building storeys, slabs, rooms and walls concludes the presented workflow. As far as model creation is concerned in general, only storey and room footprints are required alongside parameters for volumetric wall reconstruction. All of them are generated in previous steps and stored using JSON as a transition format, which means that combining them and reconstructing their spatial relations results in the final IFC model. Examples for the created models are shown in Fig. 14 for multiple single- and multi-storey laser scans.

The geometric quality of these models directly depends on the results of previous workflow steps meaning that poor models can be salvaged by adapting the corresponding parameters. As shown in the results, doors and other openings such as windows are not modelled at this point due to the workflow algorithms not yet being designed to detect them. Should the workflow be extended by additional methods, adding new model information would be simple though, as it merely requires an extension of the final stage where all extracted information is combined.

5 Discussion and Conclusion

This work presented a multi-stage workflow for the segmentation of multi-storey building point clouds and subsequent parametric reconstruction with focus on the most basic CAFM use cases related to spaces and geometries. The segmentation is split into storey-wise and room-wise segmentation steps, both of which can be executed independently from each other. The resulting point cloud segments and their rich room and storey information can potentially


synth1

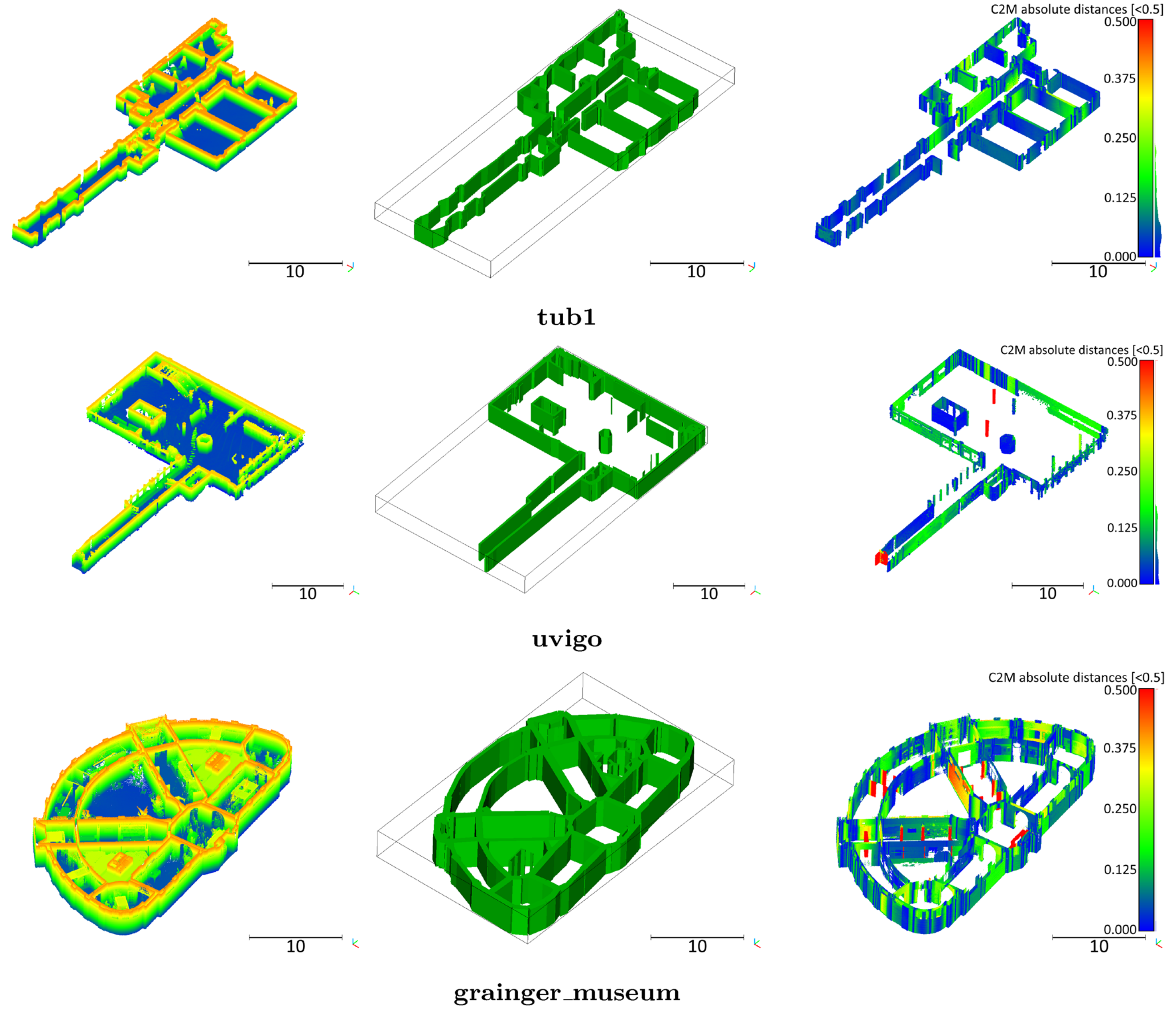
synth2

synth3

Fig. 10 Results of parametric wall reconstruction for synthetic data from the UZH dataset (Mura 2016). All ceilings have been removed for visibility. Left column: Input point cloud. Center column: Reconstructed walls with drawn-in bounding boxes. Right column: Point cloud-to-wall distances mapped to the wall structures of the input. The shown colour gradient for deviations is set to the interval of $[0.0m,\ 0.5m]$. Reconstructed walls follow the underlying geometry closely, with large deviations only being visible for floor planes and clutter objects

be used by external processing tools. Volumetric representations of the walls for each entire storey are reconstructed in a separate processing step. With the relation of storeys and rooms being extracted alongside wall storey heights and room footprints, the final step of the presented workflow is capable of combining this information into a single IFC model suitable for BIM-based CAFM.

As proven by the results, the storey and room-wise segmentation steps are highly robust and can be applied to high-quality TLS and comparably more noisy MLS data alike. All methods run very fast, with the only notable bottleneck being the optional post-processing step in the room segmentation which is used to clean up point cloud segments. Comparisons with related methods dealing with room segmentation and parametric wall reconstruction reveals that both steps are in fact performed faster by the presented workflow. Other works (Ikehata et al. 2015; Ochmann et al. 2016) present their results for point clouds with a varying number



tub1

uvigo

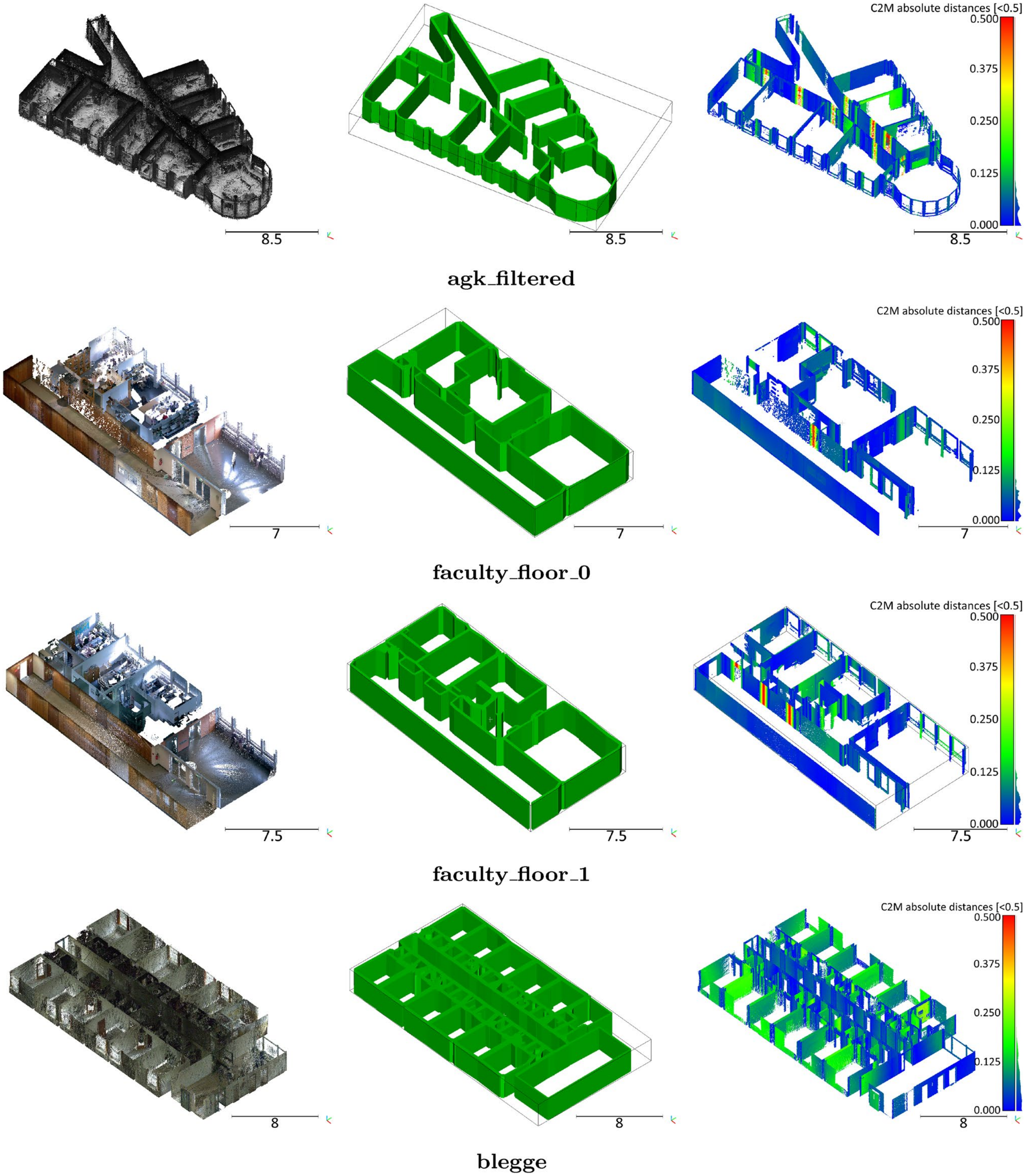
grainger_museum

Fig. 11 Results of parametric wall reconstruction for synthetic data from the ISPRS dataset (Khoshelham et al. 2017). All ceilings have been removed for visibility. Left column: Input point cloud. Center column: Reconstructed walls with drawn-in bounding boxes. Right column: Point cloud-to-wall distances mapped to the wall structures of the input. The shown colour gradient for deviations is set to the interval of [0.0m, 0.5m]. Small wall structures not present in the reconstructed model strongly contribute to the overall distance deviations

of points. However, for point clouds with a similar number of points, the combined execution times of both steps are consistently higher compared to the presented method. Shi et al. (2019) provide results for the UZH dataset (Mura 2016) and thus allow for direct comparison where the combined execution times are in favour of this work's workflow.

This makes the workflow particularly attractive for large collections of MLS-scanned facilities and allows for them to be processed and documented quickly. In addition, the accuracy of the automatically generated models is below 3.5cm in 8 out of 10 tested scenarios which seems absolutely sufficient for CAFM applications. It is only slightly worse in two scenarios, with accuracies of 5cm and 7cm respectively. Overall robustness is also quite high, even delivering solid results for point clouds captured with MLS.

Due to the way, storeys are being handled, stairwells are split and assigned to different storeys. Other potential shortcomings can be observed in the room extraction and parametric wall reconstruction steps. Both can run into problems if large vertical furniture pieces such as shelves are present,



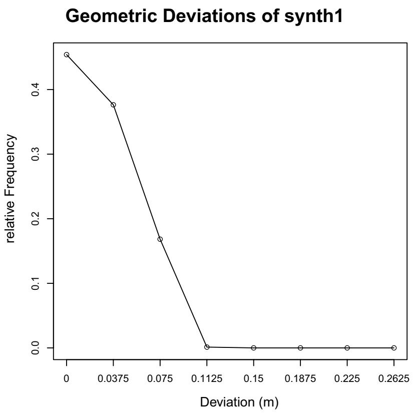
agk_filtered

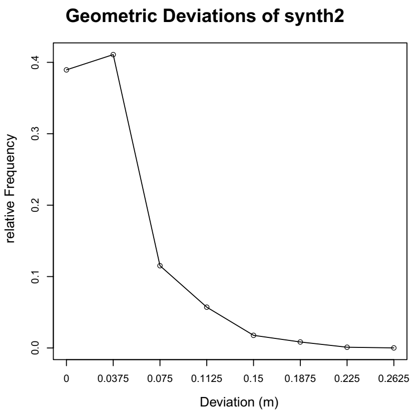
faculty_floor_0

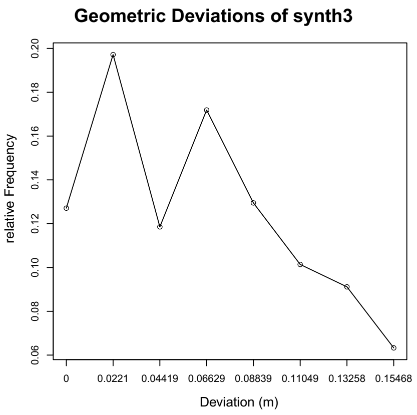
faculty_floor_1

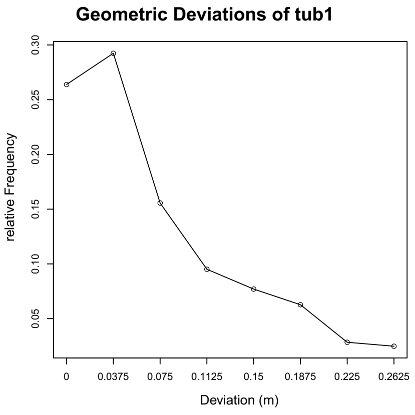
blegge

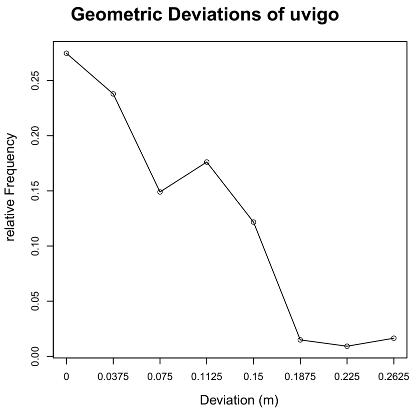
Fig. 12 Results of parametric wall reconstruction for real-world data. All ceilings have been removed for visibility. Left column: Input point cloud. Center column: Reconstructed walls with drawn-in bounding boxes. Right column: Point cloud-to-wall distances mapped to the wall structures of the input. The shown colour gradient for deviations is set to the interval of [0.0m, 0.5m]. Walls generally coincide with the provided point cloud, complex wall structures such as closets and recessed wall niches can introduce deviations though. Notable deviations occur at windows, shelves and furniture close to the walls

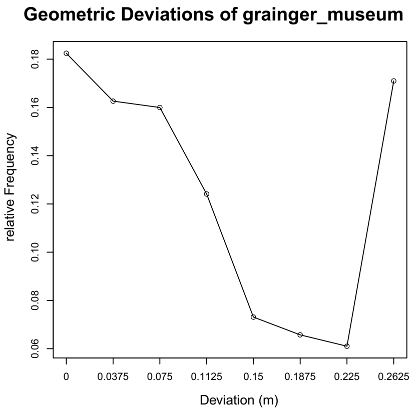

Geometric Deviations of synth1

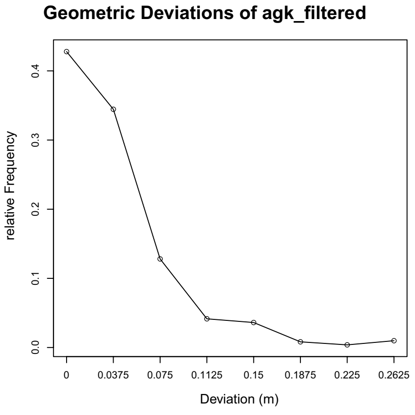

Geometric Deviations of synth2

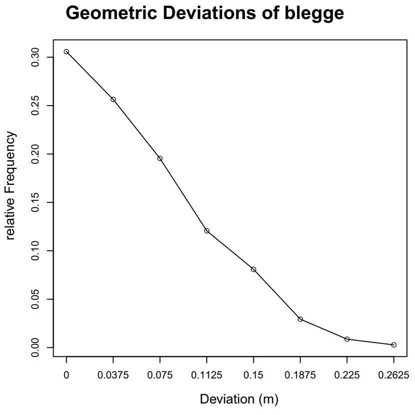

Geometric Deviations of synth3

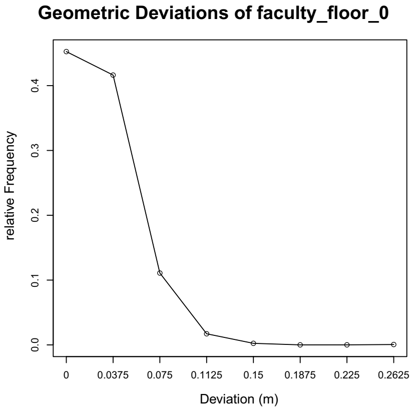

Geometric Deviations of tub1

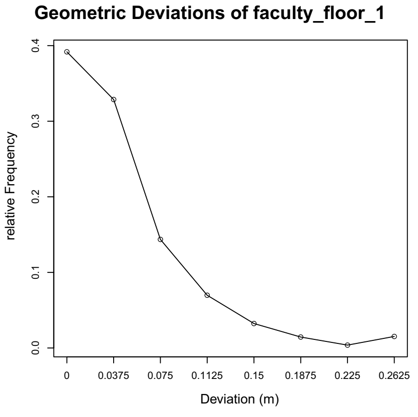

Geometric Deviations of uvigo


Geometric Deviations of grainger_museum


Geometric Deviations of agk_filtered


Geometric Deviations of blegge

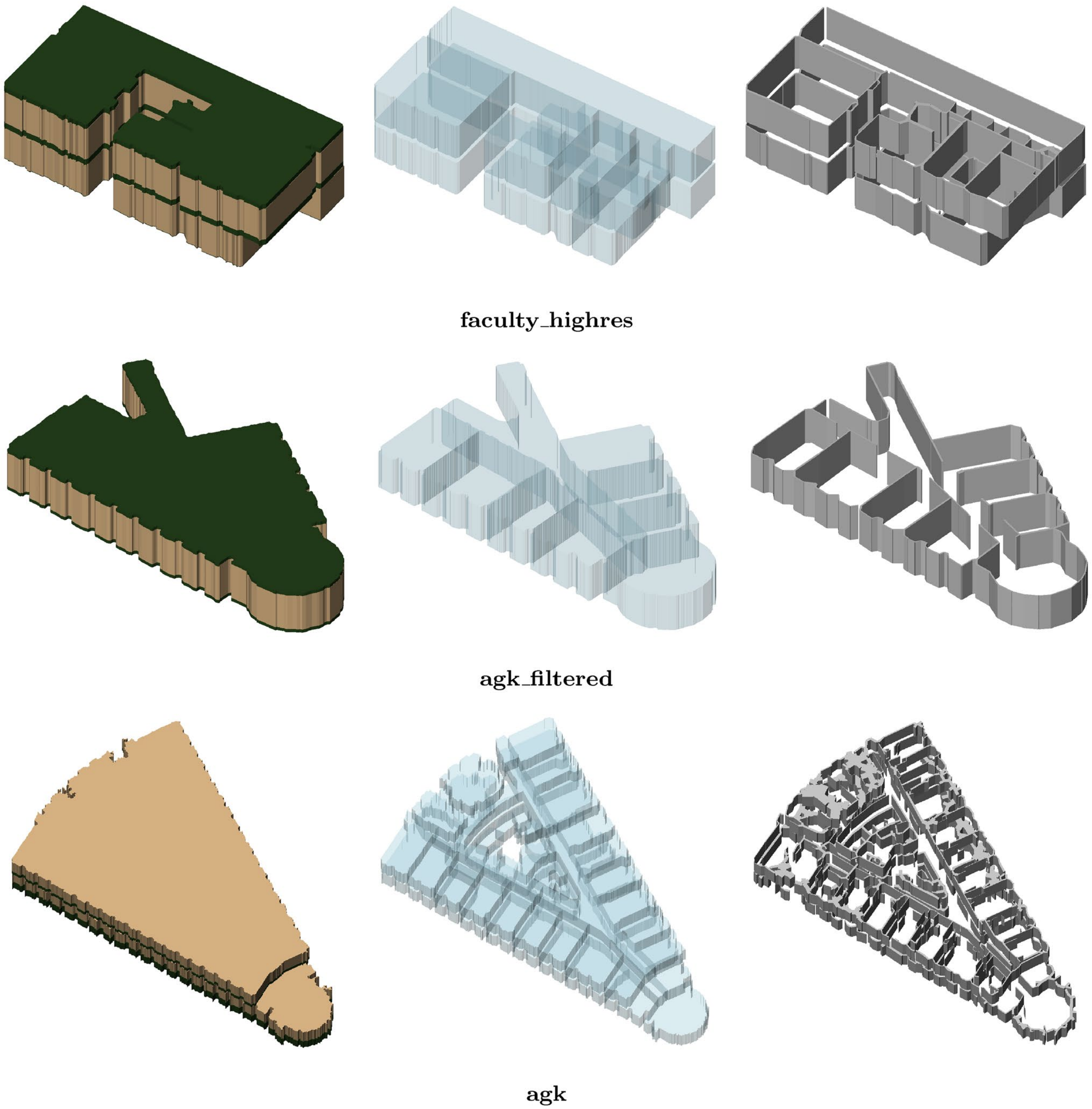

Geometric Deviations of faculty_floor_0


Geometric Deviations of faculty_floor_1

◀**Fig. 13** Distribution of geometric deviations for the point clouds shown in Table 3 within the range of 0.0m and 0.3m. Generally, the vast bulk of deviations is lower than 0.1 m

as these can be misinterpreted as walls. Algorithm parameters are quite flexible though and allow such corner cases to be taken into account.

Given the workflow's capabilities, embedding it into an automated CAFM infrastructure where changes are quickly captured using mobile reality capturing systems (e.g. MLS) and periodically integrated into the model akin to a digital twin is an attractive option (Lu et al. 2020). The relations between storeys, rooms and walls are already preserved and with pre-segmented point clouds suitable for asset detection being the result. Additionally, the reconstructed walls and floor plans could be useful for indoor navigation and minor extensions of the methods would also allow for the reconstruction of indoor network graphs akin to IndoorGML


faculty_highres

agk_filtered

agk

Fig. 14 IFC models reconstructed from single-storey and multi-storey point clouds. Building storeys and slabs are shown in the first column, spaces representing rooms in the second column and walls in the third column

Table 1 Averaged execution times of *storey segmentation algorithm* for various point clouds captured with terrestrial laser scanning (TLS) (using a Riegl VZ-400 laser scanner) and mobile laser scanning (MLS) (using a ZEB Revo RT laser scanner) after five runs, including reading and writing input and output data

Point cloud	Capturing method	Number of points	Number of storeys	Execution time (s)
tub2	MLS	21648386	2	73.532
agk	MLS	8000000	2	21.569
Faculty_highres	TLS	28680967	2	38.498
agk_filtered	MLS	5793587	1	8.927
Staircase	TLS	26555142	4	52.025

Due to the voxelization-based nature of the method, point cloud size and volume both factor into the execution time

Table 2 Averaged execution times of *room segmentation algorithm* for various point clouds captured with terrestrial laser scanning (TLS) (using a Riegl VZ-400 laser scanner) and mobile laser scanning (MLS) (using a ZEB Revo RT and backpack-based laser scanners presented by Filgueira et al. (2016)) after five runs, including reading and writing input and output data

Point cloud	Capturing method	Number of points	Unlabeled points before postproc.	Execution time without postproc. (s)	Execution time with postproc. (s)
synth1	Synthetic	19345772	6237	24.706	28.788
synth2	Synthetic	19258559	92409	30.830	71.291
synth3	Synthetic	68513293	263539	89.289	731.450
tub1	MLS	32597694	1957022	52.945	1278.433
uvigo	MLS	8805311	1212365	31.714	57.276
Grainger_museum	MLS	23454115	4113002	44.615	832.715
agk_filtered	MLS	5793587	54681	12.926	39.221
Scan1-8	TLS	17527772	468228	22.588	476.643
Offices	TLS	1425787	12984	6.705	16.226
Faculty_floor_0	TLS	2374272	15023	6.030	9.321
Faculty_floor_1	TLS	3617918	47092	8.175	70.310

Due to postprocessing not being required for IFC reconstruction and room footprint creation, results with and without this optional step are shown. The first three rows show results for synthetic point clouds. The final two rows relate to point clouds extracted from a multi-storey point cloud

Table 3 Averaged execution times and geometric deviations of *parametric wall reconstruction algorithm* for synthetic point clouds and various point clouds captured with terrestrial laser scanning (TLS) (using a Riegl VZ-400 laser scanner) and mobile laser scanning (MLS) (using ZEB Revo RT, Leica BLK2GO and backpack-based laser scanners by Filgueira et al. (2016)) after five runs, including reading and writing input and output data

Point cloud	Capturing method	Number of points	Mean deviation (m)	Median deviation (m)	Execution time (s)
synth1	Synthetic	19345772	0.005	0.005	13.133
synth2	Synthetic	19258559	0.054	0.010	16.290
synth3	Synthetic	68513293	0.077	0.052	47.525
tub1	MLS	32597694	0.088	0.030	26.579
uvigo	MLS	8805311	0.086	0.036	16.814
Grainger_museum	MLS	23454115	0.133	0.074	28.259
agk_filtered	MLS	5793587	0.056	0.008	6.513
Faculty_floor_0	TLS	2374272	0.045	0.004	3.048
Faculty_floor_1	TLS	3617918	0.062	0.012	4.022
blegge	MLS	32619768	0.077	0.028	27.329

The mean and median deviations of the reconstructed walls and point clouds have been estimated with floor and ceiling planes removed and with the maximum point-to-wall distance set to $0.3m$. The distribution of geometric deviations is depicted in Fig. 13

(Chair 2013), by analysing adjacencies of the extracted room segment boundaries. Furthermore, the modular implementation allows for other extensions and drop-in replacements of any method to achieve improved accuracy or to obtain additional semantic information. This would include sophisticated schemes for modelling surface openings such as windows and differentiating them from occluded surface sections. The integration of specialized methods for the detection and modelling of other architectural elements such as windows, doors and staircases represent another option for expanding on the established concepts. Driving home this idea of a full-fledges automated analysis suite, asset detection performed on extracted room segments using machine learning (Han et al. 2020; Qi et al. 2019) or the possible classification of rooms into functional types would represent the next logical step.

Acknowledgements This work has been supported and funded by the company “Aachener Grundvermögen Kapitalgesellschaft”.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by JM. The first draft of the manuscript was written by Jan Martens with and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

Availability of data and material Not applicable.

Code Availability Not applicable.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

Ambruş R, Claici S, Wendt A (2017) Automatic room segmentation from unstructured 3-d data of indoor environments. IEEE Robot Autom Lett 2(2):749–756. https://doi.org/10.1109/LRA.2017.2651939

Borrmann A, König M, Koch C, Beetz J (eds) (2018) Building Surveying for As-Built Modeling, Springer International Publishing, Cham, pp 393–411. https://doi.org/10.1007/978-3-319-92862-3_24

Bosché F, O'Keeffe S (2015) The need for convergence of bim and 3d imaging in the open world. In: CitA BIM Gathering Proceedings 2015, The Construction IT Alliance

Braun HP, Reents M, Zahn P, Wenzel P (2013) Facility Management: Erfolg in der Immobilienbewirtschaftung. Springer Vieweg Berlin Heidelberg. https://doi.org/10.1007/978-3-642-39083-8

Carbonari G, Stravoravdis S, Gausden C (2015) Building information model implementation for existing buildings for facilities management: A framework and two case studies. WIT Trans Built Environ 149:395–406

Chair OIS (2013) Indoorgml ogc standard for indoor spatial information. http://indoorgml.net/

Deng M, Menassa C, Kamat V (2021) From bim to digital twins: a systematic review of the evolution of intelligent building representations in the aec-fm industry. J Inform Technol Constr 26:58–83. https://doi.org/10.36680/j.itcon.2021.005

Díaz-Vilariño L, Verbree E, Zlatanova S, Diakité A (2017) Indoor modelling from slam-based laser scanner: door detection to envelope reconstruction. ISPRS Int Arch Photogr Remote Sens Spat Inform Sci XLII–2/W7:345–352. https://doi.org/10.5194/isprs-archives-XLII-2-W7-345-2017

Douglas DH, Peucker TK (1973) Algorithms for the reduction of the number of points required to represent a digitized line for its caricature. Cartogr Int J Geogr Inform Geovisualization 10(2):112–122. https://doi.org/10.3138/FM57-6770-U75U-7727

Filgueira A, Laguela S, Arias P, Bueon M (2016) Novel inspection system, backpack-based, for 3d modelling of indoor scenes. Proceedings of the International Conference on Indoor Positioning and Indoor Navigation IPIN

Gao X, Pishdad-Bozorgi P (2019) Bim-enabled facilities operation and maintenance: A review. Adv Eng Inform 39:227–247. https://doi.org/10.1016/j.aei.2019.01.005

Guennebaud G, Jacob B, et al. (2010) Eigen v3. http://eigen.tuxfamily.org

Han L, Zheng T, Xu L, Fang L (2020) Occuseg: Occupancy-aware 3d instance segmentation. 2003.06537

Hong S, Jung J, Kim S, Cho H, Lee J, Heo J (2015) Semi-automated approach to indoor mapping for 3d as-built building information modeling. Comput Environ Urban Syst 51:34–46. https://doi.org/10.1016/j.compenvurbsys.2015.01.005

Hough PV (1962) Method and means for recognizing complex patterns. US Patent 30696541962

Ikehata S, Yang H, Furukawa Y (2015) Structured indoor modeling. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) p 1323-1331, https://doi.org/10.1109/ICCV.2015.156

Itseez (2015) Open source computer vision library. https://github.com/itseez/opencv

Jung J, Stachniss C, Kim C (2017) Automatic room segmentation of 3d laser data using morphological processing. ISPRS Int J Geo-Inform. https://doi.org/10.3390/ijgi6070206

Khoshelham K, Díaz Vilariño L, Peter M, Kang Z, Acharya D (2017) The isprs benchmark on indoor modelling. Int Arch Photogramm Remote Sens Spat Inf Sci. https://doi.org/10.5194/isprs-archives-XLII-2-W7-367-2017

Li L, Su F, Yang F, Zhu H, Li D, Zuo X, Li F, Liu Y, Ying S (2018) Reconstruction of three-dimensional (3d) indoor interiors with multiple stories via comprehensive segmentation. Remote Sens. https://doi.org/10.3390/rs10081281

Lu Q, Xie X, Parlikad AK, Schooling JM (2020) Digital twin-enabled anomaly detection for built asset monitoring in operation and maintenance. Autom Constr 118:103277. https://doi.org/10.1016/j.autcon.2020.103277

Macher H, Landes T, Grussenmeyer P (2017) From point clouds to building information models: 3d semi-automatic reconstruction of indoors of existing buildings. Appl Sci 7(10):1030. https://doi.org/10.3390/app7101030

Martens J, Blankenbach J (2020) An evaluation of pose-normalization algorithms for point clouds introducing a novel histogram-based approach. Adv Eng Inform 46:101132. https://doi.org/10.1016/j.aei.2020.101132

Martens J, Blankenbach J (2021) VOX2BIM : A fast method for automated point cloud segmentation. In: EG-ICE 2021 proceedings: Workshop on Intelligent Computing in Engineering 30th June-2nd July 2021, Technische Universität Berlin, EG-ICE 2021 Workshop on Intelligent Computing in Engineering, Berlin (Germany), 30 Jun 2021 - 2 Jul 2021, Universitätsverlag der TU Berlin, Berlin, pp 442–450, https://doi.org/10.18154/RWTH-2022-02755, https://publications.rwth-aachen.de/record/842885

Matheron G (1967) Éléments pour une théorie des milieux poreux. Mason et cie Editeurs Paris; 1967, P 5 A 164

Motamedi A, Hammad A, Asen Y (2014) Knowledge-assisted bim-based visual analytics for failure root cause detection in facilities management. Autom Constr 43:73–83. https://doi.org/10.1016/j.autcon.2014.03.012

Mura C (2016) University of zurich research datasets. https://www.ifi.uzh.ch/en/vmml/research/datasets.html

Mura C, Mattausch O, Jaspe Villanueva A, Gobbetti E, Pajarola R (2014) Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Comput Graph (Pergamon) 44(1):20–32. https://doi.org/10.1016/j.cag.2014.07.005

Murali S, Speciale P, Oswald MR, Pollefeys M (2017) Indoor Scan-2BIM: Building information models of house interiors. IEEE Int Conf Intell Robots Syst. https://doi.org/10.1109/IROS.2017.8206513

Nävy J (2013) Facil Manag Grundl. Springer Vieweg Berlin Heidelberg, Computerunterstützung, Systemeinführung, Anwendungsbeispiele. https://doi.org/10.1007/978-3-662-07165-6

Ochmann S, Vock R, Wessel R, Klein R (2016) Automatic reconstruction of parametric building models from indoor point clouds. Comput Graph (Pergamon) 54:94–103. https://doi.org/10.1016/j.cag.2015.07.008

Ochmann S, Vock R, Wessel R, Tamke M, Klein R (2014) Automatic Generation of Structural Building Descriptions from 3D Point Cloud Scans. Proceedings of GRAPP 2014 - International Conference on Computer Graphics Theory and Applications January, http://cg.cs.uni-bonn.de/aigaion2root/attachments/GRAPP_2014_54_CR.pdf

Oesau S, Lafarge F, Alliez P (2013) Indoor scene reconstruction using primitive-driven space partitioning and graph-cut. Eurograph Workshop Urban Data Model Vis, UDMV 2013:9–12. https://doi.org/10.2312/UDMV/UDMV13/009-012

Okorn B, Xiong X, Akinci B, Huber D (2016) Toward automated modeling of floor plans. Proceedings of the Symposium on 3D Data Processing. Vis Transm 2(October 2014):8–15. https://doi.org/10.20431/2454-8693.0203002

Otero R, Lagüela S, Garrido I, Arias P (2020) Mobile indoor mapping technologies: A review. Autom Constr 120:103399. https://doi.org/10.1016/j.autcon.2020.103399

Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66. https://doi.org/10.1109/TSMC.1979.4310076

Patacas J, Dawood N, Kassem M (2020) Bim for facilities management: A framework and a common data environment using open standards. Autom Constr 120:103366. https://doi.org/10.1016/j.autcon.2020.103366

Perez-Perez Y, Golparvar-Fard M, El-Rayes K (2021) Scan2bim-net: Deep learning method for segmentation of point clouds for scan-to-bim. J Constr Eng Manag 147(9):04021107. https://doi.org/10.1061/(ASCE)CO.1943-7862.0002132

Perez-Perez Y, Golparvar-Fard M, El-Rayes K (2021) Segmentation of point clouds via joint semantic and geometric features for 3d modeling of the built environment. Autom Constr 125:103584. https://doi.org/10.1016/j.autcon.2021.103584

Pishdad-Bozorgi P, Gao X, Eastman C, Self AP (2018) Planning and developing facility management-enabled building information model (fm-enabled bim). Autom Constr 87:22–38. https://doi.org/10.1016/j.autcon.2017.12.004

Previtali M, Barazzetti L, Brumana R, Scaioni M (2014) Towards automatic indoor reconstruction of cluttered building rooms from point clouds. ISPRS Ann Photogramm Remote Sens Spat Inform Sci II–5:281–288. https://doi.org/10.5194/isprsannals-II-5-281-2014

Previtali M, Diaz Vilarino L, Scaioni M (2018) Towards automatic reconstruction of indoor scenes from incomplete point clouds: door and window detection and regularization. ISPRS - Int Arch Photogram Remote Sens Spat Inform Sci XLII–4:507–514. https://doi.org/10.5194/isprs-archives-XLII-4-507-2018

Qi CR, Litany O, He K, Guibas LJ (2019) Deep hough voting for 3d object detection in point clouds. In: Proceedings of the IEEE International Conference on Computer Vision

Ròka-Madaràsz L, Màlyusz L, Tuczai P (2016) Benchmarking facilities operation and maintenance management using cafm database: Data analysis and new results. J Build Eng 6:184–195. https://doi.org/10.1016/j.jobe.2016.03.007

Rusu RB (2009) Semantic 3d object maps for everyday manipulation in human living environments. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany

Sanchez V, Zakhor A (2012) Planar 3d modeling of building interiors from point cloud data. IEEE Int Conf Image Process. https://doi.org/10.1109/ICIP.2012.6467225

Schnabel R, Wahl R, Klein R (2007) Efficient ransac for point-cloud shape detection. Comput Graph Forum 26(2):214–226

Shi W, Ahmed W, Li N, Fan W, Xiang H, Wang M (2019) Semantic geometric modelling of unstructured indoor point cloud. ISPRS Int J Geo-Inform. https://doi.org/10.3390/ijgi8010009

Tang P, Huber D, Akinci B, Lipman R, Lytle A (2010) Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. Autom Constr 19(7):829–843. https://doi.org/10.1016/j.autcon.2010.06.007

Thomson C, Boehm J (2015) Automatic geometry generation from point clouds for bim. Remote Sens 7(9):11753–11775. https://doi.org/10.3390/rs70911753

Tran H, Khoshelham K, Kealy A (2019) Geometric comparison and quality evaluation of 3d models of indoor environments. ISPRS J Photogramm Remote Sens 149:29–39. https://doi.org/10.1016/j.isprsjprs.2019.01.012

Turner E, Zakhor A (2015) Multistory floor plan generation and room labeling of building interiors from laser range data. In: Battiato S, Coquillart S, Pettré J, Laramee RS, Kerren A, Braz J (eds) Computer vision. Springer International Publishing, Cham, Imaging and Computer Graphics - Theory and Applications, pp 29–44

Turner E, Zakhor A (2014) Floor plan generation and room labeling of indoor environments from laser range data. In: 2014 International Conference on Computer Graphics Theory and Applications (GRAPP), pp 1–12

Wang R, Xie L, Chen D (2017) Modeling indoor spaces using decomposition and reconstruction of structural elements. Photogramm Eng Remote Sens 83(12):827–841

Young IT (1983) Image analysis and mathematical morphology, by j. serra. academic press, london, 1982, xviii + 610 p. \$90.00. Cytometry 4(2):184–185. https://doi.org/10.1002/cyto.990040213

Zhang TY, Suen CY (1984) A fast parallel algorithm for thinning digital patterns. Commun ACM 27(3):236–239. https://doi.org/10.1145/357994.358023