



Special Issue on CAD/Graphics 2015

Automatic reconstruction of parametric building models from indoor point clouds



Sebastian Ochmann*, Richard Vock, Raoul Wessel, Reinhard Klein

Institute of Computer Science II, University of Bonn, Germany

ARTICLE INFO

Article history:

Received 7 April 2015

Received in revised form

11 July 2015

Accepted 12 July 2015

Available online 22 July 2015

Keywords:

Indoor scene reconstruction

Point cloud processing

Parametric models

ABSTRACT

We present an automatic approach for the reconstruction of parametric 3D building models from indoor point clouds. While recently developed methods in this domain focus on mere local surface reconstructions which enable e.g. efficient visualization, our approach aims for a volumetric, parametric building model that additionally incorporates contextual information such as global wall connectivity. In contrast to pure surface reconstructions, our representation thereby allows more comprehensive use: first, it enables efficient high-level editing operations in terms of e.g. wall removal or room reshaping which always result in a topologically consistent representation. Second, it enables easy taking of measurements like e.g. determining wall thickness or room areas. These properties render our reconstruction method especially beneficial to architects or engineers for planning renovation or retrofitting. Following the idea of previous approaches, the reconstruction task is cast as a labeling problem which is solved by an energy minimization. This global optimization approach allows for the reconstruction of wall elements shared between rooms while simultaneously maintaining plausible connectivity between all wall elements. An automatic prior segmentation of the point clouds into rooms and outside area filters large-scale outliers and yields priors for the definition of labeling costs for the energy minimization. The reconstructed model is further enriched by detected doors and windows. We demonstrate the applicability and reconstruction power of our new approach on a variety of complex real-world datasets requiring little or no parameter adjustment.

© 2015 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Digital 3D building models are increasingly used for diverse tasks in architecture and design such as construction planning, visualization, navigation, simulation, facility management, renovation, and retrofitting. Especially for legacy buildings, suitable models are usually not available from the initial planning. Point cloud measurements are often used as a starting point for generating 3D models in architectural software. But despite fast scanning devices and modern software, the generation of models from scratch still are largely manual and time-consuming tasks which make automatic reconstruction methods highly desirable.

Reconstruction of indoor environments poses specific challenges due to complex room layouts, clutter and occlusions. Furthermore, planning and maintenance tasks often require models which give deeper insight into a building's structure on the level of building elements such as walls, and their relations like wall connectivity. This enables high-level editing for prototyping planned changes and simulations requiring information like room

neighborhood or wall thickness. While previous reconstruction methods are able to faithfully recover partially observed surfaces from indoor point clouds and generate accurate boundary representations in the form of mesh models, a plausible decomposition into parametric, globally interrelated, volumetric building elements yet remained an open challenge. Existing approaches either represent walls, floors and ceilings as sets of unconnected planar structures detected in the point cloud [13,1,15,21,7] (Fig. 1(a)), or as collections of closed 3D boundaries of either the whole building [12], or separate rooms [4,19,18,8] (Fig. 1(b)). While the method in [20] reconstructs volumetric walls, their thickness is defined manually instead of being estimated from the input data.

To overcome the limitations of previous approaches, we propose a novel reconstruction method in which the representation of buildings using parametric, interrelated, volumetric elements (Fig. 1(c)) is an integral component. Our approach automatically reconstructs walls between adjacent rooms from opposite wall surfaces observed in the input data while simultaneously taking into account globally plausible connectivity of all elements. Together with a faithful estimation of wall thickness, the result is a high-level editable model of volumetric wall elements. The reconstruction is formulated as an energy minimization problem which simultaneously optimizes costs for assigning rooms to areal

* Corresponding author.

E-mail address: ochmann@cs.uni-bonn.de (S. Ochmann).

regions of the building, and costs for separating adjacent rooms by volumetric wall elements. In contrast to previous approaches, this has the advantage that reasonable binary costs for the assignment of pairs of room labels to adjacent areal regions of the building – and thus the selection of suitable wall elements – is directly incorporated into the global optimization. To make our method robust against large-scale clutter outside the building, outliers are automatically filtered prior to reconstruction. Finally, doors and windows are detected, classified and assigned to the respective wall elements to further enrich the model. Our evaluation using various real-world indoor scans shows that our method rapidly provides models which can be used for e.g. planning of retrofitting, especially since our method requires little or no parameter adjustment.

Applications: The distinguishing feature of our approach is that it directly captures important properties and relations of building elements. Since architectural Building Information Modeling (BIM) formats (e.g. Industry Foundation Classes, IFC) are based on similar relational paradigms, exporting our results to architectural software is straight forward. This enables a whole range of processing and analysis tasks in industry-standard software. We exemplify some applications for e.g. planning of retrofitting in Fig. 2 which can directly be implemented using our results: since the incidence and adjacency relations of walls and rooms are inherently known, selecting e.g. all walls enclosing a room or manipulating whole walls while maintaining overall room topology is easily possible (Fig. 2(a)). This allows for quick, high-level prototyping of changes on the level of semantically meaningful construction element groups. The available information also enables more complex queries for e.g. the subset of wall elements that are simultaneously incident to two adjacent rooms (Fig. 2(b)). Together with directly available properties like wall thickness, openings, room and wall areas, this provides important information for performing acoustic or thermal simulations. The global connectivity information further allows us to perform pathfinding in the whole building story (Fig. 2(c)) for e.g. simulating and optimizing escape routes.

2. Related work

Okorn et al. [13] generate 2D floor plans from 3D point clouds. A histogram of the vertical positions of all measured points is built. Peaks in this histogram are considered to be large horizontal planar structures (i.e. floor and ceiling surfaces). After removing points belonging to the detected horizontal structures, a line fitting on the remaining points is performed. The resulting line segments constituting the floor plan are not connected and do not provide e.g. closed boundaries of rooms. Budroni and Boehm [4] extract planar structures for floors, ceilings and walls by conducting a plane sweep. Using a piecewise linear partitioning of the x - y -plane, they classify cells of this partitioning as inside and outside by determining the occupancy of the cells by measured points and considering densely occupied cells as inside. The result is a 2.5D extrusion of the determined room boundary. In the approach by Sanchez and Zakhor [15], points are classified into floor, ceiling, wall, and remaining points using the point normal orientations. For floor, ceiling and wall points, planar patches are fitted and their extents are estimated using alpha shapes. Parametric staircase models are fitted to the set of remaining points. The resulting mesh models consist of unconnected planar surfaces. Monszpart et al. [7] propose a method for extracting planar structures in point clouds which follow regularity constraints. Their optimization approach balances data fitting and simplicity of the resulting arrangement of planes. A method for generating visually appealing indoor models is proposed by Xiao and Furukawa [20]. An inverse-CSG approach is used for reconstructing the building's geometry by detecting planar structures and then fitting cuboid primitives. These primitives are combined using CSG operations; the quality of the resulting model is tested using an energy functional. Finally, the resulting mesh model is textured using captured images. A drawback is that the building needs to be sufficiently well approximated by the used cuboid primitives. Adan and Huber [1] reconstruct planar floor, ceiling, and wall surfaces from multi-story point clouds by first detecting the modes of a histogram of point

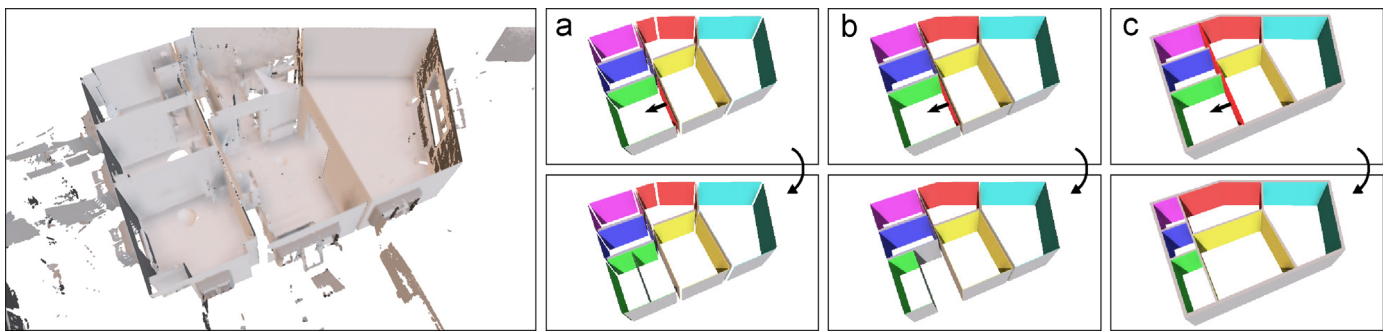


Fig. 1. Schematic of editing capabilities of different kinds of reconstructions. The input point cloud is shown on the left. The remaining columns exemplify editing operations, i.e. elements are moved in the directions of the arrows. Surface representations without (column (a)) or with (column (b)) connectivity information do not allow intuitive editing on the level of wall elements. Our reconstruction (column (c)) maintains room topology and global wall connectivity.

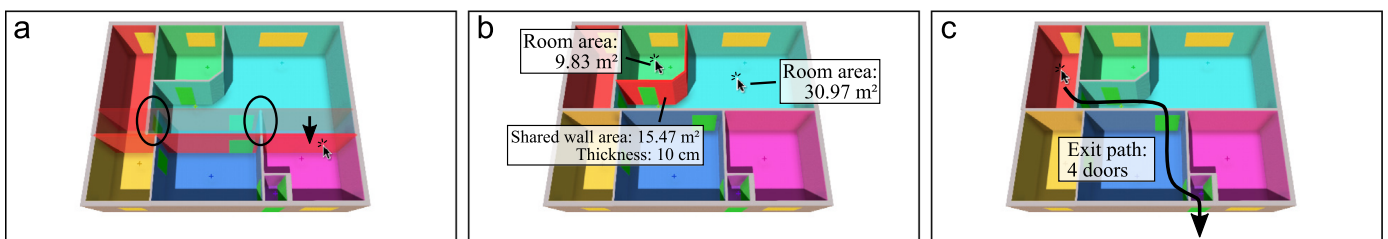


Fig. 2. Example operations which are easily implemented using our results. (a) Relations between walls and rooms enable editing while maintaining room topology. Note how incident walls are adjusted automatically. (b) Automatic determination of wall elements shared between rooms together with automatic measurements enable e.g. acoustic or thermal simulations. (c) Global connectivity enables pathfinding for e.g. simulation and optimization of escape routes.

height values to find horizontal planes, and then detecting vertical planes by means of Hough transform. They recover occluded parts of reconstructed surfaces and perform an opening detection by means of Support Vector Machine (SVM) learning. Xiong et al. [21] extend this approach by classifying detected planar patches as floor, ceiling, wall or clutter using a stacked learning approach, also taking into account contextual information of neighboring patches. Mura et al. [8] reconstruct indoor scenes with arbitrary wall orientations by building a 3D Delaunay tetrahedralization of the input dataset and partitioning inside and outside using a diffusion process governed by affinities of tetrahedron pairs. A binary space partitioning is also done by Oesau et al. [12] by first splitting the input dataset horizontally at height levels of high point densities and then constructing 2D arrangements of projections of detected wall surfaces. The space partitioning into inside and outside is performed by means of Graph-Cut. Other approaches not only perform binary space partitioning but label different rooms: Turner and Zakhor [19] generate 2.5D watertight meshes by first computing an inside/outside labeling of a triangulation of wall points and a subsequent partitioning into separate rooms using a Graph-Cut approach. This method is further developed by Turner et al. in [18], improving the texture mapping capabilities of the algorithm. The results are well-regularized, watertight, textured mesh models. Mura et al. [9] first extract candidate wall elements while taking into account possibly occluded parts of the surfaces to determine the real wall heights for filtering out invalid candidates. After constructing a 2D line arrangement, they use a diffusion embedding to establish a global affinity measure between faces of the arrangement, and determine clusters of faces constituting rooms. The result is a labeled boundary representation of the building's rooms. Many of these methods build upon a spatial partitioning defined by detected wall surfaces and a subsequent classification of regions of this partitioning. Although the resulting models have applications like visualization, navigation or energy monitoring [17], they do not realize a reconstruction of volumetric, interconnected building elements like walls.

3. Approach

The starting point of our approach is a registered point cloud of one building story consisting of multiple indoor scans including scanner positions. Registration is usually done using the scanner software and is outside the scope of this paper. The unit of measurement and up direction are assumed to be known. Surface normals for each point are estimated.

We argue that the wall structure of most building stories can be represented as a piecewise-linear, planar graph in which edges represent wall elements and vertices are locations where walls are incident (Fig. 3(e)). Wall thickness is a scalar edge attribute. Conversely, faces of this graph represent the spatial room layout. There obviously exists a duality between the story's room layout and its wall constellation, i.e. one representation can directly be derived from the other. The main idea of our approach is that – while both representations are essentially equally hard to reconstruct – we can derive important hints (priors) for the room layout from indoor point cloud scans since they are a sampling of the inner surfaces of room volumes. It is therefore meaningful to base our reconstruction on the derivation of a suitable room layout from which the constellation of walls is immediately obtained due to the duality.

We extract priors for the room layout as follows: assuming that each room was scanned from one position (or few positions), separate scans yield a coarse segmentation of the point cloud into separate rooms (Fig. 3(a)). We improve this segmentation using a

diffusion process which eliminates most overlapping regions between scans (Fig. 3(b)) and automatically filters out clutter outside of the building. As further described below, the determination of a suitable room layout is then formulated as a labeling problem of the regions of a suitable partitioning of the horizontal plane (using labels for different rooms and the outside area). This directly follows the aforementioned duality principle: after determining a suitable labeling, connected components of identically labeled cells are rooms, and edges separating differently labeled regions are wall elements.

Since our goal is to extract a piecewise-linear graph of walls, we construct a partitioning based on potential wall surfaces: we first detect vertical planes as candidates for wall surfaces and project them to the horizontal plane (Fig. 3(c)). Similar to previous approaches [9,11,12] we then construct an arrangement of (infinitely long) lines from the set of possible wall surfaces (Fig. 3(d)). In contrast to previous approaches, edges of this arrangement represent wall *centerlines* instead of wall *surfaces*. Furthermore, arrangement lines are not only constructed from *single* wall surfaces but also from *pairs* of parallel surfaces which yield candidates for walls separating adjacent rooms. This subtle but crucial difference allows us to go beyond the reconstruction of *separate* room volumes as done in previous works (Fig. 1(b)) by enabling the algorithm to reconstruct room-separating wall elements directly. In order to guide the selection of adequate wall elements, we retain the information from which supporting measured points each edge originates. This yields wall selection priors encouraging the reconstruction of wall elements which were constructed from surfaces belonging to the *same* pair of rooms that the wall separates.

The determination of a globally plausible labeling is then formulated as an energy minimization problem. This allows us to incorporate room layout priors and wall selection priors as unary and binary costs into one optimization. After an optimal labeling has been determined, only retaining edges separating differently labeled regions are the sought wall structures (Fig. 3(e)). Extruding walls according to estimated room heights and a detection and classification of openings yields the final parametric model (Fig. 3(f)).

4. Point cloud segmentation

To obtain priors for the localization of rooms in subsequent steps, each point of the input point cloud is automatically assigned a label for a room or the outside area. Our approach is based on the method by Ochmann et al. [10] which we will briefly summarize before describing our modifications: the original method assumes at least one scan within each room; multiple scans per room are merged manually such that a one-to-one mapping between (merged) scans and rooms is obtained. The initial assignment of each point to one of the (merged) scans (Fig. 3(a)) provides a coarse segmentation of the point cloud into rooms. However, openings such as open doors lead to severe overlaps between scans, causing large areas of the point cloud to contain a mix of differently labeled points. To obtain a point labeling that roughly corresponds to the building's room layout and is homogeneous within each room (Fig. 3(b)), an automatic labeling refinement is performed. The process is based on the assumption that most points that are *visible* from the position of a point p are already labeled correctly. By determining which points are visible from the position of p and averaging the observed labels, a new (soft) labeling of p is obtained. After iterating this procedure, the label with the highest confidence is assigned to p . This process can be interpreted as a diffusion of point labels between points governed by mutual visibility. In practice, a stochastic ray casting from the

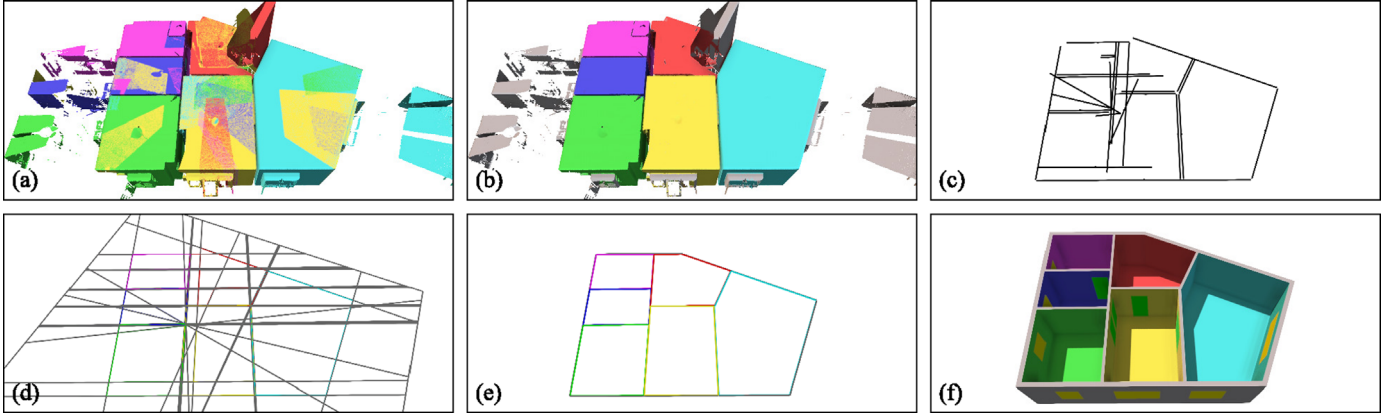


Fig. 3. Overview of our approach (see also Section 3). (a) Input point cloud; assignment of points to scans shown in different colors. (b) Refined assignment after automatic segmentation. (c) Detected vertical planes transferred to the horizontal plane. (d) Candidates for walls are derived from single and pairs of projected planes. Intersecting their centerlines yields a planar graph whose faces are subsequently assigned labels for rooms or outside area. (e) Only edges separating differently labeled faces are retained. (f) The final model with detected and classified wall openings, e.g. doors (green) and windows (yellow). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

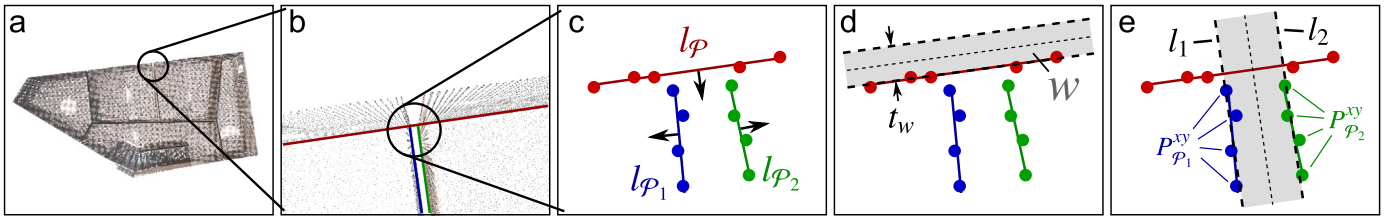


Fig. 4. Wall candidate generation. (a) and (b) Detected vertical planes in the 3D point cloud are projected into the horizontal plane. (c) Different wall surface lines including the respective (projected) support points and surface normals. (d) For each single wall surface, an infinitely long wall candidate w for a wall separating a room from outside area is generated. In this case, the thickness t_w is user-specified. (e) For each pair of approximately parallel wall surfaces, a candidate for separating adjacent rooms is generated. In this case, wall thickness is estimated from the data.

position of p into the hemisphere around the normal of p is performed.

We extend this method in two ways: first, we automatically filter out clutter outside of the building which is often caused by windows or mirrors. We argue that for a point p that is part of clutter outside of the building, most rays cast from p into the hemisphere around the normal of p do not hit any interior wall surfaces. In this case we assign a high value for an additional outside label to p . This modification proves to be highly effective in our experiments as demonstrated in Fig. 3(b) (gray points have been assigned the outside label). Second, we do not require that multiple scans per room are merged manually. Instead, we run the reconstruction using all scans as separate labels. In case of multiple scans in a room, this leads to implausible walls within rooms which are subsequently removed as described in Section 7.

5. Generation of wall candidates

Candidates for wall elements are derived from vertical surfaces observed in the scans. They constitute possible locations of walls for the optimization in Section 6. Since wall heights and lengths are not regarded in this step, the following 2D representation is used: each wall candidate $w = (t_w, n_w, d_w)$ is defined by a thickness $t_w \in \mathbb{R}^{>0}$ and an infinite centerline in the horizontal plane given in Hesse normal form $\langle n_w, x \rangle - d_w = 0$. Wall heights and lengths will be determined later.

In a first step, planes in the 3D point cloud are detected using a RANSAC implementation by Schnabel et al. [16]. Nearly vertical planes ($\pm 1^\circ$) with a sufficiently large approximate area ($\geq 1.5 \text{ m}^2$) are considered as potential wall surfaces. For a plane \mathcal{P} fulfilling these constraints, let $n_{\mathcal{P}} \in \mathbb{R}^3$ be the plane normal and $P_{\mathcal{P}}$ the set of

measured points supporting \mathcal{P} . Each extracted plane \mathcal{P} is transferred to the horizontal plane as a *wall surface line* $l_{\mathcal{P}}$ defined by $\langle n_{l_{\mathcal{P}}}, x \rangle - d_{l_{\mathcal{P}}} = 0$. A schematic example for the extraction of wall surface lines is shown in Fig. 4(a)–(c). The normal $n_{l_{\mathcal{P}}}$ is approximated by the projection of $n_{\mathcal{P}}$ into the horizontal plane,

$$n_{l_{\mathcal{P}}} := \frac{((n_{\mathcal{P}})_x, (n_{\mathcal{P}})_y)}{\|((n_{\mathcal{P}})_x, (n_{\mathcal{P}})_y)\|_2}.$$

The distance to the origin $d_{l_{\mathcal{P}}}$ is determined by least squares fitting to the set $P_{\mathcal{P}}^{xy}$ of support points projected to the horizontal plane using the fixed normal $n_{l_{\mathcal{P}}}$ such that $\sum_{p \in P_{\mathcal{P}}^{xy}} (\langle n_{l_{\mathcal{P}}}, p \rangle - d_{l_{\mathcal{P}}})^2$ is minimized. From the wall surface lines, we then generate two kinds of wall candidates as we do not know at this point which types of candidates will yield a globally plausible reconstruction:

Outside walls: For each *single* wall surface line $l_{\mathcal{P}}$, we construct a candidate for a wall separating a room from the outside area (Fig. 4(d)). Since the real wall thickness cannot be determined automatically from a single surface, a user-specified thickness is used (in our experiments, $t_w = 20 \text{ cm}$). The centerline of the candidate is constructed such that the side of the wall candidate that points towards the inside of the room is identical to $l_{\mathcal{P}}$, i.e. the centerline is defined by $\langle n_{l_{\mathcal{P}}}, x \rangle - d_w = -t_w/2$.

Room-separating walls: To generate candidates for walls separating adjacent rooms, each *pair* of wall surface lines fulfilling certain constraints is considered as two opposite surfaces of a wall separating adjacent rooms (Fig. 4(e)). Let $l_{\mathcal{P}_1}$ and $l_{\mathcal{P}_2}$ be two wall surface lines that are approximately parallel ($\pm 1^\circ$) and have opposing normal orientations. To prune invalid pairs, a coarse check is performed whether the projected support pointsets of the originating planes $P_{\mathcal{P}_1}^{xy}, P_{\mathcal{P}_2}^{xy}$ (partially) overlap. To this end, the support pointsets are projected onto the respective opposite line. If support points are present near the projected points, their

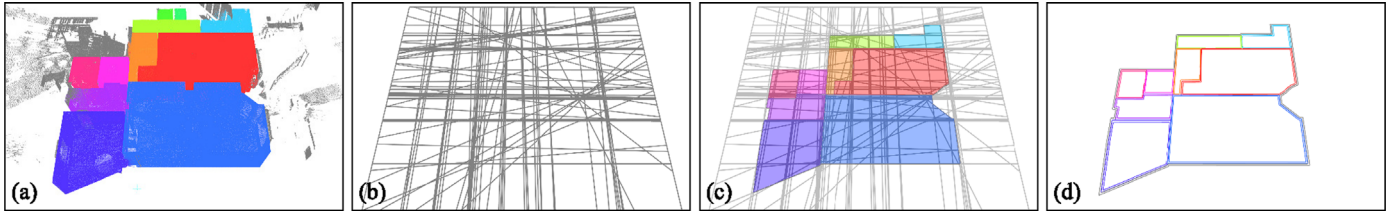


Fig. 5. Determination of suitable wall candidate segments. (a) Input point cloud after segmentation. (b) Intersecting all wall candidate centerlines yields a planar graph. We determine an assignment of all faces to rooms or outside area such that connected components of identically labeled faces are rooms and edges between differently labeled faces are wall elements. (c) Resulting labeling of faces after optimization; colors indicate room labels. (d) Retaining only edges separating differently labeled faces yields a subgraph representing the sought wall elements and their connectivity. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

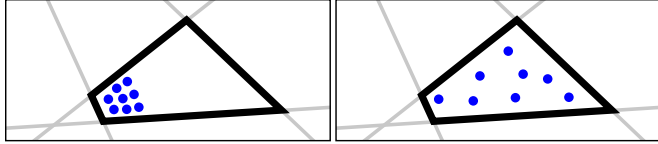


Fig. 6. Considering only the number of projected points within a face for unary costs does not take into account their spatial distribution.

support is considered overlapping. For each pair fulfilling these constraints, a wall candidate is generated by fitting to l_{p_1} and l_{p_2} simultaneously: The candidate's normal n_w is first determined as the average of the normals n_1, n_2 of l_{p_1}, l_{p_2} , weighted with the cardinality of the support pointsets,

$$n_w := \frac{|P_{p_1}^{xy}| n_1 + |P_{p_2}^{xy}| (-n_2)}{\|(|P_{p_1}^{xy}| n_1 + |P_{p_2}^{xy}| (-n_2))\|_2}.$$

Using the common normal n_w , two parallel lines $l_i, i \in \{1, 2\}$ defined by $\langle n_w, x \rangle - d_i = 0$ are fitted to the respective support pointsets such that $\sum_{p \in P_{p_i}^{xy}} (\langle n_w, p \rangle - d_i)^2$ is minimized. The centerline of the wall candidate is constructed midway between the parallel lines, $\langle n_w, x \rangle - \frac{1}{2}(d_1 + d_2) = 0$, and the candidate's thickness is defined as the distance between them, $t_w = |d_1 - d_2|$. Candidates with a thickness above a threshold are discarded (in our experiments, $t_w > 60$ cm).

6. Determination of an optimal room and wall layout

From the infinitely long wall candidates, we determine a set of wall segments which yields a plausible reconstruction of the building's walls. To this end, we consider the intersection of all wall candidate centerlines in the horizontal plane which yields a planar graph $W' = (V', E')$ (Fig. 5(b)). Faces of W' are regions of the building's layout (i.e. parts of rooms or outside area), edges E' are segments of possible walls, and vertices V' are possible locations where walls are incident. We follow the intuition that walls separate different regions, i.e. adjacent rooms, or rooms and the outside world. Consequently, a classification of the faces of W' implies locations of walls in the following sense: connected components of identically labeled faces are rooms (or outside areas), and edges between differently labeled faces are walls. Fig. 5(c) shows an example for a face labeling from which connected wall elements as shown in Fig. 5(d) are extracted. Wall thickness of an edge e is set to the thickness of the wall candidate from which e originates.

We formulate the face classification as a labeling problem which is solved using an energy minimization approach. The target functional has two terms: unary costs for the assignment of labels to faces of W' and binary costs for the assignment of label pairs to adjacent faces. Unary costs provide hints where rooms (or outside areas) are located and binary costs guide the selection of adequate edges for separating differently labeled faces. In particular, if two adjacent rooms share a

common wall, a wall candidate constructed from wall surfaces of *these* rooms should separate them. We will now formalize the problem. Let $W = (V, E)$ be the dual graph of W' and let $\{l_1, \dots, l_k, l_o\}$ be the set of labels where $l_i, i \in \{1, \dots, k\}$ are labels for each scan and l_o is the outside label. For clarity, we assume for now that each room was scanned from exactly one position and thus k equals the number of rooms; the more general case of multiple scans per room will be discussed later. Given a unary cost function $U_v(l_v)$ yielding the cost for assigning label l_v to a vertex $v \in V$, and a binary cost function $B_{v,w}(l_v, l_w)$ yielding the cost for assigning the (unordered) pair of labels l_v, l_w to $v, w \in V$, we minimize the total cost for a labeling l , i.e.

$$E(l) = \sum_{v \in V} U_v(l_v) + \sum_{(v,w) \in E} B_{v,w}(l_v, l_w) \rightarrow \min. \quad (1)$$

Applying the minimization algorithm to the dual graph W of W' allows us to determine a labeling of the faces of W' by finding an optimal labeling of the vertices of W . The problem stated in Eq. (1) is solved using the algorithm by Boykov et al. [3,6,2]. We now define unary and binary cost functions for label assignments. In the following, the notation for *label vectors*

$$\mathcal{L}(\cdot) = (c_1, \dots, c_k, c_o), \quad \forall i: c_i \geq 0, \quad \|\mathcal{L}(\cdot)\|_1 = 1$$

will be used for soft label assignments to different entities, e.g. points, faces and edges. The coefficient c_i of $\mathcal{L}(\cdot)$ corresponding to label l_i will be denoted $\mathcal{L}_i(\cdot)$. As a shorthand, let \mathcal{I}_i denote a hard label vector with $c_i = 1$, and let $\mathcal{I}_{ij} := \frac{1}{2}(\mathcal{I}_i + \mathcal{I}_j)$.

Unary costs: Intuitively, the cost $U_v(l_v)$ shall be low iff the area spanned by face f in W' is likely to belong to l_v . We first estimate a label vector $\mathcal{L}(f)$ whose coefficients reflect the probabilities that the area covered by f belongs to each room or the outside area. A naive approach would be to project all measured points into the horizontal plane and to determine how many points of each room (with respect to the point labels obtained in Section 4) are located within f . The first problem is that non-uniform distributions of measured points (Fig. 6, left) yield a similar probability estimate like a uniform distribution (Fig. 6, right) although the latter provides stronger evidence that the whole face belongs to a certain room. The second problem is that we need to estimate the probability that f is located in the outside area which is not represented by measured points.

We therefore propose a stratified sampling method which takes the spatial distribution of projected measured points into account and yields an estimate for the outside label. All measured points are projected into a uniform 2D grid in the horizontal plane. The side length of the grid cells is chosen as twice the point cloud subsampling density (see Section 8). The label vectors of all points within a grid cell are averaged and empty cells are assigned the outside label vector \mathcal{I}_o . Subsequently, the label vector $\mathcal{L}(f)$ of f is estimated by picking in the grid at uniformly sampled positions within f and averaging the resulting label vectors. The number of samples within f is chosen proportionally to the face area (at least

one sample is enforced). The unary cost function is then defined as

$$U_v(l_v) := \alpha \cdot \text{area}(f) \cdot \|\mathcal{L}(f) - \mathcal{I}_v\|_1, \quad (2)$$

where v is the vertex of W corresponding to face f in W' , and α is a weighting factor (see Section 8). $\mathcal{L}(f)$ is the estimated labeling of face f , and \mathcal{I}_v is the ideal expected label vector for label l_v . The distance between these label vectors is weighted proportionally to the area of f in order to mitigate the impact of differently sized faces in the sum of total labeling costs.

Binary costs: For the binary cost $B_{v,w}(l_v, l_w)$, consider edge e in W' to which the edge (v, w) in W corresponds. Intuitively, the cost for assigning labels l_v, l_w to $v \in V$ and $w \in V$ shall be low iff the surfaces of the wall represented by e are supported by measured points with labels l_v, l_w (in the case of wall bordering the outside area, there should be no support on the exterior side). In other words, for the separation of faces with different labels l_v, l_w , wall elements whose surfaces are supported by points with labels l_v, l_w shall be preferred. For estimating the label vector for an edge e , a sampling strategy similar to the face label vectors is used. Consider edge e originating from up to two wall surface lines l_{p_1}, l_{p_2} (see Section 5) with according projected support points $P_{p_1}^{xy}, P_{p_2}^{xy}$. If e originates from a single wall surface line l_{p_1} , we set $P_{p_2}^{xy} = \emptyset$. Analogous to the 2D grid in the horizontal plane, we construct a one-dimensional grid on e . The support points $P_{p_1}^{xy} \cup P_{p_2}^{xy}$ are projected into the grid and their point labels are averaged per cell. Empty cells are assigned the outside label. The label vector $\mathcal{L}(e)$ is now estimated by sampling uniformly distributed points on e and averaging the label vectors obtained by picking in the grid at the sample positions. We then define the binary costs as

$$B_{v,w}(l_v, l_w) := \begin{cases} \beta \cdot \text{len}(e) \cdot (\|\mathcal{L}(e) - \mathcal{I}_{vw}\|_1 + \gamma \mathcal{L}_o(e)) & \text{if } l_v \neq l_w, \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where v, w are the vertices of W corresponding to faces f, g in W' that are separated by edge e , $\text{len}(e)$ is the Euclidean length of edge e , and β, γ are weighting factors (see Section 8) respectively. Similar to the unary costs, weighting the distance between the observed and ideal label vectors by edge length mitigates the influence of different edge lengths. The additional term $\mathcal{L}_o(e)$ penalizes usage of edges with a high outside prior. We found that this term helps to select correct edges with support points on both sides for separating adjacent rooms. After the face labeling is determined, only edges which separate differently labeled faces are retained. The resulting subgraph \bar{W} of W' (Fig. 5 (d)) is used in Section 7 for reconstructing connected wall elements.

Multiple scans within one room: We previously assumed that each room was scanned from exactly one position within that room. In the case of more than one scan, one room is represented by a set of different labels. Fig. 7(a) shows an example of a hallway scanned from three positions. After segmentation (Section 4), the hallway is split into multiple regions represented by differently labeled points (Fig. 7(b)). The graph labeling optimization separates these sections by implausible walls (Fig. 7(c)). We remove

such walls (Fig. 7(d)) as part of the opening detection in the next section.

7. Model generation and opening detection

From the determined graph, the final model can now be derived in a straight forward manner. The model is further enriched by detected window and door openings.

Walls: For each edge $\bar{e} = (\bar{v}, \bar{w})$ of \bar{W} , a wall element \mathcal{W} is constructed with centerline endpoints located at \bar{v} and \bar{w} . The thickness of \mathcal{W} is determined by the thickness of the wall candidate from which \bar{e} originates. Endpoints of wall elements are connected iff the corresponding edges are incident to a common vertex. For vertical extrusion, we first estimate floor and ceiling heights for each face \bar{f} in \bar{W} separately using the following heuristic: consider all approximately horizontal planes detected during wall candidate generation (Section 5). For each plane, the number of support points located within \bar{f} is determined. The elevation of the plane with the largest support within \bar{f} and upwards- (resp. downwards-) facing normal is chosen as the floor height $h_{fl}(\bar{f})$ (resp. ceiling height $h_{cl}(\bar{f})$) of \bar{f} . The vertical extent of a wall represented by edge \bar{f} separating faces \bar{f}_1, \bar{f}_2 is then defined to span the heights of both adjacent rooms: $[\min(h_{fl}(\bar{f}_1), h_{fl}(\bar{f}_2)), \max(h_{cl}(\bar{f}_1), h_{cl}(\bar{f}_2))]$.

Opening detection: Openings in walls either arise from doors and windows, or because a reconstructed wall was artificially introduced due to multiple scans within one room as described in Section 6. By classifying detected openings accordingly, we further enrich the model by doors and windows, and determine which walls to remove for handling multiple scans within rooms. To locate potential openings, we determine intersection points between reconstructed walls and simulated laser rays from the scan positions to the measured points. The intersection points are clustered in the 2D domain of the wall surfaces (a simple greedy, single-linkage clustering based on distances between intersection points yielded satisfactory results); see Fig. 8 (b) for an example. The clusters are then classified as doors, windows, virtual (i.e. openings due to excess walls) or invalid (i.e. clutter) by means of supervised learning using libsvm [5]. Six-dimensional feature vectors with the following features are used to characterize openings: cluster bounding box width and height, distance from lower and upper wall bounds, approximate coverage by intersection points, and a binary feature indicating whether the associated wall is adjacent to outside area. Clusters recognized as doors or windows are assigned to the respective wall elements. Adjacent faces of \bar{W} separated by wall elements containing at least one virtual opening (magenta clusters in Fig. 8(b)) are merged by removing all edges to which both faces are incident. To account for changes after a wall removal, the determination of room heights, intersection points, clusters and opening classes is performed iteratively until no more virtual openings exist.

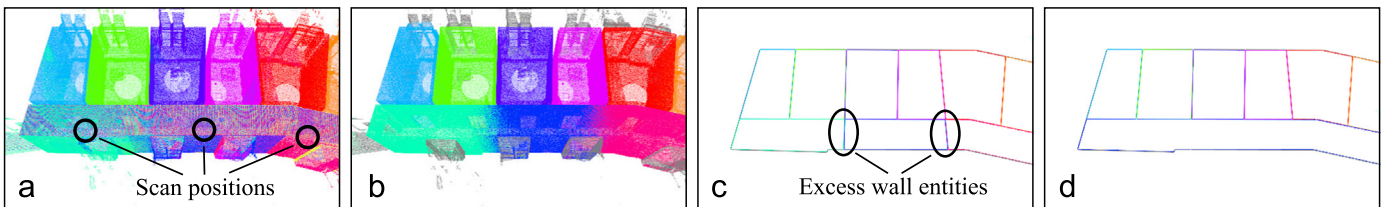


Fig. 7. Multiple scans within a single room. (a) The hallway has been scanned from three positions; room labels are mixed within that room. (b) After segmentation (Section 4), the hallway is still split into multiple sections. (c) The labeling algorithm separates these regions by wall elements that are not part of the building's true walls. (d) By detecting and removing excess wall elements, faces are merged to larger rooms.

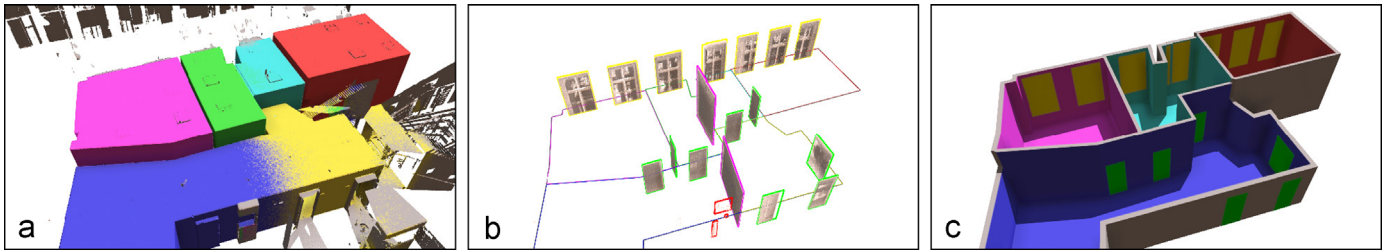


Fig. 8. Opening detection and classification. (a) The input point cloud after segmentation. (b) Detected clusters of intersections between reconstructed walls and simulated laser rays between scanner positions and measured points. Clusters are classified as doors (green), windows (yellow), “virtual” clusters indicating walls to be removed for merging multiple scans within a room (magenta), and invalid (red). (c) The final model after removal of walls containing “virtual” openings. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Table 1

Datasets used in our experiments. Figure references marked with “*” indicate that only a subset of the dataset is shown.

Dataset	Points	Scans	Time (s)	Figures
Building A, storey 1	9,524,724	23	84.9	12*
Building A, storey 2	19,365,622	33	215.7	16(a)
Building B, storey 1	826,229	5	6.0	—
Building B, storey 2	1,676,486	6	11.9	9*
Building B, storey 3	1,673,919	6	12.2	1, 3
Building B, storey 4	2,203,670	8	16.0	2
Building B, storey 5	2,470,678	11	17.7	5
Building C, storey 1	4,749,565	9	39.7	14*
Building C, storey 2	22,757,718	67	486.3	7*, 8*, 16(c)
Building C, storey 3	23,883,396	63	449.6	—
Building D, storey 1	17,712,659	34	252.2	15
Building E, storey 1	14,399,907	37	189.3	13*
Building E, storey 2	19,769,647	51	319.9	—
Building E, storey 3	17,104,101	43	241.5	16 (b)

8. Evaluation

We tested our approach on real-world point clouds of 14 stories from 5 different buildings; statistics are given in Table 1. The shown number of points is after subsampling with the Point Cloud Library [14] using a resolution of $\epsilon = 0.02$ cm (i.e. in a voxel grid with a resolution of ϵ , at most one point in each voxel is retained). Normals are estimated by means of local PCA using point patches of 48 nearest neighbors. Normals are flipped towards the respective scanner position.

Parameter selection: The first set of crucial parameters affects plane detection in the extraction of wall surfaces (Section 5). For classifying planes as vertical (wall surfaces) or horizontal (floor and ceiling surfaces) we chose a threshold on the angular deviation of $\pm 1^\circ$ from the ideal orientations. We ignore planes with less than 500 support points or an approximate areal coverage by support points below 1.5 m^2 . Also, vertical planes resulting in line segments below 0.5 m are ignored. These parameters control a tradeoff between avoiding clutter and ignoring small details: high thresholds only consider larger (but potentially more stable) planes as candidates for wall surfaces. Conversely, low thresholds may introduce clutter due to incorrectly detected planes. Fig. 9 demonstrates different choices. The second important set of parameters consists of the weights α, β, γ in Eqs. (2) and (3). In our experiments, we found that a ratio of α/β of $4/1$, and $\gamma = 4$ yielded good results (Fig. 10(c)). The effects of setting either α, β , or γ to zero are shown in Fig. 10(d)–(f). We also found that smoothing the 2D and 1D grids used for the determination of face and edge label vectors in Section 6 using a large Gaussian kernel usually improves results.

Robustness: Quality and robustness of our reconstruction depend on plane detection quality which is influenced by e.g. scanner noise, point density, registration accuracy, and clutter outside and inside of the building. As our datasets were captured

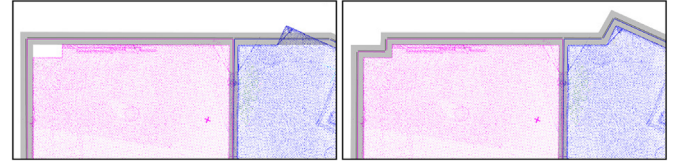


Fig. 9. Different plane detection options: Allowing smaller planes as potential wall surfaces allows for more detailed structures (right-hand side) at the cost of possibly detecting incorrect candidates in clutter.

using professional laser scanners, noise level and sampling density were no issue. Registration errors directly influence the position of detected planes and thus the generated wall candidates. Our algorithm adapts well to small misalignment; stronger translational or rotational alignment errors have specific effects as exemplified in Fig. 11. Clutter outside of the building is effectively eliminated by our automatic filtering method. Clutter inside of rooms and scan holes pose big challenges when working with indoor scans. Except for extreme cases (e.g. completely unobserved wall surfaces or objects which span the whole story height and thus yield planes that are indistinguishable from walls), our algorithm proves to be robust against e.g. furniture within rooms as shown in Fig. 12: despite the presence of large scan holes, using all available points (from e.g. furniture) as priors for room localization closes holes, and the smoothness property of the used graph-cut-based optimization yields well-regularized walls. Furthermore, as our approach uses infinitely long wall candidates, small or medium sized holes in the support pointsets of wall surfaces caused by occlusions are automatically bridged in a plausible manner (Fig. 13). We also found that the algorithm is very robust against errors in the segmentation step (Section 4), especially in the interior of the building, i.e. if overlaps between scans of adjacent rooms still exist. However, filtering out large-scale clutter outside the building is important in order to avoid erroneous classification of outside area as rooms.

Opening detection accuracy: Our method for opening detection consists of three parts (see Section 7): (1) determination of intersection points, (2) clustering of intersection points, and (3) classification of these clusters using supervised learning. Regarding (1), we found that the determined intersection points indicate locations and extents of openings well, with a low number of false positives. For (2), a simple single-linkage clustering based on point distances already yielded good results due to the low number of clutter intersection points which may cause chaining effects. However, a more sophisticated clustering method could improve results in some cases, e.g. multiple neighboring windows are sometimes recognized as a single cluster. Concerning (3), our training examples comprise 269 doors, 306 windows, 118 virtual, and 415 invalid clusters which were obtained by manually correcting a heuristic classification. During training, all stories originating from the building being classified were removed from the training set. Average cross validation rate of the training sets

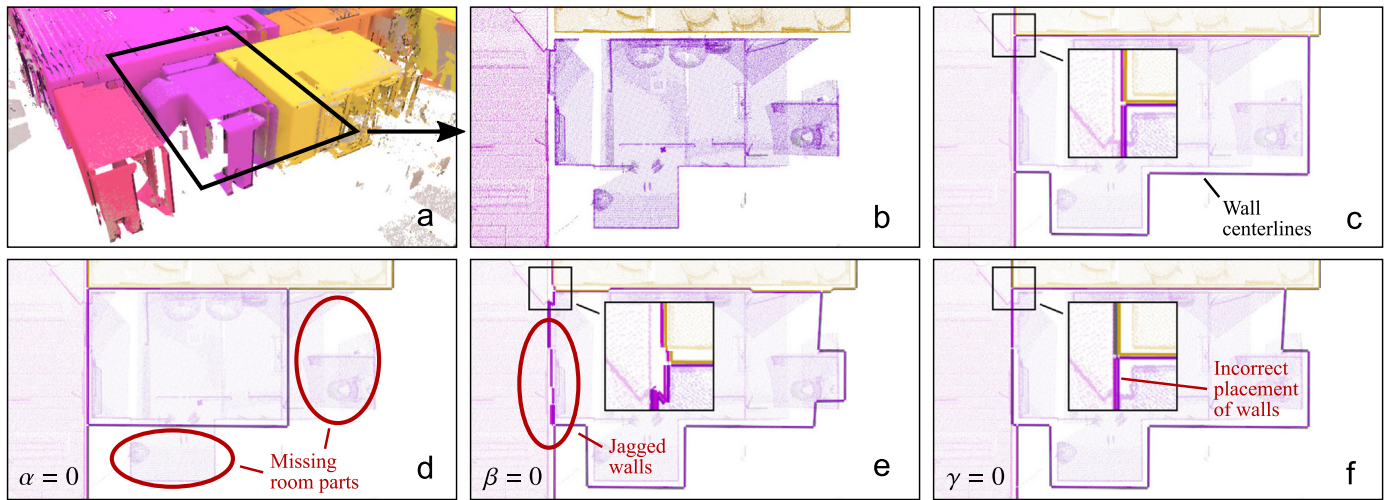


Fig. 10. Different choices for α, β, γ in Eqs. (2) and (3). (a) and (b) Perspective and orthographic view of an example situation. (c) Parameters chosen as described in Section 8. Wall centerlines are well-regularized and common wall elements have been reconstructed between rooms. (d) Without unary costs ($\alpha = 0$). While the resulting walls are well-regularized, parts of rooms are missing despite high areal support by measured points. (e) Without binary costs ($\beta = 0$). Walls are located similar to (c) but are overly complex due to missing regularization and preference for correctly labeled edges. (f) Without penalty for high outside labeling ($\gamma = 0$). The algorithm does not prefer common walls for separating adjacent rooms.

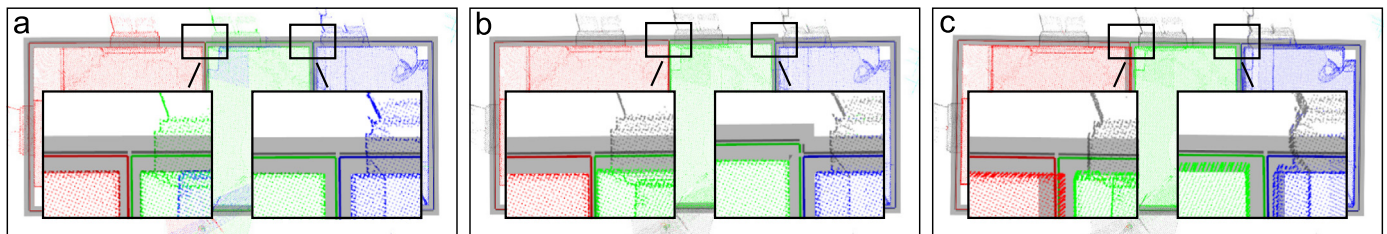


Fig. 11. Effects of registration errors. (a) Result without alignment errors. Wall volumes are shown in gray together with the respective wall centerlines. (b) Translational registration errors may result in offset walls to which the algorithm adapts accordingly (right detail view). Wall thickness may also change (wall separating the red and green rooms in the left detail view). (c) Rotational registration errors may lead to wall surface pairs not to be associated to common walls. Wall thickness is incorrect since the wall candidates do not originate from wall surfaces pairs. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

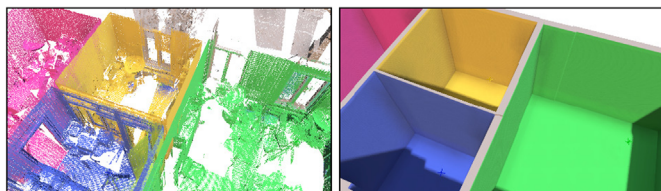


Fig. 12. Highly cluttered rooms. Left: clutter and transparent surfaces (windows) cause large scan holes; wall surfaces are only partially scanned. Right: reconstructed walls still are well regularized and separate rooms correctly.

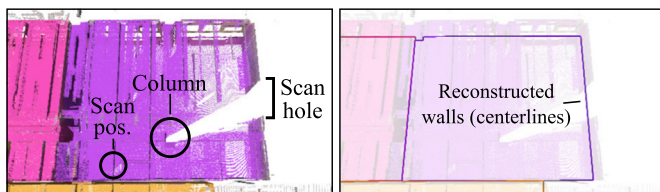


Fig. 13. Our cost minimization approach and infinitely long wall candidates automatically bridge scan holes in a plausible manner.

was 90.34%, average classification accuracy was 85.02%. This small yet significant gap indicates a generalization performance below optimum which we believe is caused by systematic differences between e.g. the used windows in different buildings, causing the feature vectors to not be i.i.d. Given the limited number of test data, we think that our approach is promising, especially since newly obtained examples can be fed back into the algorithm.

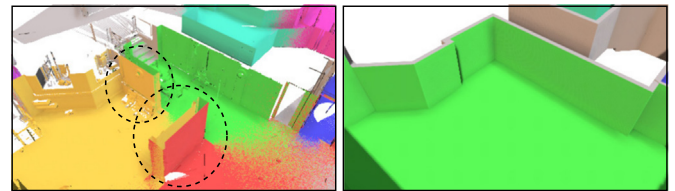


Fig. 14. Wall elements which are not connected at both ends to other walls are currently not representable by our reconstruction.

Comparison to manually generated models: A visual comparison between our reconstruction and a professional, manually generated model is shown in Fig. 15. Locations and thickness of wall elements, and locations of doors are generally good; a few walls are missing either due to the fact that (small) rooms were not scanned separately and thus room labels are missing, or because openings were misclassified as “virtual” clusters.

Time and memory requirements: Our experiments were run on a 6-core Intel Core i7-4930K (32 GB RAM) with a NVIDIA GeForce GTX 780 (3 GB RAM). Processing times of our prototypical implementation are shown in Table 1. Peak RAM usage (incl. visualization) for the largest dataset (Fig. 16c) was about 16 GB.

Limitations: If rooms are not completely enclosed by walls (e.g. balconies or partially scanned staircases), points might erroneously be classified as outside area during the segmentation step which may lead to missing parts in the reconstruction. Due to the current formulation of our approach, wall elements which are not connected to other walls at both ends cannot be represented.

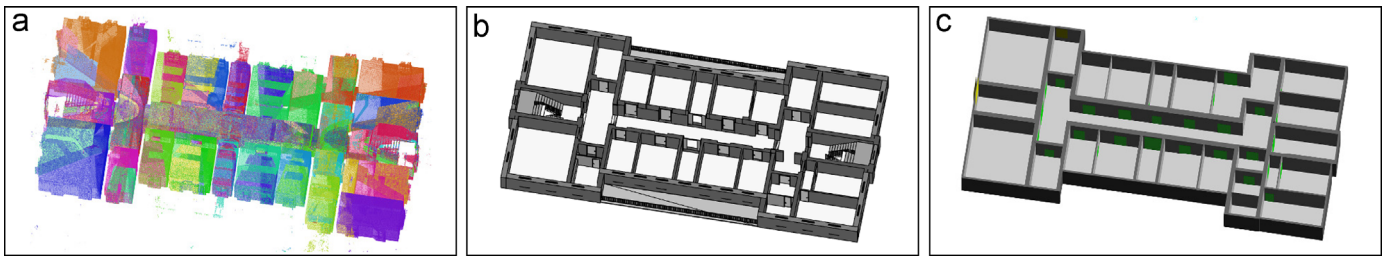


Fig. 15. Visual comparison of reconstruction and manually generated model. (a) Input point cloud; scans are shown in different colors. (b) Professionally, manually generated model. (c) Reconstructed model. Locations and thickness of walls, and locations of doors are generally good. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

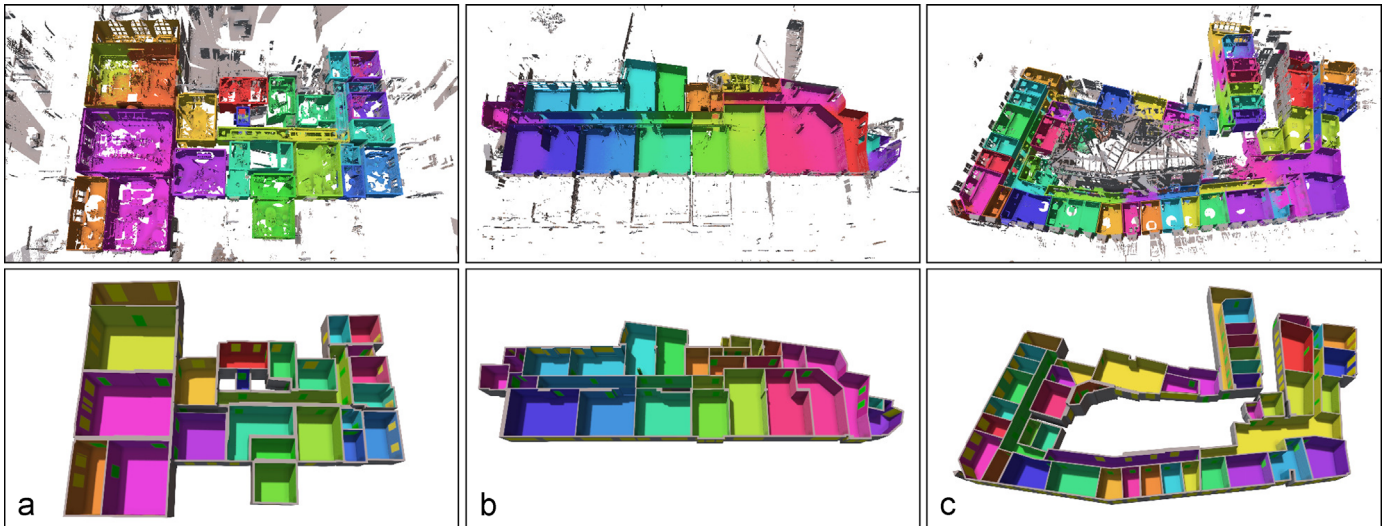


Fig. 16. Example results on point clouds with 33, 43, and 67 scans. Upper row: point clouds after segmentation step; most ceiling points (i.e. points with downwards-facing normals) are removed for visualization. Lower row: reconstructed models; detected windows are shown in yellow, doors are shown in green. Most wall elements are faithfully reconstructed; some excess walls have not been removed (see e.g. the large room in the lower-right corner of the second column). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

As a consequence, they are either missing (Fig. 14), or erroneously connected to other wall elements. Also, since we only consider planar wall surfaces and linear wall candidates, only piecewise linear wall structures can be reconstructed.

9. Conclusion and future work

We presented the first automatic method for the reconstruction of high-level parametric building models from indoor point clouds. The feasibility of our approach was demonstrated on a variety of complex real-world datasets which could be processed with little or no parameter adjustments. In the future, a more thorough comparison of reconstruction results with existing, manually generated models would help to analyze reconstruction results quantitatively. A generalization to multiple building stories poses specific challenges but would enable the reconstruction of multi-story models without the need to process stories separately. Also, the usage of different capturing devices (e.g. mobile devices) and real-time handling of streamed data are topics for future investigation.

Acknowledgments

This work was partially funded by the German Research Foundation (DFG) under grant KL 1142/9-2 (MoD), as well as by the European Community under FP7 grant agreements 600908 (DURAARK) and 323567 (Harvest4D).

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version at <http://dx.doi.org/10.1016/j.cag.2015.07.008>.

References

- [1] Adan A, Huber D. 3d reconstruction of interior wall surfaces under occlusion and clutter. In: Proceedings of international conference on 3D imaging, modeling, processing, visualization and transmission (3DIMPVT); 2011. p. 275–81.
- [2] Boykov Y, Kolmogorov V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans Pattern Anal Mach Intell* 2004;26(9):1124–37.
- [3] Boykov Y, Veksler O, Zabih R. Fast approximate energy minimization via graph cuts. *IEEE Trans Pattern Anal Mach Intell* 2001;23(11):1222–39.
- [4] Budroni A, Boehm J. Automated 3d reconstruction of interiors from point clouds. *Int J Archit Comput*; 2010. <http://dx.doi.org/10.1260/1478-0771.8.1.55>.
- [5] Chang CC, Lin CJ. LIBSVM: A library for support vector machines. *ACM Trans Intell Syst Technol* 2011;2:27(27):1–27 Software available at: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [6] Kolmogorov V, Zabin R. What energy functions can be minimized via graph cuts? *IEEE Trans Pattern Anal Mach Intell* 2004;26(2):147–59.
- [7] Monzpart A, Mellado N, Brostow G, Mitra N. RAP ter: rebuilding man-made scenes with regular arrangements of planes. In: ACM SIGGRAPH 2015; 2015.
- [8] Mura C, Jaspe Villanueva A, Mattausch O, Gobbetti E, Pajarola R. Reconstructing complex indoor environments with arbitrary wall orientations. In: Proceedings of EG posters; 2014.
- [9] Mura C, Mattausch O, Jaspe Villanueva A, Gobbetti E, Pajarola R. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. *Comput Graph* 2014;44.
- [10] Ochmann S, Vock R, Wessel R, Tamke M, Klein R. Automatic generation of structural building descriptions from 3D point cloud scans. In: GRAPP; 2014.

- [11] Oesau S, Lafarge F, Alliez P. Indoor scene reconstruction using primitive-driven space partitioning and graph-cut. In: Eurographics workshop on urban data modelling and visualisation; 2013.
- [12] Oesau S, Lafarge F, Alliez P. Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. *ISPRS J Photogramm Remote Sens* 2014;90 (April (68–82)).
- [13] Okorn B, Xiong X, Akinci B, Huber D. Toward automated modeling of floor plans. In: Proceedings of the symposium on 3D data processing, visualization and transmission, vol. 2; 2010.
- [14] Rusu RB, Cousins S. 3D is here: point cloud library (PCL). In: IEEE international conference on robotics and automation (ICRA), Shanghai, China; May 9–13, 2011.
- [15] Sanchez V, Zakhor A. Planar 3d modeling of building interiors from point cloud data. In: 2012 19th IEEE international conference on image processing (ICIP); 2012. p 1777–80.
- [16] Schnabel R, Wahl R, Klein R. Efficient RANSAC for point-cloud shape detection. In: Computer graphics forum, vol. 26. Wiley Online Library; 2007. p. 214–226.
- [17] Tamke M, Blümel I, Ochmann S, Vock R, Wessel R. From point clouds to definitions of architectural space. In: eCAADe 2014; September 2014.
- [18] Turner E, Cheng P, Zakhor A. Fast automated scalable generation of textured 3d models of indoor environments. *IEEE J Sel Top Signal Process* 2015;9(April (3)):409–21.
- [19] Turner E, Zakhor A. Floor plan generation and room labeling of indoor environments from laser range data. In: GRAPP; 2014.
- [20] Xiao J, Furukawa Y. Reconstructing the world's museums. In: Computer vision, ECCV 2012. Springer; 2012. p. 668–81.
- [21] Xiong X, Adan A, Akinci B, Huber D. Automatic creation of semantically rich 3d building models from laser scanner data. *Autom Constr* 2013;31:325–37.