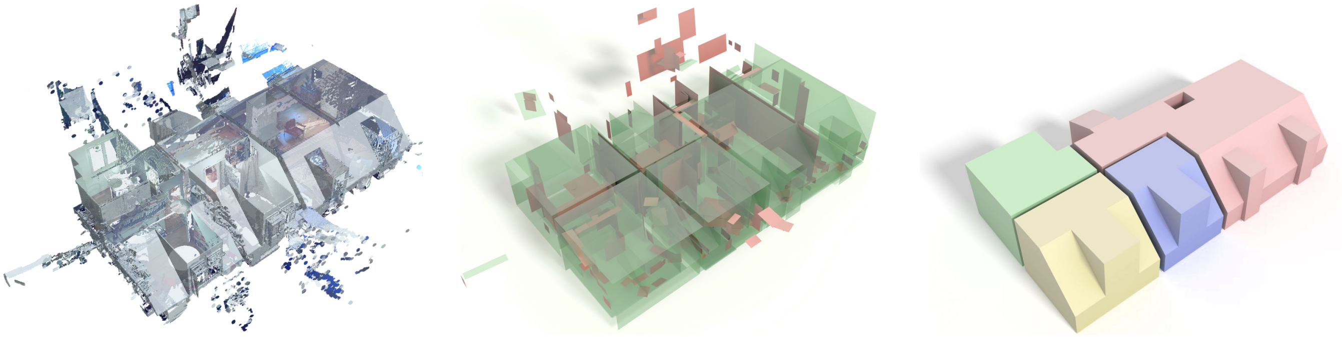
Piecewise-planar Reconstruction of Multi-room Interiors with Arbitrary Wall Arrangements

C. Mura[1], O. Mattausch[1,2] and R. Pajarola[1]

[1]Visualization and MultiMedia Lab, Department of Informatics, University of Zürich   [2]Computer Vision Lab, ETH Zürich



Input point clouds   Detection of permanent components   Volumetric reconstruction of rooms

Figure 1: *Given a set of cluttered 3D point clouds of building interiors (LEFT), we use fitting rectangles of the planar components as simplified scene description, and separate those belonging to permanent structures (green) from clutter (red) (MIDDLE). From the dominant planes of the permanent components we build a 3D cell complex, whose cells are partitioned to create individual room polyhedra (RIGHT).*

Abstract

Reconstructing the as-built architectural shape of building interiors has emerged in recent years as an important and challenging research problem. An effective approach must be able to faithfully capture the architectural structures and separate permanent components from clutter (e.g. furniture), while at the same time dealing with defects in the input data. For many applications, higher-level information on the environment is also required, in particular the shape of individual rooms. To solve this ill-posed problem, state-of-the-art methods assume constrained input environments with a 2.5D or, more restrictively, a Manhattan-world structure, which significantly restricts their applicability in real-world settings. We present a novel pipeline that allows to reconstruct general 3D interior architectures, significantly increasing the range of real-world architectures that can be reconstructed and labeled by any interior reconstruction method to date. Our method finds candidate permanent components by reasoning on a graph-based scene representation, then uses them to build a 3D linear cell complex that is partitioned into separate rooms through a multi-label energy minimization formulation. We demonstrate the effectiveness of our method by applying it to a variety of real-world and synthetic datasets and by comparing it to more specialized state-of-the-art approaches.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Boundary representationsCurve, surface, solid, and object representations

1. Introduction

In the fields of architecture, construction, and civil engineering there is an increasing need for computer-assisted pipelines to extract accurate and semantically rich 3D models of buildings from raw measured data. Such models are meant to represent the "as-built" condition of structures (as opposed to the way they were designed) and are therefore largely focused on the permanent components of a building. Better known as *as-built Building Information Models* (BIMs) [THA*10], they can be used to evaluate how faithfully structures were actually built with respect to their original design, to plan renovations and modifications to the layout, and also for advanced tasks such as energy performance analysis. These tasks require not only an accurate model of the permanent elements of the building, but also higher-level information about the environment – first and foremost the subdivision into different rooms.

However, the development of such pipelines is made difficult by a number of factors. Realistic building interiors are typically cluttered with furniture and other objects. Not only is it necessary to separate these elements from the structural shape of a building, but they also generate viewpoint occlusions resulting in missed sam-

To appear at *Pacific Graphics (2016)* and in *Computer Graphics Forum*

pling of the permanent structures. Moreover, under these conditions traditional criteria for detecting boundaries between different rooms are error-prone and unreliable. In spite of recent research efforts, a satisfactory solution to this ill-posed problem has not been proposed yet. Specialized indoor reconstruction pipelines typically try to reduce the complexity of the problem by assuming that walls are vertical [OLA14,CF14,TZ14,MMJV*14,OVWK16] or by relying on the Manhattan-World assumption [TZ12,IYF15]. While such simple assumptions increase robustness, many architectural elements of real-world buildings deviate from vertical walls with perpendicular arrangements. More recent methods [TZ14,MMJV*14,IYF15], which are otherwise capable of detecting individual rooms in the environments processed, still suffer from a 2.5D restriction. On the other hand, methods for general surface reconstruction [BTS*14] can represent arbitrary geometry, but do not distinguish between permanent components and clutter and generally assume occlusions-free input data.

This paper introduces a novel pipeline for reconstructing the architectural shape of multi-room/story building interiors that is capable of handling a significantly larger class of real-world environments. As a sufficiently expressive model to capture the majority of real-world interiors, our method lifts the restrictive Manhattan-World and 2.5D assumptions and uses piecewise-linear structures and structural soundness as the only priors on the environment. To our knowledge, this is the first method that can reconstruct prevalent 3D structures such as slanted walls and sloped ceilings *robustly* and in a *unified manner* from cluttered indoor datasets.

Our pipeline considers as input data a set of 3D point clouds, aligned to a common reference space and enriched with basic information like normal, splat size, viewpoint and segmentation into co-planar patches [MMJV*14], and produces as output a set of watertight polyhedra representing the boundaries of the rooms in the environment. The reconstruction is driven by the main architectural structures, while fine-grained details (e.g. small cavities and protrusions in the walls) are not in the focus of this work and do not appear in the final model. The processing is split into the following two main stages (see also Fig. 1).

Detection of permanent components. The input point clouds are converted into a graph-based representation that encodes the planar components of the scene and their adjacency relations. The permanent components (i.e., the static architecture) are then separated from clutter (non-structural components like furniture) by reasoning on the structural relations between adjacent components based on six rules. In a cluttered environment, this is essential for making the further processing steps feasible.

Volumetric reconstruction of rooms. The planes of the permanent components are used to build a 3D cell complex that partitions the scene into polyhedral cells. A visibility-based clustering is then applied to find the approximate location of the individual rooms, which are finally reconstructed by applying a multi-label optimization to the set of polyhedral cells.

To remove the ambiguities that can naturally arise from such a general model, the first step can be complemented by an optional interactive step, in which the user is allowed to correct the results generated automatically through simple sketches. We further introduce non-trivial extension of consolidated techniques for indoor reconstruction that make a fully-3D pipeline feasible. These include: an approach inspired by structural analysis to extracting the permanent components of an indoor scene from its adjacency graph; an automatic approach based on the Markov Cluster Algorithm [VD08] driven by visibility information to detect the number and an initial location estimate of the rooms of an indoor environment; a multi-label energy minimization formulation for the room reconstruction problem that enforces clearly separated and regular room shapes in complex multi-room environments.

2. Related Work

The topic of indoor reconstruction is closely related to the more well-studied problem of recovering the outer shape of urban structures, though the issues to be addressed are different. In fact, geometric regularity and occlusion-free data are common assumptions when reconstructing outdoor structures, both at a city-scale [VLA15] and at the level of individual buildings. In this context, the very restrictive Manhattan-world assumption (also widely used for interiors) has been used to reconstruct buildings as unions of box-like structures [VAB12]. Some approaches are able to capture less regular geometries for more generic piecewise-planar scene reconstruction by using *binary space partitioning* (BSP) [CLP10,BdLGM14] or tetrahedral space partitionings [LA13]. Such methods rely on the intersection parity of visibility rays to perform the inside/outside segmentation, which is bound to fail in cluttered and heavily occluded settings such as real-world indoor environments. Furthermore, none of these methods performs a separation between structural components and clutter nor deals with the detection of rooms.

In recent years several methods specialized on interiors have been proposed. For virtual reality and interactive exploration, panoramic images can be used as input sources, often in conjunction with 3D data from multi-view stereo, to produce geometrically simple yet textured models. In this context, various starting assumptions are used, ranging from the Manhattan-world prior [CF14] to the simple verticality of wall structures [PGG*16]; however, the final models are normally obtained by vertical extrusion of a 2D floorplan of the environment with respect to the ceiling height. Even in domains in which accuracy is vital (e.g. architecture and civil engineering) and dense 3D measurements are used as input, the top-view of the building is often the primary goal of the reconstruction. Some of these methods are restricted to piecewise-linear floorplans [BB10, MMJV*14], while others also capture rounded walls [TZ12]. In contrast, our pipeline allows for arbitrary orientations of the permanent structures, by using a generic 3D BSP scene representation. With some notable exceptions [AH11,MMJV*14], many of these methods disregard the problem of occlusions and missing regions, or use only a simple 2D neighborhoods analysis to distinguish between clutter and permanent structures [OLA14], which would not work in full 3D environments. Hence we impose a graph-based scene representation and use a more sophisticated structural analysis to find a reliable classification into permanent structures and clutter.

In more recent approaches, the attention has shifted towards the extraction of semantic properties, such as a segmentation into individual rooms [TZ14, MMJV*14], in some cases together with

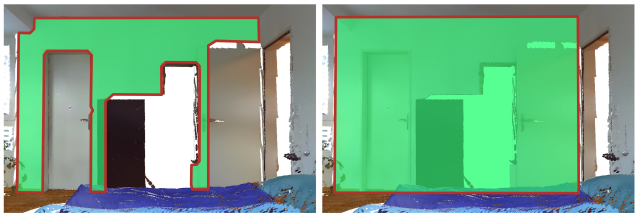

(a) (b)

Figure 2: *For cluttered environments, alpha-shapes adhere to region boundaries generated by occlusions (a). Oriented rectangles represent a more reliable proxy for the shapes of interest (b).*

their mutual arrangements [IYF15, ASZ*16, OVWK16], but are again restricted to 2.5D environments. Aiming at a multi-room reconstruction poses the additional problem of estimating the number of rooms, where previous work relies on iterative heuristics [MMJV*14] or subsequent merging steps [OVWK16] in a 2.5D setting. Instead, to get a proper first estimate for the room shapes in a general 3D environment, we employ a visibility clustering method with respect to given view points [DBGBR*14] and adapt it to the context of indoor reconstruction. For the final room labeling, we introduce novel energy terms that are intuitive and significantly improve the room segmentation results.

3. Detection of Permanent Components

The input to the first step of our pipeline is a set of 3D point clouds acquired at known positions of an indoor environment (from at least one location in every room) and registered in a single reference frame. The input point clouds are enriched with per-point normals and splat-sizes and with a segmentation into patches of co-planar points, as described in related work [LA13,MMJV*14,OVWK16]. To reduce the complexity of subsequent steps in the pipeline, in particular the construction of a space partitioning, it is important to separate the components that belong to the permanent architecture from clutter in a meaningful way that incorporates information about structural soundness. The input point clouds are not well-suited for reasoning on the structure of the environment and only convey information about local geometric features. Hence they are converted into a higher-level graph structure that encodes the planar components of the scene, along with their adjacency relations.

3.1. Building the Adjacency Graph

We convert the input point-based models into a more compact and semantically richer *adjacency graph* $G_{\text{adj}} = (V,E)$. In this structure, used in many shape analysis approaches [LMS13], the set of vertices V encodes the individual parts of the model considered, while E represents the adjacency relationships between such parts, i.e. is composed of all the pairs (v_1, v_2) for which the corresponding parts $v_1, v_2 \in V$ are adjacent (with respect to a threshold θ_{adj}). Our construction follows the approach described by Mattausch et. al [MPM*14], who use consistently oriented *fitting rectangles* as shape proxies for the parts of the scene. The use of these primitives in our pipeline is motivated by two main arguments. First, oriented fitting rectangles have proven to work well in practice in indoor settings and on more general urban models [vLvKV11]. Second, in the presence of large missing regions, more expressive representations (e.g. *alpha shapes* [EM94]) do not ensure a better approximation of the actual features (see Fig. 2).

3.2. Detection of Structural Paths

To apply reasoning to the adjacency graph, we use the observation that an indoor environment is composed of three main structural elements, namely *ceilings*, *walls* and *floor*, which are arranged from top to bottom in a consistent fashion, i.e. ceilings (on top) unload their weight onto the floor (bottom), typically transitioning through walls. Furthermore we introduce the concept of *structural paths* in the adjacency graph G_{adj}. A structural path is a sequence $W_S = (v_1, v_2, \ldots, v_{n-1}, v_n)$, where $v_i \in G_{adj}$, v_1 corresponds to a rectangle in the ceiling, v_n to a rectangle on the floor and every edge (v_i, v_{i+1}) is an edge in G_{adj} that is *structurally valid*. We consider an edge (v_i, v_{i+1}) to be *structurally valid* if the two fitting rectangles r_i and r_{i+1} corresponding to the nodes v_i and v_{i+1} express a transition that is coherent with the top-bottom arrangement described above. Valid transitions are encoded using a set of six spatial configurations, denoted as *structural patterns* and depicted in Fig. 3. Patterns **1**-**4** capture the transition from the ceiling downwards to the floor (patterns **TopDown1** and **TopDown2**), including the special case of large alcoves that jut out of the main room structure (**Alcove1** and **Alcove2**). Patterns **5**-**6** (**Side1** and **Side2**) encode *lateral adjacencies* between walls. In particular, the mutual rectangle positions for these patterns can be described as follows.

1. **TopDown1:** at least one of r_i and r_{i+1} is not vertical and there exists a pair of edges s'_i, s'_{i+1} that are spatially close (i.e. their minimum distance is $< \theta_{\text{adj}}/2$) and parallel to a same line; additionally, the projections of s'_i and s'_{i+1} onto said line overlap;
2. **TopDown2:** r_i rests on r_{i+1}, i.e., the top edge of r_{i+1} lies on the plane of r_i and its projection onto such plane is contained in r_i;
3. **Alcove1:** the top edge of r_{i+1} (any if r_{i+1} is horizontal) lies on r_i and the other edge lies on the positive half space of Π_i;
4. **Alcove2:** the bottom edge and either the left or the right edge of r_{i+1} intersect r_i;
5. **Side1:** r_i and r_{i+1} are both non-horizontal and have the same vertical slant (i.e., are either both vertical or are parallel); additionally, the left edge of r_i and the right edge of r_{i+1} (or vice-versa, the left edge of r_{i+1} and the right edge of r_i) are adjacent in the sense of **TopDown1**;
6. **Side2:** the intersection of the planes of r_i and r_{i+1} crosses the left edge of r_i and the right edge of r_{i+1} (or vice-versa, the left edge of r_{i+1} and the right edge of r_i).

To find structural paths in the contact graph, we apply the following base algorithm. For every ceiling node v_{ceil} (corresponding to a rectangle on a ceiling) we perform a region growing in G_{adj}, using v_{ceil} as starting node and only expanding along edges that are structurally valid in the sense of patterns **1**-**4**. If one or more ground patches are reached, we backtrack from each of them until v_{ceil} to extract the structural paths found. Fig.4 shows a typical structural path that reaches the floor from a ceiling node, crossing edges that conform to **TopDown1**. In the same picture, the fixtures on the ceiling are not connected to the floor and are thus marked as clutter. Similarly, the cabinet represents an orthogonal extrusion from the

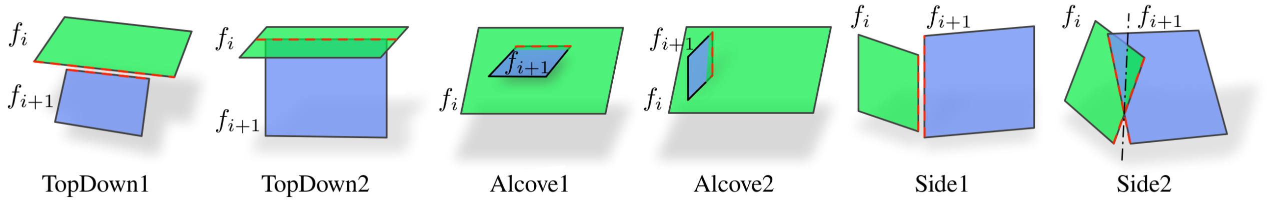


Figure 3: *The six structural patterns for pairs of neighboring fitting rectangles that are used in our structural region growing algorithm.*

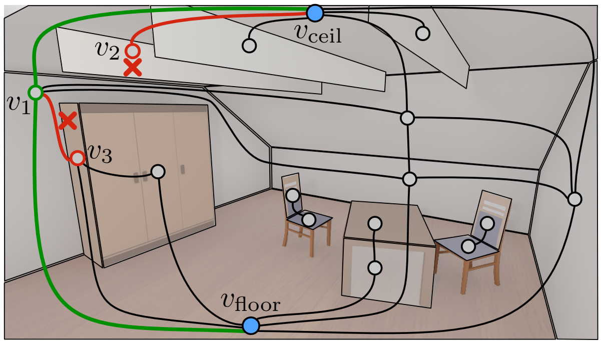


Figure 4: *Node v_1 belongs to a structural path ($(v_{\text{ceil}}, v_1, v_{\text{floor}})$, in green) in the adjacency graph and is therefore marked as permanent. Nodes v_2 and v_3, reachable from the ceiling but not on a valid path to the floor, are classified as clutter.*

nearby wall and does not represent structural support for the ceiling elements (TopDown1-2), hence it is not part of any structural path.

While this procedure works well when the whole extent of the structures of interest is visible, in practice furniture placed along the line of sight of the acquisition device often projects large shadows onto the scene, especially onto the lower parts of the walls. This means that the nodes in G_{adj} corresponding to such walls may have no edge connecting them to the floor. We therefore perform a second selection step, starting the region growing from the structural patches found in the first pass and considering the lateral adjacencies of **Side1** and **Side2** (see Fig. 5). Note that objects like cupboards and cabinets (see also Fig. 4) are not wrongly added to a structural path by the side relations, as they do not follow a sequence of side-based adjacencies between structural elements (e.g. walls), but rather constitute blocks that protrude out of them.

The result of this algorithm is a set of oriented fitting rectangles $\mathcal{R}_{\text{struct}}$ corresponding to the detected structural components of the environment. As shown in Fig. 6, our approach is highly effective in separating permanent structures from clutter. In many cases (marked by the red circles), the method of Mura et al. [MMJV*14] wrongly classifies clutter and artifacts as permanent components or results in over-segmented primitives. Note that the exact shape of the rooms is recovered in a final step based on energy minimization; hence it is justified to perform a first, non globally-optimal pruning, as it greatly simplifies the problem for subsequent stages while a globally correct final reconstruction can still be recovered. An error analysis with respect to noise and changes in the adjacency graph is given in Fig. 12 and 13, respectively.

Detecting floor/ceiling rectangles. Horizontal floors and ceilings are detected by analyzing the z-histogram of the scene [TZ12, OLA14]; however, since we also target buildings with non-horizontal roofs, we also analyze the rectangles that are non-horizontal. A non-horizontal rectangle is marked as belonging to a ceiling if the vertical projection of all the rectangles above it do not intersect it; here a rectangle r_i is considered to be *above* rectangle r_j if the vertices if r_i lie in the upper half-space defined by the plane of r_j.

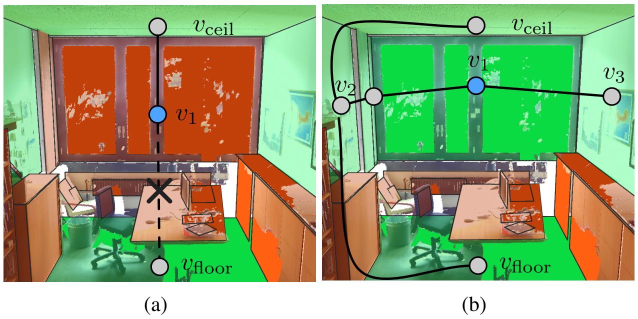


Figure 5: *The patch corresponding to node v_1 is adjacent to a ceiling patch, but, due to viewpoint occlusions, it is not connected to the floor (a). However, using structural pattern **Side1**, v_1 can correctly be classified as structural, as it is connected by a lateral adjacency path between v_2 and v_3 that both belong to a structural path (b).*

Interactive refinement. In case that errors in the classification (e.g., due to ambiguous configurations that stem from missing data) lead to unwanted final results, we provide an option to use interactive operations to correct them and to enforce specific completions of occluded regions. Unlike full-fledged semi-automatic modeling pipelines [ASF*13], the number of primitives affected by manual refinements is assumed to be negligible compared to the model complexity. For instance, only 21 rectangles are altered in the reconstruction of ‘Building D’ (Tab. 1), representing only 4% of the total number of rectangles discovered in the model. Throughout this process, all fitting rectangles are shown in color-coding according to their current label (permanent structures are visualized in green, clutter is shown in red). The user can intervene using the following three operations (shown in Fig. 7).

(a) Label flip. Selecting a rectangle and inverting its label. A rectangle marked as permanent becomes clutter and vice-versa.

(b) Extension. Sketching a line that starts from one permanent rectangle and ends on another green rectangle. The two rectangles are extended to the intersection of their planes.

(c) Orthogonal extension. Sketching a line that starts from one permanent rectangle r_i and ends on another green rectangle r_{i+1}, crossing a boundary line segment s_i' of r_i. A new rectangle orthogonal to the plane of r_i and extending from s_i' until the plane of r_{i+1}

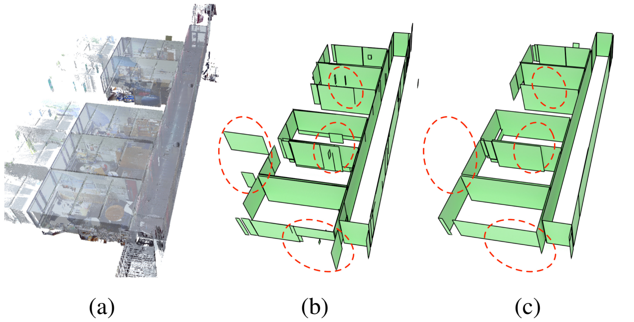


Figure 6: *Selection of permanent components by a state-of-the art method [MMJV*14] (b) and our new approach based on structural paths in 3D (c), which produces comparable or better results on a 2.5D dataset (a) targeted to the former approach. The result shown in (c) was obtained without any user refinement.*

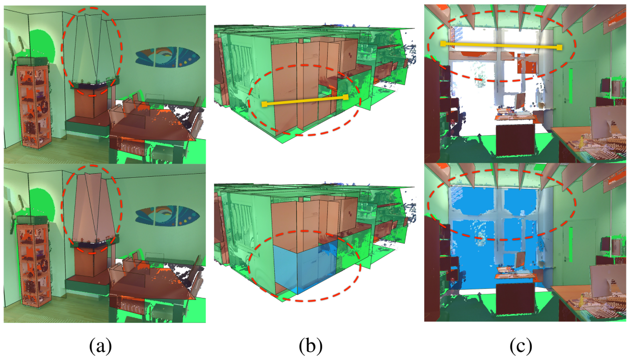


Figure 7: *Examples of interactive refinement of the automatic results: label flip (a); extension (b); orthogonal extension (c).*

is added as a new permanent component, while r_{i+1} is extended until reaching such extension.

4. Volumetric Segmentation

In this second stage, we use the detected structures to build a space partitioning of the scene and then reconstruct the individual rooms as unions of volumetric sub-regions of this structure. We can identify three main steps in the volumetric segmentation process: (i) construction of the space partitioning, (ii) detection of the individual rooms, and (iii) reconstruction of the detected rooms.

4.1. Space Partitioning

As a preprocess, the dominant planes of the building are detected by clustering the permanent rectangles according to their normal direction and to their position along that direction, using the approach of Mura et al. [MMJV*14]. For each cluster obtained, we perform an additional *least-median of squares* (LMS) fit on the points associated to its rectangles to avoid averaging effects due to the clustering procedure. A BSP tree of the scene is now built by intersecting its bounding box (expanded by a small factor) with all dominant planes. The set of polyhedral cells $\mathcal{C}$ associated to the leaves of this tree corresponds to the partitioning of the space induced by the dominant planes. We assemble such cells according to their adjacency relationships $\mathcal{N} = \{(c,c')|c,c' \in \mathcal{C} \wedge c,c' \text{are adjacent}\}$ into a *3D cell complex* $(\mathcal{C},\mathcal{N})$. An adjacency $(c,c') \in \mathcal{N}$ corresponds to a polygonal facet $f_{c,c'}$ shared by the polyhedra corresponding to c and c'. During the construction of the complex we associate to each such facet the ID of the dominant plane that generates it and the *coverage* of the facet (denoted by $cov(f_{c,c'})$), which corresponds to the fraction of its area covered by scanned points. It is worth mentioning that the coverage of a facet can be used as a measure of *dissimilarity* between its two adjacent cells, since a facet with high coverage is likely to lie on the surface of a permanent structure (e.g. a wall) and thus separate cells that belong to different environments.

4.2. Room Detection

In our pipeline, the shape of each room is obtained as the volumetric union of a specific subset of the cells of the complex $(\mathcal{C},\mathcal{N})$. In fact, the reconstruction is cast as a *multi-label optimization* problem [BK04], in which each cell is assigned one out of $N_{\text{labels}} = N_{\text{rooms}} + 1$ labels, that is, N_{rooms} labels for the rooms, plus an additional one for the outer space and the space occupied by walls. To reliably estimate N_{rooms}, we adopted the work of DiBenedetto et al. [DBGBR*14]. In their method, aimed at generating panoramas for image-based rendering, multiple view probes are clustered using a *Markov cluster algorithm* (MCL) [VD08] driven by the amount of visible surface overlap. We adapt their approach by considering as view probes the polyhedral cells $\mathcal{C}^{vp} = \{c_1^{vp}, \ldots, c_{n_{vp}}^{vp}\}$ of the complex that contain a scan position. As stated in Sec. 3, we assume that every room contains at least one scan position, which implies that the number of rooms in the environment can be obtained by correctly clustering the set $\mathcal{C}^{vp}$.

In the absence of an exact mathematical definition of a room [TZ14], we define it as a sub-space of an environment mostly separated from the rest of the space by permanent components. From this it follows that two locations placed in different rooms should see very different parts of the scene, while the visibility from positions within the same room is very similar. We define the amount of *visible surface overlap* between two cells in terms of the visible structural fitting rectangles $\mathcal{R}_{\text{struct}}$ (see Sec. 3.2). In particular, let vis_i^r denote the visible area of $r \in \mathcal{R}_{\text{struct}}$ as seen from cell c_i and let $overlap^r(c_i,c_j)$ be the visibility overlap between cells c_i and c_j relative to rectangle r, defined as:

$$overlap^r(c_i,c_j) = \frac{\max(vis_i^r, vis_j^r) - \min(vis_i^r, vis_j^r)}{\max(vis_i^r, vis_j^r)} \quad (1)$$

We then define the visibility overlap $overlap(c_i,c_j)$ between the viewpoint cells c_i and c_j as follows:

$$overlap(c_i,c_j) = \frac{\sum_{r \in \mathcal{R}_{\text{struct}}} w_{i,j}^r \cdot overlap^r(c_i,c_j)}{\sum_{r \in \mathcal{R}_{\text{struct}}} w_{i,j}^r} \quad (2)$$

Here $w_{i,j}^r$ is a term that balances the importance of each rectangle taking into account all rectangles as seen from every viewpoint cell:

$$w_{i,j}^r = \frac{\max(vis_i^r, vis_j^r)}{\max_{r \in \mathcal{R}_{\text{struct}}, i=1\ldots n_{vp}} vis_i^r} \quad (3)$$

Given this definition of visibility overlap, we construct a weighted graph G_{vp} that contains one node n_i for every viewpoint cell c_i^{vp}.

This graph is undirected and complete; each edge (n_i, n_j) is assigned the weight $g(overlap(c_i^{vp}, c_j^{vp}))$, where $g(x)$ is a monotonically increasing concave function that suppresses the contribution of low values. We defined it as $g(x) = 1 - e^{-((1-x)/0.5)^2}$, as this function proved to work well in all our test cases.

Applying the Markov clustering [VD08] to G_{vp} yields a set of clusters $\Gamma = \{\Gamma_1, \dots, \Gamma_{N_{\text{rooms}}}\}$, grouping the viewpoint cells according to their visibility overlap. The algorithm automatically selects the number of room clusters, which, thanks to the similarity measure chosen, corresponds to the number of rooms N_{rooms} of the environment. It is worth stressing that this clustering step is only applied to the cells that contain a viewpoint (the viewpoint cells). While this process could be applied to all cells in $\mathcal{C}$ to obtain the shapes of the rooms, the resulting reconstruction would only be based on visibility information and lack geometric and structural regularity. For this reason, we extract the final room models using a more expressive multi-label energy minimization approach.

4.3. Room Reconstruction

The previous steps of the pipeline yield a polyhedral cell complex $(\mathcal{C}, \mathcal{N})$ and N_{rooms} clusters $\Gamma_1, \dots, \Gamma_{N_{\text{rooms}}}$, each corresponding to a room of the environment and containing the viewpoint cells located inside that room. In this final step, we assign each polyhedral cell of the complex one label from the set $\mathcal{L} = \{l_1, \dots, l_{N_{\text{rooms}}}, l_{\text{out}}\}$, which includes one label for each room plus the additional label l_{out} for the outer space. The problem can be modeled naturally as a multi-label *Markov random field* (MRF) [BVZ01]; in particular, we seek the optimal label assignment $L^* = \{\, L^*_c \mid L^*_c \in \mathcal{L}\,,\, c \in \mathcal{C} \,\}$ that minimizes an energy function of this kind:

$$E_{\text{label}}(L) = E_{\text{data}}(L) + E_{\text{smooth}}(L)\ . \tag{4}$$

These two terms (denoted in the literature as *data* and *smoothness* terms) correspond to the energy associated, respectively, to an initial, coarse labeling of the cells and to the coherency of the label assignments to pairs of adjacent cells.

Data term. E_{data} consists of a sum of unary functions, each representing a penalty for assigning label $L_c \in \mathcal{L}$ to a cell $c \in \mathcal{C}$:

$$E_{\text{data}}(L) = \sum_{c \in \mathcal{C}} D_c(L_c) \tag{5}$$

To derive the data terms, we treat the viewpoint cells $c' \in \Gamma_i$ of a room cluster Γ_i as representatives of the visibility for that room; we then compute the penalty for assigning a label l_i to any cell c in terms of the visibility overlap with the viewpoint cells in Γ_i (using Eq. 2). Since the notion of viewpoint cell does not apply to the cluster of outer space, the penalty for labeling a cell c as l_{out} is defined as the fraction $O(c)$ of the scene (as seen from the center of c) that corresponds to empty space (i.e., not occupied by any structural rectangle). The term $D_c(L_c)$ can therefore be expressed as follows:

$$D_c(L_c) = \begin{cases} 1 - \max_{c' \in \Gamma_{L_c}} overlap(c, c') & \text{if } l_c \neq l_{\text{out}} \\ O(c) & \text{otherwise} \end{cases} \tag{6}$$

Smoothness term. The effect of E_{data} is balanced by the smoothness energy E_{smooth}, which aims at regularizing the labeling by penalizing the assignment of different labels to adjacent cells. We express this term as a sum of four sub-terms, which enforce not only fidelity to the measured data, but also geometric simplicity of the resulting model, structural coherency of the rooms and separation between rooms:

$$E_{\text{smooth}}(L) = \lambda_{cov}\, E_{cov} + \lambda_A\, E_A + \lambda_G\, E_G + \lambda_{sep}\, E_{sep} \tag{7}$$

Here the dependency of the energy sub-terms on the labeling L has been omitted for brevity. Each of these energies are defined by a sum of pairwise potentials involving the pairs $(c, c') \in \mathcal{N}$ of adjacent cells in the complex, and have the following general form:

$$E_{\dots}(L) = \sum_{(c,c') \in \mathcal{N}} V^{\dots}_{c,c'}(L_c, L_{c'}) \tag{8}$$

The individual terms are defined as follows:

- Coverage term E_{cov}: penalizes assigning the same label to a pair (c, c') if the facet $f_{c,c'}$ is densely covered by scanned points; its potential is defined as follows:

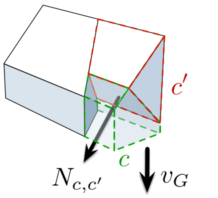
$$V^{cov}_{c,c'}(L_c, L_{c'}) = \mathbf{1}(L_c \neq L_{c'}) \cdot cov(f_{c,c'}) \tag{9}$$

  where $cov(f_{c,c'})$ is the coverage defined in Sec. 4.1;
- Area term E_A: favors geometric simplicity by penalizing the total area of the interface surface between two rooms; its interaction potential is defined as:

$$V^{A}_{c,c'}(L_c, L_{c'}) = \mathbf{1}(L_c \neq L_{c'}) \cdot area(f_{c,c'}) \tag{10}$$

  where $area(f_{c,c'})$ is the surface area of the facet.
- Gravity term E_G: penalizes label assignments to a pair (c, c') that leave one cell "floating", i.e. without a cell below to sustain its weight (see Fig. 8); this condition is formulated with the following potential:

$$V^{G}_{c,c'}(L_c, L_{c'}) = \mathbf{1}(L_c \neq L_{c'}) \cdot (v_G \cdot N_{c,c'}) \cdot s(c, c', L_c, L_{c'}) \tag{11}$$



  where v_G is the vector $(0,0,-1)$, $N_{c,c'}$ is the normal of $f_{c,c'}$ and $s(c, c', L_c, L_{c'})$ is a function that takes value 1 if the bottom cell in the pair (c, c') is labeled l_{out} and the upper cell has a label $\neq l_{\text{out}}$, and 0 otherwise (see inset, which describes the case $s(c, c', L_c, L_{c'}) = 1$);
- Room separation term E_{sep}: enforces the presence of thick walls between the clusters of any two rooms, by penalizing the assignment of different labels to adjacent cells unless one of the two labels is L_{outer}; the corresponding potential is as follows:

$$V^{sep}_{c,c'}(L_c, L_{c'}) = \begin{cases} 1 & \text{if } L_c = L_{\text{outer}} \vee L_{c'} = L_{\text{outer}} \\ 0 & otherwise \end{cases} \tag{12}$$

The relative importance of these sub-terms (also with respect to the data term) can be adjusted by varying the values of the λ in Eq. 7 in the range $[0 \dots 1]$. Since all the pairwise potentials used satisfy by definition the submodularity property and are *semi-metrics*, the energy function of Eq. 4 can be minimized using the $\alpha - \beta$ *swap*

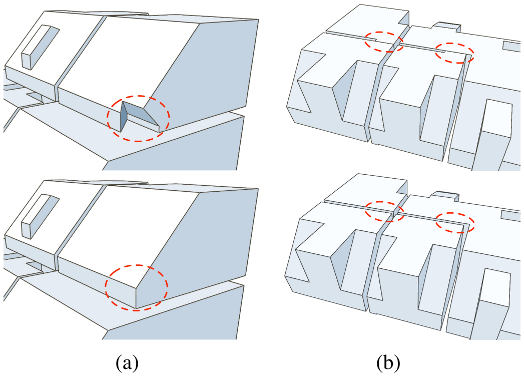


Figure 8: *Our gravity term helps obtain plausible reconstructions by favoring stable cells configurations (a), while our room separation term ensures that rooms are separated by empty space (b).*

algorithm [BVZ01]. The minimum energy is associated to the optimal labeling L^* and the shape of the individual rooms can be reconstructed by volumetric union of the cells with the same label.

5. Results

We tested our pipeline on a set of 6 real-world and 2 synthetic datasets of building interiors. Relevant information on all datasets and on the reconstruction results are summarized in Tab. 1. Our test suite includes several models from state-of-the-art methods [MMJV*14, IYF15, OVWK16]; however, to evaluate the full capabilities of our method, we introduced 5 new building models that clearly violate the 2.5D and Manhattan-World assumptions. Our models come from three different sources: high-quality terrestrial laser range-scanning (label LiDAR in Tab. 1); a proprietary acquisition system based on Kinect-like cameras (label RGB-D); software simulation of time-of-flight scanning on hand-modeled geometry using software ray-casting (label Synth.).

Implementation. Our software is written in C++ and uses the publicly available implementations of the MCL [VD08] and $\alpha-\beta$ *swap* [BK04] algorithms. All tests were performed on a MacBook Pro with an Intel Core i7 (2.5GHz), 16GB DDR3 RAM and an NVIDIA GeForce GT 750M. Processing times for all models are given in Tab. 1 and vary from about 20 seconds (for dataset 'Cottage') to about 5 minutes (for 'House'). The input planar segmentation is obtained using a region-growing algorithm driven by point-to-plane offset η_{off} and deviation η_{ang} from plane normal [CLP10, MMJV*14, BdLGM14]. Both the coverage of a facet and the visible area of a rectangle are computed using the standard rasterization pipeline. Our pipeline depends on the following parameters. During the construction of the adjacency graph, we consider two patches to be adjacent if their distance θ_{adj} is less than 20cm. The MCL clustering is used with inflation 1.1 and thresholding the links with weight < 0.1. The energy minimization depends on the weights λ_{cov}, λ_A, λ_G, λ_{sep} (see Eq. 7), that we fixed to 0.2, 0.05, 0.1 and 0.2, respectively. Finally, we set the default values of η_{off} and η_{ang} (used in the planar segmentation) to 0.5cm and 0.975, respectively, and pruned the fitting rectangles with a diameter < 20cm and a ratio between smaller and greater side < 0.001.

Comparison with state-of-the-art. We compared our approach against state-of-the-art methods using their original 2.5D interior datasets. As shown in Fig. 9, our approach, although designed to capture more general 3D structures, is able to process this type of environments with comparable results to the more restrictive original methods. In particular, the reconstruction results for 'Office 2' and 'Apartment 1' show that our approach is able to detect ceilings of different heights without requiring any ad-hoc steps, which is not possible with the method of Mura et al. [MMJV*14]. Compared to ours, the reconstruction of 'Apartment 1' obtained by Ikehata and colleagues [IYF15] is richer in fine-grained geometric detail, whereas our result only includes larger architectural features. Our approach can correctly reconstruct environments with a high room count like 'Building D' (28 rooms). The results of our room detection algorithm are comparable to those of previous approaches, with only minor differences occurring in ambiguous cases such as wide passages between rooms and corridors.

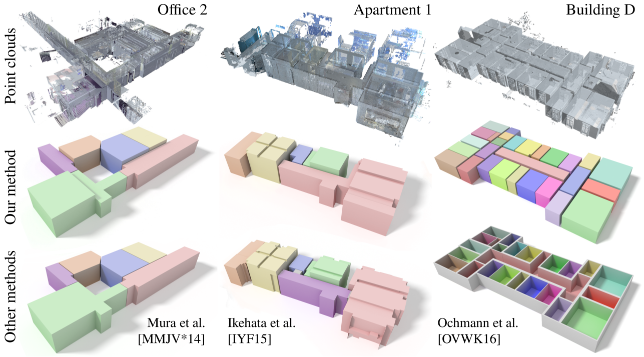


Figure 9: *Comparison between the reconstructions obtained by our method (MIDDLE ROW) and by previous approaches (BOTTOM). For each method, the comparison is based on the input models (TOP ROW) introduced in the corresponding paper.*

In Fig.10, we show a direct comparison with the approach of Mura et al. on a dataset exhibiting several features that can not be described in 2.5D (i.e., the first floor of 'Maisonnette'). While their method can reconstruct an approximate floorplan, many large wall structures, as well as entire sub-spaces (e.g. the alcove in the green room) are lost. On the other hand, our novel method achieves a relatively faithful reconstruction fully automatically, which can be further improved with a few operations of our interactive refinement. Note that the complete dataset with fine details highlighted is shown in Fig. 11. We also run the 2.5D method on building models with relatively high irregularities like 'Modern', but the results produced barely reflected the actual shapes of the rooms due to errors in the selection of the wall candidates.

General 3D datasets. Reconstruction results for a set of more general model are shown in Fig. 11 and in Fig. 1. All environments are composed of multiple rooms and exhibit many slanted ceilings and wall surfaces, which makes them well-suited to evaluate the capabilities of our pipeline. 'Cottage' consists of two hotel rooms (each with a bedroom, a bathroom and a small passage) and a corridor in relatively regular arrangements. 'Penthouse' is a more challenging model, with less regular room shapes, more structural detail (e.g. window alcoves) and high levels of clutter. In spite of this, our method allows to reconstruct an accurate model of its rooms.

Dataset	#Points	#Scans	#Rooms	Type	Time (step1/step2)	#Rect. (perm./clut.)	#Alt.	#Planes	#BSP cells	Figs.
Office 2	27.8M	10	6	LiDAR	79.9s (29.6/50.3)	629 (121/508)	25	53	4958	9
Apartment 1	7.2M	16	5	RGB-D	177.5s (6.3/171.2)	377 (143/234)	27	71	9634	9
Building D	63.1M	36	28	LiDAR	304.0s (99.8/204.2)	501 (252/249)	21	68	8317	9
Penthouse	22.2M	8	5	LiDAR	66.5s (16.0/50.5)	332 (93/239)	17	51	7188	1,8(b)
Maisonnette	19.4M	7	5	LiDAR	112.9s (20.8/92.1)	296 (115/181)	8	63	12202	8(a),10,11,14
Cottage	12.4M	7	7	LiDAR	19.3s (10.2/9.1)	159 (51/108)	6	29	836	11
House	52.8M	19	9	Synth.	317.7s (55.6/262.1)	519 (153/336)	13	66	11784	11
Modern	22.2M	8	3	Synth.	70.6s (22.8/47.8)	266 (57/209)	0	49	5717	11, 12, 13

Table 1: *Information on the input data (no. points, no. scans, data source) and on the environment (no. rooms), overall computation time (in brackets: partial times for step 1/step 2); no. of fitting rectangles extracted automatically (in brackets: structural rectangles/clutter rectangles); no. of rectangles altered in the refinement; no. of dominant planes; size of the BSP complex (i.e., no. of cells).*

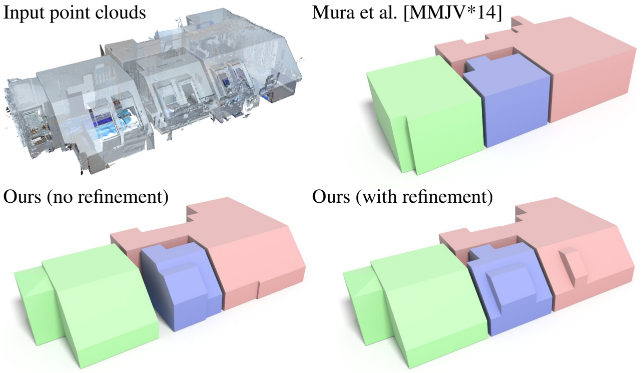


Figure 10: *Comparison between the results of Mura et al. [MMJV*14] and those produced by our method without and with user intervention for a model with several slanted surfaces.*

Datasets 'Maisonnette' and 'House' represent more complex environments with multiple stories. 'House' (synthetic dataset) corresponds to a 3-story house containing many rooms and interior details (as shown in the inset). 'Maisonnette' contains many structures that violate both the Manhattan-World and the 2.5D assumption, it exhibits irregular room shapes and is rich in geometric detail. In particular, the large window alcove in the green room has a roof with several different orientations; as shown in the inset, these are preserved in the final reconstruction. The ability to capture fine-grained orientations is also demonstrated by 'Modern', a synthetic dataset containing many complex wall arrangements and configurations, all captured correctly in our reconstruction.

Interactive refinement. We evaluated the number of operations and the time needed in the interactive refinement to complete a typical model with high levels of clutter, using the first floor of 'Maisonnette' as shown in Fig. 10. In this case, the user intervention helps reconstruct the correct boundaries for the blue room in almost complete absence of scanned evidence for a wall surface. We asked 1 expert and 2 novice users to complete the task. The expert user completed the refinement in 43 seconds performing 6 operations, while the 2 novice users (who received a complete introduction and training to the system) performed, respectively, 12 and 11 operations in 360 and 339 seconds. Note that the unexperienced users produced results comparable to those obtained by the expert and shown in Fig. 10. In addition to this evaluation, we report in Tab. 1 the number of operations performed on each test model.

Rectangles classification. The first floor of 'Maisonnette' was also used to evaluate the detection step (Sec. 3) in terms of precision and recall. The values obtained, respectively 97% and 93%, show that the approach is effective in selecting the main architectural features of the environment. Additionally, we quantitatively compared our classification step with that of Mura et al. [MMJV*14] using 'Office 3', thus complementing the visual comparison in Fig. 6. Our method obtained a precision of 85% and a recall of 77%, whereas their approach scored, respectively, 90% and 48%. Our lower precision value is due to false positives corresponding to spurious ceiling rectangles caused by window reflection artifacts; as these are scattered and isolated by nature, they do not affect the reconstruction.

Robustness. We analyzed the robustness of our approach by corrupting the synthetic model 'Modern' with increasing levels of noise, using the noise model of Mura et al. [MMJV*14]. In our first test, we varied the noise level while keeping all parameters of our pipeline fixed; the input planar decomposition was obtained by adapting the threshold η_{off} of the region growing to the noise level of the input dataset. As shown in Fig. 12, our method is fairly robust for noise levels $\sigma \leq 5$cm. We also evaluate the robustness with respect to the adjacency threshold θ_{adj} used to construct the adjacency graph G_{adj}. We did this for every noise level considered, but for $\sigma = 1cm, 2.5cm$ we did not detect meaningful differences with the results in Fig. 11; we therefore show in Fig. 13 only the results obtained at the highest noise level ($\sigma = 5cm$). Decreasing θ_{adj} from the default value ($20cm$) to $10cm$ leads to an incorrect classification of some structural rectangles and eventually to an erroneous reconstruction of the blue room. For $\theta_{\text{adj}} = 5cm$ even more rectangles are misclassified, resulting in the green room being attached to the central space. It is worth noticing, however, that in this case the adjacency threshold θ_{adj} corresponds to the noise level considered. Besides studying the influence of σ and θ_{adj} on the final reconstruction, we analyzed how they influence the precision and recall values for the detection of the permanent components. For $\sigma = 1cm, 2.5cm$ both measures attain at 100%, whereas for $\sigma = 5cm$ we obtained a precision of 94% and a recall of 88%. The influence of G_{adj} is stronger: for $\theta_{\text{adj}} = 20cm, 10cm, 5cm$, the precision/recall pairs are, respectively, 94%/88%, 96%/85%, 94%/60%.

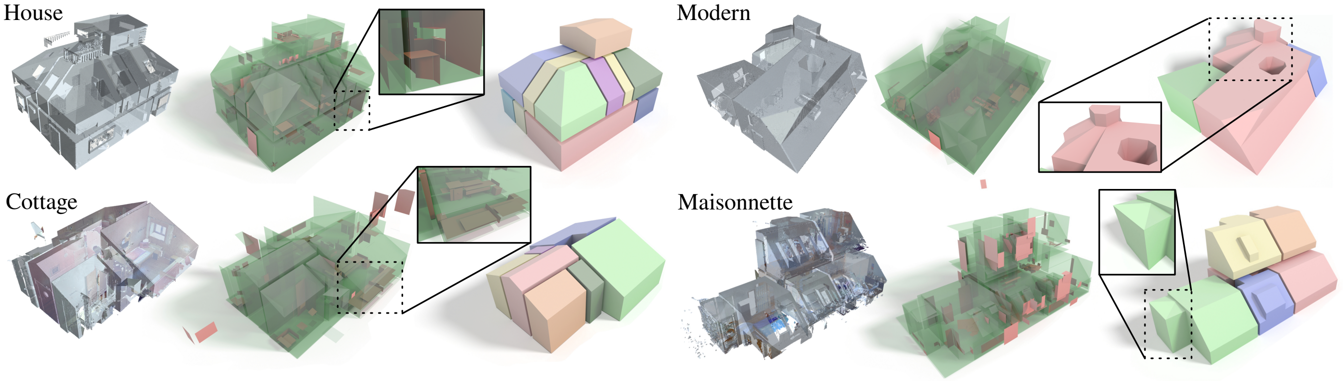


Figure 11: *Input models and reconstruction results. For each dataset, we show the input point cloud (LEFT), the classification into structural (green) and clutter (red) rectangles (MIDDLE) and the reconstructed room polyhedra (RIGHT). Insets show either details of the complexity of the planar components or of the final reconstruction.*

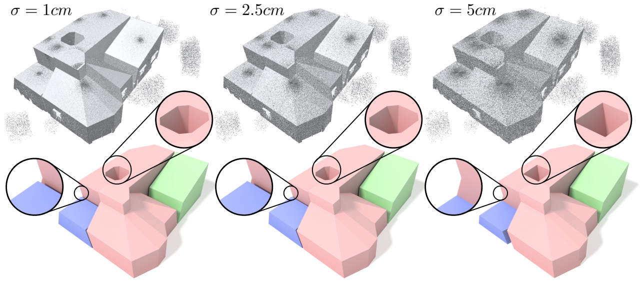


Figure 12: *Robustness with respect to increased measurement noise. The progressive decrease in reconstruction accuracy is shown by the details in the in the insets.*

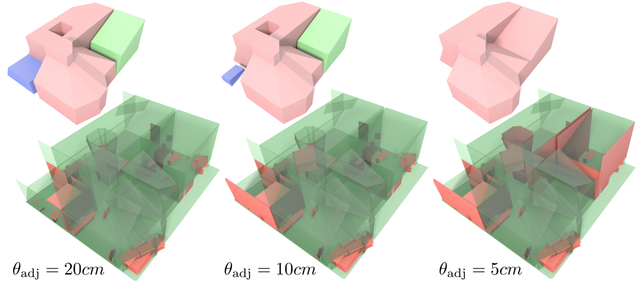

$\theta_{\text{adj}} = 20cm$
$\theta_{\text{adj}} = 10cm$
$\theta_{\text{adj}} = 5cm$

Figure 13: *Reconstruction robustness with respect to varying adjacency threshold in the graph. The model shown is 'Modern', corrupted with gaussian noise of* $\sigma = 5cm$ *(see Fig. 12).*

6. Limitations

Most round surfaces and ceilings can be approximated well in practice by our method; nevertheless, due to the use of fitting rectangles and of a piecewise-planar approach, in presence of occlusions we can not guarantee that non-flat surfaces are recovered in a reliable and uniformly smooth manner (see Fig. 14). Very high levels of clutter, as in the case of dataset 'Penthouse' shown in Fig. 1, can make the reconstruction problem ill-posed. Due to the presence of built-in bookshelves or cabinets, some boundary walls are almost entirely missing in the scanned model (see detail of 'Maisonnette' in Fig. 14) and can only be recovered with user interaction. Due to the generality of building shapes that we would like to model, some reconstruction solutions are intrinsically ambiguous without additional prior information. For instance, the chimney in Fig. 7 could be considered not to be part of the main architectural shape of the environment, but its rectangles are detected as structural by the automatic classification. User input is needed to disambiguate these cases. The simple scheme used for producing the input planar patches will eventually fail for very high levels of measurement noise and could be replaced by dedicated techniques [MMBM15, OLA16], which are however orthogonal to our work. Moreover, the use of the scan positions in the room detection process results in two restrictions. First, a room is detected only if it was acquired from at least one location inside it (as in [MMJV*14, OVWK16]). Second, if two scans corresponding to the same (very large) room do not have enough overlap, the visibility-based clustering may assign them to different clusters, leading to over-segmentation. However, these issues only occur if the input scans do not cover the scene in an adequate way; the problem did not occur in any of our real-world datasets.

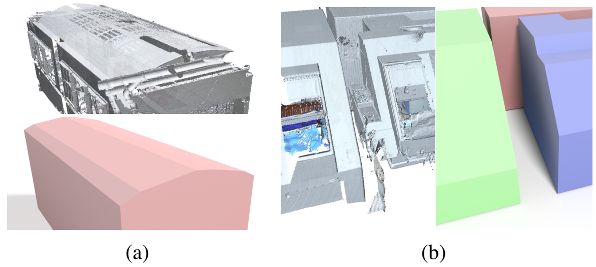


Figure 14: *Limitations of our method. Using a piecewise-planar approach, curved surfaces may not be smoothly approximated (a). If a surface is almost entirely missing in the input model, the automatic reconstruction can be incomplete (b).*

7. Conclusions and Future Work

We have presented the first pipeline for architectural reconstruction of multi-room building interiors that goes beyond the Manhattan-World or 2.5D assumptions. Our approach allows to reconstruct the individual room polyhedra in fairly complex 3D environments with arbitrary wall orientations, which is not possible using existing pipelines. While more flexible in terms of the room shapes that can be captured, our method is still robust to clutter and missing data and is capable of detecting separate rooms. We regard our method to be a significant step forward towards the creation of a flexible and effective pipeline for the *scan-to-BIM* problem. Nevertheless, our work leaves various avenues open for future work. Being a piecewise-linear method, our approach should be extended to reconstruct curved surfaces in a consistent and reliable manner. Although the structural patterns chosen work well in real-world scenes, it would be interesting to explore a data-driven approach to extract a larger set of configurations for the structural components. In this context, the addition of higher-level primitives would be a straightforward extension. Moreover, we would like to extend our work to process efficiently very large-scale multi-story datasets, investigating new techniques for the construction of the cell complex that allow such complexity.

Acknowledgments. The authors would like to express their gratitude to Prof. Yasutaka Furukawa, Satoshi Ikehata and Sebastian Ochmann for their assistance with the comparison.

References

[AH11] Adan A., Huber D.: 3D reconstruction of interior wall surfaces under occlusion and clutter. In *Proceedings Symposium on 3D Data Processing, Visualization, and Transmission* (2011), pp. 275–281. 2

[ASF*13] Arikan M., Schwärzler M., Flöry S., Wimmer M., Maierhofer S.: O-Snap: Optimization-based snapping for modeling architecture. *ACM Transactions on Graphics 32*, 1 (January 2013), 6:1–15. 4

[ASZ*16] Armeni I., Sener O., Zamir R. A., Jiang H., Brilakis I., Fischer M., Savarese S.: 3D semantic parsing of large-scale indoor spaces. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 1534–1543. 3

[BB10] Budroni A., Böhm J.: Automated 3D reconstruction of interiors from point clouds. *International Journal of Architectural Computing 8*, 1 (January 2010), 55–73. 2

[BdLGM14] Boulch A., de La Gorce M., Marlet R.: Piecewise-planar 3D reconstruction with edge and corner regularization. *Computer Graphics Forum 33*, 5 (August 2014), 55–64. 2, 7

[BK04] Boykov Y., Kolmogorov V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 9 (September 2004), 1124–1137. 5, 7

[BTS*14] Berger M., Tagliasacchi A., Seversky L. M., Alliez P., Levine J. A., Sharf A., Silva C. T.: State of the art in surface reconstruction from point clouds. In *Proceedings Eurographics (STARs)* (2014), pp. 161–185. 2

[BVZ01] Boykov Y., Veksler O., Zabih R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 11 (November 2001), 1222–1239. 6, 7

[CF14] Cabral R., Furukawa Y.: Piecewise planar and compact floorplan reconstruction from images. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 628–635. 2

[CLP10] Chauve A.-L., Labatut P., Pons J.-P.: Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition* (2010), pp. 1261–1268. 2, 7

[DBGBR*14] Di Benedetto M., Ganovelli F., Balsa R. M., Jaspe V. A., Scopigno R., Gobbetti E.: Exploremaps: Efficient construction and ubiquitous exploration of panoramic view graphs of complex 3d environments. *Computer Graphics Forum 33*, 2 (May 2014), 459–468. 3, 5

[EM94] Edelsbrunner H., Mucke E. P.: Three-dimensional alpha shapes. *ACM Transactions on Graphics 13*, 1 (January 1994), 43–72. 3

[IYF15] Ikehata S., Yan H., Furukawa Y.: Structured indoor modeling. In *Proceedings IEEE International Conference on Computer Vision* (2015), pp. 1323–1331. 2, 3, 7

[LA13] Lafarge F., Alliez P.: Surface reconstruction through point set structuring. *Computer Graphics Forum 32*, 2 (May 2013), 225–234. 2, 3

[LMS13] Laga H., Mortara M., Spagnuolo M.: Geometry and context for semantic correspondences and functionality recognition in manmade 3D shapes. *ACM Transactions on Graphics 32*, 5 (September 2013), 150:1–16. 3

[MMBM15] Monszpart A., Mellado N., Brostow G. J., Mitra N. J.: RAPter: Rebuilding man-made scenes with regular arrangements of planes. *ACM Transactions on Graphics 34*, 4 (August 2015), 103:1–12. 9

[MMJV*14] Mura C., Mattausch O., Jaspe Villanueva A., Gobbetti E., Pajarola R.: Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. *Computers & Graphics 44* (November 2014), 20–32. 2, 3, 4, 5, 7, 8, 9

[MPM*14] Mattausch O., Panozzo D., Mura C., Sorkine-Hornung O., Pajarola R.: Object detection and classification from large-scale cluttered indoor scans. *Computer Graphics Forum 33*, 2 (May 2014), 11–21. 3

[OLA14] Oesau S., Lafarge F., Alliez P.: Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. *ISPRS Journal of Photogrammetry and Remote Sensing 90* (April 2014), 68–82. 2, 4

[OLA16] Oesau S., Lafarge F., Alliez P.: Planar shape detection and regularization in tandem. *Computer Graphics Forum 35*, 1 (February 2016), 203–215. 9

[OVWK16] Ochmann S., Vock R., Wessel R., Klein R.: Automatic reconstruction of parametric building models from indoor point clouds. *Computers & Graphics 54* (February 2016), 94–103. 2, 3, 7, 9

[PGG*16] Pintore G., Garro V., Ganovelli F., Agus M., Gobbetti E.: Omnidirectional image capture on mobile devices for fast automatic generation of 2.5D indoor maps. In *Proceedings IEEE Winter Conference on Applications of Computer Vision (WACV)* (2016), pp. 1–9. 2

[THA*10] Tang P., Huber D., Akinci B., Lipman R., Lytle A.: Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Elsevier Automation in Construction 19*, 7 (November 2010), 829–843. 1

[TZ12] Turner E., Zakhor A.: Watertight as-built architectural floor plans generated from laser range data. In *Proceedings Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission* (2012), pp. 316–323. 2, 4

[TZ14] Turner E., Zakhor A.: Floor plan generation and room labeling of indoor environments from laser range data. In *Proceedings International Conference on Computer Graphics Theory and Applications* (2014). 2, 5

[VAB12] Vanegas C. A., Aliaga D. G., Benes B.: Automatic extraction of Manhattan-world building masses from 3D laser range scans. *IEEE Transactions on Visualization and Computer Graphics 18*, 10 (October 2012), 1627–1637. 2

[VD08] Van Dongen S.: Graph clustering via a discrete uncoupling process. *SIAM Journal on Matrix Analysis and Applications 30*, 1 (February 2008), 121–141. 2, 5, 6, 7

[VLA15] Verdie Y., Lafarge F., Alliez P.: LOD generation of urban scenes. *ACM Transactions on Graphics 34*, 3 (April 2015), 30:1–14. 2

[vLvKV11] van Lankveld T., van Kreveld M., Veltkamp R. C.: Identifying rectangles in laser range data for urban scene reconstruction. *Computers & Graphics 35*, 3 (June 2011), 719–725. 3