

Robust Reconstruction of Interior Building Structures with Multiple Rooms under Clutter and Occlusions

Claudio Mura* Oliver Mattausch* Alberto Jaspe Villanueva† Enrico Gobbetti† Renato Pajarola*

*Visualization and MultiMedia Lab, University of Zürich, Switzerland

†Visual Computing Group, CRS4, Sardinia, Italy

Abstract—We present a robust approach for reconstructing the architectural structure of complex indoor environments given a set of cluttered input scans. Our method first uses an efficient occlusion-aware process to extract planar patches as candidate walls, separating them from clutter and coping with missing data. Using a diffusion process to further increase its robustness, our algorithm is able to reconstruct a clean architectural model from the candidate walls. To our knowledge, this is the first indoor reconstruction method which goes beyond a binary classification and automatically recognizes different rooms as *separate* components. We demonstrate the validity of our approach by testing it on both synthetic models and real-world 3D scans of indoor environments.

Keywords-indoor scene reconstruction ; LIDAR reconstruction ; point cloud processing

I. INTRODUCTION

In the architecture and engineering domains there is a substantial need for semantically rich 3D models of buildings. 3D acquisition devices such as laser range scanners are now mature enough for fast, accurate, and cost-effective gathering of 3D measurements. However, efficient methods must be devised to extract such higher-level models from the acquired raw point-cloud data.

Of particular interest is the problem of determining the architectural shape of indoor environments (e.g., rooms bounded by walls, floors, and ceilings). While akin to the more well-studied topic of buildings and outdoor settings reconstruction, indoor reconstruction exhibits a number of distinctive features that make it significantly harder to manage (see Sec. II). First of all, methods for indoor reconstruction must be significantly more tolerant to missing data than their outdoor counterparts, since environments such as offices and apartments exhibit extremely high levels of clutter, which typically results in heavy occlusions of walls and other structures of interest (see Fig.1). Secondly, windows and other highly reflective surfaces are often present in such scenes. Due to the critical interaction properties of such elements with the input devices, large-scale artifacts, noise and missing data affect the acquired model (see Fig. 1). Finally, typical indoor structures such as apartments and office buildings pose the challenge of recognizing the structure in terms of a graph of connected rooms and corridors.

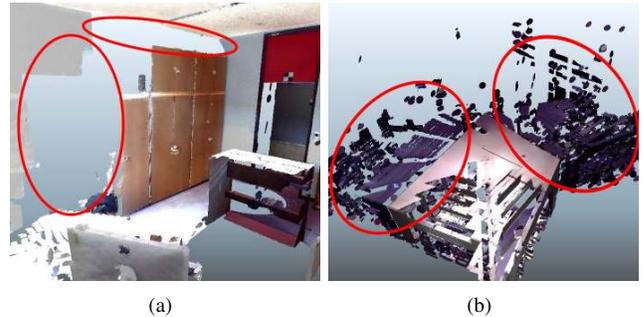


Figure 1. Heavy occlusions (a) and large-scale artifacts (b) often occur in scanned models of interiors.

Several approaches have been proposed to recover the architecture of indoor scenes. Some methods focus on reconstructing architectural floor plans [1], while others [2] aim at directly extracting wall boundaries. However, most current algorithms rely on the implicit assumption that the architectural components are well sampled, are only able to tolerate small amounts of clutter and can therefore fail in many real world situations. Moreover, many of the existing solutions are targeted at simply connected environments such as corridors and cannot reconstruct the shape of individual rooms within more complex environments.

This paper presents a robust approach to reconstructing the architectural shape of indoor environments given a set of cluttered input range scans that partially cover the scene of interest (typically 1 or 2 scans per room). Our method assumes that the environment is composed of multiple rooms bound by vertical walls, which holds true for the majority of buildings. An occlusion-aware process extracts vertical planar patches as candidates for genuine wall segments, separating them from clutter and coping with missing data, by using efficient *viewpoint-based* visibility computations on a per-scan basis. Starting from these candidates, we reconstruct a clean multi-room architectural model, by segmenting the area into multiple rooms using a robust diffusion process (see Fig. 2).

Our system extends and combines a number of state-of-the-art results in a non-trivial way. Its main novel contributions are the following:

- An efficient yet robust approach to extract architectural

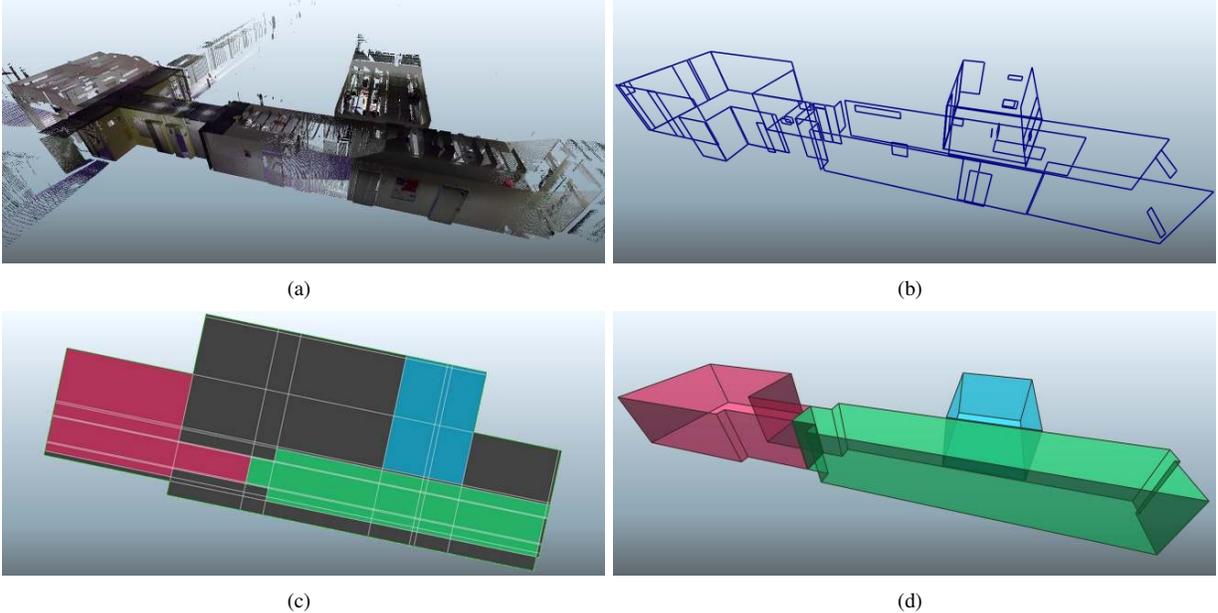


Figure 2. The main phases of our algorithm: From the input model (a) we robustly extract wall candidates (b). These are used to construct a cell complex in the 2D plane, from which we obtain a room segmentation using recursive clustering (c) and finally the individual room polyhedra (d).

elements corresponding to genuine wall segments that effectively takes missing data into account, a predominant issue in real-world indoor scan data.

- An automatic segmentation of the input model into the correct number of rooms, which is robust with respect to imperfect data due to the used diffusion distances.
- A robust reconstruction of each detected room as a single watertight polyhedron.

To our knowledge, this is the first indoor reconstruction method which goes beyond a binary in/out classification and automatically recognizes different rooms as *separate* components. Such a room labeling is useful in many real-world applications, such as the definition of thermal zones for energy simulation. As demonstrated in Sec. VII, the method is applicable to environments with an extremely high level of clutter, and is robust in terms of tolerance to scanning noise and artifacts from reflecting surfaces.

II. RELATED WORK

Classical methods for reconstructing building structures from 3D laser scanner data have focused on creating visually realistic models [3], [4], rather than structured 3D building models. Even though some of these 3D reconstruction algorithms extract planar patches from data [5], this has the goal of finding simplified representations of the models, rather than identifying walls, ceilings, and floors. In this context, clutter is dealt with specialized hole-filling techniques [4], [6], [7], which can only manage small-scale occlusions.

More recently, focus has shifted on the creation of more structured 3D models, with the purpose of simplifying the process of converting point cloud data into a building information model (the *scan-to-BIM* problem). One of

the key challenges is an effective handling of occlusion. Most previous work in this area assumes almost complete visibility, or that parts of the building occluded from one view are available from another viewpoint [4], [6], [1]. This assumption is not verified in most practical situations, which have to deal with complex occlusions and heavily cluttered environments. Most recent work thus exploits prior knowledge on building structure to achieve robustness.

Using the heavily constrained *Manhattan World* (MW) assumption, which forces planar and axis-aligned orthogonal walls, Furukawa et al. [8] reconstruct the 3D structure of moderately cluttered interiors by fusing multiple depth maps (created from images) through the solution of a volumetric Markov Random Field, while Vanegas et al. [9] reconstruct buildings from laser scans by detecting box structures and shooting visibility rays to label the volumes as either inside or outside. We focus, instead, on less constrained environments with vertical, but non-orthogonal walls.

In this setting, inside/outside labeling, possibly combined with visibility computations and energy minimization techniques, is often used to perform volumetric segmentation of scanned models. Chauve et al. [10] build a BSP-like space partitioning structure from an input point cloud and then solve a minimum *st*-cut on its cell-adjacency graph, using visibility criteria for the labelling of the arcs. Similarly, Lafarge et al. [11] compute a 3D Delaunay triangulation of a filtered version of the input point set and solve a minimum cut problem on its dual structure. The arcs of the graph are weighted using visibility sampling. Oesau et al. [2] generate a 3D space partitioning by stacking lines detected in the vertical projection of the input model, then label the volumetric cells into inside/outside using a visibility-driven

energy minimization. Sanchez and Zakhor [12] focus on the simultaneous detection of both large-scale and small-scale architectural structures, while Adan et al. [13], [14] proposed a method that discriminates between empty space and occlusions, and that can fill the latter. All of these methods assume moderately clean environments and simply perform a binary classification of space, while we propose an automatic segmentation of the input model into the correct number of rooms, which is robust with respect to imperfect data due to the used diffusion distances.

III. METHOD OVERVIEW

The input to our algorithm is a set of point clouds taken at known locations and representing one or more rooms of the interior of a building. We assume that the scans are registered in the same reference frame and, without loss of generality, that the up-vector is $\mathbf{v}_{up} = (0, 0, 1)$. We consider only buildings with planar, vertical walls, but, like Lafarge et al. [2], we drop the more restrictive Manhattan World assumption. The method produces a set of k closed polyhedra as output, one for each room in the input scene.

Although we target the reconstruction of indoor environments with vertical walls, our pipeline does not purely work in a 2D projection in the xy -plane, but we perform operations both in the 3D space and in the 2D projection. In particular, the patch detection and the occlusion-based pruning are performed in 3D space, since this captures the shape of the patches more faithfully, resulting in effective wall regions selection. Similarly, the final wall fitting is performed directly on the points in 3D space to make the estimate of their position more accurate. The subsequent diffusion-based room segmentation is performed entirely in the 2D projection (i.e., the floor plan), as the assumption of vertical walls makes the use of the third dimension redundant.

In the following, we summarize the main steps of our proposed approach. A visual overview of the method is given in Fig. 2.

Occlusion-aware selection of candidate walls. Vertical planar regions that are potential wall patches are extracted from the input scans. For each scan, occluding patches are then projected onto the potential wall patches to recover their actual (unoccluded) vertical extent and hence get a robust indicator of the likelihood that they are genuine wall segments, pruning those which are likely to be clutter.

Room segmentation. This step is performed entirely in the 2D projection in the xy -plane. First of all, projected candidate walls are clustered to get a smaller number of good representative lines for walls. Secondly, a cell complex is built from the intersections of the representative lines and its edges are weighted according to the likelihood of being genuine walls. Diffusion distances are then computed on the cell-graph of the complex

and they are used to drive a recursive clustering of the cells that extracts the separate rooms.

Model reconstruction. The accurate wall geometry is computed for each room by robustly fitting the extracted planes to the inlier points in 3D. Finally, each room polyhedron is created by intersecting the reconstructed wall planes with those of floor and ceiling.

These three steps are described in more detail in the following sections.

IV. OCCLUSION-AWARE SELECTION OF CANDIDATE WALLS

In this phase we extract from the input point clouds a set of planar patches that correspond to candidate walls. We first grow planar regions in the 3D point cloud. To keep only segments which potentially correspond to candidate walls, we select only those regions which are classified as vertical. A lightweight 3D occlusion check is used to further prune the vertical patches, discarding those that have a low unoccluded vertical extent.

A. Low-level segmentation into planar patches

Since our input models are raw and unstructured point clouds, the very first step of our pipeline must identify some structured evidence of the architectural shape of interest. A natural choice for buildings composed of planar elements is to use planar patches, as done in many previous approaches [14], [10], [11]. The use of 3D patches, as opposed to, e.g., 2D line projections [13], [2], is well-suited for our occlusions-based pruning algorithm. We perform patch growing on a per-scan basis, so that every patch contains points that belong to a same scan. This way, when looking for potential occluders of a patch, we can restrict the search to the patches extracted from the same scan.

We extract patches using a simple region growing process based on normal deviation and plane offset. Like Chauve et al. [10] we have found this scheme to work well; more robust and elaborate methods [15] were not needed in our applications. Since a correct choice of the seed points is very important, we start the growing from the points that have planar and low-noise neighborhoods. The quality of a candidate seed is evaluated by fitting a plane to its k -neighbors with the *Least-Median-of-Squares* (LMS) algorithm [16] and by then computing the sum of the residuals.

For the next steps we need a simplified patch representation, and we found that an oriented bounding box (OBB) gives us a reasonable trade-off between simplicity and shape approximation quality. The OBB is aligned with the two principal components of the xy -plane projection of the patch and gives a good fit for structures like long and thin walls that are not aligned with the main axes (see Figure 3).

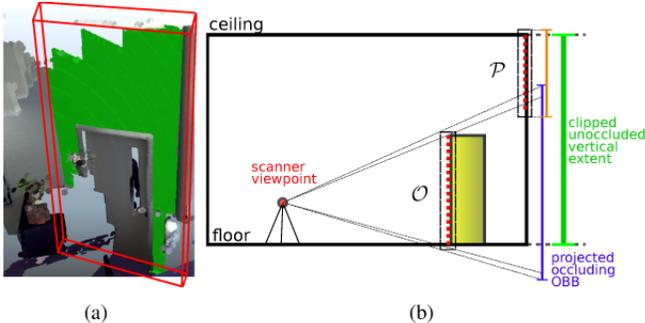


Figure 3. (a) OBB corresponding to a candidate wall patch. (b) The OBB of an occluding patch is projected onto the plane of a candidate wall patch to recover its unoccluded vertical extent.

B. Selection of candidate wall patches

In order to restrict our method to architectural structures of interest, we only consider vertical regions as *potential* wall patches. Thus, we only consider patches for which $|\mathbf{n} \cdot \mathbf{v}_{up}| > 1 - \epsilon$, where the patch normal \mathbf{n} is computed using *Principal Component Analysis* (PCA). This effectively rules out flat structures like tables. We also discard small cluttering patches for which the horizontal extent (i.e., the longer of the two horizontal sides of their OBB) is smaller than 40cm .

This process does not yet exclude large vertical cluttering elements such as large cupboards from the potential wall patches. We therefore perform a further pruning, following the intuition that genuine wall structures must cover a vertical extent that is almost equal to the distance between the floor and the ceiling. Checking for this condition in real-world inputs such as scans of offices or apartments is problematic. Obstacles located between the camera and the walls (see Figure 1, left) can severely limit the amount of structure visible. Taking more scans from additional viewpoints can only partially solve this problem and it can not be considered a viable solution for static laser scanning.

For this purpose, we employ a lightweight visibility test to estimate the expected *unoccluded* vertical extent of each potential wall patch \mathcal{P} . In our technique, an occlusion happens if the OBB of a patch \mathcal{P} and that of an occluder \mathcal{O} overlap when seen from the scan position from which they were taken. We construct the *infinite shadow volume* [17] of each \mathcal{O} by casting rays from the scan position through the vertices of its OBB. We then compute the intersection of this shadow volume with the plane induced by \mathcal{P} through its normal and the center of its OBB. Finally, the projection of the shadow volume is tested for overlap with the bounding rectangle of \mathcal{P} (i.e., the projection of its OBB). This process is shown in Figure 3. When an occlusion between \mathcal{P} and \mathcal{O} occurs, we consider the vertical extent of the projection of \mathcal{O} onto \mathcal{P} and merge it to the vertical extent of \mathcal{P} itself. By repeating this check for every \mathcal{O} , we obtain the combined height h of \mathcal{P} . We then prune \mathcal{P} from the candidate list if

the following condition is satisfied: $h \leq (1 - \eta) \cdot h_{rooms}$. Here η is a small number which we set to 0.05 and h_{rooms} is the distance between the floor and the ceiling. Note that we obtain an accurate measure for h_{rooms} as a byproduct of the robust fitting of wall and ceiling planes described in Section VI.

By repeating this check for every potential wall patch we obtain a pruned list of actual candidate walls that are likely to belong to wall structures. The use of the unoccluded vertical extent significantly improves the selection of candidate walls in cluttered environments and all subsequent steps of our algorithm benefit from this. Our method works well even though it is based on an approximation of the visibility problem. On the other hand, we believe that a more sophisticated analysis would be inadequate for this application due to the imperfect nature of real world input data, which contain large holes and missing parts.

V. AUTOMATIC ROOM SEGMENTATION

The following steps are computed entirely in the 2D projection in the xy -plane, which leads to a simplified description as compared to a full 3D model. We first merge the 2D projections of candidate wall patches corresponding to a same wall structure to get a reduced number of *representative lines*. Next, we construct a description of the area of interest consisting of polygonal faces. By computing the diffusion distances between faces, we get a global measure of similarity that can be used to recursively cluster this cell complex into the individual rooms.

A. Computing representative lines

Similar to the approach of Oesau et al. [2], the computed candidate wall patches are projected in 2D to obtain a set of *line segments*, which are then clustered using *mean-shift*. A first directional clustering yields the main orientations of the walls and is followed by a 1D mean-shift clustering [18] which identifies possible multiple offsets of parallel wall segments along a same direction. This way we obtain a set of clusters of line segments $\mathcal{C} = \{\mathcal{C}_0, \dots, \mathcal{C}_n\}$. Each cluster \mathcal{C}_k corresponds to a particular wall structure and is associated with a *representative line* l_k . We explicitly store the list of representative lines $\mathcal{L} = \{l_0, \dots, l_n\}$ as well as the associated clusters of line segments \mathcal{C} , as this helps us compute the weights of the edges in the subsequent cell complex construction step.

B. Cell complex construction

From the representative lines \mathcal{L} we build a partition of the plane that represents the floor plan of the processed indoor environment. Our plane partition is a standard 2D cell complex induced by \mathcal{L} , also known in the literature as *arrangement of lines* [19]. We discard from the complex the vertices and the edges that lie outside the bounding box of the input model. An example of such a cell complex can be

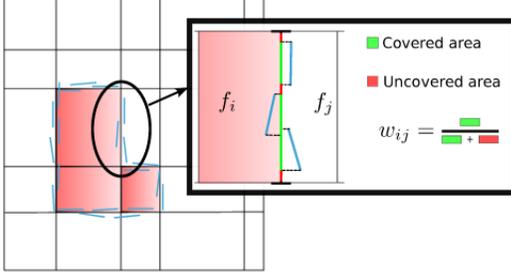


Figure 4. Computation of the coverage of an edge of the 2D cell complex. The 2D line segments of candidate walls (in light blue) are first projected onto the edge, then the ratio of the occupied length to the total length is assigned as a weight to the segment.

seen in Figure 5. Note that, during the construction process, we associate to each edge of the complex the representative line l_k from which the edge originated, together with the corresponding cluster of 2D line segments \mathcal{C}_k .

Given an edge e_{ij} between two faces f_i and f_j of the complex, we want to assign it a weight w_{ij} that corresponds to its likelihood of being a real wall structure. To do so, we consider all nearby candidate walls (i.e., all line segments of the cluster \mathcal{C}_k associated to e_{ij}) and project them onto e_{ij} itself. The weight w_{ij} is defined as the fraction of the extent of the edge that is covered by such projections. The computation of the weight w_{ij} of the edge between two faces f_i and f_j is shown in Figure 4.

Note that we also keep a so called *infinity face* f_∞ , which corresponds to the outside and has an edge incident to each face on the boundary of the cell complex. This infinity face will become important for the termination criterion during the recursive clustering.

C. Diffusion embedding

Once the cell complex representing the environment is built, we establish a global affinity measure among its faces. We use the coverage weights w_{ij} of the edges to derive a graph Laplacian matrix \mathbf{L} , with entries \mathbf{L}_{ij} defined as follows:

$$\mathbf{L}_{ij} = \begin{cases} e^{-w_{ij}/2\sigma} & \text{if } i \neq j \wedge f_i, f_j \text{ are adjacent,} \\ 1 & \text{if } i = j, \\ 0 & \text{otherwise} \end{cases}$$

From matrix \mathbf{L} we define a Markov probability transition matrix as $\mathbf{M} = \mathbf{D}^{-1}\mathbf{L}$, with $\mathbf{D} = \text{diag}(\sum_{j=1}^n \mathbf{L}_{ij})$. Each element \mathbf{M}_{ij} can be seen as a local affinity value between faces f_i and f_j , as it is defined by considering only direct connectivity between faces. We propagate these local affinities by means of *diffusion maps* [20], which are known to be robust against noise [21] and therefore well suited for our task. The idea behind the diffusion framework is to compute a mapping Φ that embeds the original data in a multidimensional space. The standard Euclidean distance in this space is a measure of dissimilarity between the faces

of the cell complex. In other words, if $\|\Phi(f_i) - \Phi(f_j)\|_2$ is low then the faces f_i and f_j are likely to be in the same room. This process has also a physical interpretation as heat diffusion: it can be seen as a measure of how much heat can flow from f_i to f_j in a given diffusion time. The closer the faces are in the Euclidean embedding, the faster the heat can flow from one to the other.

Given a face f_i , its corresponding coordinate in the Euclidean embedding is

$$\Phi(f_i) = (\lambda_1^t \phi_1(f_i), \lambda_2^t \phi_2(f_i), \dots, \lambda_m^t \phi_m(f_i)),$$

where λ_k and ϕ_k are the k -th eigenvalue and eigenvector of \mathbf{M}^t respectively. Two parameters control the diffusion process: the *diffusion time* t , a measure of smoothness that determines how much the affinities are propagated; and the number m of eigenvalues/eigenvectors of \mathbf{M} used in the diffusion map, corresponding to the dimensionality of the embedding. We have experimented with different parameter settings to discover that the influence of changes in these values is small. All the results shown in this paper have been obtained using $t = 10$, $m = 4$, and $\sigma = 0.01$.

D. Recursive segmentation

We use the embedding distances described in Section V-C to compute a segmentation of the 2D complex into separate rooms. The desired partitioning is achieved by repeatedly applying a binary version of the k -medoids algorithm to the set of faces of the complex. At each step, the clustering algorithm selects the two faces that are maximally far apart in terms of diffusion distance and uses them as centers (*medoids*) of two new clusters. Each step of the recursive partitioning creates a cluster corresponding to a new room and another cluster that contains the faces not yet labeled. We continue segmenting the unlabeled set until *all* the scan positions have been assigned to a cluster which does not contain f_∞ . This way we *automatically* create a room partitioning without the need to explicitly know the number of rooms. An alternative approach would be to directly perform a multi-way segmentation; however, this would require computing the exact number of rooms beforehand. Our approach achieves the same result without needing such an explicit step.

The introduced heuristics work well because of the properties of the diffusion embedding. At each step, the method puts into different clusters faces that are maximally far apart in terms of diffusion distances. The faces that correspond to the inside and those that represent the outside are connected by many edges (the whole perimeter wall of the building) and each of them contributes to put the inside and the outside faces closer in terms of diffusion distances. On the other hand, the separation between individual rooms is either due to the physical distance or, in the case of adjacent rooms, to edges associated to candidate walls. As long as such candidate walls are sufficiently solid, the faces inside a room

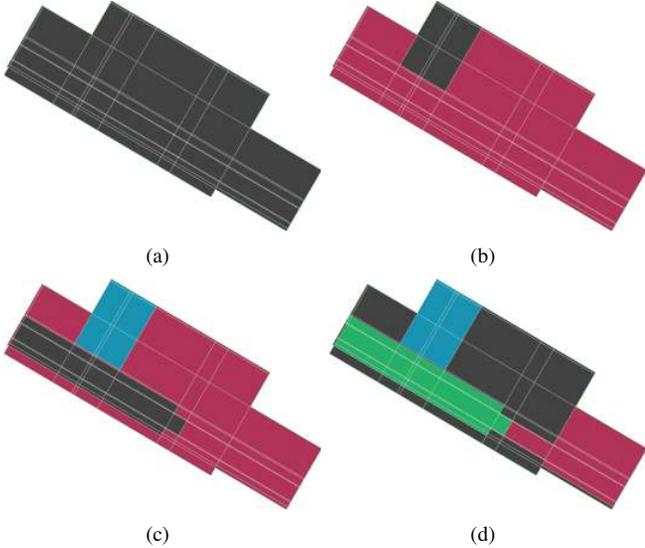


Figure 5. An illustration of the recursive clustering process: the input cell complex (a); first (b) and second (c) levels of partitioning, with the unlabeled space shown in magenta; final results of the partitioning (d), with the gray faces representing the outside space and the rooms shown in color. The white lines indicate the underlying cell complex built from the representative lines.

are likely to be much closer in the diffusion embedding to the outside faces than to the faces in other rooms. Hence our method finishes separating the inside from the outside when all individual rooms have been detected and put in separate clusters.

VI. MODEL RECONSTRUCTION

Eventually we reconstruct the 3D polyhedra from the detected room clusters. For each edge in the boundary of a cluster, we access the projected candidate walls \mathcal{C}_k associated to the edge (see Section V-A). We then select the points of the corresponding 3D patches and employ a LMS fitting scheme (known to have a 50% breakdown point with respect to outliers) to get the inlier points for that plane. To get the best possible estimate of the plane parameters, a standard Principal Component Analysis (PCA) is additionally performed on the inlier set. We also use a robust pipeline for reconstructing the floor and ceiling planes (see below). The polygons of the final polyhedra are obtained by intersecting pairs of adjacent wall planes with the floor and ceiling planes. An example of the complete room polyhedra resulting from a given segmentation can be seen in Figure 2 (bottom row).

Since we assume that floor and ceiling are planar and orthogonal to the up-vector, we find during the patches extraction the two horizontal patches \mathcal{P}_{floor} and \mathcal{P}_{ceil} with respectively minimum and maximum z value (Section IV-A). To increase the accuracy and robustness of the estimation, we employ the following strategy to fit the final planes. Given \mathcal{P}_{floor} (respectively \mathcal{P}_{ceil}), we take the horizontal patches whose distance from \mathcal{P}_{floor} (\mathcal{P}_{ceil}) is less than a

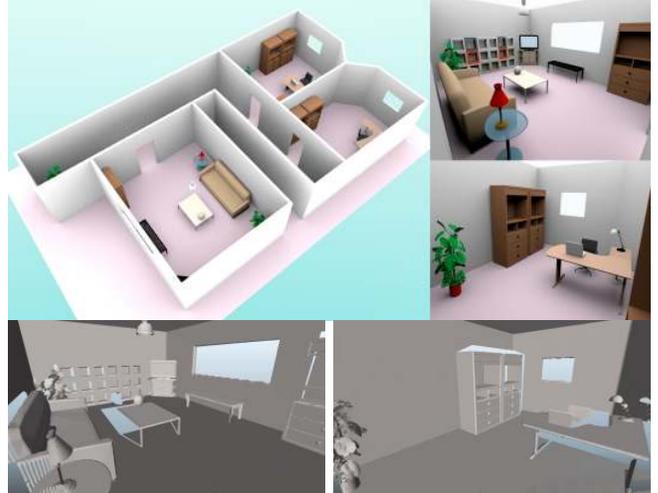


Figure 6. We use a synthetic model of an indoor environment to test the accuracy of our algorithm. The model (top) is re-sampled in software using ray-casting to simulate the behavior of a 3D laser range scanner. Notice the large areas in shadow in the bottom pictures.

threshold and use their points as support set for a LMS fit. Note that throughout this process we only consider patches whose diagonal is larger than 0.5 meters.

VII. RESULTS AND DISCUSSION

All results were computed on an Intel Core i7 960 processor clocked at 3.2 GHz. The computation is performed entirely on the CPU, using OpenMP directives to parallelize some processing stages, including planar regions growing and weighting of the cell complex.

We have tested our algorithm on three real-world datasets. These datasets were acquired by LIDAR laser scanning with a sampling resolution of 24mm at 10 meters. The first dataset, which we denote as ROOM, corresponds to a single room. The other two (OFFICE 1 and OFFICE 2) correspond to corridors with attached rooms acquired in two different office environments.

We also created and tested a synthetic model (SYNTHETIC, shown in Figure 6). The model has been generated manually using a 3D modeling software and has virtually been scanned from 7 positions to simulate the result of 3D laser scanning. The sampling resolution in this virtual scanning process was set to the one used in real-world acquisitions. To make the simulation more realistic, we corrupted the depth measurements using Gaussian noise with $\sigma = 0.25\text{cm}$.

Relevant statistics about the used datasets together with experimental results and timings are listed in Table I. Beyond the basic data, the table shows the number of extracted candidate wall patches (column 4), the number of rooms in the input data set (column 5), the number of rooms detected by the method (column 6) and the number of rooms missed (column 7). Overall we can see that there is a single misclassification error in all our datasets (the

Dataset	Points	Scans	Walls	Rooms	Detected	Missed	Phase 1	Phase 2	Phase 3	Sum
ROOM	8.3M	3	21	1	1	0	6.5s	13.8s	1.4s	21.7s
OFFICE 1	11.1M	4	50	3	3	0	7.1s	25.1s	1.5s	33.7s
OFFICE 2	13.8M	5	57	2	3	0	8.8s	31.7s	1.6s	42.1s
SYNTHETIC	19.4M	7	125	4	4	0	14.1s	47.0s	0.9s	62.0s

Table I

DESCRIPTION OF OUR DATASETS WITH STATISTICS AND RUNNING TIMES. COLUMN Walls CORRESPONDS TO THE NUMBER OF EXTRACTED WALL PATCHES, Detected AND Missed TO THE NUMBER OF ROOMS RESPECTIVELY DETECTED AND MISSED BY OUR METHOD. Phase 1 CORRESPONDS TO WALL CANDIDATE DETECTION, Phase 2 TO ROOM SEGMENTATION, AND Phase 3 TO WALL RECONSTRUCTION.

above-mentioned over-segmentation) and that no room was missed by our method. The timings show that the algorithm is reasonably fast for up to 20M input points. From the overall times we deduce that the computation time grows slightly faster than linearly with respect to the number of points in the input model.

The ROOM dataset is shown in the first row of Figure 7. Due to the large window front this is an exceptionally difficult setting for the laser scanner, resulting in many reflection artifacts and thus in a huge number of outliers. Nevertheless, our algorithm is able to correctly extract the shape of the room. OFFICE 1 (row 2) represents a more complex environment composed of multiple rooms. Our algorithm accurately detects the individual rooms and the associated polyhedra. The highly anisotropic shape of the corridor in OFFICE 2 (row 3) is also reconstructed well by our algorithm and correctly separated from the neighboring room (shown in red). In this case, the corridor is split into two different clusters (shown in green and blue), resulting in over-segmentation. However, such an error can be easily corrected interactively by means of simple user sketching. SYNTHETIC (row 4) has a complex building structure and shows a case that does not comply with the *Manhattan World* assumption as OFFICE 1. Nevertheless, our algorithm works flawlessly for this input and can retrieve the bounding polyhedra of all 4 rooms.

VIII. CONCLUSIONS AND FUTURE WORK

We have presented a system for the automatic reconstruction of complex indoor environments that can correctly segment the input model into separate rooms. Our method is robust to clutter and occlusions and performs well in real-world scenarios. We regard our algorithm as a first step towards going beyond simple geometric reconstruction to extract semantic information from the input dataset.

As for all current approaches, our approach has also some limitations. In particular, we consider only buildings that have planar, vertical walls as well as horizontal ceilings. We focus on the robust extraction of basic room shapes, and do not attempt to recognize fine architectural details. As a future goal we would like to create a fully parametrized architectural model that could serve as a basis for the editing work of a designer. We also plan to incorporate slanted walls, curved surfaces and other typical architectural structures into our robust pipeline.

Acknowledgments. This work is partially supported by the People Programme (Marie Curie Actions) of the European Union’s Seventh Framework Programme FP7/2007-2013/ under REA grant agreement n°290227 (DIVA) and Sardinian Regional Authorities.

REFERENCES

- [1] E. Turner and A. Zakhor, “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] S. Oesau, F. Lafarge, and P. Alliez, “Indoor scene reconstruction using primitive-driven space partitioning and graph-cut,” in *Proceedings Eurographics Workshop on Urban Data Modeling and Visualisation*, 2013, pp. 9–12.
- [3] S. F. El-Hakim, P. Boulanger, F. Blais, and J.-A. Beraldin, “System for indoor 3D mapping and virtual environments,” in *Proceedings SPIE Videometrics V*, vol. 3174, 1997, pp. 21–35.
- [4] C. Frueh, S. Jain, and A. Zakhor, “Data processing algorithms for generating textured 3D building facade meshes from laser scans and camera images,” *International Journal of Computer Vision*, vol. 61, no. 2, pp. 159–184, February 2005.
- [5] I. Stamos, G. Yu, G. Wolberg, and S. Zokai, “3D modeling using planar segments and mesh elements,” in *Proceedings IEEE Symposium on 3D Data Processing, Visualization, and Transmission*, 2006, pp. 599–606.
- [6] F. Dell’Acqua and R. Fisher, “Reconstruction of planar surfaces behind occlusions in range images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 569–575, April 2002.
- [7] A. D. Sappa, “Improving segmentation results by studying surface continuity,” in *Proceedings Conference on Pattern Recognition*, vol. 2, 2002, pp. 929–932.
- [8] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, “Reconstructing building interiors from images,” in *Proceedings International Conference on Computer Vision*, 2009, pp. 80–87.
- [9] C. A. Vanegas, D. G. Aliaga, and B. Benes, “Automatic extraction of Manhattan-world building masses from 3D laser range scans,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 10, pp. 1627–1637, October 2012.
- [10] A.-L. Chauve, P. Labatut, and J.-P. Pons, “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.

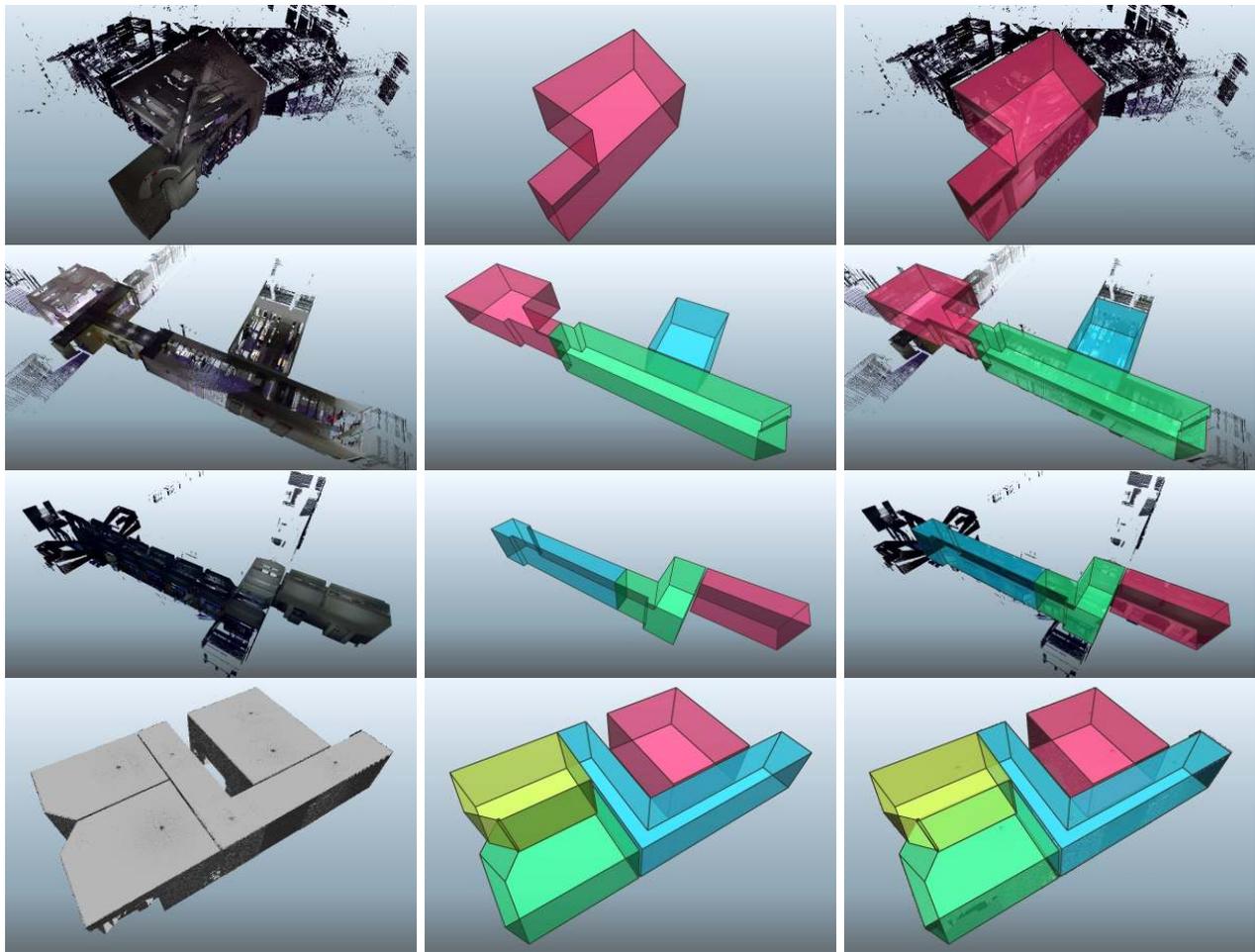


Figure 7. From top to bottom: ROOM, OFFICE 1, OFFICE 2 and SYNTHETIC datasets. Notice that our algorithm can cope well with large-scale outlier artifacts originating from reflections (ROOM) as well as with complex environments (OFFICE 1) and nontrivial structures (corridor in OFFICE 2, room structure in SYNTHETIC). From left to right: input model, extracted room polyhedra, superimposed input and output models.

- [11] F. Lafarge and P. Alliez, "Surface reconstruction through point set structuring," in *Proceedings Eurographics*, 2013, pp. 5–17.
- [12] V. Sanchez and A. Zakhor, "Planar 3D modeling of building interiors from point cloud data," in *Proceedings IEEE International Conference on Image Processing*, 2012, pp. 1777–1780.
- [13] A. Adan and D. Huber, "3D reconstruction of interior wall surfaces under occlusion and clutter," in *Proceedings Symposium on 3D Data Processing, Visualization and Transmission*, 2011, pp. 275–281.
- [14] A. Adan, X. Xiong, B. Akinci, and D. Huber, "Automatic creation of semantically rich 3D building models from laser scanner data," *Automation in Construction*, vol. 31, pp. 325–337, May 2013.
- [15] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, June 2007.
- [16] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. John Wiley & Sons, 1987.
- [17] F. C. Crow, "Shadow algorithms for computer graphics," *Proceedings ACM SIGGRAPH*, vol. 11, no. 2, pp. 242–248, July 1977.
- [18] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Manhattan-world stereo," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1422–1429.
- [19] H. Edelsbrunner, J. O'Rourke, and R. Seidel, "Constructing arrangements of lines and hyperplanes with applications," *SIAM Journal on Computing*, vol. 15, no. 2, pp. 341–363, May 1986.
- [20] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, July 2006.
- [21] Y. Lipman, X. Chen, I. Daubechies, and T. Funkhouser, "Symmetry factored embedding and distance," *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 103:1–12, July 2010.