Computers & Graphics 44 (2014) 20–32

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts ☆

Claudio Mura [a,*], Oliver Mattausch [a], Alberto Jaspe Villanueva [b], Enrico Gobbetti [b], Renato Pajarola [a]

[a] Visualization and MultiMedia Lab, University of Zürich, Switzerland
[b] Visual Computing Group, CRS4, Sardinia, Italy

ARTICLE INFO

Article history:
Received 9 May 2014
Received in revised form
6 July 2014
Accepted 8 July 2014
Available online 1 August 2014

Keywords:
Indoor scene reconstruction
LIDAR reconstruction
Point cloud processing

ABSTRACT

We present a robust approach for reconstructing the main architectural structure of complex indoor environments given a set of cluttered 3D input range scans. Our method uses an efficient occlusion-aware process to extract planar patches as candidate walls, separating them from clutter and coping with missing data, and automatically extracts the individual rooms that compose the environment by applying a diffusion process on the space partitioning induced by the candidate walls. This diffusion process, which has a natural interpretation in terms of heat propagation, makes our method robust to artifacts and other imperfections that occur in typical scanned data of interiors. For each room, our algorithm reconstructs an accurate polyhedral model by applying methods from robust statistics. We demonstrate the validity of our approach by evaluating it on both synthetic models and real-world 3D scans of indoor environments.

1. Introduction

In architecture and engineering, there is a substantial need for semantically rich 3D models of buildings. As 3D designs are most often not available, or significantly different from the "as-is" condition of a given building, technology for creating models from observations is of primary importance. 3D acquisition devices such as laser scanners are now available for fast, accurate and cost-effective acquisition of 3D data. However, efficient methods must be devised to extract higher-level models from the acquired raw point-cloud data.

Of particular interest is the problem of determining the architectural structure of indoor environments (e.g., room walls, floors and ceilings). Indoor reconstruction exhibits a number of distinctive challenges that make it significantly harder to manage than the more well-studied problem of building shape reconstruction from outdoor scans (see also Section 2). First of all, indoor reconstruction methods 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. This typically results in heavy occlusions of walls and other structures of interest (see also Fig. 1). Secondly, windows and other highly reflective surfaces are often present in such scenes. As a result, the acquired model is heavily affected by large-scale artifacts, measurement noise and missing data, due to the critical interaction properties of the reflective elements with the measurement devices (see also Fig. 1). Finally, creating structured 3D models of typical indoor environments, such as apartments and office buildings, poses the challenge of recognizing their interior structure in terms of a graph of connected rooms and corridors.

Much of the work on interior environments has focused so far on the analysis and classification of the objects in the scene [1,2], while the problem of recovering architectural components is less developed, and has concerned mostly floor plan reconstruction and wall boundary determination (see Section 2). Most current methods rely on the implicit assumption that the architectural components are well sampled. Even those approaches that include an explicit filtering stage in their pipeline are only able to tolerate small amounts of clutter and can fail in many situations that are commonly found in real world scenes. 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.

In this paper, we present a robust pipeline for reconstructing a clean architectural model of an indoor environment from a set of cluttered 3D input scans that partially cover the scene of interest

☆This article was recommended for publication by Hongbo Fu.

* Corresponding author. Tel.: +41 44 6354365.
E-mail address: claudio@ifi.uzh.ch (C. Mura).

http://dx.doi.org/10.1016/j.cag.2014.07.005

(typically 1 or 2 panoramic scans per room). Our method only assumes that the scanner positions are known, and the environment is composed of multiple rooms bound by vertical walls, which holds true for a vast majority of buildings, and is capable to recover a room graph, as well as an accurate polyhedral representation of each room.

The whole pipeline is depicted in Fig. 2. 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 a space partitioning induced by the candidate walls, we use a robust heat diffusion process to propagate similarities between cells of the partitioning which belong to the same room. We then cluster the area into multiple rooms using an iterative binary subdivision algorithm. Unlike standard methods like *k*-means, our solution automatically finds the correct number of rooms without a termination threshold by exploiting the knowledge of the scanner positions.

This work is a significantly extended version of our *CADCG 2013* contribution [3]. Besides supplying a more thorough exposition, we provide here significant new material and a number of important novel contributions. Our main improvements are the following:

- a thorough description of the room detection process, which shows in more detail the properties of the computed diffusion embedding and provides a comprehensive analysis of the subdivision scheme;
- important methodological improvements, including a post-processing stage that corrects possible imperfections in the clustering results and a more effective robust technique based on *M-estimators* [4] for the reconstruction of the final wall planes;
- an extended evaluation, where we perform both a qualitative and quantitative analysis on a wider set of inputs, including two large real-world datasets and two new synthetic datasets that feature more complex room layouts and high variability in the shapes of the rooms.

The overall approach is the first indoor reconstruction pipeline capable of coping with heavy occlusions and missing data, while automatically recognizing different rooms as *separate* components. Such a room labeling is useful in many real-world applications, such as room asset planning and management or the definition of thermal zones for energy simulation. As demonstrated in Section 7, the method is applicable to large real-world environments with an extremely high level of clutter and is robust to scanning noise and large artifacts originating from reflecting surfaces.

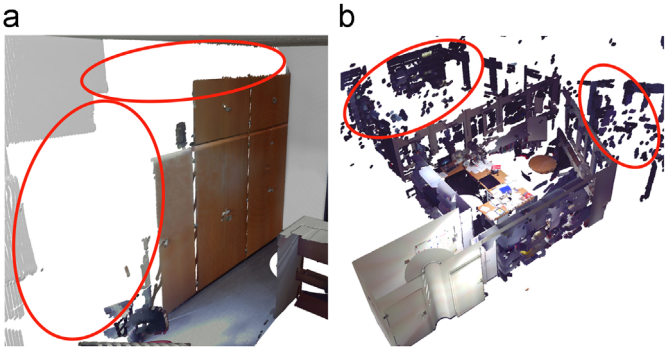


Fig. 1. Heavy occlusions (a) and large-scale artifacts (b) often occur in scanned 3D models of interior rooms.

2. Related work

Many researchers have studied the problem of reconstructing building structures from 3D laser range scan data. In this section, we briefly discuss only the approaches that most closely relate to ours.

Classical methods have often focused on creating visually realistic models [5,6], rather than structured 3D building models. Even though some of these 3D reconstruction algorithms extract planar patches from data [7], 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 [6,8,9], which can only manage small-scale occlusions.

More recently, focus has shifted to 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). In this sense, an important step towards the production of semantically rich models is the detection of the rooms in the input environment. Our previous work [3], which is extended and complemented by this paper, presents an occlusion-aware method that employs a diffusion process to automatically extract multiple rooms from interiors models. Recently, Turner and Zakhor [10] solved the same problem by first over-segmenting the 2D floor plan into portions of rooms and then merging adjacent segments to obtain the final partitioning. However, their approach does not perform an effective handling of occlusions. Ochmann et al. [11] also proposed a method for segmenting laser-scanned indoor models into different rooms, but their approach simply classifies the input points and does

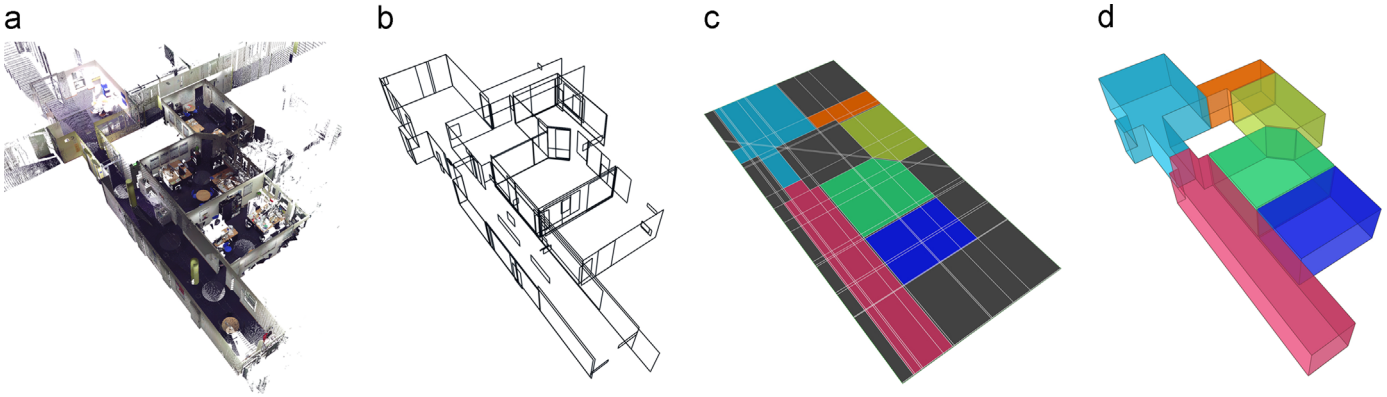


Fig. 2. The main phases of our algorithm: from the input model (a) we robustly extract candidate walls (b). These are used to construct a cell complex in the 2D floor plane. From this we obtain a partitioning into individual rooms (c) and finally the individual room polyhedra (d). Note that in (a) the ceiling has been removed for the sake of visual clarity.

Please cite this article as: Mura C, et al. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Comput Graph (2014), http://dx.doi.org/10.1016/j.cag.2014.07.005

not aim at reconstructing an accurate model of the sole architectural components. Moreover, their algorithm cannot successfully deal with non-convex shapes.

Surprisingly enough, the problem of occlusions is disregarded in much of the previous work in this area. Most existing approaches assume either that the scene is almost completely visible or that parts of the building occluded from one view are available from another viewpoint [6,8,12,13]. 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. [14] 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. [15] reconstruct buildings from 3D laser range 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 and non-convex room boundaries.

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. [16] 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 labeling of the arcs. Similarly, Lafarge and Alliez [17] compute a 3D Delaunay triangulation of a filtered version of the input point set and solve a min-cut problem on its dual structure. The arcs of the graph are weighted using visibility sampling. Oesau et al. [18] 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 [19] focus on the simultaneous detection of both large-scale and small-scale architectural structures, while Adan et al. [20,21] 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 real expected number of rooms, which is robust with respect to imperfect data due to the used diffusion distances.

In contrast to the fully automatic methods described earlier, other approaches include human intervention in their workflow [22–24]. This is orthogonal to our method, which could also be employed as a component of an interactive solution.

3. Method overview

The input to our algorithm is a set of 3D point clouds representing one or more rooms of the interior of a building and taken at known locations (with at least one scan inside each room). We assume that the scans are registered in the same reference frame and, without loss of generality, that the up-vector is $\boldsymbol{v}_{\mathrm{up}} = (0, 0, 1)$. We consider only buildings with planar, vertical walls, but, like Lafarge et al. [18], we drop the more restrictive Manhattan World assumption, and we also allow for non-convex floor-plan room boundaries. 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.

Automatic room segmentation: This step is performed entirely in the 2D projection of 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 an iterative 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 will planes with the planes of the floor and ceiling.

4. Occlusion-aware selection of candidate walls

In the first phase of our method we extract a set of planar patches from the input 3D point clouds that correspond to candidate walls. We first grow planar regions in the 3D point cloud, and to keep only segments which potentially correspond to candidate walls, we select only those regions that 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.

4.1. Low-level segmentation into planar patches

Since our input models are raw and unstructured 3D 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 primarily composed of planar elements is to use planar patches, as done in many previous approaches [21,16,17]. The use of 3D patches, as opposed to, e.g., 2D line projections [20,18], 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 laser range scan. This way, when looking for potential occluders of a patch, we can restrict the search to the patches extracted from that same scan.

We extract patches using a simple region growing process based on normal deviation and plane offset. Like Chauve et al. [16] we have found this scheme to work well; more robust and elaborate methods [25] were not needed in our application. Since a correct choice of the seed points is very important, we start the growing from the points that have most planar and low-noise neighborhoods. The quality of a candidate seed is evaluated by fitting a plane to its k nearest neighbors with the *Least-Median-of-Squares* (LMS) algorithm [4] and by then computing the sum of the residuals.

Please cite this article as: Mura C, et al. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Comput Graph (2014), http://dx.doi.org/10.1016/j.cag.2014.07.005

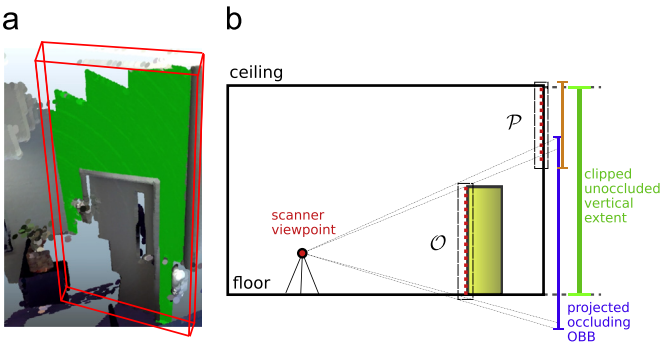


Fig. 3. (a) OBB corresponding to a candidate wall patch. (b) The OBB of an occluding patch is projected onto the plane of a candidate wall patch to recover its unoccluded vertical extent.

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 main 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 also Fig. 3).

4.2. 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 $|\boldsymbol{n} \cdot \boldsymbol{v}_{\text{up}}| < \epsilon$, where the patch normal $\boldsymbol{n}$ is computed using *Principal Component Analysis* (PCA). This effectively rules out other 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 40 cm.

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 3D scans of offices or apartments is problematic. Obstacles located between the camera and the walls (see also Fig. 1, left) can severely limit the amount of structure visible. Taking more scans from additional viewpoints can only partially solve this problem and it cannot be considered a viable solution for static 3D laser range 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* [26] 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). In practice, we scale each OBB by a factor of 1.05 to ensure intersection between the shadow volume of $\mathcal{O}$ and the bounding rectangle of a potentially occluded patch $\mathcal{P}$. The process is illustrated in Fig. 3. If 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 based on the following condition: $h \leq (1-\eta) \cdot h_{\text{rooms}}$. Here η is a small number which we set to 0.05 and h_{rooms} is the distance between the floor and the ceiling. An accurate measure for h_{rooms} is obtained as a byproduct of the robust fitting of wall and ceiling planes as described in Section 6.

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 scan visibility problem. On the other hand, we believe that a more sophisticated analysis would be inadequate for this task due to the imperfect nature of real world input data, which contain large holes and missing parts.

5. Automatic room segmentation

The following steps are carried out entirely in the 2D projection of the *xy*-plane, which leads to a simplified process as compared to a full 3D approach. We first merge the 2D projections of candidate wall patches corresponding to the same wall structure to get a reduced number of *representative lines*. From the intersections of these representative lines we construct a 2D cell complex, i.e., a description of the area of interest consisting of a partitioning of the plane into polygonal faces. By computing the diffusion distances between faces, we get a global measure of affinity that can be used to iteratively cluster this cell complex into individual rooms.

5.1. Computing representative lines

Similar to the approach of Oesau et al. [18], the computed candidate wall patches are projected into 2D to obtain a set of *line segments*, which are then clustered using *mean-shift*. A first directional clustering yields the main orientations of walls; for each such orientation we then perform a 1D mean-shift clustering [27] which identifies all possible offsets of parallel wall segments of that orientation. This way we obtain a set of clusters of line segments $\mathcal{C} = \{\mathcal{C}_0, \ldots, \mathcal{C}_n\}$. 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 $\mathcal{L} = \{l_0, \ldots, l_n\}$ as well as the associated clusters of line segments $\mathcal{C}$, as this helps us to compute the weights of edges in the subsequent cell complex construction step.

5.2. Cell complex construction

From the representative lines $\mathcal{L}$ we build a partition of the plane that models the floor plan of the processed indoor environment. Our plane partitioning is a standard 2D cell complex induced by $\mathcal{L}$, also known in the literature as an *arrangement of lines* [28]. An example of such a cell complex can be seen in Fig. 2(c). 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 $\boldsymbol{e}_{ij}$ between two cell 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 contributing candidate walls (i.e., all line segments of the cluster $\mathcal{C}_k$ associated to $\boldsymbol{e}_{ij}$) and project them onto $\boldsymbol{e}_{ij}$ itself. Let us denote with $\text{cov}(\boldsymbol{e}_{ij})$ the fraction of the extent of $\boldsymbol{e}_{ij}$ that is covered by such projections. We define the weight w_{ij} as follows:

$$w_{ij} = \frac{\text{cov}(\boldsymbol{e}_{ij})}{\text{length}(\boldsymbol{e}_{ij})} \tag{1}$$

The computation of the weight w_{ij} of the edge between two faces f_i and f_j is shown in Fig. 4.

Please cite this article as: Mura C, et al. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Comput Graph (2014), http://dx.doi.org/10.1016/j.cag.2014.07.005

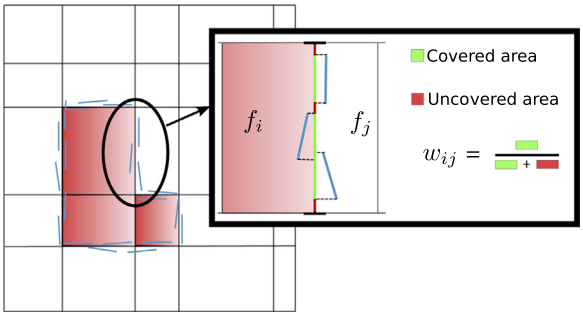


Fig. 4. Computation of the coverage of an edge of the 2D cell complex. The 2D line segments of candidate walls (in light blue) are first projected onto the edge, then the ratio of the occupied length to the total length is assigned as a weight to the segment.

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 iterative clustering.

5.3. Diffusion embedding

Once the cell complex representing the environment is built and the coverage weights w_{ij} of the edges between neighboring faces are determined, we establish a global affinity measure for all pairs of faces. We use the coverage weights to derive a sparse affinity matrix $\mathbf{L}$, similar to a Laplacian, with entries $\mathbf{L}_{ij}$ defined as follows:

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

From matrix $\mathbf{L}$ we define a Markov probability transition matrix as $\mathbf{M} = \mathbf{D}^{-1}\mathbf{L}$, with $\mathbf{D} = \mathrm{diag}(\sum_{j=1}^{n} \mathbf{L}_{ij})$. 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* [29], which are known to be robust against noise [30] and therefore well suited for our task. The diffusion map Φ embeds the faces in a multidimensional Euclidean space. Given a face f_i, its corresponding coordinate in the embedding space is

$$\Phi(f_i) = (\lambda_1^t \phi_1(f_i), \lambda_2^t \phi_2(f_i), \ldots, \lambda_m^t \phi_m(f_i)), \quad (3)$$

where λ_k and ϕ_k are the k-th eigenvalue and eigenvector of $\mathbf{M}$ respectively. Two parameters control this diffusion process: the *diffusion time* t (a measure of smoothness that determines how much the affinities are propagated) and the number m of eigenvectors of $\mathbf{M}$ used in the diffusion map (corresponding to the dimensionality of the embedding). The Euclidean distance in this multidimensional 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 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. This can be clearly understood by examining Fig. 5, where the diffusion distances from a reference face to all the other faces are visualized as a heatmap. The heat diffuses quickly to the faces that belong to the same room, as they are close to the reference face in the Euclidean embedding. On the other hand, heat propagates slower to outer faces (indicated by a green color) and to faces that belong to other rooms, which in the embedding are far from the reference face (indicated by a blue/cyan color).

Visualizing the position of the faces in their embedding space is helpful towards understanding the importance of the diffusion process. However, this is not directly possible when $m > 3$. To give an intuition of how the Euclidean embedding is shaped, separate 3D projections of the embedded points could be considered, one at a time. In Fig. 6 we show the projection of a diffusion embedding generated in our tests onto three of its dimensions, selected manually for the sake of illustration. Each object in the plot corresponds to a face of the 2D cell complex and is colored according to the room it is assigned as a result of the segmentation stage. While this offers only a partial view of the embedding space, it can be still understood how faces belonging to the same room are close in the diffusion embedding. The remaining $m-3$ coordinates, not shown in this plot, make the separation between the faces belonging to different rooms fully accurate.

The properties of the embedding can be analyzed indirectly by considering the matrix of pairwise distances that results from the diffusion process. In this matrix, the i-th row contains the diffusion distances from face f_i to all other faces in the complex. In Fig. 7 a color-coded version of the distance matrix is shown. Note that the order of the rows has been modified so that the rows corresponding to faces in a same room appear one after another. The matrix has a block structure, with the red squares along the diagonal denoting the distances between faces of a same room. The first six squares correspond to rooms, while the last square represents the distances between the outer faces. It is worth noticing how the squares correspond to uniformly low distance values, which confirms that faces within a same room are close in terms of diffusion distances. The diffusion process corrects many errors due to missing data, clutter, and wrong candidate walls, emphasizing the room separations.

These arguments show that our diffusion formulation is very effective in highlighting the similarities between faces within the same room. We also noticed that the process is only slightly influenced by variations in the parameter settings. All the results shown in this paper have been obtained using $t=40$ and $\sigma = 0.0625$. The maximum dimensionality of the embedding (that is, the maximum number of eigenvalues that can be used in the diffusion maps) is bounded from above by the number of faces n_f of the complex. To obtain correct results, larger models require using a number of eigenvalues m that is higher than the value n_f for simpler environments. For this reason, to accommodate for the more complex environments, we have set $m = \min(n_f, 80)$.

Algorithm 1. Iterative clustering algorithm.

Input : Set $\mathcal{F}$ of the faces in the Euclidean embedding
Input : Set $\mathcal{V}$ of the faces containing viewpoints
Output: Set $\mathcal{K}$ of clusters defining the rooms

```
1 ClusterRooms (F, V)
2     K ← ∅
3     K_∞ ← F
4     do
          // create clusters K_R (new room) and
          // K_∞ (unlabeled faces containing f_∞)
5         (K_R, K_∞) ← KMedoids(K_∞)
          // update set of room clusters
6         K ← K ∪ K_R
7     while ∃v ∈ V : v ∈ K_∞
8     return {K}
9 end
```

Please cite this article as: Mura C, et al. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Comput Graph (2014), http://dx.doi.org/10.1016/j.cag.2014.07.005

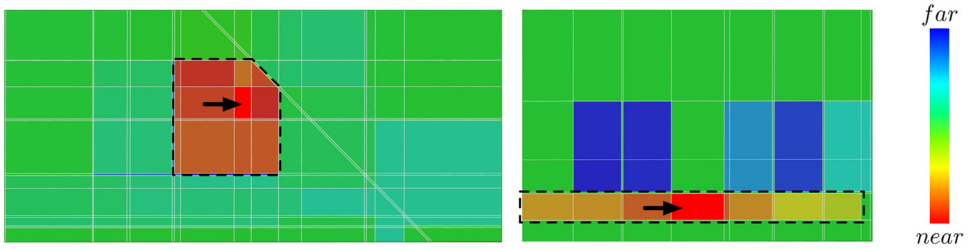


Fig. 5. Heatmap visualization of the diffusion distances for two different datasets. In both cases, the distances from a reference face (marked by the arrow) to all other faces in the complex are shown. Note how the cells in the same room (highlighted by the black dotted line) as the source are closer in terms of diffusion distances than the cells that belong to the outer space or to other rooms (shown in green and blue).

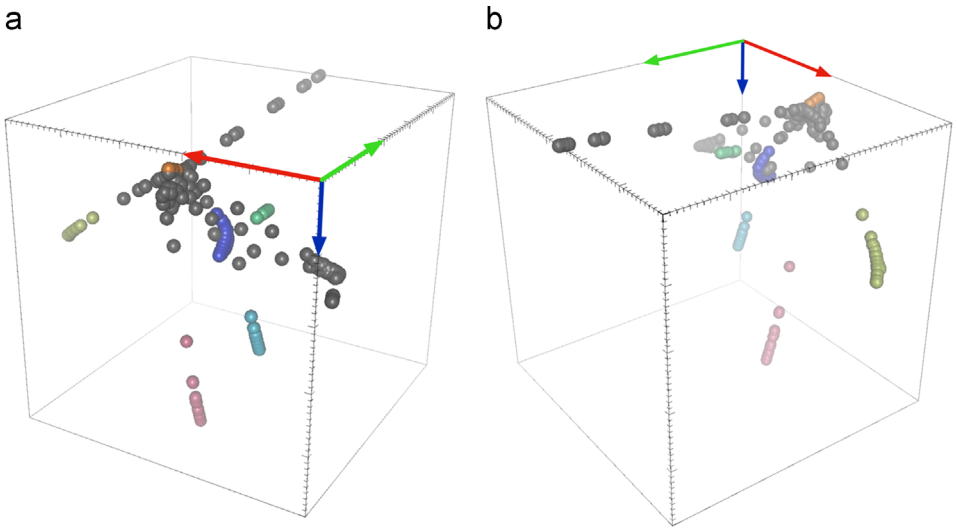


Fig. 6. Scatterplots showing two rotated views of a 3D projection of the Euclidean embedding. Each point in the plot represents a face of the 2D cell complex and is colored according to its final room assignment. The plot refers to the dataset OFFICE 2 (Fig. 2(d)) and the three dimensions have been manually selected for illustration purposes only.

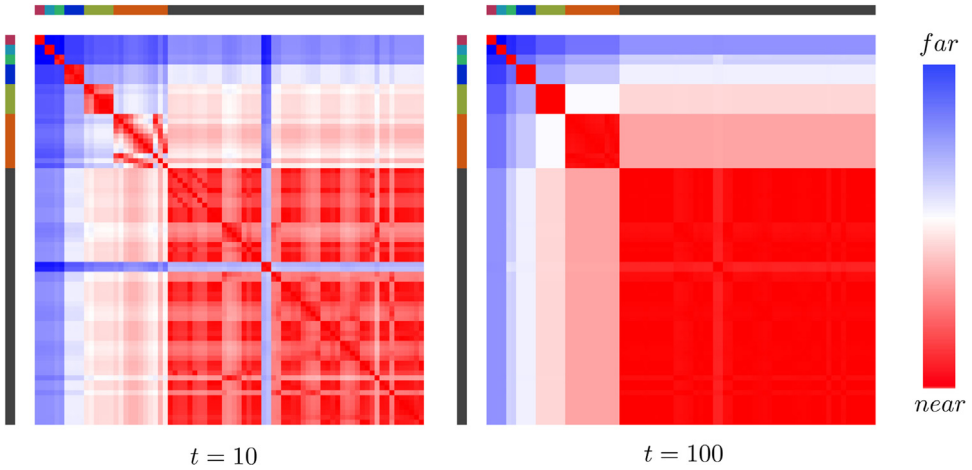


Fig. 7. Color-coded visualization of the pairwise distances for the cell complex of dataset OFFICE 3 (Fig. 13(e)) using two different values of the diffusion time t. The matrices have been rearranged so that rows corresponding to faces in a same cluster are adjacent; the colored bars to the sides of the matrices highlight the correspondence with the detected rooms. Each red square on the diagonal corresponds to the pairwise distances between faces in the same room cluster. Note how faces within the same cluster are close, and how increasing the diffusion time removes noise and emphasizes the structure of the clusters.

5.4. *Iterative clustering*

We use the embedding distances between the faces of the 2D complex described above to partition the set of faces into a number of clusters, each corresponding to a single room of the environment. Since the location at which each input point cloud was taken is known (see Section 3), we exploit the viewpoint information to *automatically* extract the correct number of rooms.

Please cite this article as: Mura C, et al. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Comput Graph (2014), http://dx.doi.org/10.1016/j.cag.2014.07.005

A pseudo-code of the algorithm is given in Algorithm 1. Our partitioning scheme works in a iterative manner, as depicted in Fig. 8. At each step of the algorithm we apply a binary version of the k-medoids clustering [31], that is, setting $k=2$ (line 5 of Algorithm 1). The k-medoids algorithm works by alternating between two steps: computation of the cluster centers (the *medoids*) and assignment of the data points to the best cluster. Differently from the more commonly used k-means algorithm, the cluster centers are restricted to be items of the input dataset.

In particular, given two clusters $\mathcal{K}_0$ and $\mathcal{K}_1$, each medoid k_i is updated according to the rule $k_i = \arg\min_{\mathcal{K}_i} \sum_{k_j \in \mathcal{K}_i} \|k_i - k_j\|^2$. At each step, the k-medoids is initialized by setting as initial medoids the two faces that are farthest away from each other in terms of diffusion distance. Of the two clusters that result from each partitioning step, one always corresponds to a new single room, denoted as $\mathcal{K}_R$ in Algorithm 1. For example, the split labeled 1 in Fig. 8 extracts the room shown in red. For environments with genus ≥ 1 (such as the one shown in Fig. 2), $\mathcal{K}_R$ can correspond to a *hole* (i.e. a set of outside faces disjoint from f_∞); since we assume that each room is scanned from at least one viewpoint (Section 3), we discard $\mathcal{K}_R$ if it does not include a face containing a viewpoint. The second cluster extracted by the k-medoids contains the faces not yet labeled (shown in gray in Fig. 8 and denoted as $\mathcal{K}_\infty$ in Algorithm 1) and always includes the face f_∞. This behavior is linked to the properties of the diffusion embedding, which are analyzed in detail in the next paragraph. As the algorithm proceeds to iteratively cluster $\mathcal{K}_\infty$ (line 5 in Algorithm 1), it creates a split sequence similar to the one shown in the bottom part of Fig. 8.

The iterative partitioning has to be stopped when all rooms have been detected. To do so, we exploit the viewpoint information that comes with the input point clouds. In particular, we can assert that the faces that contain a scan position ($v \in \mathcal{V}$ in Algorithm 1) are certainly inside a room. Conversely, we can assume that each room was scanned from at least one location inside it (see Section 3). Hence, to check if $\mathcal{K}_\infty$ contains yet unlabeled rooms, it is sufficient to check whether it contains a face $v \in \mathcal{V}$ (line 7 in Algorithm 1). Otherwise we can conclude that all rooms have been detected, in which case the partitioning process is terminated. In Fig. 8, this termination criterion is met after the last remaining room has been extracted in the split labeled 10. The advantage of this method is that it stops the subdivision when all rooms have been extracted, without the need

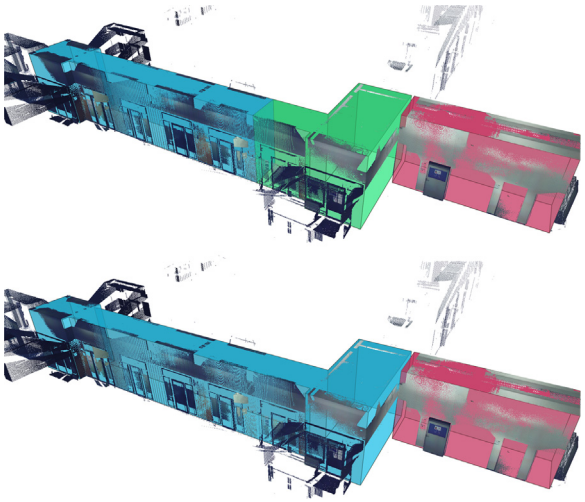


Fig. 9. Large rooms with an elongated shape such as corridors may be incorrectly split into multiple clusters (top). By applying our robust post-processing step, we are able to detect these cases and recover the correct shape of any over-segmented room (bottom).

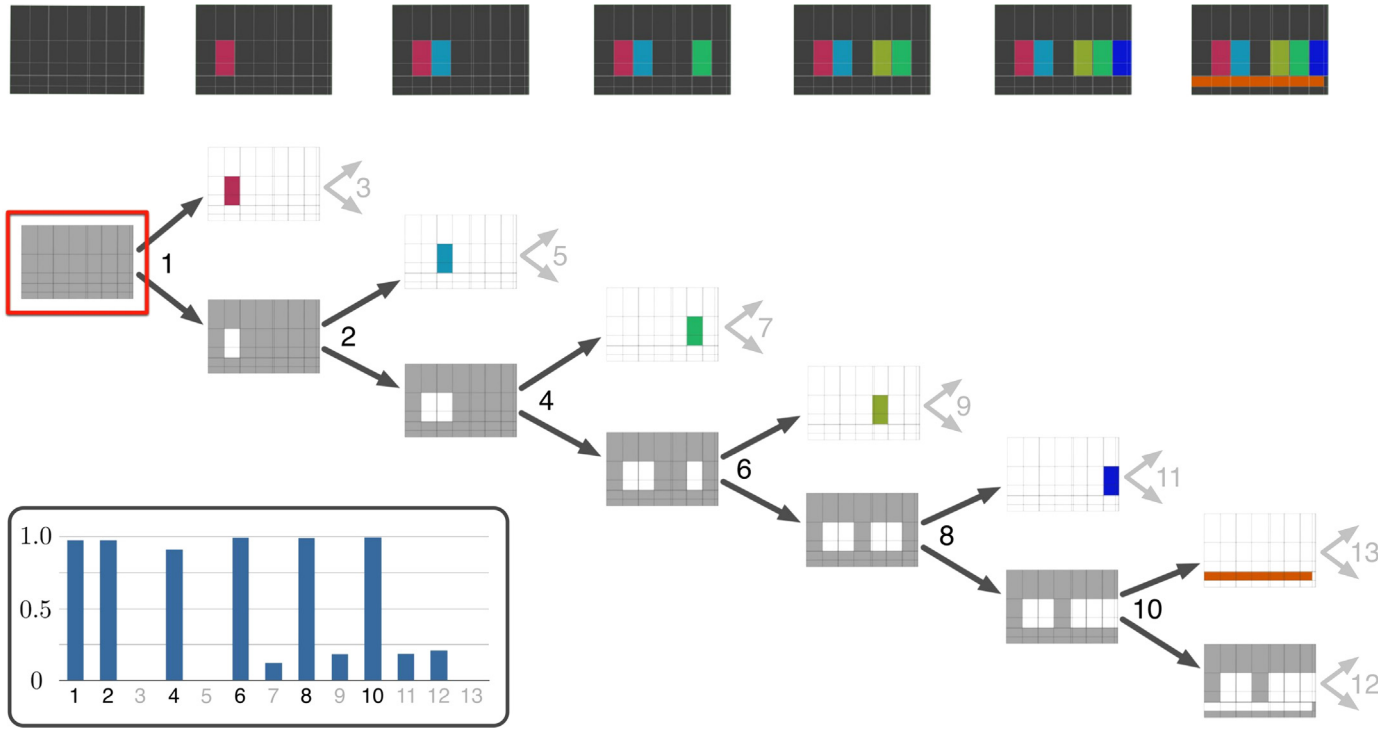


Fig. 8. Illustration of the iterative partitioning process. The input cell complex (framed in red) is iteratively divided into two clusters, generating an (unbalanced) binary tree of splits. At each split (identified by a number), the set of input faces is partitioned into a room cluster and a set of unlabeled faces (shown in gray). The partial partitioning corresponding to each split step can be seen in the top part of the figure. Note that the only splits that are actually performed in our algorithm are the ones numbered in black; the gray numbers denote hypothetical splits that we only examine to analyze the properties of the partitioning. In particular, we show a measure of the quality of each split (including the hypothetical ones) in the bar plot in the inset. The numbers in the x-axis correspond to the identifiers of the splits. It is easy to see how only the splits actually performed have meaningful quality values.

Please cite this article as: Mura C, et al. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Comput Graph (2014), http://dx.doi.org/10.1016/j.cag.2014.07.005

of setting a threshold or specifying the target number of rooms in advance, which is a great advantage over other room clustering alternatives.

Analysis: In the following analysis of our clustering algorithm we will explain the systematic behavior of extracting a single room in each iteration. We identified two factors for this behavior – the specific properties of the diffusion embedding and of the k-medoids clustering – which we examine here in detail.

A relevant property of the embedding is that the faces representing the outside (and including f_∞) are closer than the faces inside any other room. Evidence for this is provided by the results in Fig. 5: the outer faces are closer to the room where the source face is located (indicated by a green color) than the faces in other rooms (indicated by a blue color). This property is even more evident if we consider the visualization of the pairwise distance matrix in Fig. 7. The matrix has been rearranged so that the colored bands to the left of the larger red square on the diagonal represent the distances between the outer faces and the faces of each room.

This happens because the faces that correspond to the inside and those that represent the outside are connected by many edges of the complex (the whole perimeter wall of the building); each edge contributes to putting 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 of a room are much closer in the diffusion embedding to the outside faces than to the faces in any other room.

During each iteration of the clustering process, the first assignment of faces to the medoids will create two *unbalanced* clusters, with medoids k_1 and k_2. This is because the medoid that is closer to f_∞ (let us assume this is k_2) will become the pivot for most of the faces of the complex. As a consequence, k_2 will move towards f_∞. Since every room is closer to the outside than to any other room, every face in the complex will be assigned to the cluster of k_2, with the exception of the faces that belong to the room containing k_1.

The use of the k-medoids algorithm, which requires the cluster centers to be faces of the complex, further increases the robustness and the stability of this partitioning process. This is because a cluster center will either be a face inside a room (and hence the cluster is strongly bound to this particular room) or a face close to f_∞ (possibly, f_∞ itself). We have also experimented with an alternative formulation based on applying the more traditional k-means clustering, where cluster centers can correspond to arbitrary positions in the embedded space. We discovered that k-means is much more dependent on the diffusion parameters (in particular, on the diffusion time t) and requires considerable adjustment of their values to yield correct results.

To further prove the effectiveness of our algorithm we have evaluated the quality of each split *a posteriori*. Let $\mathcal{B} = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$ be the set of edges that separate the two components extracted in a single binary split. We define the *split quality* Q_{split} of $\mathcal{B}$ as follows:

$$Q_{\text{split}}(\mathcal{B}) = \frac{\sum_{i=1}^{n} w(\boldsymbol{e}_i) \cdot \text{length}(\boldsymbol{e}_i)}{\sum_{i=1}^{n} \text{length}(\boldsymbol{e}_i)}, \tag{4}$$

where $w(\boldsymbol{e})$ denotes the coverage of $\boldsymbol{e}$ due to candidate wall segments and follows the definition provided in Eq. (1). This quality function measures how solid the boundary between two adjacent clusters is in the range [0…1], where 1 corresponds to a split along a perfectly solid wall. Note that the edge lengths act here as weights, ensuring that each edge contributes to the split cost proportionally to its importance.

The quality of the splits generated in the sequence of Fig. 8 is shown as a bar plot in the inset of the same picture. The splits actually performed by our algorithm are identified by the black numbers. According to the quality measure, each of those splits generates a boundary that corresponds to an actual wall with high confidence. On the other hand, forcing a further (hypothetical) split when the algorithm would normally terminate would yield a boundary with clearly low quality. These hypothetical splits are denoted by light gray numbers. Since we are taking the viewpoints into account to check for the proper termination depth, no threshold has to be set to avoid such bad splits.

Fig. 10. To test our algorithm we created three synthetic models (SYNTH 1 in the top, SYNTH 2 in the middle and SYNTH 3 in the bottom) of indoor environments, all representing office settings. In these models we selected a set of positions that were used as viewpoints in a virtual 3D scanning process. The stacked pictures on the right show different detail views of the original model (top) and of the corresponding virtually scanned model (bottom).

5.5. Post-processing

The clustering algorithm works well as long as the diffusion process correctly embeds in spatial proximity those faces that belong to the same room. However, this assumption may fail when the input environment contains long corridors. This means that the specified diffusion time was not sufficient to propagate the affinity values between the distant faces of such a structure. Hence corridors may be incorrectly split into several separate clusters (see Fig. 9), as reported in our previous paper [3]. This issue could be solved by increasing the diffusion time t in Eq. (3), which would allow affinities to propagate across the whole extent of the over-segmented room. However, to avoid the need for an interactive and adaptive tuning of this parameter, we have designed a simple yet effective post-processing technique that is based on explicitly checking the goodness of the boundary between adjacent cluster. This is based on the fact that such incorrect splits can be easily

Please cite this article as: Mura C, et al. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Comput Graph (2014), http://dx.doi.org/10.1016/j.cag.2014.07.005

and robustly detected since they do not correspond to an actual wall segment.

We first discover the adjacencies between the detected clusters of 2D cells, that is, we find the pairs of clusters whose boundaries touch in at least one edge. For each such pair, we compute the (possibly multiple) connected components of the shared boundary $\mathcal{B}$, then evaluate the split quality $Q_{\text{split}}(\mathcal{B})$ as defined in Eq. (4).

We merge two adjacent clusters if for each of their boundary segments Q_{split} is lower than a threshold, that we have set to 0.5 in our experiments. This method robustly and correctly merges connected structures like the corridor in Fig. 9. Note that, in practice, the choice of this threshold did not require any fine tuning, as we have observed a clear-cut distinction between erroneous and correct splits. In all our experiments, fake boundaries always had $Q_{\text{split}} < 0.22$, while real ones always had $Q_{\text{split}} > 0.97$ (as shown in the plot of Fig. 8).

6. Model reconstruction

Eventually we reconstruct the 3D polyhedra from the detected room clusters. First of all, the boundary edges of each cluster are extracted, merging adjacent edges that are collinear. Then, the full 3D extent of the walls is recovered. We could simply extrude the 2D boundary edges vertically, but we choose a different approach based on robust statistics to obtain a more accurate estimation of the wall parameters.

For each edge in the boundary of a cluster, we access the projected candidate walls $\mathcal{C}_k$ associated to the edge (see Section 5.1) and we select the points of the corresponding 3D patches. A robust plane fitting algorithm is then applied to these points to extract the final wall planes. We use Iteratively Re-weighted Least Squares (IRLS) [4] instead of the Least Median of Squares (LMS) algorithm, which has been previously used for plane fitting [3]. Like LMS, IRLS is known to be robust with respect to outliers, but is much faster to compute (between 11 and 18× in our experiments, see Table 2). The IRLS method consists in solving a sequence of weighted least squares problems until convergence; robustness is achieved by using a suitable weight function, chosen so that outliers (which correspond to large residuals) have a reduced influence in the estimation. In our experiments, we used the function $w(x) = 1/|x|$ as weight function for the IRLS, which corresponds to minimizing the L_1 norm of the residuals. As shown by the error plots in Fig. 11, the two methods achieve a comparable accuracy.

We use a similar fitting procedure for reconstructing the floor and ceiling planes. 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 $\mathcal{P}_{\text{floor}}$ and $\mathcal{P}_{\text{ceil}}$ with respectively minimum and maximum z values (Section 4.1). To increase the accuracy and robustness of the estimation, we employ the following strategy to fit the final planes. Given $\mathcal{P}_{\text{floor}}$ (respectively $\mathcal{P}_{\text{ceil}}$), we take the horizontal patches whose distance from $\mathcal{P}_{\text{floor}}$ (and $\mathcal{P}_{\text{ceil}}$) is less than a threshold and use their points as support set for an IRLS fit. Note that throughout this process for practical purposes we only consider patches with a diagonal larger than 50 cm.

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 also be seen in Fig. 2(d).

7. Results and discussion

Qualitative evaluation: We have tested our algorithm on 5 different real-world datasets. These datasets were acquired by LIDAR laser range scanning, using a sampling resolution of 24 mm at 10 m. The scanned input models and the resulting room polyhedra are shown in Fig. 13.

Datasets ROOM 1 and ROOM 2 (shown in Fig. 13(a) and (b), respectively), contain two single rooms. However, they represent exceptionally difficult settings for the laser scanner, as the large window fronts result in many reflection artifacts and thus in a huge number of outliers. Nevertheless, our algorithm is able to correctly extract the shape of the rooms.

The other three datasets (OFFICE 1, OFFICE 2 and OFFICE 3) represent different office environments which are composed of corridors with attached rooms. The highly anisotropic shape of the corridor in OFFICE 1 (Fig. 13(c)) is reconstructed correctly and separated from the neighboring room. Thanks to our post-processing step, the over-segmentation problems that affected our original pipeline [3] are solved. OFFICE 3 (Fig. 13(e)) represents a larger environment composed of several rooms, all attached to a central corridor. While OFFICE 3 exhibits a clear arrangement of the rooms, the environment in OFFICE 2 (Fig. 13(d)) has a more complex and irregular structure, lacking a single central corridor and containing some empty space completely surrounded by other rooms. Despite these challenges, all the rooms are correctly detected and reconstructed.

In order to test our algorithm on more general architectural shapes, we generated three synthetic models (SYNTH 1, SYNTH 2 and SYNTH 3, shown in Fig. 10). The models have been generated manually using a 3D modeling software and have been virtually scanned from several positions to simulate the results of 3D laser range 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 with additive Gaussian noise, using $\sigma = 1$ mm to obtain a quality similar to that of the real scans. Moreover, we simulated the scattered outliers outside of the actual architecture

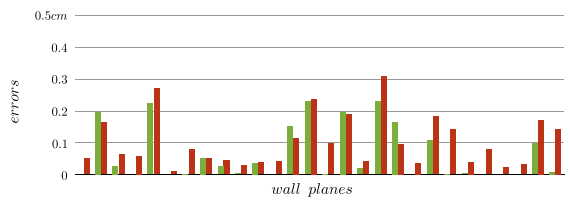


Fig. 11. Accuracy of the final wall planes reconstruction using the new IRLS method (green) and the LMS method (red). The plots refers to dataset SYNTH 1. For each wall plane of the reconstructed model (on the horizontal axis) the maximum distance from the plane to the set of points used for the fit is shown. The accuracy of IRLS is comparable and generally better than the one obtained using LMS.

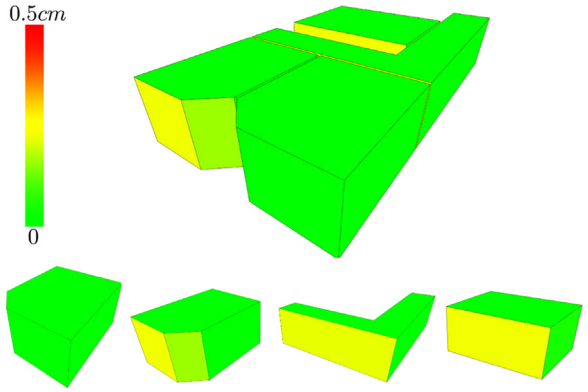


Fig. 12. Quantitative evaluation of the results for dataset SYNTH 1. Color-coded visualization of the maximum fitting error for the visible wall faces. The colors show that our reconstruction achieves very good accuracy, with the largest errors being around 3 mm. A detailed visualization of the error for each reconstructed face is shown in the bar plot in Fig. 11.

Please cite this article as: Mura C, et al. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Comput Graph (2014), http://dx.doi.org/10.1016/j.cag.2014.07.005

due to window reflections (as shown in Fig. 1(b)). To generate a similar effect, we do not discard all the rays that exit through a window. Instead, we generate depth values from a uniform distribution in the range 6 m–8 m (chosen so that the resulting points are outside of the room) for 0.5% of these rays.

The virtually scanned synthetic models and the resulting room polyhedra are shown in Fig. 14. SYNTH 1 (Fig. 14(a)) and SYNTH 2 (Fig. 14(b)) show two cases that do not comply with the *Manhattan World* assumption; in particular, the rooms in SYNTH 2 exhibit a very irregular boundary and a high variation of incident angles between walls. Even more challenging is the dataset SYNTH 3 (Fig. 14(c)), which contains a higher number of rooms, many of which having a non-trivial shape composed of many separate wall segments (see, for instance, the central room colored in blue). Thanks to our pipeline, environments like these can be faithfully reconstructed in a completely automatic way. In such cases, more traditional modeling workflows based on human sketching would require a significant amount of manual work.

To further analyze the robustness of our method we have corrupted model SYNTH 1 with high levels of noise (using

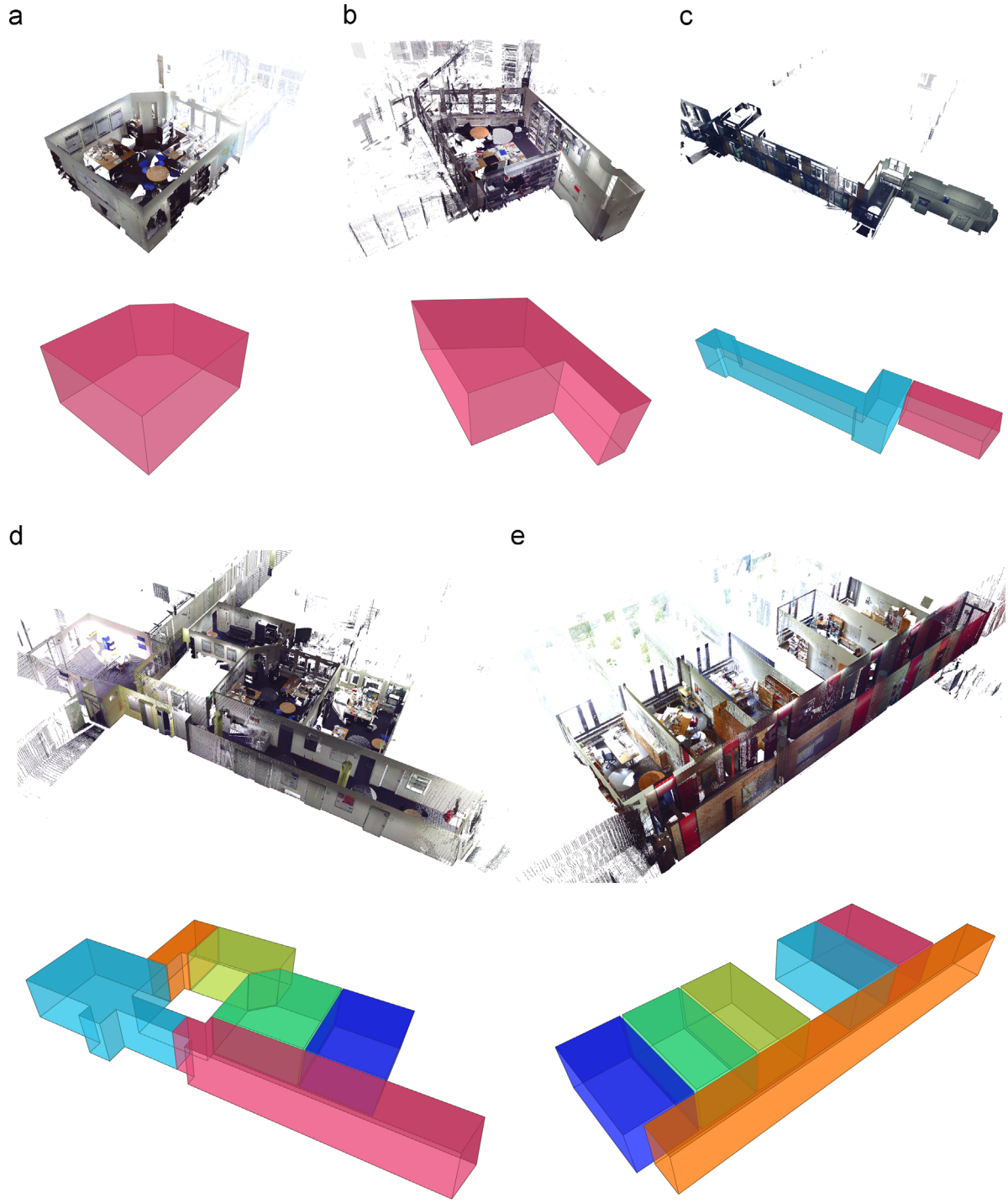


Fig. 13. Reconstruction results for real-world datsets. From top to bottom, left to right: ROOM 1 (a), ROOM 2 (b), OFFICE 1 (c), OFFICE 2 (d) and OFFICE 3 (e). Notice that our algorithm can cope well with large-scale outlier artifacts originating from reflections (e.g. ROOM 2) as well as with complex room arrangements (e.g. OFFICE 2). For clarity of visualization, the input models are shown with the ceiling removed.

Please cite this article as: Mura C, et al. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Comput Graph (2014), http://dx.doi.org/10.1016/j.cag.2014.07.005

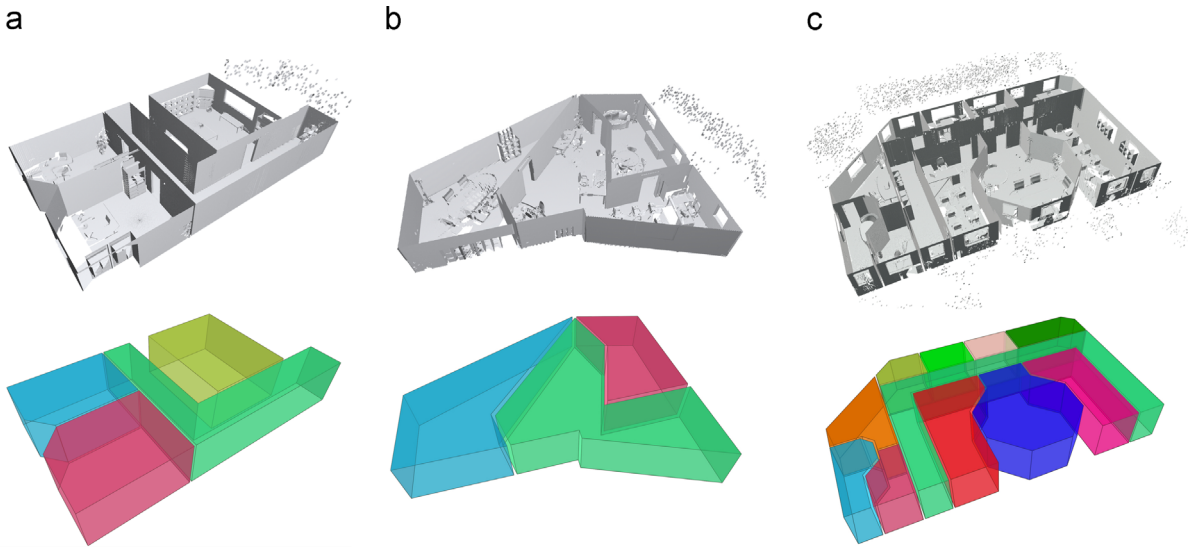


Fig. 14. Reconstruction results for the SYNTH 1 (a), SYNTH 2 (b) and SYNTH 3 (c) datasets. It can be seen how our algorithm can successfully deal with environments that do not satisfy the Manhattan-World assumptions and that contain rooms with complex and irregular boundaries. Notice the presence of scattered points outside of the windows, added to the virtual scans to simulate the artifacts from laser rays hitting reflective surfaces (e.g. glass). As in Fig. 13 we have removed the top part of the input model for the sake of clarity.

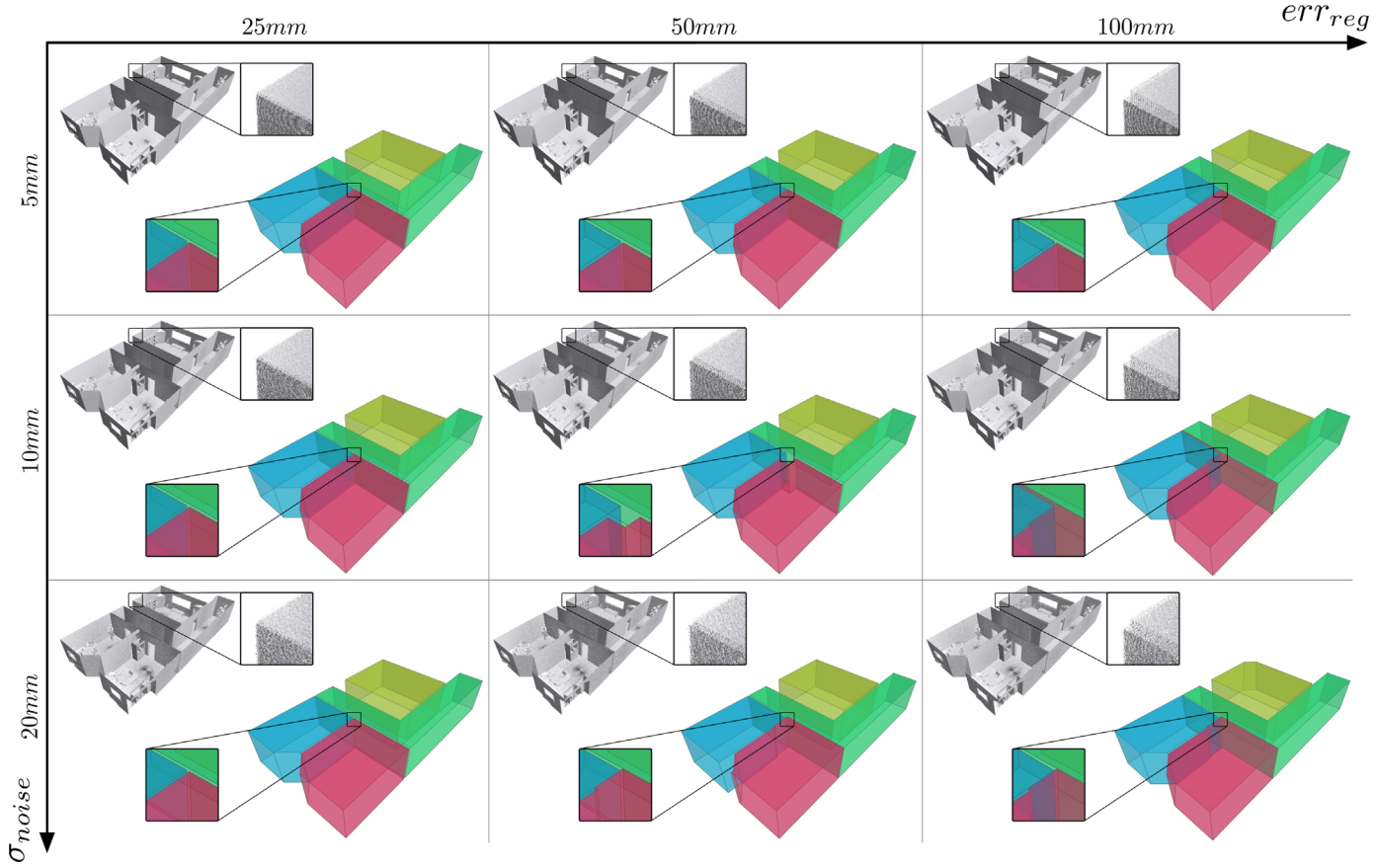


Fig. 15. Reconstruction results for dataset SYNTH 1 corrupted with increasing levels of measurement noise (σ_{noise}, in the vertical axis) and registration error (err_{reg}, in the horizontal axis). Although the combined effect of high noise and registration error (bottom-left part of the grid) leads to artifacts in the reconstructed models, our method is able to correctly detect all the rooms of the environment.

$\sigma = 5, 10, 20$ mm) and introduced an artificial registration error (25, 50, 100 mm). While such levels of degradation do not appear in data obtained by static laser scanning, they are not uncommon when using other acquisition technologies, such as hand-held cameras or cart and vehicle-based systems. As shown in Fig. 15, for a combination of high levels of noise and high alignment error

Please cite this article as: Mura C, et al. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Comput Graph (2014), http://dx.doi.org/10.1016/j.cag.2014.07.005

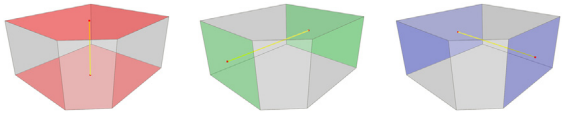
Table 1
Description of our datasets with statistics and running times. Columns *#S*, *#W*, *#R* show, respectively, the number of scans, the number of extracted wall candidates and the number of rooms in the environment. Columns *Ph. 1*, *Ph. 2*, *Ph. 3* show the timings for the three phases of our algorithm (i.e. wall candidate detection, room segmentation and final walls reconstruction).

Model	Pts (M)	*#S*	*#W*	*#R*	*Ph.* 1 (s)	*Ph.* 2 (s)	*Ph.* 3 (s)	Tot. (s)
ROOM 1	5.5	2	21	1	5.2	1.3	0.2	6.7
ROOM 2	8.3	3	25	1	6.5	1.8	0.3	8.6
OFFICE 1	13.8	5	58	2	7.5	2.2	0.8	10.5
OFFICE 2	27.7	10	113	6	11.9	6.0	1.6	19.5
OFFICE 3	38.8	14	134	6	22.8	8.7	3.0	34.4
SYNTH 1	19.4	7	52	4	12.8	5.0	1.6	19.4
SYNTH 2	19.4	7	60	3	11.1	6.2	1.3	18.6
SYNTH 3	69.4	25	202	11	37.0	38.4	10.0	85.3

Table 2
Timings for the two robust fitting schemes employed for the final reconstruction. The use of the iteratively re-weighted least squares (IRLS) provides a significant speedup (up to 18 ×) over an LMS-based fit.

Dataset	LMS (s)	IRLS (s)	Speedup
ROOM 1	2.25	0.17	13.16 ×
ROOM 2	4.0	0.31	12.85 ×
OFFICE 1	12.63	0.78	16.10 ×
OFFICE 2	21.47	1.64	13.07 ×
OFFICE 3	39.36	2.99	13.16 ×
SYNTH 1	27.43	1.55	17.74 ×
SYNTH 2	14.92	1.31	11.40 ×
SYNTH 3	182.61	9.95	18.36 ×

Table 3
Quantitative evaluation for dataset ROOM 1. For each pair of parallel walls we show their real-world distance (*Real*), acquired using a laser measuring device, and their distance in the reconstructed model (*Model*). The last row (*Disp.*) shows, for each wall plane in each pair of parallel walls, the dispersion of the points used for the fit about the plane itself. The discrepancies between real-world and reconstructed distances are proportional to the dispersion values.



Real	3.086 m	6.170 m	5.700 m
Model	3.090 m	6.160 m	5.689 m
Disp.	4.81–4.91 cm	5.71–7.51 cm	3.22–5.11 cm

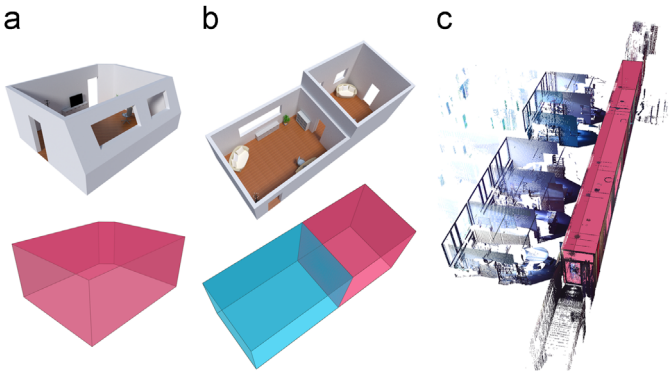


Fig. 16. Some limitations of our reconstruction approach. Environments with slanted walls (a) and different ceiling heights (b) cannot be faithfully reconstructed by our pipeline. We assume that each room is covered by at least one scan taken inside it; the rooms attached to the corridor in (c), which are partially covered by scans taken in the corridor, violate this assumption and are not recognized by our reconstruction method.

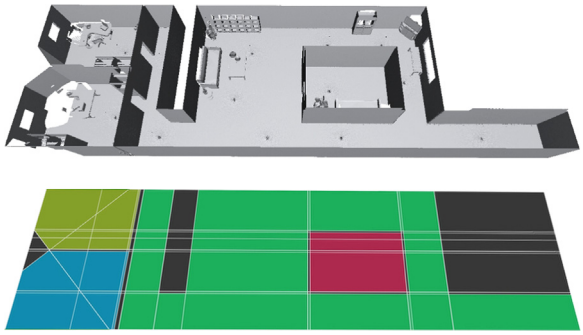


Fig. 17. Results produced by our iterative room partitioning algorithm for the case of an environment with a room completely contained in another room (cluster of red faces in the bottom picture). Note that, in the top picture, the ceiling has been removed to improve visual clarity. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

some reconstruction artifacts appear (e.g., walls separating adjacent rooms are attached to the actual rooms). Even under these challenging conditions our algorithm is able to successfully detect all the rooms of the environment.

Performance: All relevant statistics about the datasets together with timings are listed in Table 1. For each dataset, the table shows the number of points (column 2), the number of scans (column 3), the number of extracted candidate walls (column 4) and the number of rooms of the environment (column 5).

The timings shown in the last column refer to our C++ implementation of the method. All tests were run on an Intel Xeon E5-2670 2.6 GHz processor. We have used OpenMP to parallelize some processing stages, including planar regions growing and weighting of the cell complex. As it can be seen from the timings in Table 1, the method takes less than 90 s even for inputs with almost 70 M points. Without compromising on the quality as compared to LMS (see Fig. 11), the use of our new IRLS-based formulation for the final wall fit significantly reduces the overall computational times. As shown in Table 2, this technique is significantly faster than an LMS scheme, which dominated the computational times in our original formulation [3].

Quantitative evaluation: In addition to the results shown above, we have performed a quantitative evaluation of our method using both synthetic (SYNTH 1) and real-word inputs (ROOM 1).

In the first case, a ground truth model is explicitly given. We have generated two virtual scans of the input model, with and without noise. This way, for every face of the reconstructed noisy model we could evaluate its distance from the real inlier points. As shown by the color-coded visualization in Fig. 12, the accuracy of the reconstruction is very good. By analyzing the more detailed plot in Fig. 11, which shows the errors for all the faces of the model, one can see that the maximum error exceeds 3 mm for a few faces only, while for most of them is well below 1 mm.

Performing a quantitative evaluation for real-world datasets like ROOM 1 is more problematic, as a reliable ground-truth model of the acquired environment is unavailable. We therefore evaluated the distance between pairs of parallel walls of the real environment using a manually-operated laser distance measure device. We then compared the measurements with the distance between the corresponding wall faces in the reconstructed model. The results shown in Table 3 confirm that also on real-world inputs our method is able to achieve very good accuracy levels.

Please cite this article as: Mura C, et al. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Comput Graph (2014), http://dx.doi.org/10.1016/j.cag.2014.07.005

8. Conclusions and future work

We have presented a system for the automatic reconstruction of complex indoor architectural environments that can correctly partition the input model into the appropriate number of separate rooms. Our method is robust against clutter and occlusions, and performs well in real-world scenarios. With respect to the previous solution [3], we have improved the accuracy of the room detection as well as the performance of our method. We also added an in-depth analysis of our algorithm and provided both additional visual and quantitative evaluations of a wider range of more complex models.

Most of the limitations of our approach are directly linked to our starting assumptions – we target only buildings that have planar, vertical walls and horizontal ceilings; structures that cannot be fully explained with a 2D cell complex, such as slanted walls (see Fig. 16(a)), ceilings of different heights (see Fig. 16(b)) or structures like staircases cannot be correctly reconstructed by our pipeline. We focus on the robust extraction of the basic room shapes, and do not attempt to recognize fine architectural details. We rely on the assumption that each room contains at least one scan position; rooms that do not satisfy this condition (Fig. 16(c)) cannot be extracted by the iterative clustering procedure. All environments shown in the results are composed of rooms which share a border with the outside (i.e. the cluster $\mathcal{K}_\infty$ of unlabeled faces, see Section 5.4); additional tests (see Fig. 17) showed that our method can reconstruct an environment containing one room completely contained in another, but more complex cases (e.g. multiple, adjacent rooms surrounded by other rooms) should be investigated in more detail in future work.

We view our algorithm as a first step towards going beyond simple geometric reconstruction to extract semantic information from the input dataset. We plan to combine this with methods for the automatic detection of indoor objects like furniture [32] in an indoor reconstruction pipeline. As a future goal we would like to create a fully parametrized architectural CAD model that could serve as a basis for subsequent editing work of a designer. We plan to incorporate general slanted walls, curved surfaces and other typical architectural structures into our pipeline. Finally, we are interested in extending our virtual scanning prototype so that it realistically models the behavior of time-of-flight scanners.

Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful comments. 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 no. 290227 (DIVA) and Sardinian Regional Authorities.

References

[1] Kim YM, Mitra NJ, Yan D-M, Guibas LJ. Acquiring 3D indoor environments with variability and repetition. ACM Trans Gr 2012;31(6):138 1–11.
[2] Nan L, Xie K, Sharf A. A search-classify approach for cluttered indoor scene understanding. ACM Trans Gr 2012;31(6):137 1–10.
[3] Mura C, Mattausch O, Jaspe AV, Gobbetti E, Pajarola R. Robust reconstruction of interior building structures with multiple rooms under clutter and occlusions. In: Proceedings of computer-aided design and computer graphics; 2013. p. 52–9.
[4] Huber PJ, Ronchetti EM. Robust statistics, Wiley series in probability and statistics. Hoboken, New Jersey: Wiley; 2009.
[5] El-Hakim SF, Boulanger P, Blais F, Beraldin J-A. System for indoor 3D mapping and virtual environments. In: Proceedings of SPIE videometrics V, vol. 3174; 1997. p. 21–35.
[6] Frueh C, Jain S, Zakhor A. Data processing algorithms for generating textured 3D building facade meshes from laser scans and camera images. Int J Comput Vis 2005;61(2):159–84.
[7] Stamos I, Yu G, Wolberg G, Zokai S. 3D modeling using planar segments and mesh elements. In: Proceedings of IEEE symposium on 3D data processing, visualization, and transmission; 2006. p. 599–606.
[8] Dell'Acqua F, Fisher R. Reconstruction of planar surfaces behind occlusions in range images. IEEE Trans Pattern Anal Mach Intell 2002;24(4):569–75.
[9] Sappa AD. Improving segmentation results by studying surface continuity. In: Proceedings of conference on pattern recognition, vol. 2; 2002. p. 929–32.
[10] Turner E, Zakhor A. Floor plan generation and room labeling of indoor environments from laser range data. In: Proceedings of international conference on computer graphics theory and applications; 2014.
[11] Ochmann S, Vock R, Wessel R, Tamke M, Klein R. Automatic generation of structural building descriptions from 3d point cloud scans. In: Proceedings of international conference on computer graphics theory and applications; 2014.
[12] Turner E, Zakhor A. Watertight as-built architectural floor plans generated from laser range data. In: Proceedings of conference on 3D imaging, modeling, processing, visualization and transmission; 2012. p. 316–23.
[13] Turner E, Zakhor A. Watertight planar surface meshing of indoor point-clouds with voxel carving. In: Proceedings of 3DV; 2013. p. 41–8.
[14] Furukawa Y, Curless B, Seitz SM, Szeliski R. Reconstructing building interiors from images. In: Proceedings of international conference on computer vision; 2009. p. 80–7.
[15] Vanegas CA, Aliaga DG, Benes B. Automatic extraction of Manhattan-world building masses from 3D laser range scans. IEEE Trans Vis Comput Gr 2012;18(10):1627–37.
[16] Chauve A-L, Labatut P, Pons J-P. Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data. In: Proceedings of IEEE conference on computer vision and pattern recognition; 2010. p. 1261–8.
[17] Lafarge F, Alliez P. Surface reconstruction through point set structuring. In: Proceedings of eurographics; 2013. p. 5–17.
[18] Oesau S, Lafarge F, Alliez P. Indoor scene reconstruction using primitive-driven space partitioning and graph-cut. In: Proceedings of eurographics workshop on urban data modeling and visualisation; 2013. p. 9–12.
[19] Sanchez V, Zakhor A. Planar 3D modeling of building interiors from point cloud data. In: Proceedings of IEEE international conference on image processing; 2012. p. 1777–80.
[20] Adan A, Huber D. 3D reconstruction of interior wall surfaces under occlusion and clutter. In: Proceedings of symposium on 3D data processing, visualization and transmission; 2011. p. 275–81.
[21] Adan A, Xiong X, Akinci B, Huber D. Automatic creation of semantically rich 3D building models from laser scanner data. Autom Constr 2013;31:325–37.
[22] Arikan M, Schwärzler M, Flöry S, Wimmer M, Maierhofer S. O-Snap: optimization-based snapping for modeling architecture. ACM Trans Gr 2013;32(1):6 1–15.
[23] Sankar A, Seitz S. Capturing indoor scenes with smartphones. In: Proceedings of ACM symposium on user interface software and technology; 2012. p. 403–12.
[24] Kim YM, Dolson J, Sokolsky M, Koltun V, Thrun S. Interactive acquisition of residential floor plans. In: Proceedings of IEEE international conference on robotics and automation; 2012. p. 3055–62.
[25] Schnabel R, Wahl R, Klein R. Efficient RANSAC for point-cloud shape detection. Comput Gr Forum 2007;26(2):214–26.
[26] Crow FC. Shadow algorithms for computer graphics. Proceedings of ACM SIGGRAPH 1977;11(2):242–8.
[27] Furukawa Y, Curless B, Seitz SM, Szeliski R. Manhattan-world stereo. In: Proceedings of IEEE conference on computer vision and pattern recognition; 2009. p. 1422–9.
[28] Edelsbrunner H, O'Rourke J, Seidel R. Constructing arrangements of lines and hyperplanes with applications. SIAM J Comput 1986;15(2):341–63.
[29] Coifman RR, Lafon S. Diffusion maps. Appl Comput Harmon Anal 2006;21(1):5–30.
[30] Lipman Y, Chen X, Daubechies I, Funkhouser T. Symmetry factored embedding and distance. ACM Trans Gr 2010;29(4):103 1–12.
[31] Kaufman L, Rousseeuw P. Clustering by means of medoids. In: Yadolah D, editor. Statistical data analysis based on the L1-norm and related methods. Amsterdam: North-Holland; 1987.
[32] Mattausch O, Panozzo D, Mura C, Sorkine O, Pajarola R. Object detection and classification from large-scale cluttered indoor scans. Comput Gr Forum 2014;33(2):11–21.

Please cite this article as: Mura C, et al. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Comput Graph (2014), http://dx.doi.org/10.1016/j.cag.2014.07.005