

# IMPROVING AUTOMATIC RECONSTRUCTION OF INTERIOR WALLS FROM POINT CLOUD DATA

E. Maset\*, L. Magri, and A. Fusiello

DPIA – University of Udine, Via delle Scienze, 206 – 33100 Udine, Italy  
maset.eleonora@spes.uniud.it, magri.luca.l@gmail.com, andrea.fusiello@uniud.it

Commission IV, WG IV/5

**KEY WORDS:** Indoor modelling, Wall reconstruction, Point cloud, Line fitting, Min-hashed J-Linkage

## ABSTRACT:

In this paper we deal with the automatic reconstruction of interior walls from point clouds, an active research topic with several practical applications. We propose an improved version of the method presented in (Magri and Fusiello, 2018), where the overall structure of the environment is extracted by fitting lines to the main building features, using Min-hashed J-Linkage as a multi-model fitting technique. Our variation has the merit of producing more accurate results, both in terms of wall reconstruction and room segmentation, and greatly reducing the need for user-defined thresholds.

## 1. INTRODUCTION

Nowadays we are witnessing an increasing in the demand of updated and detailed 3D models of indoor environments, that find application in several fields such as navigation, emergency response, building maintenance and monitoring (Zlatanova et al., 2013). In particular, Building Information Models (BIMs) require to represent a facility in a semantically rich manner, combining accurate geometric information with object identification and adjacency relationships among the elements (Tang et al., 2010).

While it is easy to retrieve a BIM from a CAD-based model of a structure's design, updating this model or constructing it from scratch according to the as-built conditions of a building is a challenging task (Volk et al., 2014). On the other hand, the growing availability of instruments such as terrestrial or mobile laser scanners allows to obtain high quality 3D data, that represent the basis for the as-built indoor models reconstruction. For these reasons, generating 3D indoor models from point clouds has become an active research topic in recent years, leading to the development of algorithms that are able to reduce time and cost of the manual processing (Wang et al., 2017).

Among the methods proposed in the literature, we focus on the one presented in (Magri and Fusiello, 2018), that is able to automatically reconstruct the interior wall of a building, relying on a multi-model fitting technique known as *J-Linkage* (Toldo and Fusiello, 2008). More in detail, the method operates on the 2D floor plane, where 3D points are projected, and extracts the overall structure of the environment by fitting line segments to the main building features. The aim of this work is to propose an improved version of (Magri and Fusiello, 2018), taking into account also the points distribution along the vertical axis, avoiding the usage of several manually tuned thresholds and leveraging on a cell-complex subdivision of the plane to enhance the topological correctness of the result.

The paper is organized as follows. In the next section, recent algorithms for the automatic generation of indoor models are reviewed. Section 3 describes in detail the proposed

method, while Sec. 4 shows the results obtained on the dataset of the ISPRS benchmark on indoor modelling (Khoshelham et al., 2017, Khoshelham et al., 2018). Finally, Sec. 5 draws the conclusion.

## 2. RELATED WORK

In the last years several methods have been proposed for the automated creation of indoor models from 3D data generated from terrestrial or mobile laser scanners, RGB-D cameras or multi-view stereo.

The main task of indoor modelling is the geometric reconstruction of the primary surfaces of the building, namely floors, ceilings and walls. The core of any method for indoor scene reconstruction is represented by the robust extraction of multiple geometric primitives from noisy and outlier-contaminated measurements, deriving from the presence of furniture causing clutter and complex walls arrangements that determine occlusions. In this context, three different classes of algorithms can be identified, according to the searched primitives (Previtali et al., 2018): linear, planar or volumetric.

The method presented in (Magri and Fusiello, 2018) falls in the first category, where the overall structure of the environment is extracted by fitting lines to the main building features, using *Min-hashed J-Linkage* as a multi-model fitting technique. RANSAC is instead exploited in (Wang et al., 2017) to identify linear primitives, and the result is refined using a 1D mean shift clustering algorithm.

Many works makes use of RANSAC to fit multiple planes in space. Among them, we recall (Previtali et al., 2014) and (Previtali et al., 2018), that combine a plane fitting step to retrieve floors, ceilings and walls with a further analysis based on a *ray-tracing labelling* that allows to identify and classify openings and occlusions. The method presented in (Budroni and Böhm, 2010) finds instead the main planar structures by sweeping a plane along the vertical and the two main horizontal directions. (Oesau et al., 2016) developed a method that performs planar

\*Corresponding author

shape detection through region growing, interleaved with a regularization step that enforces regularities such as parallel, coplanar and orthogonal relationships among the primitive shapes, improving in this way robustness against noisy data.

The use of volumetric primitives has not been extensively investigated yet. We can cite the method by (Xiao and Furukawa, 2014), that proposed a 3D Inverse Constructive Solid Geometry, using the results obtained for 2D sections to generate candidate primitives. (Khoshelham and Díaz-Vilariño, 2014) implemented a grammar-based algorithm that generates a 3D parametric model by repeatedly placing and merging cuboids into spaces enclosed by points. The same method is applied in (Tran et al., 2017) to automatically extract topological relations between indoor spaces for indoor navigation.

An important step in the generation of indoor 3D models is room level segmentation. (Bormann et al., 2016) presents a comprehensive review of methods suited for segmenting 2D maps into separate entities. The problem is often formulated in terms of a cell labeling process, where the cells are a subdivision of the floor plane, and solved using an energy based algorithm. (Wang et al., 2017) and (Ambruş et al., 2017), e.g., minimize the energy function through a graph-cut algorithm. In (Ikehata et al., 2015) room segmentation is formulated as a clustering problem and solved using the k-medoids algorithm.

Recently, some algorithms have been proposed in the literature also to detect and reconstruct doors and windows, usually relying on both 3D points and RGB images. In (Díaz-Vilariño et al., 2015), e.g., the geometry of the walls is used to generate orthoimages, to which the Generalized Hough Transform is applied to detect closed doors. The method developed by (Michailidis and Pajarola, 2017), on the contrary, extracts windows and doors only from point clouds, constructing a cell-complex representation for each wall, which is then used for the wall object segmentation via a graph-cut method. However, identifying structural wall openings such as windows and doors in point clouds is still a difficult task and represents a hot research topic.

### 3. PROPOSED METHOD

In the next sections we will present the proposed method, that can be divided into three main steps. At first, the 3D point cloud is reduced to a set of planar points that are subsequently organized into linear structures (Sec. 3.1). The detected set of lines is then improved following the procedure described in Sec. 3.2 and the points belonging to the retained lines are arranged into segments, corresponding to the reconstructed walls. Finally, the space enclosed by the detected walls is segmented into rooms (Sec. 3.3).

Please note that, unlike many state-of-the art procedures (e.g. (Turner and Zakhor, 2014, Previtali et al., 2018)), our method requires neither the knowledge of the scanner positions nor of the sensor trajectory. It can be therefore directly applied to datasets acquired by terrestrial laser scanners or mobile mapping systems, as well as to point clouds generated from images through multi-view stereo algorithms.

#### 3.1 Wall Samples Extraction and Clustering

The first part of the method closely follows the procedure proposed in (Magri and Fusiello, 2018), which aims at reducing the 3D point cloud to a set of sampled planar points, referred

to as *wall samples*, enriched with information about their local orientations.

Similarly to (Turner and Zakhor, 2012), as a preliminary step, an histogram of point heights is computed and the floor/ceiling bins are identified as the bottom-most and top-most local maxima. Planes are fitted via Iterative Reweighted Least Squares on the points belonging to these bins, the corresponding inliers are labeled as floor and ceiling respectively and are then discarded from subsequent analysis (and the histograms are updated accordingly). The remaining 3D points are projected onto the floor plane which is uniformly discretized in a grid of ground-cells, whose width should be chosen according to the desired resolution of the final model. In our implementation the cell size is set to 7.5 cm.

If enough points (default is 20) fall inside a ground-cell, their median position is taken as a *wall sample* representative for that cell and a normal vector is locally estimated using Principal Component Analysis, together with a confidence measure of the co-planarity of the corresponding points. Only normals with a confidence higher than a certain threshold are retained.

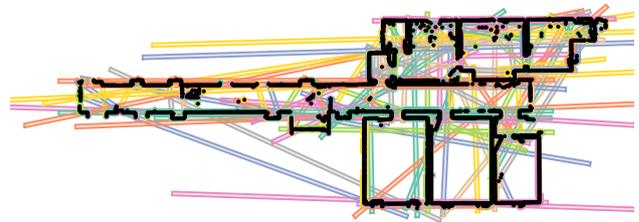


Figure 1. Lines extracted by Min-hashed J-Linkage, depicted with their inlier band.

The wall samples are then organized into linear structures, taking into account also their normal vectors, in order to better distinguish points that lie on the same wall and to gain robustness against clutter. In particular, wall samples are grouped into lines exploiting the Min-hashed J-Linkage algorithm, a multi-model fitting technique that is able to overcome some limitations of RANSAC and Hough Transform (Toldo and Fusiello, 2008).

At high-level, J-Linkage implements a two steps *first-represent-then-clusterize* scheme: at first, data are represented by the preferences they grant to a pool of provisional model hypotheses, then a greedy bottom-up agglomerative clustering is performed to obtain a partition of the data merging points with similar preferences according to the *Jaccard* distance (Jaccard, 1901).

The concept of preferences can be formulated as follows. Let  $X$  be the set of  $n$  data points and  $\text{err}: X \times H \rightarrow \mathbb{R}$  an error function to measure residuals between data and models (in our case, lines). Moreover, let  $\epsilon$  be an inlier threshold provided as input. The method starts by generating a set of random provisional models  $H = \{h_1, \dots, h_m\}$  by drawing  $m$  subsets of data points with the minimum cardinality necessary to instantiate a model. Then a  $n \times m$  binary matrix  $P$  is built by defining its  $(i, j)$ -th entry as

$$P(i, j) = \begin{cases} 1 & \text{if } \text{err}(x_i, h_j) \leq \epsilon \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Each row  $P_i$  can be easily identified with the *preference set*  $\text{PS}(x_i)$  of a given point  $x_i$ , i.e., the subset of structures in  $H$

that support that point. The rationale is that points belonging to the same model will have similar preference sets, and therefore can be clustered in the conceptual space  $\{0, 1\}^m$  to reveal the structures hidden in the data.

The preference set of a subset  $Y \subseteq X$  is composed by the models that fit all the points in  $Y$ :

$$PS(Y) = \bigcap_{x \in Y} PS(x). \quad (2)$$

The clustering algorithm proceeds in a bottom-up manner. First, every data is put in its own cluster. The distance between clusters is computed as the Jaccard distance between the respective preference sets. The Jaccard distance between two sets  $U, V$  is defined as  $1 - J(U, V)$ , where

$$J(U, V) = \frac{|U \cap V|}{|U \cup V|}, \quad (3)$$

denotes the Jaccard similarity. Starting from singletons, each sweep of the algorithm merges the two clusters with the smallest Jaccard distance, until all the preference sets of clusters are disjoint. The parameters of the returned structures are then estimated by least squares fitting on each cluster of points. Please note that this preference approach is robust to outliers, that can be recognized as observations whose preferences deviate significantly from the rest of the data, and tend to emerge as micro-clusters, that can be removed *a posteriori*.

To speed up the computation of the Jaccard distance, (Magri and Fusiello, 2018) proposed the use of min-Hash, a local sensitive hashing procedure. Min-Hash allows to approximate the Jaccard similarity between two preference set, greatly boosting the efficiency, without sacrificing the accuracy and the robustness of J-Linkage.

An example of the result provided by Min-hashed J-Linkage is illustrated in Fig. 1. As previously mentioned, this stage requires the definition of the inlier threshold  $\epsilon$  (set to 15 cm in the experiments): a point belongs to the supporting set of a fitting line if its residual is below this threshold.

From this point we depart from (Magri and Fusiello, 2018) and propose the procedure described in the following.

### 3.2 Line Refinement and Segment Generation

The lines yielded by Min-hashed J-Linkage are associated to wall samples that come not necessarily from actual wall, but also from furniture and clutter, which can negatively affect the results. For this reason, we exploit the point heights histogram associated to each wall sample to cluster them into three classes corresponding to uniform (U), low-thicker (L) and high-thicker (H) histograms (see Fig. 2), and prune the J-Linkage result.

The histogram is first binarized (empty cell  $\rightarrow 0$ ) and clustering is subsequently performed using *Hamming* distance and *k-means* with three seeds corresponding to the classes archetypes (result is shown in Fig. 2). Then, we deem as outliers those lines that are supported by a majority of L or H samples. The remaining lines are scrutinized for outliers with a *inward testing* procedure (Davies and Gather, 1993): samples that are classified as L or H are removed one at a time and the line is fitted on the remaining ones. If the residual of the removed sample is

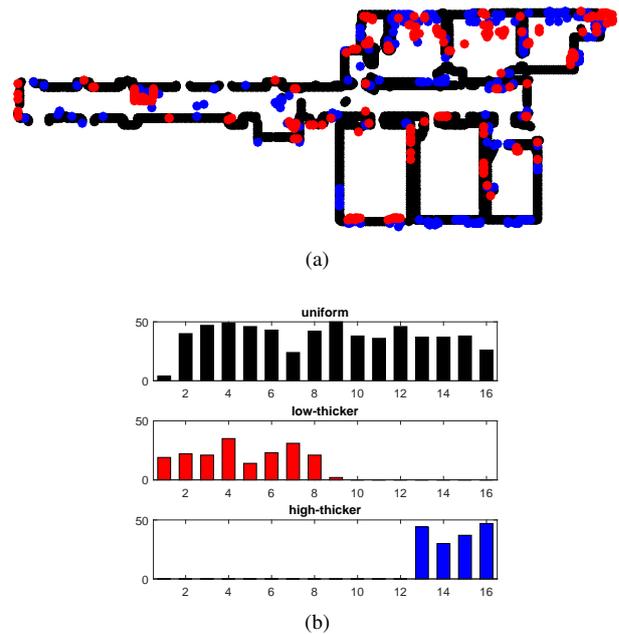


Figure 2. Classification of the wall samples (a) into U/L/H with examples of histograms (b).

greater than  $\epsilon$ , it is considered as an outlier and removed from the supporting set of that line.

To further reject lines that do not correspond to actual walls, we discard the ones that do not conform to the so called *Manhattan World* assumption, according to which walls are positioned along two main orthogonal directions (Coughlan and Yuille, 1999). Following (Magri and Fusiello, 2018), lines are grouped by fitting vanishing points with J-Linkage and retaining only the lines that belong to the two clusters with the larger cardinality, as shown in Fig. 3. Aligning the dominant directions with the axes greatly simplifies subsequent computations.

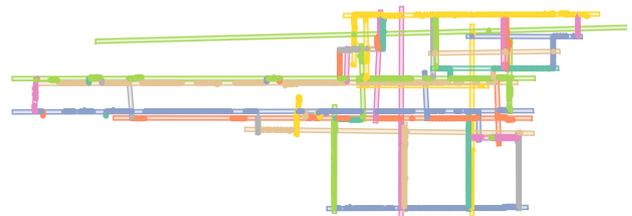


Figure 3. Results of the vanishing point clustering. Lines that do not conform to the dominant orientations have been discarded.

To finally group wall samples into segments, it is convenient to exploit the subdivision of the plane induced by the arrangement of the detected lines (see Fig. 4), formally defined as a 2D *cell-complex* and stored in a *Doubly Connected Edge List* (DCEL), a structure commonly used in computational geometry that consists of vertices, edges, and faces (Edelsbrunner et al., 1986). To each edge of the DCEL we assign the wall samples that i) are included in the supporting set of the line to which the edge belongs and ii) whose coordinates fall in the interval defined by the edge's endpoints. Then we project these wall samples onto the corresponding edge.

In this stage, in order to fill gaps caused by the presence of furniture and clutter, we consider the outlier lines previously discarded (because mainly supported by H or L samples) that are

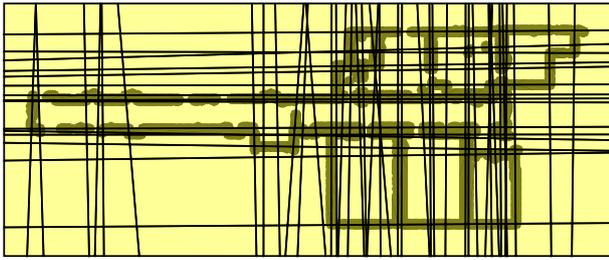


Figure 4. Cell-complex determined by the line arrangement.

aligned with the dominant orientations. Their samples are then projected onto the closest edge along the direction orthogonal to the line itself, within a threshold of 60 cm.

Following the linear order along an edge, a new segment is instantiated when the distance between two consecutive points exceeds a threshold, computed for each line  $l$  as follows:

$$t(l) = \bar{d} + 3.5 \operatorname{med}_i |d_i - \bar{d}|$$

where  $d_i$  is the distance between consecutive points belonging to the supporting set of the line  $l$  and  $\bar{d} = \operatorname{med}_i(d_i)$ .

In this way, we constructed a hierarchy: point  $\rightarrow$  segment  $\rightarrow$  edge  $\rightarrow$  line where all the maps are injective. As a result, thanks to the subdivision of the wall segments induced by the cell-complex, the topological correctness of the segments is guaranteed, i.e., there are no intersections other than at the endpoints (like T-junctions or overlaps).

### 3.3 Space Segmentation

To enrich the obtained model with semantic information useful in subsequent analyses that can, e.g., facilitate the process of creating as-built BIMs, we propose a space segmentation method to first distinguish between indoor and outdoor, and to subsequently clusterize the faces of the cell-complex into rooms.

More in detail, our procedure computes the connected components of the dual graph of the cell-complex, where vertices correspond to faces and edges connect adjacent faces. Adjacency is defined in two different ways, according to the task.

The first task is to distinguish between internal and external faces. For every line we identify its first and last segments and consider as *separators* all the edges in between. Then we define two faces as adjacent if they do not share a separator. The connected component of the graph that includes the faces adjacent to the bounding box is labelled as external. Figure 5 shows the outcome of this stage for the working example.

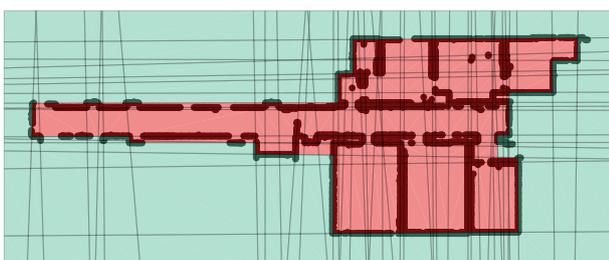


Figure 5. Identification of indoor and outdoor faces.

A different definition of separator is used instead to identify rooms among the indoor cells. Empirical tests demonstrated that it is not sufficient to simply label as *separators* the edges occupied by a wall segment. In fact, gaps between segments belonging to the same wall, that have not been filled in the preceding stages, can negatively influence the room segmentation, resulting in a low number of connected components and consequently in an under-segmentation. For this reason, we analyze the gaps between *wall* segments along a line and create a *gap* segment (represented in red in Fig.6) for those that are shorter than a specified threshold (set to 1.2 m in the experiments). Gap segments are only used in the face clustering step and will not appear in the final model. Next, an edge is labeled as *separator* if the length of *wall* and *gap* segments covering the edge is greater than 10% of its total length. The adjacency matrix of the graph is consequently constructed following the previously described rule and each connected component is considered a separated room, as shown in Fig.7.

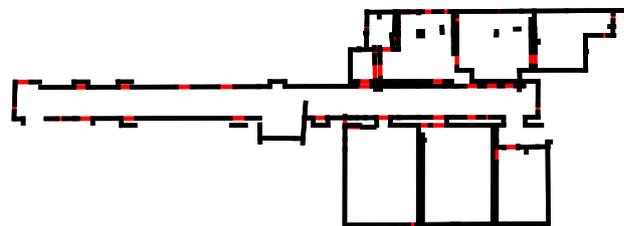


Figure 6. Wall segments (in black) and gap segments (in red) used in the face clustering step.

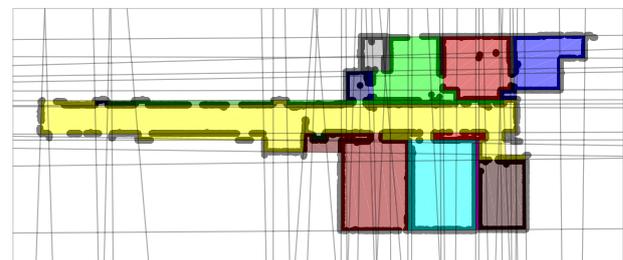


Figure 7. Result of the space segmentation algorithm: every room correspond to a different colour (this figure is best viewed in colours).

The outcome of the complete workflow presented in these sections is illustrated in Fig. 8(a).

## 4. RESULTS AND DISCUSSION

We tested the proposed method on the dataset of the ISPRS benchmark on indoor modelling (Khoshelham et al., 2017). The dataset consists of five point clouds acquired by different sensors in various scenarios. It represents therefore a challenging test that can highlight advantages and limitations of the developed algorithm.

In Fig. 8(a) we show the floor plan generated with the proposed workflow for the point cloud **TUB1**. Comparing the result with the one obtained by (Magri and Fusiello, 2018), illustrated in Fig. 8(b), one can notice the higher accuracy and completeness reached by the novel method. Moreover, the space clustering algorithm implemented in (Magri and Fusiello, 2018) produced an over-segmentation, resulting in an incorrect splitting of the

rooms into several small spaces. This is particularly evident for the long hallway.

Similar considerations can be applied also to the results obtained for the dataset **TUB2**, represented in Figs. 9 and 10. The majority of the walls were reconstructed by our method, and the room segmentation appears closer to reality than the outcome of (Magri and Fusiello, 2018). Please note that in this case the two floors were processed separately, manually specifying the height of the ceiling and the floor for the first and the second level, respectively.

The datasets **Fire Brigade** and **UVigo** present a high level of clutter, gaps caused by occlusions and curtain walls. Identifying and reconstructing the walls in these scenarios is a difficult task and, as shown by Fig. 11 and 12, the obtained results are not satisfactory. Although an improvement can be noticed in the floor plan generated with the method proposed in this paper, clutter led to the creation of wall segments that do not correspond to actual walls. Moreover, our method was not able to reconstruct a curtain wall of the point cloud **Fire Brigade** (see Fig. 11(a)). In fact, the samples corresponding to that wall were labeled as L or H according to the associated height histograms, as described in Sec. 3.2, and the fitting line was thus discarded for the subsequent processing steps.

Finally, Fig. 13 shows the results for point cloud **UoM**. The presence of several pieces of furniture makes the dataset challenging. Nevertheless, in this case the proposed method was able to cope with the significant level of clutter, generating a more accurate floor plan than the competing algorithm.

Overall, we are aware that the obtained results are far from being perfect. However, it should be underlined that the proposed algorithm is completely automatic, without user interaction, and independent of the acquiring system. The use of several manually-tuned thresholds have been avoided and for the remaining ones the same values were adopted for all the point clouds, showing robustness to the change of the data characteristics. The outcome of our method could therefore ease the expensive and time consuming process of manually generating indoor models.

## 5. CONCLUSION AND FUTURE WORK

In this paper we proposed a variation of the method presented in (Magri and Fusiello, 2018), that has the merit of producing more accurate results (both in terms of detected walls and room segmentation), and reducing the need for user-defined thresholds. With respect to this issue we have been following these principles, in cascade: i) to avoid free parameters at all; ii) to make them data-dependent; iii) to make user-specified parameters intelligible and subject to an educated guess. This guarantees the processing to be completely automatic in the majority of cases, as demonstrated by the experiments carried out on challenging datasets, acquired in different scenarios.

The results produced by our method could serve as a starting point for more accurate analyses that could improve the architectural model. Relying on the room segmentation generated by our algorithm and the detected openings on the walls, as a future work we are planning to add a more elaborated 3D analysis to locate doors and windows. Moreover, we would like to investigate other segmentation algorithms that could refine the semantic and topological information associated to the model.

## ACKNOWLEDGMENTS

The authors would like to thank 3Dflow srl for the support.

## REFERENCES

- Ambruş, R., Claiaci, S. and Wendt, A., 2017. Automatic room segmentation from unstructured 3-d data of indoor environments. *IEEE Robotics and Automation Letters* 2(2), pp. 749–756.
- Bormann, R., Jordan, F., Li, W., Hampp, J. and Hägele, M., 2016. Room segmentation: Survey, implementation, and analysis. In: 2016 IEEE international conference on robotics and automation (ICRA), IEEE, pp. 1019–1026.
- Budroni, A. and Böhm, J., 2010. Automatic 3d modelling of indoor manhattan-world scenes from laser data. *ISPRS Archives* 38(5), pp. 115–120.
- Coughlan, J. M. and Yuille, A. L., 1999. Manhattan world: Compass direction from a single image by bayesian inference. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Vol. 2, IEEE, pp. 941–947.
- Davies, L. and Gather, U., 1993. The identification of multiple outliers. *Journal of the American Statistical Association* 88(423), pp. 782–792.
- Díaz-Vilariño, L., Khoshelham, K., Martínez-Sánchez, J. and Arias, P., 2015. 3d modeling of building indoor spaces and closed doors from imagery and point clouds. *Sensors* 15(2), pp. 3491–3512.
- Edelsbrunner, H., O'Rourke, J. and Seidel, R., 1986. Constructing arrangements of lines and hyperplanes with applications. *SIAM Journal on Computing* 15(2), pp. 341–363.
- Ikehata, S., Yang, H. and Furukawa, Y., 2015. Structured indoor modeling. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1323–1331.
- Jaccard, P., 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles* 37, pp. 547–579.
- Khoshelham, K. and Díaz-Vilariño, L., 2014. 3d modelling of interior spaces: Learning the language of indoor architecture. *ISPRS Archives* 40(5), pp. 321–326.
- Khoshelham, K., Díaz-Vilariño, L., Peter, M., Kang, Z. and Acharya, D., 2017. The ISPRS benchmark on indoor modelling. *ISPRS Archives* 42(2/W7), pp. 367–372.
- Khoshelham, K., Tran, H., Díaz-Vilariño, L., Peter, M., Kang, Z. and Acharya, D., 2018. An evaluation framework for benchmarking indoor modelling methods. *ISPRS Archives* 42(4), pp. 297–302.
- Magri, L. and Fusiello, A., 2018. Reconstruction of interior walls from point cloud data with min-hashed J-linkage. In: 2018 International Conference on 3D Vision (3DV), IEEE, pp. 131–139.
- Michailidis, G.-T. and Pajarola, R., 2017. Bayesian graph-cut optimization for wall surfaces reconstruction in indoor environments. *The Visual Computer* 33(10), pp. 1347–1355.

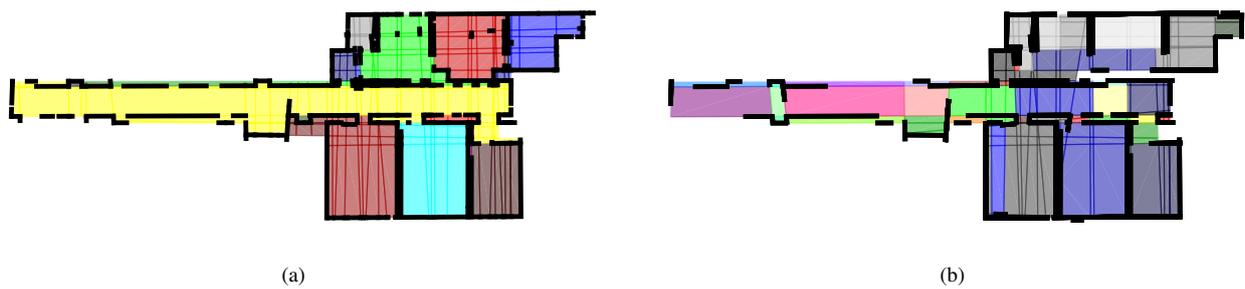


Figure 8. Dataset **TUB1**: floor plans obtained with (a) the proposed method and (b) the method by (Magri and Fusiello, 2018). This figure is best viewed in colours.

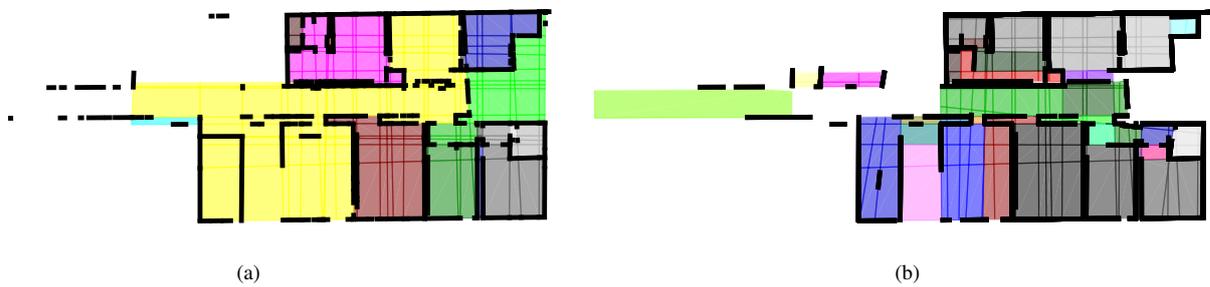


Figure 9. Dataset **TUB2** (first floor): floor plans obtained with (a) the proposed method and (b) the method by (Magri and Fusiello, 2018). This figure is best viewed in colours.

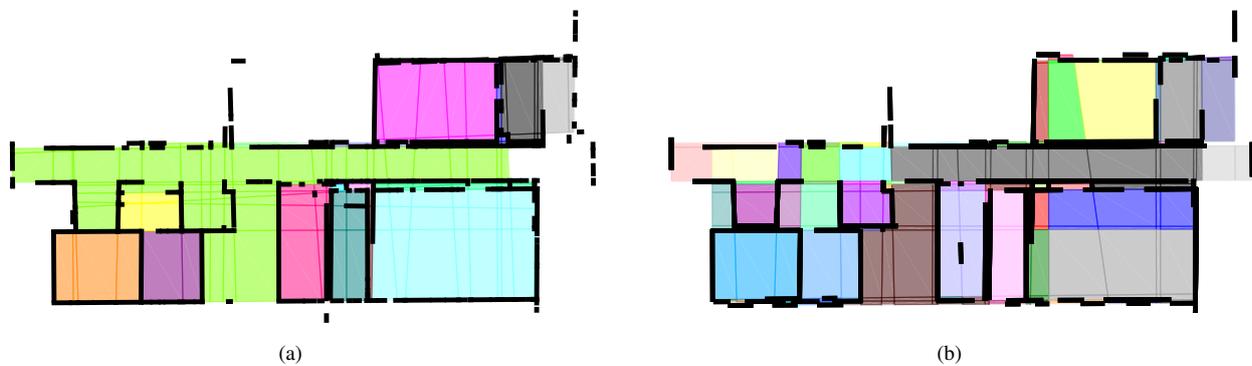


Figure 10. Dataset **TUB2** (second floor): floor plans obtained with (a) the proposed method and (b) the method by (Magri and Fusiello, 2018). This figure is best viewed in colours.

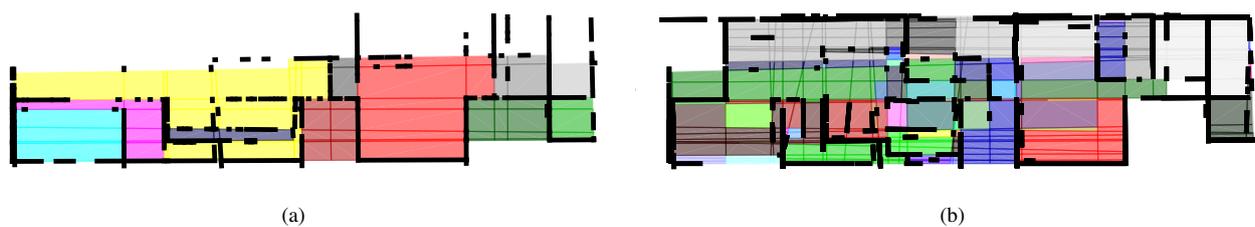


Figure 11. Dataset **Fire Brigade**: floor plans obtained with (a) the proposed method and (b) the method by (Magri and Fusiello, 2018). This figure is best viewed in colours.

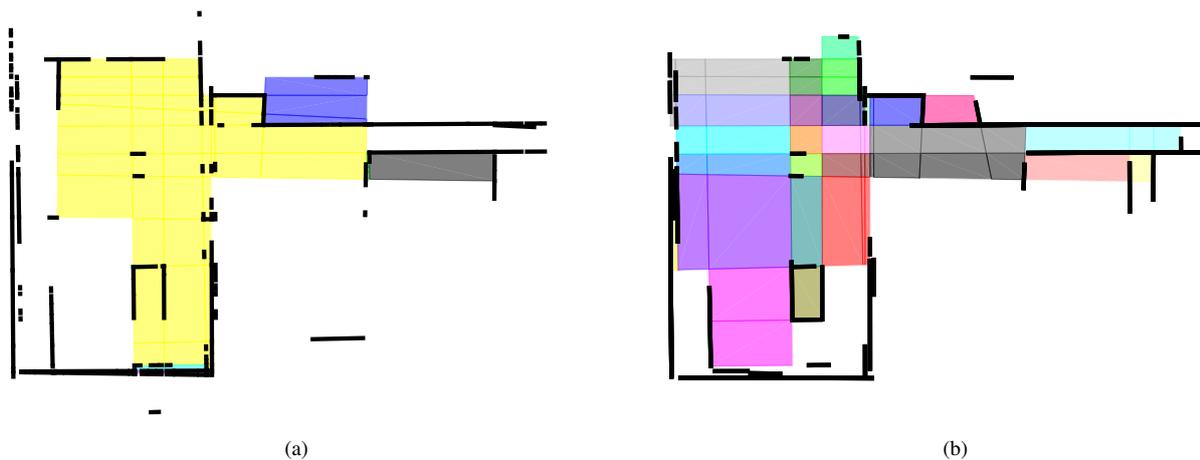


Figure 12. Dataset **UVigo**: floor plans obtained with (a) the proposed method and (b) the method by (Magri and Fusiello, 2018). This figure is best viewed in colours.

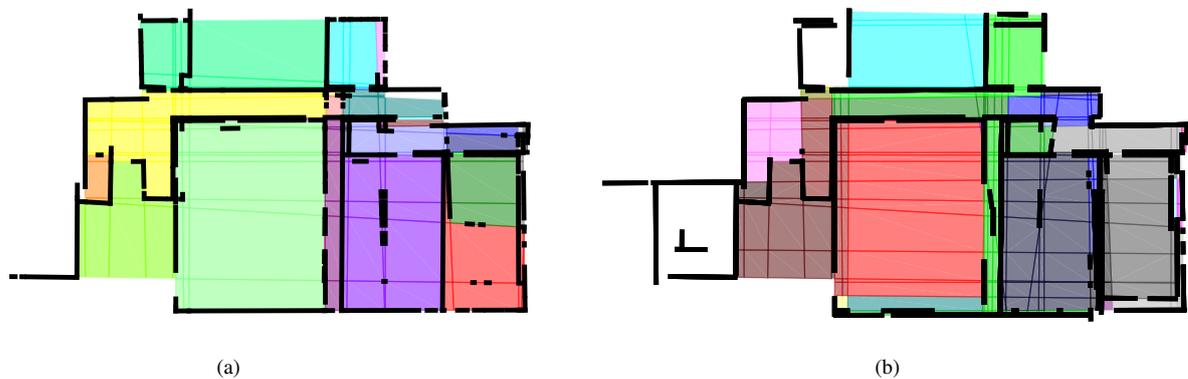


Figure 13. Dataset **UoM**: floor plans obtained with (a) the proposed method and (b) the method by (Magri and Fusiello, 2018). This figure is best viewed in colours.

Oesau, S., Lafarge, F. and Alliez, P., 2016. Planar shape detection and regularization in tandem. 35(1), pp. 203–215.

Previtali, M., Barazzetti, L., Brumana, R. and Scaioni, M., 2014. Towards automatic indoor reconstruction of cluttered building rooms from point clouds. ISPRS Annals 2(5), pp. 281–288.

Previtali, M., Díaz-Vilariño, L. and Scaioni, M., 2018. Indoor building reconstruction from occluded point clouds using graph-cut and ray-tracing. Applied Sciences 8(9), pp. 1529.

Tang, P., Huber, D., Akinci, B., Lipman, R. and Lytle, A., 2010. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. Automation in construction 19(7), pp. 829–843.

Toldo, R. and Fusiello, A., 2008. Robust multiple structures estimation with J-Linkage. In: Proceedings of the European Conference on Computer Vision, pp. 537–547.

Tran, H., Khoshelham, K., Kealy, A. and Díaz-Vilariño, L., 2017. Extracting topological relations between indoor spaces from point clouds. ISPRS Annals 4, pp. 401–406.

Turner, E. and Zakhor, A., 2012. Watertight as-built architectural floor plans generated from laser range data. In: 3D

Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on, IEEE, pp. 316–323.

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

Volk, R., Stengel, J. and Schultmann, F., 2014. Building information modeling (bim) for existing buildings – literature review and future needs. Automation in construction 38, pp. 109–127.

Wang, R., Xie, L. and Chen, D., 2017. Modeling indoor spaces using decomposition and reconstruction of structural elements. Photogrammetric Engineering & Remote Sensing 83(12), pp. 827–841.

Xiao, J. and Furukawa, Y., 2014. Reconstructing the worlds museums. International journal of computer vision 110(3), pp. 243–258.

Zlatanova, S., Sithole, G., Nakagawa, M. and Zhu, Q., 2013. Problems in indoor mapping and modelling. ISPRS Archives 42(4/W4), pp. 63–68.