Eurographics Workshop on 3D Object Retrieval (2014)
B. Bustos, H. Tabia, J.-P. Vandeborre, and R. Veltkamp (Editors)

Towards the Extraction of Hierarchical Building Descriptions from 3D Indoor Scans

S. Ochmann[1], R. Vock[1], R. Wessel[1], and R. Klein[1]

[1]Institute of Computer Science II, University of Bonn, Germany

Abstract

We present a new method for the hierarchical decomposition of 3D indoor scans and the subsequent generation of an according hierarchical graph-based building descriptor. The hierarchy consists of four basic levels with according entities, building - storey - room - object. All entities are represented as attributed nodes in a graph and are linked to the upper level entity they are located in. Additionally, nodes of the same level are linked depending on their spatial and topological relationship. The hierarchical description enables easy navigation in the formerly unstructured data, measurement takings, as well as carrying out retrieval tasks that incorporate geometric, topological, and also functional building properties describing e.g. the designated use of single rooms according to the objects it contains. In contrast to previous methods which either focus on the segmentation into rooms or on the recognition of indoor objects, our holistic approach incorporates a rather large spectrum of entities on different semantic levels that are inherent to 3D building representations. In our evaluation we show the feasibility of our method for extraction of hierarchical building descriptions for various tasks using synthetic as well as real world data.

Categories and Subject Descriptors (according to ACM CCS): H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing—H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—I.3.m [Computer Graphics]: Miscellaneous—I.5.m [Pattern Recognition]: Miscellaneous—

1. Introduction

With the availability of fast and cheap 3D acquisition devices, digital point clouds have replaced analog methods of site measuring as the favored means for documenting the as-is state of buildings, especially regarding their interior. However, due to their inherent lack of structure, point clouds only serve as a starting point for tasks like retrofitting or renovation. The need for easy navigation, targeted (textual) search, manipulation, taking of measurements, and efficient rendering usually forces architects and construction companies to manually generate additional metadata information or even 3D Building Information Modeling (BIM) overlays of the point cloud. By that, the advantages that come along with the digital capturing devices are partially lost again as a large amount of manual postprocessing is still required.

The first step to a better usability of indoor measurements for architectural, engineering, and construction purposes is a proper segmentation into semantically meaningful parts including storeys and rooms. Several most recently introduced methods have taken on this task [MMJ*13,OVW*14,TZ14]. While the main goal in most of these publications is to reconstruct a per-room boundary representation of the captured building which facilitates rendering, such methods also partially improve navigating and manipulating the data, e.g. by hiding or removing certain rooms. Additionally, although not explicitely mentioned in the publications, such methods would theoretically allow for basic measurement tasks like e.g. determining the area or height of a room.

The second step to better usability of 3D indoor scans is to equip segmented rooms with an appropriate set of metadata that allows targeted retrieval of information pointing to the function of a room according to the objects it contains. For example, if a room contains objects like a sink or a shower, its designated function is likely to be that of a bathroom. The above-mentioned methods are restricted to geometric properties of the room; they are not able to de-

DOI: 10.2312/3dor.20141054

rive function-related information. Apart from the field of robotics where recognition of shapes in indoor scenes is a very important task (see e.g. [RBMB09, KAJS11]), identifying objects in rooms has been addressed by several more architecture and construction-related approaches lately, see e.g. [NXS12, KMYG12]. However, current methods for indoor analysis relying on point cloud representations treat the two semantic levels of segmentation into rooms and the interior of a room as being rather isolated from each other. Thereby, the full potential of an integrated building representation is not used, especially for tasks like targeted retrieval. In this work we try to overcome the aforementioned drawbacks. Our holistic hierarchical graph-based building description incorporates rather coarse levels including storeys and rooms, but it also captures the objects that are located in each room. By that, we are able to combine search queries relying on the topological arrangement of rooms fulfilling certain constraints (e.g. area) with queries that target the semantically high-level function of a room. In our evaluation we show several exemplary queries and retrieval results in non-trivial real-world data. Summarizing the contributions of our work, they are:

- A method for segmenting buildings into storeys & rooms
- Extraction of room neighborhood and room connectivity
- A holistic concept for a hierarchical building descriptor

2. Related Work

The analysis of architectural data is a wide field with different problem statements and applications. Our method combines methods for *Scene Segmentation* (Section 2.1), *Scene Understanding and Object Recognition* (Section 2.2), and *Topological Structuring* (Section 2.3) of a building.

2.1. Segmentation and Geometric Structure Extraction

Since point cloud data is inherently unstructured it is mandatory to structure and segment the scene before attempting to extract the structure of a building. [MMJ*13] aim to extract a building model which is segmented on a per-room level. They construct a polyhedral model by projecting, clustering and finally intersecting wall candidates in a two-dimensional cell complex. In [BB10] the authors propose extracting and projecting floor and ceiling structures and finding a segmentation of the plane into cells in order to derive the building's ground plan. A method combining 3D point cloud data and ground-level photographs is presented in [XF12]. The authors reconstruct a CSG (constructive solid geometry) representation and use an "Inverse-Constructive-Solid-Geometry" approach to determine the *observed empty* space. [TZ14] present a method for creating 2D floor plans and 2.5D building models including a segmentation into rooms by reducing the room-labeling to a *Graph-Cut* problem on a 2D Delaunay triangulation where triangles are labeled as *interior* or *exterior*. A similar approach is suggested in [OLA*13] for the reconstruction of permanent structures by using a *Graph-Cut* approach to decompose and label the space into interior and exterior. Probabilistic clustering of points based on their mutual visibility is used in [OVW*14] for obtaining a segmentation of the point cloud into rooms, followed by a detection of openings. In contrast to [BB10,XF12,OLA*13] our method does not aim to reconstruct purely geometric properties of the scanned scene, but to perform a hierarchical, semantically meaningful segmentation. While [TZ14, MMJ*13] perform a segmentation into rooms, they do not determine connections between them. Also, our method works directly on the point cloud without reconstructing a mesh model. In comparison to [OVW*14] our method is able to cope with highly non-convex rooms, determines room neighborhoods and provides a more robust opening detection.

2.2. Object Recognition and Scene Understanding

In contrast to focusing on the coarse structure of rooms and their connectivity a lot of research went into the recognition of objects and their relations within a room. Regarding the overwhelming amount of publications targeting this task especially in the field of robotics, we restrict ourselves on approaches that focus on architecture and construction using static scenes. [RBMB09] propose a hybrid approach to understand indoor scenes by using geometric, as well as surface models in order to segment the scene into objects. A method based on an oversegmentation of the scene based on the smoothness and continuity of surfaces with subsequent labeling based on Markov Random Fields is described in [KAJS11]. Template learning using the large amount of freely available synthetic 3D models to enable object recognition is described in [LF10]. In [KMYG12], the authors use learned graph-based models for objects in a *learning phase* by matching stable primitive parts across measurements, and then try to fit these models in a *recognition phase*. A similar approach that tries to avoid the problem of prior segmentation is presented in [NXS12]. The authors propose a *Search-Classify* method where an oversegmented scene is iteratively simplified while simultaneously maximizing classification likelihood for previously computed feature descriptors. [SXZ*12] propose an interactive method for the segmentation of indoor scenes into semantic entities (e.g. furniture elements) from RGBD images. Having obtained semantic labels in each image using a Conditional Random Field model, they reconstruct the scene with objects from a model database. While these methods provide tools for the segmentation and understanding of scenes on a room level, they do not take into account the overall building topology as proposed in this paper.

2.3. Topology Extraction

There is a wide variety of research work that focuses on extraction and applications of a building's topology (i.e. the structure, connectivity and accessibility of rooms), although

almost all of the proposed methods work with representations different from 3D point clouds. [ALWD12] use image segmentation and OCR (optical character recognition) techniques to extract room structure and semantics in 2D floor plans. In [AWL*14] the authors enhance this approach by adding a semantic analysis based on SURF (speeded up robust features) which yields a graph representation of rooms used for retrieval of room configurations. Another method based on image representations of floor-plans is proposed in [MLVT10], which is based on a recursive decomposition of the image to retrieve nearly convex regions. [WBK08] extract topological information from low-level 3D CAD representations of buildings by comparing 2D plans extracted at different cut-heights for each storey. In [LWL*13], a building's topology is derived from high-level BIM models by analyzing certain entity constellations in the model. In contrast to these approaches, our method does not rely on the availability of 2D or 3D models of the building but works purely on point cloud scans.

3. Method Overview

This section provides a concise overview of our approach for generating a hierarchical building descriptor from indoor point cloud scans. Starting with registered point cloud scans of a building, the steps of our approach are as follows.

- Planar structures are detected. This yields an initial, coarse structuring of the point cloud.
- Using the given initial assignment of points to scanners, a semi-automatic method for segmenting the point cloud into rooms is performed (Section 4). This assigns each point to exactly one room and resolves ambiguities in regions where multiple scans overlap.
- Using this segmentation, the neighborhood relation of rooms (Section 5.1) and their connectivity (Section 5.2) is determined and encoded in a graph structure.
- The point subset of each room is further segmented into objects like furniture elements (Section 6.1). For each object, shape descriptors are computed and connected to the respective room node (Section 6.2).
- After the hierarchical descriptor for a building has been built, structural queries may be performed by means of matching attributed query graphs representing room and object constellations (Section 7).

Note that the presented processing chain is very modular in that different algorithms may be plugged in for performing the individual steps.

4. Segmentation Into Rooms

The first step of our approach is the segmentation of the point cloud into subsets corresponding to rooms. The approach uses a coarse initial guess for point-to-room assignments which is subsequently refined. We initially generate a preliminary room label r_i' for each scanner location s_i and assign all points belonging to s_i the label r_i'. Because a room may have been scanned from multiple positions, these scan positions must be merged in the first phase. In the scope of this paper, this step is done manually using an intuitive graphical interface. In the second phase, an automatic relabeling procedure is carried out which is based on the following assumption. Let x be a point which belongs to room r (even though the initial room assignment of x may be different from r). We hypothesize that most points which are (directly or "almost" directly) *visible* from the position of x belong to the same room as x and that those points tend to be labeled correctly because only a relatively small fraction of points is located in regions where scans overlap.

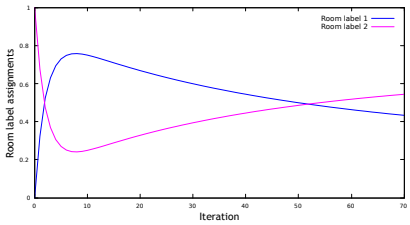

Figure 1: *Label diffusion for a point which is initially labeled incorrectly (the correct label is 1). The correct label is assumed after a few iterations, however it may once again become incorrect with increasing number of iterations.*

The relabeling procedure is formulated as a diffusion process in which the "transfer" of point-to-room labels between points is governed by the mutual *visibility* between point pairs. The rationale behind this formulation is that it not only allows transfers between points which are *directly* visible but also allows *indirect* connections via a few ray "bounces". The importance of this is that occlusion effects (either due to non-convex rooms or clutter) are mitigated. We model the transfer probability between points as a Markov chain with the row-stochastic transition matrix

$$T := \begin{pmatrix} \frac{v_{11}}{k_1} & \cdots & \frac{v_{1n}}{k_1} \\ \vdots & \ddots & \vdots \\ \frac{v_{n1}}{k_n} & \cdots & \frac{v_{nn}}{k_n} \end{pmatrix}, \tag{1}$$

where n is the number of points, $v_{ij} = 1$ iff x_j is visible from x_i and 0 otherwise, and $k_i = \sum_{j=1}^{n} v_{ij}$. In addition, we define $v_{ii} = 1$ for all i. The value $T_{i,j}^k$ yields the probability of "moving" from point x_i to x_j via line-of-sight rays in exactly k steps. In addition, let an initial (hard) point-to-room assignment be given as the label matrix

$$L := \begin{pmatrix} l_{11} & \cdots & l_{1m} \\ \vdots & \ddots & \vdots \\ l_{n1} & \cdots & l_{nm} \end{pmatrix}, \tag{2}$$

where m is the number of rooms, l_{ij} equals 1 iff point x_i is associated with room label j and 0 otherwise. The product

$$L_k := T^k L \tag{3}$$

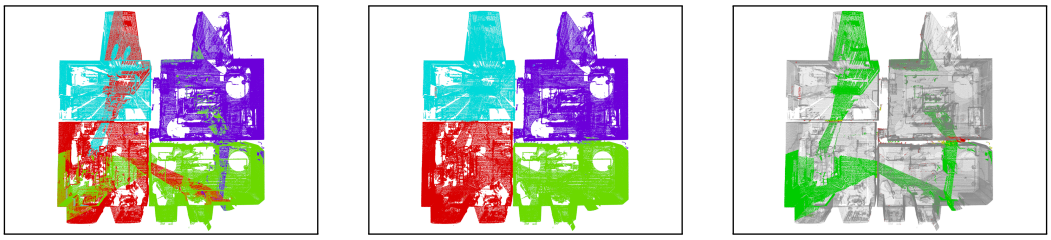

Figure 2: *Room labeling before (left) and after (middle) relabeling. Right: Points which were relabeled are highlighted.*

yields a new label distribution which takes into account the average labels of points encountered after k bounces of line-of-sight rays between points. Consider the progression

$$L_k, \qquad k = 1, \ldots, \tag{4}$$

whose limit for $k \to \infty$ yields the distribution of point-to-room assignments after an infinite number of ray bounces. Figure 1 shows the labeling progression for a particular point. Two effects can be observed in the plot. Firstly, it is sometimes necessary to run a few iterations until the point assumes its correct labeling (in this case, three iterations were necessary). The main reason for this are occlusions within the building such that the majority of the room the point belongs to may only be seen *indirectly* after a few ray bounces. Secondly, the label assignments in the limit of the progression may once again become incorrect as the diffusion spreads throughout building. In the extreme case, if there exists a path between all pairs of points (and thus T is an irreducible transition matrix), all points will assume the same label distribution in the limit. As a compromise between allowing multiple iterations and avoiding the limit case, we decide for a room label $room(x_i)$ for point x_i by integrating over the first N iterations for each label:

$$room(x_i) := \operatorname*{argmax}_{j \in \{1,\ldots,m\}} \sum_{k=1}^{N} (L_k)_{i,j}. \tag{5}$$

In our experiments, a value of $N = 10$ yielded satisfactory results. For the practical implementation, a set of scans is initially given, together with the respective scan origins. The scans are assumed to be registered in a common coordinate system (this step is usually done by the scanner software). As a prerequisite for the relabeling, point normals are estimated by means of local PCA (principal component analysis) of point patches and planar structures are detected using a RANSAC (random sample consensus) implementation by Schnabel et al. [SWK07]. Each detected plane is also assigned the set of points which constitute it, as well as an occupancy bitmap which is used for perfoming fast, approximate intersection tests with the building structure. Each bitmap pixel may take a continuous value in $[0,1]$ and is initially set to 1 iff the projection of at least one point lies within the boundary of that pixel and 0 otherwise. The bitmaps are subsequently smoothed using a box filter in order to fill small holes.

For approximating L_k, a stochastic, iterative ray voting scheme is used. Instead of averaging the labels of *all* points that are visible from x, k sample rays are generated whose directions are randomly sampled on the hemisphere around the normal of x. For each sample ray r_i, the nearest intersection p_{isect} with the set of planes is determined (taking into account the respective occupancy bitmaps). If the nearest intersection is with a plane whose normal points into the same hemisphere as the ray direction, the sample is not counted. In each iteration, the new label soft assignment vector $l(x)$ of x is determined by averaging the label vectors of points located within the area of each occupancy bitmap pixel intersected by a ray as well as averaging over all sample rays:

$$l_{new}(x) := \frac{1}{h+1}\left(l(x) + \sum_{i=1}^{h}\left(\frac{1}{b_i}\sum_{j=1}^{b_i} l(y_{i,j})\right)\right), \tag{6}$$

where h is the number of sample rays which intersected some plane, b_i is the number of points within the bitmap pixel b hit by ray r_i, and $y_{i,j}$ is the jth point within b. Note that if not a single sample ray successfully intersected a plane, the definition yields $l_{new}(x) = l(x)$. Figure 2 shows part of the point cloud before and after relabeling as well as an overview of which parts of the point cloud have been relabeled. For each room r, the approximate room area is determined which may later be used as a constraint when querying for room constellations. Let P_h be the set of points belonging to (approximately) horizontal planes, and let P_r be the set of points of room r. The point set $P_r \cap P_h$ is projected into a regular grid in the x-y-plane with cell size c. The number of cells n containing at least one projected point yields the approximate room area $area_r := nc^2$. We also estimate the floor elevation of each room which is later used for aligning the z position of object descriptors with the floor elevation. Let $\mathcal{P} = \{p_1, \ldots, p_n\}$ be the set of (approximately) horizontal planes whose normals are pointing upwards, let P_{p_i} be the respective point sets belonging to plane p_i, and P_r as above. We define the best floor candidate plane as

$$p_{floor} := \operatorname*{argmax}_{i \in \{1,\ldots,n\}} |P_r \cap P_{p_i}|. \tag{7}$$

The mean z position of the point set $P_r \cap P_{p_{floor}}$ is chosen as the floor elevation $elev_r$ of room r. Having estimated each room's floor elevation, a simple binning procedure is used to group rooms into storeys. We start with an empty set of bins.

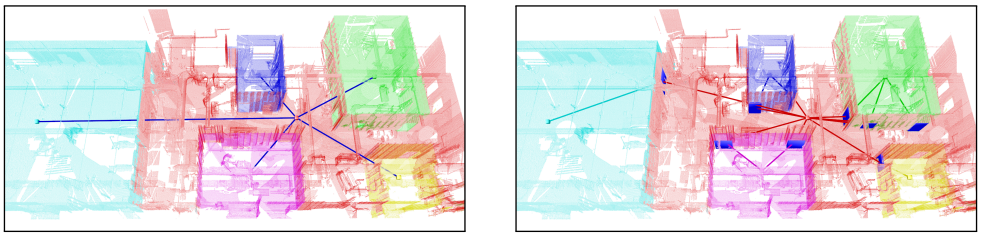

Figure 3: *Room neighbors (left) and connections extracted from a real-world dataset.*

For each room r, it is checked whether there exists a bin b in which all rooms have a floor elevation which is close enough to $elev_r$ with respect to a threshold. If b exists, the room is inserted into b; otherwise a new bin containing r is created. The result up to this point is a graph in which a *storey node* is inserted for each group of rooms which share approximately the same floor elevation, each connected to a (root) *building node*. For each room label, a *room node* is inserted, connected to the respective storey node, and assigned its point subset, area, and floor elevation.

5. Detection of Room Neighbors and Connections

This Section describes our method for determining relations between rooms. Examples for the extraction of the room neighborhood relation and room connectivity in a real-world dataset are shown in Figure 3.

5.1. Room Neighborhood

For determining which rooms are adjacent and shall thus be connected by a *room neighbor edge*, we assume that two rooms are adjacent iff they share at least one wall. The task is to find those walls together with the information which pairs of rooms are separated by them. The top-left image in Figure 4 shows the room labeling after the relabeling step. The process starts with the extraction of plane pairs which are close enough regarding a given threshold and whose normals approximately point away from each other (top-right). A particular plane might belong to more than one room (i.e. the point set associated with the plane contains points associated with different rooms; bottom-left). Therefore the point set of each plane is segmented into point sets belonging to the individual rooms. For each candidate pair of point sets A and B, the associated points are projected into a common plane (either of the involved planes may be used) and an approximate intersection point set is computed by testing for each point $a \in A$ whether there exists a point $b \in B$ with $||a-b||_2 < thresh$ (and vice versa). If the number of points in the approximate intersection set exceeds a given threshold, an edge is inserted and attributed with the information which plane primitive pair was involved. Note that practice, more than two detected planes may constitute a wall due to noise and clutter such that a binning approach is used and a set of planes is assigned to the room neighbor edge; technical details have been omitted here for brevity. The bottom-right image shows the resulting graph for the example.

5.2. Room Connectivity

The opening detection is based on the observation that certain rays that were cast during the scanning process indicate the existence of openings in the building's structure. In particular, if the origin of a ray (scanner position) is located in another room than the measured point, there must be an opening located inbetween the two points. This set of rays is extracted by considering the point-to-room labels *before* and *after* the relabeling step. If the label of a point x was *changed* by the relabeling procedure, the ray which captured x is assumed to pass through an opening. The top-left image in Figure 5 shows the regions observed by the individual scanners as well as their positions. Note that the "red" room consists of multiple scans which have been merged, however the original scanner positions are used for ray generation. Let r be a (laser) ray which measured point x and let $room_{old}(x)$ and $room_{new}(x)$ denote the room label of x before and after the relabeling step. If $room_{old}(x) \neq room_{new}(x)$ and there exists a room neighbor edge $e = (room_{old}(x), room_{new}(x))$,

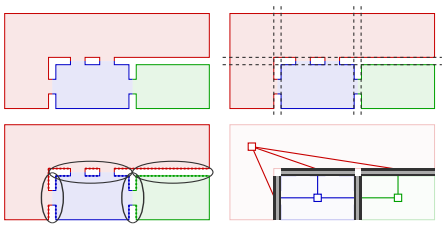

Figure 4: *Steps of the room neighborhood graph generation.*

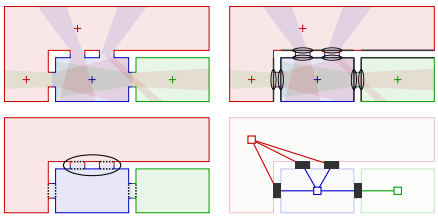

Figure 5: *Steps of the room connectivity graph generation.*

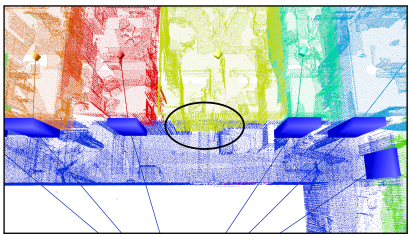

Figure 6: *A failure case of the door detection.*

the intersection of r with all planes associated with e is computed. The top-right image shows the positions where these intersections are located in the example. Because a pair of rooms may in general be connected by more than one door (bottom-left), each point set is split into connected components using a point distance threshold. Finally, the extracted point sets are used for approximating the positions and sizes of the openings. The resulting openings are shown in the bottom-right image. Note that the aforementioned methods do not make assumptions on the orientation of planes and thus not only allow horizontal connections (e.g. doors) but also vertical connections (e.g. stairways) as long as overlaps between the respective scans exist.

A failure case of the door detection is shown in Figure 6. The reasons for the missing door are twofold. Firstly, the door was closed when the scan inside of the "yellow" room was performed. Because the algorithm only requires rays to be shot through an opening from one side, it would normally be able to cope with this situation. However, the other side of the opening was only scanned from sharp angles and thus almost no rays were shot through the door.

6. Assignment of Objects to Rooms

We now extend the graph descriptor by information about objects contained within the individual rooms. By assigning each extracted object a shape descriptor, we enable example-based queries for objects in combination with queries for topological constellations of rooms.

6.1. Extraction of Objects

We now separate objects from broader building structures. Let R be the set of points associated with room r and let $P = \bigcup_{p \in \mathcal{P}} p$ be the set of points associated with detected planes. Points belonging to planes are removed from the room's set of points, $R' := R \setminus P$. Subsequently, connected components in R' are determined, that is two points are assumed to belong to the same component iff their distance is below a threshold. The point cloud is filtered beforehand by considering the mean μ and standard deviation σ of the distance between a point and its k nearest neighbors and filtering out points for which the average distance to its neighbors lies above $\mu + \alpha\sigma$ where α is a user-defined constant (see [Rus09]). The obtained segmentation tends to over-segment the point cloud. However, the object descriptor as described in the next Section also takes into account combinations of nearby segments and thus mitigates this problem.

6.2. Object Shape Descriptors

A global shape descriptor is constructed for each of the extracted object components. Note that we restrict ourselves to a relatively simple object descriptor in the scope of this paper, but it may easily be exchanged with other kinds of descriptors. The three-dimensional space around an extracted point subset is segmented into Θ horizontal slices, Φ concentric shells and Ψ sectors. Each descriptor is built around a local, vertical axis whose x-y-position is centered at the mean position of all points associated with the segment. The z position of the bottom end of the descriptor is set to the previously determined floor elevation $elev_r$ of room r in which the object is located. The (unnormalized) descriptor $D''_o(\theta,\phi,\psi)$ for an object o is defined as

$$D''_o(\theta,\phi,\psi) := \sum_{x \in (\theta,\phi,\psi)} \frac{1}{2\phi}, \tag{8}$$

where $x \in (\theta,\phi,\psi)$ are the points located within the respective bin defined by slice θ, shell ϕ, and sector ψ. The normalization factor within the sum accounts for the increase of volume of shells located farther away from the center. For values of θ outside of the range $[0,\Theta-1]$ or ϕ outside of the range $[0,\Phi-1]$, the respective parameters are set to the nearest valid value. Values of ψ outside of the range $[0,\Psi-1]$ are repeated periodically (modulo Ψ). The bin values are subsequently smoothed using a box filter according to

$$D'_o(\theta,\phi,\psi) := \sum_{\alpha,\beta,\gamma \in \{-1,0,1\}} D''_o(\theta+\alpha,\phi+\beta,\psi+\gamma). \tag{9}$$

The descriptor is normalized according to

$$D_o(\theta,\phi,\psi) := \frac{D'_o(\theta,\phi,\psi)}{\sum_{\theta',\phi',\psi'} D'_o(\theta',\phi',\psi')}. \tag{10}$$

For comparing two object descriptors D_q and D_o, a symmetric version of the χ^2 distance is used. In order to enable rotation invariance along the z-axis, all possible shifts of the sectors of one of the descriptors are evaluated which yields the descriptor distance

$$d(D_q,D_o) := \operatorname*{argmin}_{\delta \in \{0,\ldots,\Psi-1\}} \left(\sum_{\theta,\phi,\psi} \frac{(D_q(\theta,\phi,\psi) - D_o(\theta,\phi,\psi+\delta))^2}{D_q(\theta,\phi,\psi) + D_o(\theta,\phi,\psi+\delta)} \right). \tag{11}$$

Note that summands with a zero denominator are set to zero. As pointed out in the previous section, one object may be separated into multiple components; see Figure 7 for an example. To allow matching of complete query objects against objects which are comprised of multiple segments, additional object nodes consisting of combinations of *up to* three nearby segments are added to the graph (this approach is loosely based on ideas from [NXS12]). For each combination of segments, an *object node* is added to the graph, as-

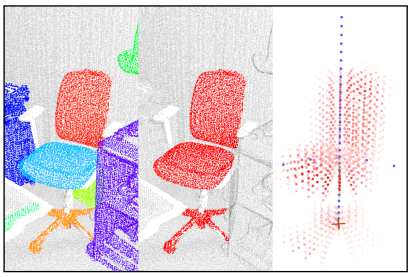

Figure 7: *Left: A chair consisting of multiple segments. Middle: One of the segment combinations added to the graph descriptor. Right: Visualization of descriptor coefficients.*

signed its shape descriptor and connected to the respective room node.

7. Graph Matching, Pruning and Scoring

This Section describes how combined topological and example-based object queries are performed. A query is given in form of an attributed graph $G = (V,E)$ consisting of a set of nodes V (of type storey, room, or object, possibly attributed with room areas or object descriptors), and a set of edges E (horizontal/vertical opening, or object edge). For determining subgraph matches, *subgraph monomorphisms* are sought (using the VF2 implementation of the Boost Graph Library). In general, the types of nodes and edges must be the same in order to match. In addition, room nodes in the query graph may be attributed with a minimum and/or maximum area which must match the target room's area, if given. When matching two object nodes, the *dissimilarity* of the associated shape descriptors is determined. Other kinds of hard and soft constraints for nodes and edges are possible, however for the experiments in this paper we restrict ourselves to the aforementioned constraints. Apart from binary compatibility decisions for nodes and edges, we used the object dissimilarities for scoring each match. For a match m, let q_i be the object nodes in the query graph and t_i the respective matching nodes in the target graph, then the score is defined as $score(m) := -\sum_i d(D_{q_i}, D_{t_i})$. Because we include pairs and triples of object segments as object nodes in the graph, an additional pruning step is performed to avoid matches in which a particular segment is used multiple times.

8. Evaluation

In this section, we present results on part of a real-world laser scan of the *Risløkka Trafikkstasjon* (Oslo). The dataset consists of 33 scans which were merged to 28 room labels. The cloud was coarsely cropped in order to remove some surrounding clutter like trees, and subsampled such that there is at most one point within a voxel of 1 cm^3, resulting in a total number of 25.7 million points. Apart from intuitively defining arbitrary attributed room and storey configurations, our approach allows to attach certain objects to the room nodes that are incorporated in the query. To this end, the user may either select an object that was identified during our segmentation process, or he may also include external mesh models. In the latter case, the mesh model is uniformly sampled upon loading in order to obtain a point cloud for which a shape descriptor is computed as described in Section 6.2. Figure 8 shows exemplary results of subgraph queries on the dataset. Mesh models were used as the input representation for the query objects. As can be seen from our preliminary results, our method on the one hand allows for improved navigation of the formerly unstructured point cloud data by segmenting it into storeys and rooms. On the other hand, it also allows to get a hint on the intended usage of single rooms by identifying function-related fixtures, see e.g. the detected basin, which can be used to constitute the base for further generation of high-level textual room attribution.

9. Conclusion

We presented a holistic approach for the extraction of hierarchical building descriptors purely from 3D indoor point clouds which incorporate topological and functional properties of a building. The outlined processing chain is very modular such that individual parts may easily be exchanged and improved. The current chain comprises a segmentation of the point cloud into storeys and rooms which is subsequently used for determining room neighbors and connections. A segmentation of the rooms' point subsets into objects contained within each room further enriches the graph structure with object shape descriptors. After the building descriptor has been built, combined queries for room constellations and contained objects may be performed. The current implementation allows to include constraints like room area, example-based object shape and connection type in the queries. Our approach has been demonstrated on a large-scale real-world dataset. In the future we want to investigate the relation between our diffusion-based segmentation and the approach recently suggested by Mura et al. [MMJ*13] who used a GPS embedding suggested by Rustamov [Rus07] which is also closely related to a diffusion process on a mesh.

10. Acknowledgements

We would like to thank Dag Fjeld Edvardsen from Catenda, Norway, for providing scans of Risløkka trafikkstasjon, Oslo as well as Henrik Leander Evers and Martin Tamke for scans of the Technical University of Denmark, and scans of Kronborg Castle, Denmark. This work was partially funded by the German Research Foundation (DFG) under grant KL 1142/9-1 (Mapping on Demand), and by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 600908 (DURAARK) 2013-2016.

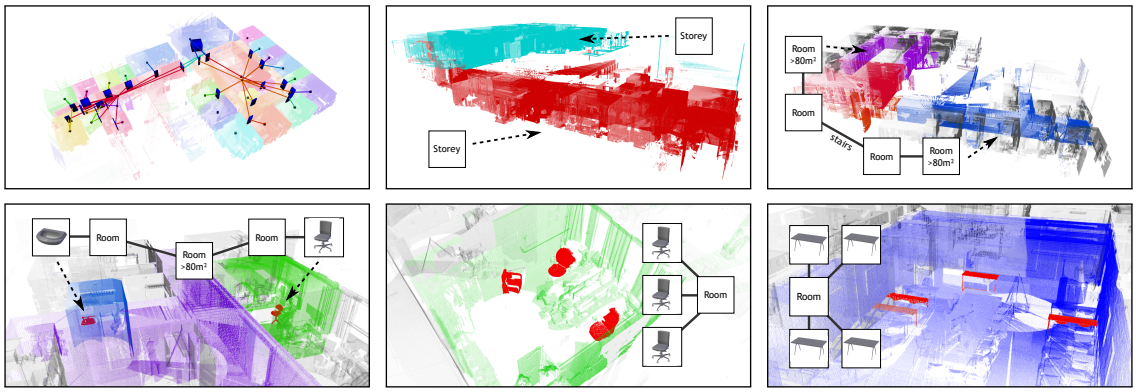

Figure 8: *Results of subgraph queries. Top-left: The room connections extracted from the dataset. Our approach not only allows queries for topological constellations (top middle, top right), but also for combined queries including objects (bottom row).*

References

[ALWD12] AHMED S., LIWICKI M., WEBER M., DENGEL A.: Automatic Room Detection and Room Labeling from Architectural Floor Plans. In *Document Analysis Systems* (2012), pp. 339–343. 3

[AWL*14] AHMED S., WEBER M., LIWICKI M., LANGENHAN C., DENGEL A., PETZOLD F.: Automatic analysis and sketch-based retrieval of architectural floor plans. *Pattern Recognition Letters 35*, 0 (2014), 91 – 100. 3

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

[KAJS11] KOPPULA H. S., ANAND A., JOACHIMS T., SAXENA A.: Semantic labeling of 3d point clouds for indoor scenes. In *Advances in Neural Information Processing Systems* (2011), pp. 244–252. 2

[KMYG12] KIM Y. M., MITRA N. J., YAN D.-M., GUIBAS L.: Acquiring 3D indoor environments with variability and repetition. *ACM Transactions on Graphics (TOG) 31*, 6 (2012), 138. 2

[LF10] LAI K., FOX D.: Object recognition in 3d point clouds using web data and domain adaptation. *International Journal of Robotics Research 29*, 8 (July 2010), 1019–1037. 2

[LWL*13] LANGENHAN C., WEBER M., LIWICKI M., PETZOLD F., DENGEL A.: Graph-based retrieval of building information models for supporting the early design stages. *Advanced Engineering Informatics* (May 2013). 3

[MLVT10] MACÉ S., LOCTEAU H., VALVENY E., TABBONE S.: A system to detect rooms in architectural floor plan images. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems* (New York, NY, USA, 2010), DAS '10, ACM, pp. 167–174. 3

[MMJ*13] MURA C., MATTAUSCH O., JASPE VILLANUEVA A., GOBBETTI E., PAJAROLA R.: Robust reconstruction of interior building structures with multiple rooms under clutter and occlusions. In *Proceedings of the 13th International Conference on Computer-Aided Design and Computer Graphics* (November 2013). 1, 2, 7

[NXS12] NAN L., XIE K., SHARF A.: A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics 31*, 6 (Nov. 2012), 137:1–137:10. 2, 6

[OLA*13] OESAU S., LAFARGE F., ALLIEZ P., ET AL.: Indoor scene reconstruction using primitive-driven space partitioning and graph-cut. In *Eurographics Workshop on Urban Data Modelling and Visualisation* (2013). 2

[OVW*14] OCHMANN S., VOCK R., WESSEL R., TAMKE M., KLEIN R.: Automatic generation of structural building descriptions from 3d point cloud scans. *Proceedings of the 9th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (GRAPP 2014)*, Jan. 2014. 1, 2

[RBMB09] RUSU R. B., BLODOW N., MARTON Z. C., BEETZ M.: Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on* (2009), IEEE, pp. 1–6. 2

[Rus07] RUSTAMOV R. M.: Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2007), SGP '07, Eurographics Association, pp. 225–233. 7

[Rus09] RUSU R. B.: *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009. 6

[SWK07] SCHNABEL R., WAHL R., KLEIN R.: Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum 26*, 2 (June 2007), 214–226. 4

[SXZ*12] SHAO T., XU W., ZHOU K., WANG J., LI D., GUO B.: An interactive approach to semantic modeling of indoor scenes with an RGBD camera. *ACM Transactions on Graphics 31*, 6 (Nov. 2012), 12. 2

[TZ14] TURNER E., ZAKHOR A.: Floor plan generation and room labeling of indoor environments from laser range data. *Proceedings of the 9th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (GRAPP 2014)*, January 2014. 1, 2

[WBK08] WESSEL R., BLÜMEL I., KLEIN R.: The room connectivity graph: Shape retrieval in the architectural domain. In *The 16-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision* (2008). 3

[XF12] XIAO J., FURUKAWA Y.: Reconstructing the world's museums. In *Computer Vision–ECCV 2012*. Springer, 2012, pp. 668–681. 2