Exploiting the Room Structure of Buildings for Scalable Architectural Modeling of Interiors

Claudio Mura
University of Zurich
claudio@ifi.uzh.ch

Renato Pajarola
University of Zurich
pajarola@acm.org

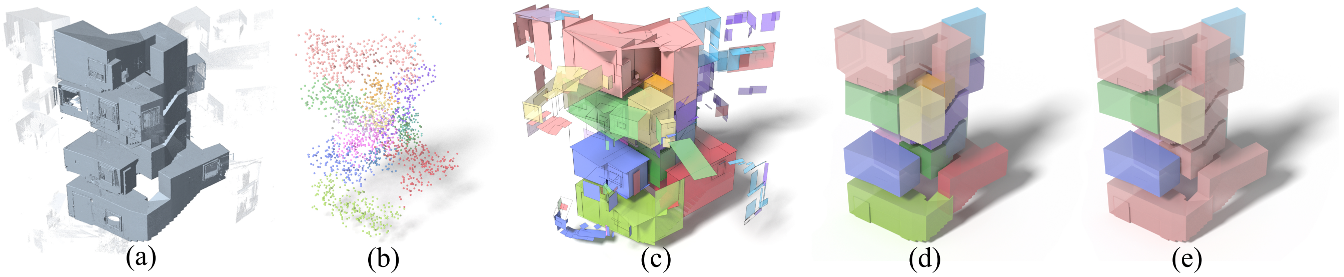

Figure 1: Overview of our pipeline. From left to right: input point clouds (a); room-based clustering of viewpoints (b); room-based clustering of planar scene parts (c); reconstructed room models before (d) and after (e) merging of overlapping rooms.

ABSTRACT

We propose a scalable strategy for the architectural modeling of large-scale interiors from 3D point clouds. We exploit the fact that buildings are structured into different rooms to cast the modeling of a large, multi-room environment as a set of simpler and independent reconstruction problems. This drastically reduces the complexity of the computation and makes the processing of large-scale datasets feasible even without using restrictive priors that affect the precision of the final output.

CCS CONCEPTS

• **Computing methodologies** → Mesh geometry models; • **Theory of computation** → Computational geometry;

KEYWORDS

Boundary representations, point clouds, architecture modeling

ACM Reference format:
Claudio Mura and Renato Pajarola. 2017. Exploiting the Room Structure of Buildings for Scalable Architectural Modeling of Interiors. In *Proceedings of SIGGRAPH '17 Posters, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
DOI: 10.1145/3102163.3102213

1 MOTIVATION AND BACKGROUND

With the ever-increasing diffusion of fast and accurate 3D acquisition systems, new frontiers have opened in domains such as engineering, architecture and facility management. A number of application scenarios rely on the use of compact 3D models of interiors that describe their architectural shape. In practice, such models are often created by hand using interactive modeling tools, leading to labor-intensive and time consuming workflows. For this reason, there is a strong need for automatic pipelines to extract compact and structured 3D models of interiors from raw input data.

The challenges involved in this task are manifold and range from the robust handling of defects in the input data to the precise representation of architectural structures. To manage the complexity of the problem, many state-of-the-art methods sacrifice this latter aspect and employ restrictive priors such as the 2.5D assumption [Ochmann et al. 2016], resulting in methods that can only faithfully model environments with vertical walls and horizontal floors and ceilings. Some *full-3D* methods lift such limitations and allow to model wall structures with arbitrary orientations, but their complexity either makes them only applicable to small environments [Boulch et al. 2014] or requires the use of strategies that can remove small-scale structures [Mura et al. 2016].

We present an approach that exploits the simple fact that indoor environments are subdivided into separate rooms to make the use of full-3D methods feasible on large-scale inputs. We show that performing the room detection – a standard step in many modeling pipelines [Mura et al. 2016; Ochmann et al. 2016] – at the beginning of the processing pipeline allows to reconstruct each room separately, drastically reducing the complexity of the problem and enabling the use of more expressive formulations.

2 ROOM-BASED MODELING PIPELINE

Our pipeline (Fig. 1) takes as input a 3D point cloud with oriented normals and abstracts its planar parts into a set of bounding rectangles [Mura et al. 2016]; the output consists in a set of 3D polyhedra, each describing the boundary surface of a room of the environment.

2.1 Room Detection

Unlike most recent approaches [Mura et al. 2016; Ochmann et al. 2016], which perform the room detection at a late stage in their

SIGGRAPH '17 Posters, Los Angeles, CA, USA
© 2017 Copyright held by the owner/author(s). 978-1-4503-5015-0/17/07…$15.00
DOI: 10.1145/3102163.3102213

pipelines and rely on viewpoint information in the input model, we detect the rooms from the initial planar abstraction and without using the locations of the device at acquisition time.

To do so, we generate a set of synthetic viewpoints by embedding the bounding rectangles in an octree of a pre-defined resolution (25cm in our tests) and by selecting the centers of the leaf cells for which most of the visible scene, as defined by the bounding rectangles, corresponds to front-facing rectangles. In this context, the rectangles act as proxies for the structures of the scene; their winding is derived from the normals of the input 3D points.

We then compute for each viewpoint P_i and rectangle R the area of the surface of R seen by P_i, which we denote by $vis(P_i, R)$. This information allows to group the viewpoints based on their visible surface overlap, using a state-of-the-art procedure based on the Markov cluster algorithm [Mura et al. 2016]. The result (shown in Fig. 1(b)) is a set of n_{rooms} clusters of viewpoints $C = \{C_1, \ldots, C_{n_{\text{rooms}}}\}$, each providing an estimate of the spatial occupancy of a room.

2.2 Room-based Grouping of Rectangles

In the previous step, we used the values $vis(P_i, R)$ to cluster the viewpoints according to the room to which they belong. Conversely, in this step such values are used to define the visibility $vis(C_k, R)$ of a rectangle R from a cluster of viewpoints C_k, which in turn allows to define the probability $\mathscr{P}(R, C_k)$ of R being relevant to the modeling of the room of C_k. In particular, we define $vis(C_k, R) = \max_{P_i \in C_k} vis(P_i, R)$ and compute $\mathscr{P}(R, C_k)$ as follows:

$$\mathscr{P}(R, C_k) = \frac{vis(C_k, R)}{\sum_{i=1}^{n_{\text{rooms}}} vis(C_i, R)}$$

Based on these values, we compute the set $\mathcal{R}_k$ of rectangles that will be used to reconstruct the room defined by cluster C_k as $\mathcal{R}_k = \{R \mid \mathscr{P}(R, C_k) > \theta\}$, where θ (set to 0.25 in our tests) defines a trade-off between efficiency and reconstruction safety. This *soft* assignment of rectangles to the clusters of viewpoints (shown in Fig. 1(c), where each rectangle is colored based on the cluster from which it is most visible) ensures that primitives that are barely visible from the cluster of a room do not burden its reconstruction.

2.3 Room Reconstruction

Each set $\mathcal{R}_k$ is used to reconstruct a 3D mesh that bounds the space occupied by a room. Following a well-established scheme [Boulch et al. 2014], we reconstruct this mesh by labeling the cells of a *cell complex* built from the rectangles in $\mathcal{R}_k$ as inside or outside space and by selecting the facets that separate cells with a different label. Our cell complex is based on a 3D *Binary Space Partitioning* (BSP) constructed from a set of 3D planes that correspond to the main surfaces of the environment and that are computed by clustering the rectangles in $\mathcal{R}_k$. A 3D BSP allows to represent architectural (planar) elements with arbitrary orientations, but incurs in a significant overhead for its construction; this limits its applicability to small environments [Boulch et al. 2014] or requires the use of aggressive pruning strategies to reduce the number of planes used in its construction [Mura et al. 2016].

Since our room-based pipeline considers each room separately, the set of planes used to construct the 3D BSP remains small, thus making the use of this structure feasible without requiring error-prone techniques to reduce the number of primitives used in the computation. In particular, we prune from $\mathcal{R}_k$ only those rectangles that are clearly detached from the ceiling and wall structures (e.g. furniture in the center of a room), without relying on more aggressive strategies [Mura et al. 2016], and extract the dominant planes for the construction of each BSP using a conservative version of the PEARL algorithm. The extraction of the shape of each room from its cell complex employs a state-of-the-art formulation [Mura et al. 2016], adapted to our binary labeling setting.

The result of the room-based reconstruction is shown in Fig. 1(d). Since some of the detected rooms can correspond to sub-spaces of a same environment (e.g. a long staircase), we merge spatially overlapping rooms in a post-processing step, which yields the final room models (Fig. 1(e)).

3 RESULTS AND DISCUSSION

As shown in Fig. 1(e), our room-based pipeline can produce accurate models of large multi-room interiors with complex shapes. Note, in particular, the presence of details like steps and fixtures and the successful recognition of the individual rooms, including the large, central staircase, correctly reconstructed as a single environment.

Even more interestingly, the entire processing of the input model (9.5M points and 827 rectangles) took only about 2 minutes (125.6s) on a MacBook Pro with an Intel Core i7 (2.5GHz) and 16GB DDR3 RAM. This is made possible by our room-based strategy approach, which avoids the bottleneck of the construction of a single, global 3D BSP-based complex. We specifically compared the computational cost of the construction of this structure with that of building the $n_{\text{rooms}} = 13$ cell complexes created by our method. The overall time spent in this computation by our algorithm was 95.8s (split among multiple threads), with the largest complex (built from 119 input planes) requiring 32.4s. On the other hand, the construction of the global BSP, built from 602 planes, was aborted after the insertion of the first 460 planes and over 8 hours of computation, as it exceeded the memory capabilities of our test machine.

These results show that our strategy effectively avoids the bottleneck of constructing a global 3D BSP while still producing a reconstruction that captures the full-3D nature of an environment. Some aspects remain however open to further research. In particular, modeling each room separately implies that the final model is produced without using any regularizers that account for the interactions between adjacent sub-environments; this is particularly critical when merging spatially overlapping rooms that are initially reconstructed as separate environments (currently done in post-processing). Moreover, we want to adopt a more optimized strategy for the generation of the viewpoints from the input model.

REFERENCES

Alexandre Boulch, Martin de La Gorce, and Renaud Marlet. 2014. Piecewise-Planar 3D Reconstruction with Edge and Corner Regularization. *Computer Graphics Forum* 33, 5 (August 2014), 55–64.

Claudio Mura, Oliver Mattausch, and Renato Pajarola. 2016. Piecewise-planar Reconstruction of Multi-room Interiors with Arbitrary Wall Arrangements. *Computer Graphics Forum* 35, 7 (October 2016), 179–188.

Sebastian Ochmann, Richard Vock, Raoul Wessel, and Reinhard Klein. 2016. Automatic Reconstruction of Parametric Building Models from Indoor Point Clouds. *Computers & Graphics* 54 (February 2016), 94–103.