

AUTOMATIC CREATION OF SEMANTICALLY RICH 3D BUILDING MODELS FROM LASER SCANNER DATA

Antonio Adan¹, Xuehan Xiong², Burcu Akinci³, and Daniel Huber^{2*}

¹*Department of Electrical Engineering, Electronics, and Automation,
Castilla La Mancha University, Ciudad Real, Spain*

²*The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA*

³*Department of Civil and Environmental Engineering, Carnegie Mellon University, USA*

* *Corresponding author (dhuber@cs.cmu.edu)*

ABSTRACT: Laser scanners are increasingly used to create semantically rich 3D models of buildings for civil engineering applications such as planning renovations, space usage planning, and building maintenance. Currently these models are created manually – a time-consuming and error-prone process. This paper presents a method to automatically convert the raw 3D point data from a laser scanner positioned at multiple locations throughout a building into a compact, semantically rich model. Our algorithm is capable of identifying and modeling the main structural components of an indoor environment (walls, floors, ceilings, windows, and doorways) despite the presence of significant clutter and occlusion, which occur frequently in natural indoor environments. Our method begins by extracting planar patches from a voxelized version of the input point cloud. We use a conditional random field model to learn contextual relationships between patches and use this knowledge to automatically label patches as walls, ceilings, or floors. Then, we perform a detailed analysis of the recognized surfaces to locate windows and doorways. This process uses visibility reasoning to fuse measurements from different scan locations and to identify occluded regions and holes in the surface. Next, we use a learning algorithm to intelligently estimate the shape of window and doorway openings even when partially occluded. Finally, occluded regions on the surfaces are filled in using a 3D inpainting algorithm. We evaluated the method on a large, highly cluttered data set of a building with forty separate rooms yielding promising results.

Keywords: *interior modeling, 3D modeling, scan to BIM, lidar, object recognition, wall analysis, opening detection.*

1. INTRODUCTION

In the Architecture, Engineering, and Construction (AEC) industry, semantically rich 3D models are increasingly used throughout a building’s lifecycle, from the design phase, through construction, and into the facility management phase. These models, which are generally known as building information models (BIMs), are used for many purposes, including planning and visualization during the design phase, detection of mistakes made during construction, and simulation and space planning during the management phase. Unfortunately, the BIM created at design time vary significantly from what is actually built, due to ensuing renovations, for example. For most buildings, no design BIM exists at all. As a result, there is

an increasing interest in creating BIMs of the actual “as-built” or “as-is” state of a building (both terms are referred to as “as-is” hereafter).

Currently, as-is BIMs are created through a manual process, typically using data from laser scanners as input (Figure 1) [1]. Laser scanners are placed in various locations throughout and around a building, and 3D measurements from each location are registered and combined to form a cloud of 3D points in a common coordinate system. Geometric surface or volumetric primitives are fitted to the 3D point cloud to model walls, floors, ceilings, columns, beams, and other structures of interest. The modeled primitives are annotated with identity labels (e.g., wall) and meta-data, such as the

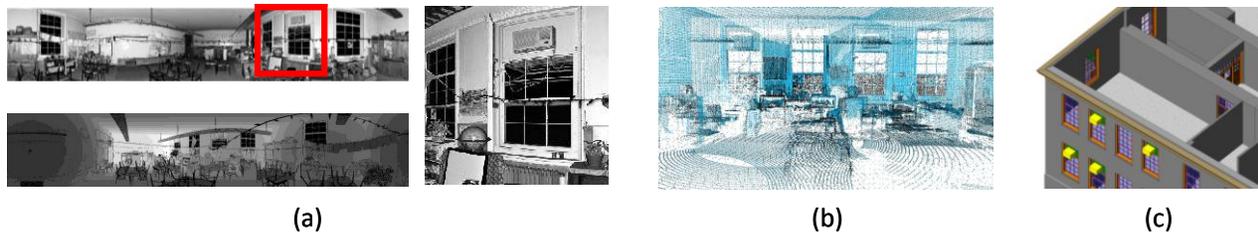


Figure 1. As-is BIM creation. a) The input data from a single scanner location – reflectance image (top), range image (bottom), and close up of highlighted region (right). b) The point data is aligned and combined into a cloud of points. Data from four scans was combined to form this point cloud. c) Geometric primitives are modeled to form the BIM. This BIM was created manually by a professional service provider.

surface material (e.g., concrete), and spatial and functional relationships between nearby structures and spaces are established. Together, this information forms the BIM.

The as-is BIM creation process is a labor-intensive and error-prone operation. Even with training, the result produced by one modeler may differ significantly from that produced by another person. Our goal is to develop tools to help automate this process using techniques from computer vision and machine learning [2, 3]. In this paper, we summarize our work on automatically creating as-is BIMs from laser scan data. Our method takes as input a set of registered 3D point clouds obtained from various locations in a room, and automatically identifies and models the walls, floor, ceiling, and any significant rectangular openings (e.g., doorways and windows). Applied to all the rooms in a building, our method will automatically produce a compact, semantically rich, 3D model which, while not strictly a BIM in the traditional sense, contains the geometric and identity information that substantially makes up the BIM. The remaining steps of labeling windows and doorways and converting the model from a surface representation to a volumetric representation are the subject of ongoing work.

One of the key challenges to automating the as-is BIM creation process is the problem of occlusions. Building modeling algorithms are frequently demonstrated on simple examples like hallways that are devoid of furniture or other objects that would obscure the surfaces to be modeled. To be practical, modeling algorithms need to be capable of functioning in natural, unmodified environments,

since it is usually not feasible to remove the furniture prior to scanning. Not only do occluding objects block visibility of the surfaces of interest, they may also be inadvertently interpreted as parts of the model themselves. For example, a large cabinet against a wall may look very similar to a wall. An automated modeling algorithm must be capable of reliably distinguishing between such “clutter” objects and the target surfaces.

Our algorithm addresses the challenges of clutter and occlusion by explicitly reasoning about them throughout the process. To distinguish clutter from non-clutter, we learn a model of what clutter “looks like” and how it is different from walls, ceilings, and floors. To understand occlusions, we use a ray-tracing algorithm to identify regions that are occluded from every viewpoint and to distinguish these regions from openings in the surface (e.g., due to doorways or windows).

Broadly, our algorithm consists of two phases. In the first phase, planar patches are extracted from the point cloud and a context-based machine learning algorithm is used to label the patches as wall, ceiling, floor, or clutter. These patches are intersected with one another to form a simple surface-based model of the room. In the second phase, each planar surface is analyzed to identify and model the occluded regions and openings. A learning algorithm is used to encode the characteristics of opening shape and location, which allows the algorithm to infer the shape of an opening even when it is partially occluded. An inpainting algorithm is used to fill in the occluded regions with a realistic surface for visualization purposes. These

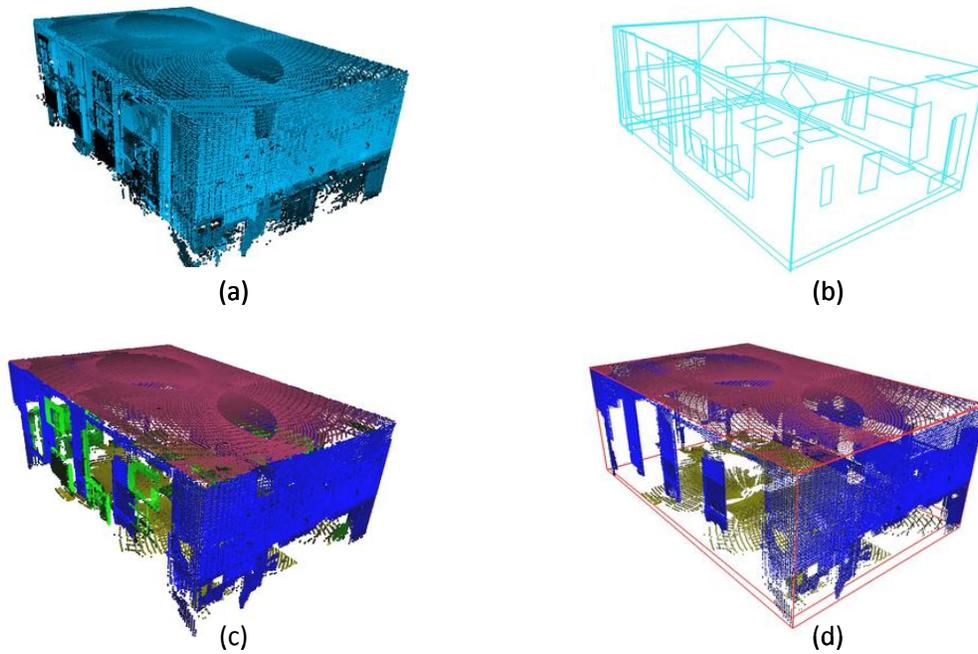


Figure 2. The context-based modeling algorithm consists of four steps, shown for the point cloud from Figure 1: a) voxelization; b) patch detection; c) patch classification; d) patch intersection and clutter removal.

two phases are described in more detail in the next two sections. Further details may be found in [4] and [5].

2. CONTEXT-BASED MODELING

We hypothesize that context can play an important role in recognizing the structures in a building. Distinguishing between different types of objects and between clutter and non-clutter can be difficult or impossible if those objects are seen only in isolation. The context of an object within the room can provide the clues needed to recognize it successfully. For example, a vertical surface may be a wall or an open door (which is considered clutter in this case). Both surfaces are large and planar, but a door typically does not connect to the ceiling, whereas a wall does. Furthermore, a wall is more likely to be parallel or perpendicular to other walls in the room. In this way, the more easily recognized structures can provide the scaffolding that enables the recognition of other, more challenging instances. This phase of the algorithm consists of four steps (Figure 2).

Step 1 – Voxelization. The input point cloud is discretized in a uniformly spaced 3D grid data structure, which is known as a voxel space (Figure 2a). This

discretization serves to reduce the density of data in areas where it is overly dense while maintaining the original data density in sparse areas.

Step 2 – Patch detection. Planar patches are extracted from the voxelized data (Figure 2b). Patches are found using a region growing algorithm to connect nearby points that have similar surface normals and that are well-described by a planar model. The boundary of a patch is described by the minimum area bounding rectangle of the constituent points.

Step 3 – Patch classification. The modeled patches are classified according to the possible labels – wall, ceiling, floor, or clutter – using a context-based algorithm that is modeled using a conditional random field (CRF). The classifier uses local features computed on each patch in isolation as well as features describing the relationship between each patch and its nearest neighbors. We experimented with a number of local features, including a patch’s orientation with respect to the vertical axis, surface area, and height. The contextual relations we considered include orthogonal, parallel, adjacent, and coplanar (Figure 2c).

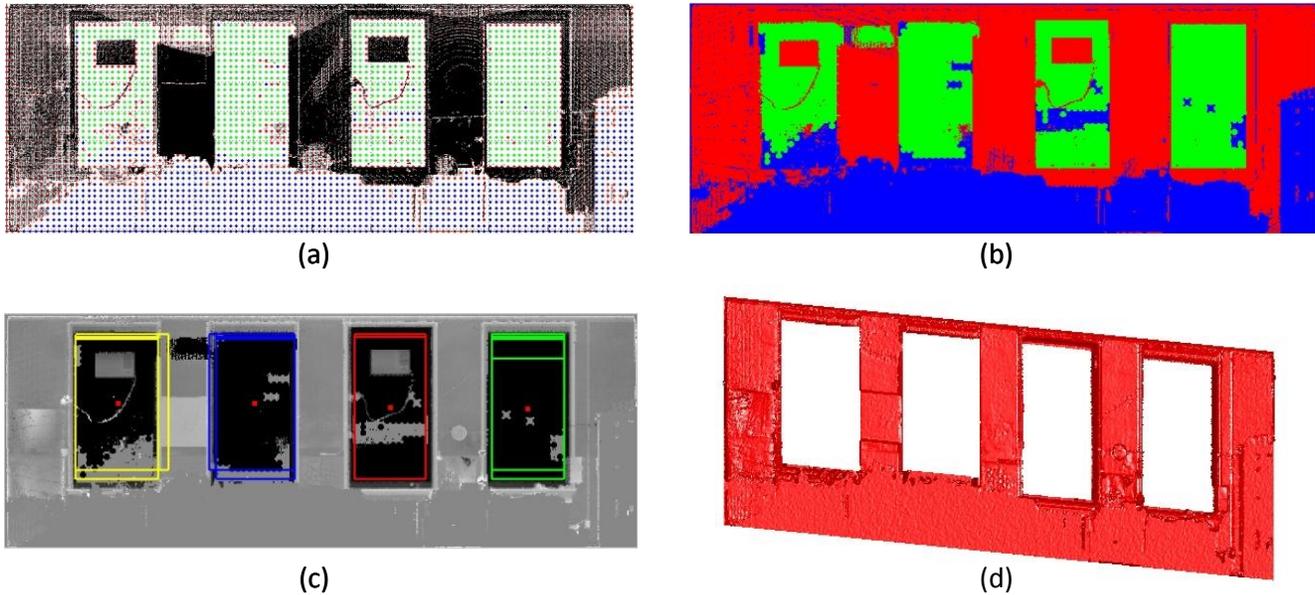


Figure 3. The detailed surface modeling algorithm, shown for one wall using the results of Figure 2: a) occlusion labeling; b) iterative region growing; c) opening detection; d) occlusion reconstruction.

Step 4 – Patch intersection. Finally, the boundaries of the patches are cleaned up by intersecting adjacent patches, and patches labeled as clutter are removed (Figure 2d).

The result of this process is a compact model of the walls, floor, and ceiling of a room, with each patch labeled according to its type.

3. DETAILED SURFACE MODELING

The surfaces produced by the first phase represent idealized surfaces that are perfectly planar and unoccluded and with no openings. Real surfaces in unmodified environments, particularly walls, are usually heavily occluded, and it is important to understand where those occlusions occur. For example, if an engineer wants to measure a dimension of a room, he or she would likely prefer to make the measurement at a location that was actually observed, rather than at an occluded position, which might be incorrectly modeled.

The detailed surface modeling phase of the algorithm operates on each planar patch produced by the context-based modeling process, detecting the occluded regions and regions within openings in the surface. The process involves three steps for each patch (Figure 3).

Step 1 – Occlusion Labeling. A voxel space encompassing the patch and oriented with it is established. A corresponding uniform 2D grid is also established on the

surface of the planar patch. Information stored in this 2D grid can be treated like an image consisting of pixels. The pixels in this image are classified into one of three categories using a ray-tracing algorithm. By tracing a line from the origin of the laser scanner to a detected measurement, a pixel is classified as empty if the ray passes through the plane of the patch, as occupied if the ray stops approximately at the plane, and occluded if the ray stops before reaching the plane. The labels from each scanner position that observes the patch are combined into a single labeling in which a pixel is labeled as occluded if it is occluded from every viewpoint (Figure 3a). Additionally, some pixels may be unobserved due to the sampling resolution of the sensor. These unobserved pixels are inferred using an iterative region growing algorithm, resulting in a high-resolution labeled image (Figure 3b).

Step 2 – Opening detection. Detecting openings in unoccluded surfaces can be achieved by analyzing the data density and classifying low density areas as openings. Occlusions make this problem more difficult, since we must infer the state of the unseen data. Furthermore, some regions within an opening may be labeled as occupied, for example, due to a window-mounted air-conditioner. We address this problem by learning a model of the appearance of openings from training examples. We use a support

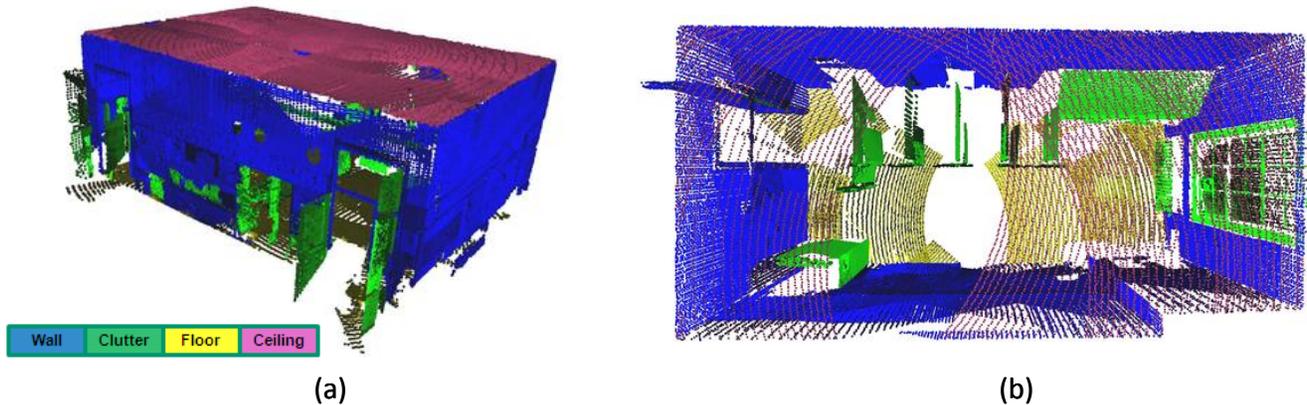


Figure 4. Examples of the context-based modeling algorithm for some other rooms. The right example is a top view of a bathroom, in which the stalls and some large cabinets were all successfully distinguished from the walls.

vector machine (SVM) classifier to label candidate openings based on features derived from the label image (Figure 3c). Candidate openings are enumerated from essential vertical and horizontal lines extracted from a composite depth image of the data within the patch. We use a number of features as input to the SVM, including opening width, height, and distance from the boundaries of the patch (sides, and top, and bottom). The resulting detections are grouped and a prototype representative is selected by averaging the boundary estimates within each cluster.

Step 3 – Occlusion reconstruction. Once the openings are determined, the occluded regions of the wall are filled in with an inpainting algorithm. This step is not strictly necessary, but it improves the visualization of the results (Figure 3d).

4. EXPERIMENTAL RESULTS

We conducted experiments using data from a building that was manually modeled by a professional laser scanning service provider. The facility is a two-story schoolhouse containing 40 rooms. The facility was scanned from 225 locations resulting in over 3 billion 3D measurements.

We evaluated the context-based modeling algorithm using this data set. The patch detection resulted in 389 patches in the interior of the building. The labeling algorithm achieved 89% classification accuracy on this

data. The highest confusion was between walls and clutter. Our experiments suggest that the context aspect of our algorithm improves recognition performance by about 6% and that the most useful contextual features are coplanarity and orthogonality. Figure 4 shows some example results. The main failures occur in the interiors of low built-in closets and in stairwells, both of which are atypical situations. Detailed experiments can be found in cite [4].

We also evaluated the detailed wall modeling algorithm using the patches produced by the context-based modeling algorithm. The algorithm was able to detect window and door openings with 88% (61/69) accuracy. We also evaluated the accuracy of the sizes of the modeled openings and found that the average error in opening size was 4.35 cm. Figure 5 shows some example results, and Figure 6 shows one floor of the entire reconstructed model. Further experiments may be found in [5].

5. FUTURE WORK

We are currently working on completing the points-to-BIM pipeline by implementing an automated method to convert the surface-based representation produced by our algorithm into a volumetric representation that is commonly used for BIMs. We are also working on extending the algorithm to distinguish windows, doorways, and other openings.

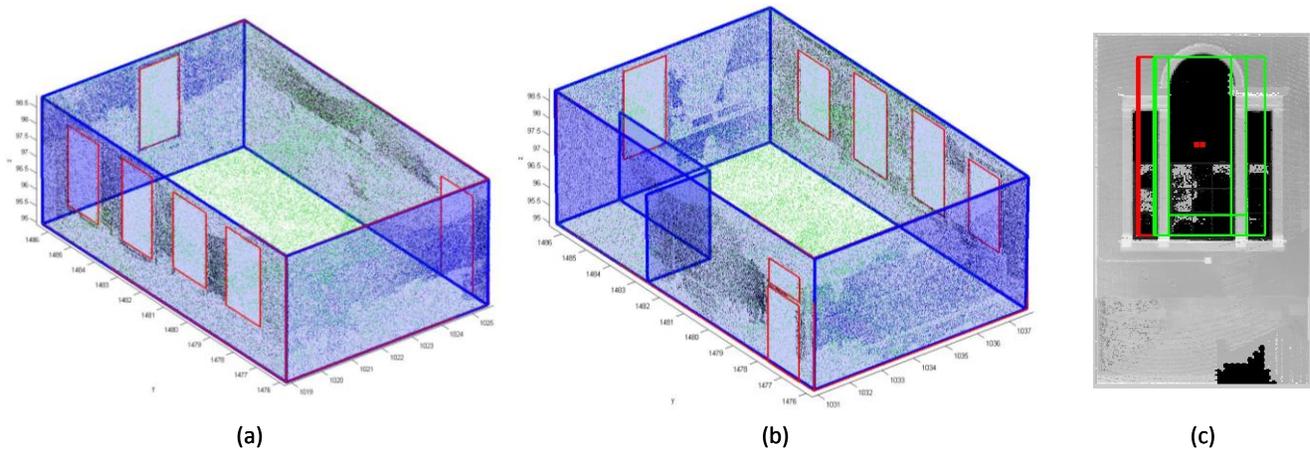


Figure 5. Example results of the detailed surface modeling algorithm for entire rooms. a) The room shown in Figure 2. Walls, ceilings, and floors are outlined in blue, and windows and doors are outlined in red. b) Another room. c) An example where the opening detection failed due to an arched window, which violates the assumptions of the algorithm.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 0856558 and by the Pennsylvania Infrastructure Technology Alliance. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We thank Quantapoint, Inc., for providing experimental data.

REFERENCES

[1] P. Tang, D. Huber, B. Akinci, R. Lipman, et al.,

“Automatic Reconstruction of As-built Building Information Models from Laser-Scanned Point Clouds: A Review of Related Techniques,” *Automation in Construction*, vol. 19, pp. 829-843, November 2010.

[2] D. Huber, B. Akinci, P. Tang, A. Adan, et al., “Using Laser Scanners for Modeling and Analysis in Architecture, Engineering, and Construction,” in *Proceedings of the Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, 2010.

[3] B. Okorn, X. Xiong, B. Akinci, and D. Huber, “Toward Automated Modeling of Floor Plans,” in *Proceedings of the Symposium on 3D Data Processing, Visualization and Transmission*, Paris, France, 2010.

[4] X. Xiong and D. Huber, “Using Context to Create Semantic 3D Models of Indoor Environments,” in *Proceedings of the British Machine Vision Conference (BMVC)*, 2010.

[5] A. Adan and D. Huber, “3D Reconstruction of Interior Wall Surfaces Under Occlusion and Clutter,” in *Proceedings of 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, Hangzhou, China, 2011 (in press).

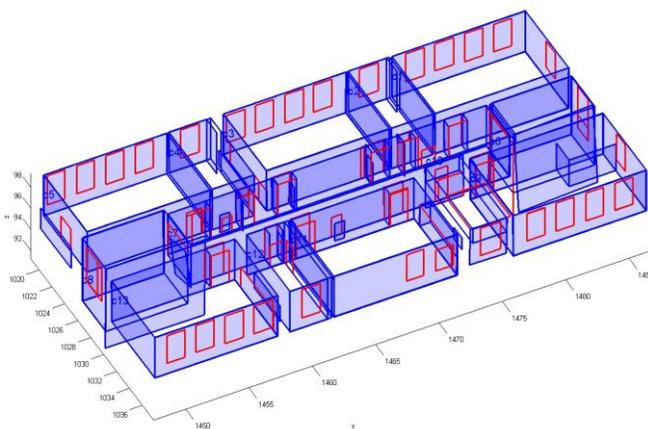


Figure 6. Results for all rooms on the second floor. The color scheme is the same as in Figure 5a.