3D Reconstruction of Interior Wall Surfaces Under Occlusion and Clutter

Antonio Adan
Department of Electrical Engineering,
Electronics, and Automation
Castilla La Mancha University
Ciudad Real, Spain
Antonio.Adan@uclm.es

Daniel Huber
The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
dhuber@cs.cmu.edu

***Abstract*—Laser scanners are often used to create 3D models of buildings for civil engineering applications. The current manual process is time-consuming and error-prone. This paper presents a method for using laser scanner data to model predominantly planar surfaces, such as walls, floors, and ceilings, despite the presence of significant amounts of clutter and occlusion, which occur frequently in natural indoor environments. Our goal is to recover the surface shape, detect and model any openings, and fill in the occluded regions. Our method identifies candidate surfaces for modeling, labels occluded surface regions, detects openings in each surface using supervised learning, and reconstructs the surface in the occluded regions. We evaluate the method on a large, highly cluttered data set of a building consisting of forty separate rooms.**

***Keywords*-3D model, laser scanner, point cloud, opening detection, occlusion reasoning**

I. Introduction

Laser scanners are increasingly being used to create 3D models of buildings and other facilities in the architecture, engineering, and construction (AEC) domain for a variety of purposes, including building renovation, cultural heritage preservation, and facility management [5], [12]. These models are usually manually constructed – a labor-intensive and tedious process. Our long-term goal is to develop methods to automate this process using computer vision and machine learning techniques (Figure 1).

While there has been much research on mapping and modeling of building interiors and exteriors using robots [8], aerial platforms [25], and terrestrial laser scanners [10], relatively little attention has been given to the detailed modeling of wall surfaces [2], [18]. Exterior façade modeling methods operate under the assumption that the surface being modeled is relatively unobstructed. In indoor environments, objects like furniture and wallhangings, frequently occlude the wall surfaces, making the modeling problem more challenging. In this paper, we address the problem of detailed modeling of wall surfaces in natural, cluttered environments. Our goal is to recover the wall surface, to identify and model openings, such as windows and doorways, and to fill occluded surface regions.

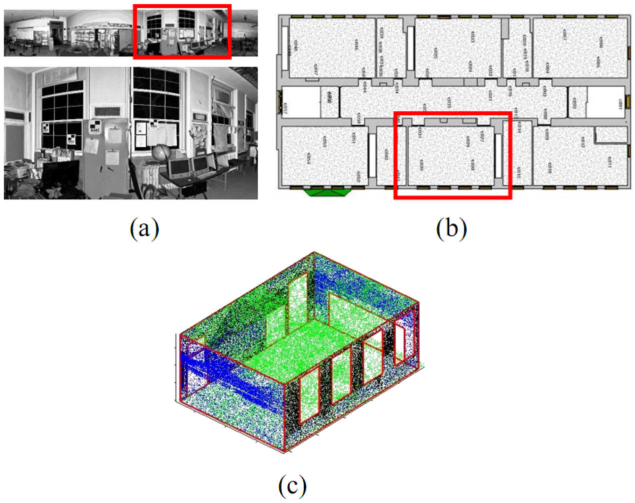


Figure 1. Overview. We model surfaces from 3D data of complex scenes with occlusion and clutter (a) portion of the range and reflectance images from one laser scan; (b) ground truth floor plan for the building with modeled room highlighted by the rectangle; (c) reconstructed model of the room produced by our algorithm.

Our approach utilizes 3D data, such as that produced by a terrestrial laser scanner operating from one or more locations within a room. Although we focus on wall modeling, the method can be applied to the easier case of floors and ceilings as well. The method consists of four main steps: 1) *Wall detection* – The approximate planes of the walls, ceiling, and floor are detected using projections into 2D followed by a Hough transform (Section III-A). 2) *Occlusion labeling* – For each wall surface, ray-tracing is used to determine which surface regions are sensed, which are occluded, and which are empty space (Section III-B). 3) *Opening detection* – A learning-based method is used to recognize and model openings in the surface based on the occlusion labeling (Section III-C). 4) *Occlusion reconstruction* – Occluded regions not within an opening are reconstructed using a hole-filling algorithm (Section III-D).

The primary contribution of this research is the overall approach, which focuses on addressing the problem of clutter and occlusions and explicitly reasons about the missing information. Our approach is unique in that it distinguishes between missing data from occlusion versus missing data in an opening in the wall. Secondly, we propose a learning-based method for detecting and modeling openings and distinguishing them from similarly shaped occluded regions. Finally, we propose and use methods for objectively evaluating reconstruction accuracy, whereas previous façade modeling work has focused on primarily on subjective visual quality.

II. Related Work

There is a large body of research on reconstruction of building interiors and exteriors using laser scanners [1]–[4], [10], [13], [18], [19], [23], [24] as well as imagery and video [6], [17]. Much of the emphasis in previous work has been on creating visually realistic models rather than geometrically accurate ones. Examples in this category include methods by El-Hakim et al., which focused on indoor environments [9], and Frueh et al., which focused on outdoor environments [10].

Image- and video-based approaches for modeling buildings are also well established. Early work by Debevec used images to semi-automatically model building exteriors [6]. More recently, Pollefeys et al. used video to model urban environments from a moving vehicle [17]. So far, these approaches are not accurate enough for the AEC domain, but recent advances, such as Manhattan world stereo [11], show promise.

As laser scanner technology has progressed, researchers have begun using these devices to construct detailed models of walls and building façades. Thrun et al. developed a plane extraction method based on the expectation-maximization algorithm [24], and other researchers have proposed plane sweep approaches to find planar regions [4], [13]. Stamos et al. combine planar patch modeling with meshes in complex areas [23]. Some façade modeling methods also extract window openings, which are normally detected based on regions where the data density is zero or very low and then modeled using rectangles [3], [18]. Model-based approaches can also be used to predict patterns in façades using top-down processing [1], [19].

Occlusions sometimes occur in façade reconstruction (e.g., by trees, parked cars, etc.), but most existing work assumes that occluded regions can be observed from another viewpoint [10], [18]. Such methods would not work in situations with significant, unavoidable occlusions or windows with objects on or in them (e.g., air-conditioners, pictures, etc.). An alternative approach is to identify another region on a façade that matches the occluded region and substitute the missing data with the observed data [2], [26]. The more general problem of reconstructing occluded surfaces in 3D is also known as hole filling or surface completion, and there are many proposed solutions [5], [21]. Surfaces in buildings often have a more constrained structure, and, consequently, specialized approaches can be applied in many cases [7], [10], [16], [22].

One key distinction between our proposed method and previous work is that our method is explicitly reasons about occlusions and is therefore capable of operating in natural, heavily occluded environments. In contrast, methods such as Budroni's do not consider the occlusion [4]. In that work, and in other previous work (e.g., [13], [24]), the test data consists of hallways with no furniture or other potentially occluding objects. In our data, on average, only 50% of the wall surface area was observed (see Section IV), and previous methods would likely perform poorly in the face these occlusion levels.

III. Wall Surface Modeling and Reconstruction

Our wall surface modeling algorithm uses 3D data obtained from fixed, known locations throughout a facility. The data from one location is known as a scan. We assume the scans are already registered (i.e., aligned in a common coordinate system), and that the "up" direction is known. Data registration is a well-studied problem, and methods to manually or automatically register scans are commercially available.

Laser scanners produce enormous data sets – our test data contains over 500 million points. To cope with this quantity of data, we adopt a voxel-based scheme for the algorithm's early stages. We denote the space of 3D points $\mathbf{P}$ and the overlaid space of voxels as $\mathbf{V}$. Each point p_i in the set of 3D points $\mathbb{P}$ is quantized into a voxel in $\mathbf{V}$, and the centers of the occupied voxels form a new, sparser set of points $\mathbb{V}$. We implicitly assume that the walls are aligned with an axis of $\mathbf{V}$, but it is straightforward to define separate voxel spaces for each wall being modeled. We also assume that the surfaces to be modeled are planar. The extension to non-planar surfaces is the subject of ongoing work. The next several sub-sections detail the steps of our algorithm.

A. Wall Detection

The first step in our algorithm is to detect and estimate the surfaces to be modeled. This detection is performed by a method similar to that described in [15]. The modes of a histogram of height values z_i of each point v_i in $\mathbb{V}$ determine the height of the ceiling and floor. Similarly, projecting the points' horizontal coordinates (x_i, y_i) onto a horizontal plane gives a 2D histogram from which wall surfaces are extracted using a Hough transform (Figure 2). The result is a set of surfaces $\mathbb{S}$ to be further processed. Each surface $S_j \in \mathbb{S}$ is modeled by the set of voxels bounded by a rectangle encompassing the occupied voxels used to define the surface plane. Note that S_j can be treated as a 2D image I_j by projecting orthographically along the normal direction. The remaining steps in the algorithm operate on each surface $S_j \in \mathbb{S}$ individually.

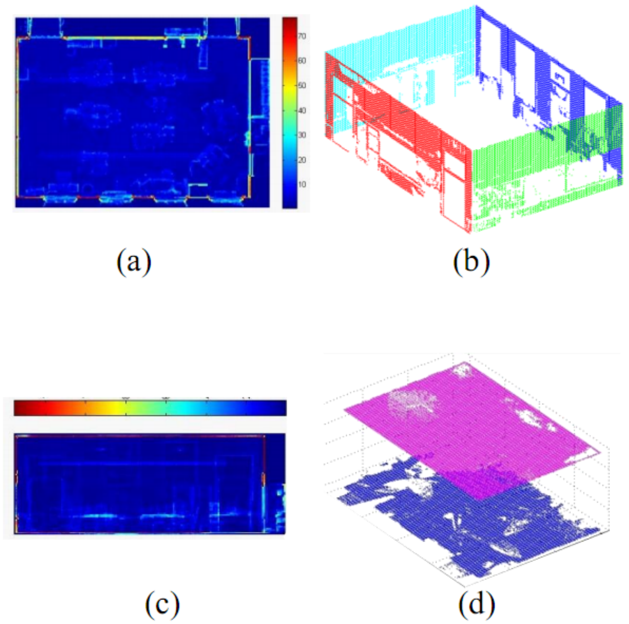


Figure 2. Wall detection. Histograms obtained from vertical (a) and horizontal (c) projections of the data and points corresponding to the walls (b), floor and ceiling (d) of the room.

B. Occlusion Labeling

In this step, each voxel in surface S_j is assigned one of three labels: occupied (F – for full), empty (E), or occluded (O). The labeling is performed once for each scanning position that observes the surface, and then the information is merged together into a unified representation (Figure 3).

The previous step labeled voxels as empty or occupied based on whether a surface was detected at that location. We denote this labeling as L_0. Without additional information, it is not possible to distinguish between a voxel that is truly empty and one that is merely occluded. We resolve this problem by using ray-tracing to explicitly reason about occlusions between the sensor and the surface being modeled.

Let $\mathbb{O} = \{O_1, O_2, \ldots O_K\}$ be the set of scans from which the surface S_j is visible. For each scan O_k, a labeling L_k is generated by tracing a ray from the origin of the scanner to each measured point that lies within the frustum formed by the scanner origin, the bounds of the surface being modeled, and a maximum distance located a fixed distance beyond the surface. The labeling is initialized using the L_0. Voxels between the origin and an occupied voxel are labeled as empty (E), and voxels between an occupied voxel and the maximum distance are labeled as occluded (O).

After this ray-tracing labeling, we have K labels for each voxel in the surface S_j. We integrate these labels with the labels from the initial labeling L_0 to form a combined labeling L_F. L_F is initialized using L_0, and then the following rule is applied for each voxel v in S_j:

$$\text{If } L_0(v) = E \text{ and } L_j(v) = O, \forall j = 1, 2 \ldots K,$$

$$\text{then } L_F(v) = O \tag{1}$$

In other words, a voxel is considered occluded if it is occluded from every viewpoint. This integration process is similar in spirit to an evidence grid [14], but in this case, the accuracy of the sensor is very good, so it is not beneficial to model the sensor readings probabilistically.

The voxel space $\mathbf{V}$ is well suited for processing the initial very large data sets efficiently. However, for detailed

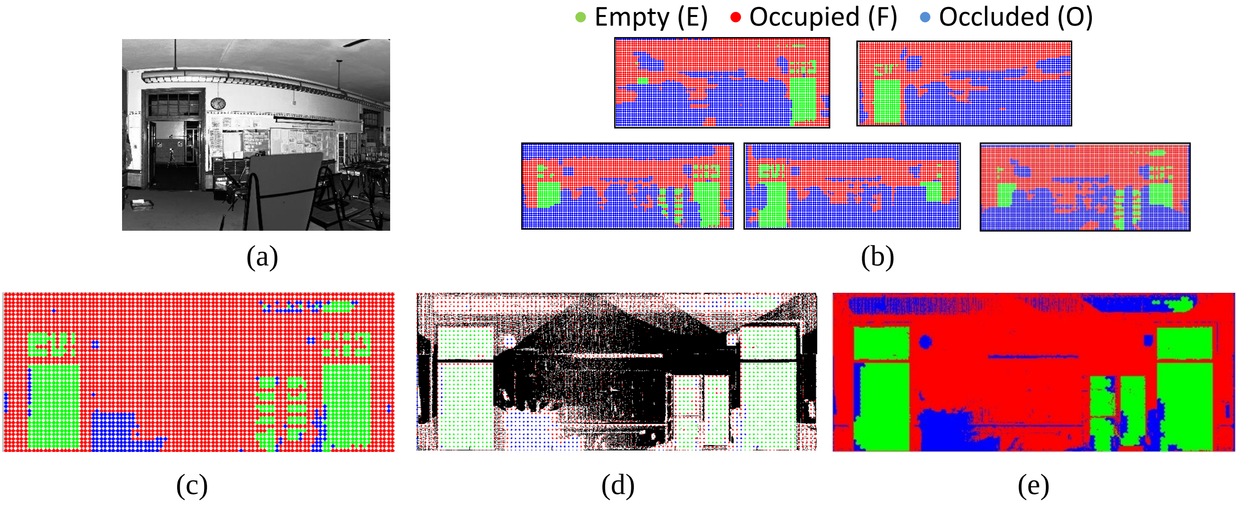


Figure 3. Occlusion labeling. (a) Reflectance image from one of five scans used to model the wall. (b) Ray-tracing is used to label occlusion regions on the surface. (c) Labels from multiple laser scans are integrated into a single representation. (d-e) A high resolution labeling (e) is inferred using a region-growing algorithm based on seeds from the low resolution data (d).

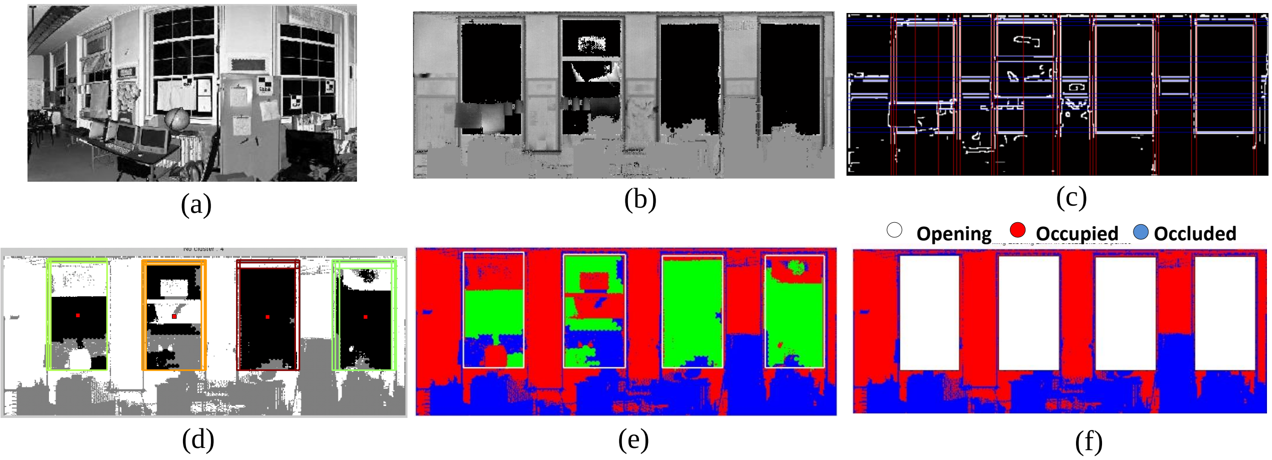


Figure 4. Opening detection. (a) Reflectance image of a wall surface. (b) Corresponding depth image. (c) Detected lines overlaid on edge image. (d) Openings detected by the SVM-based detector, with each cluster of openings in a different color. (e) Prototype openings after clustering, superimposed on I_F. (f) Final labeling with openings masked out.

modeling of wall surfaces, we would like to use the maximum resolution of data possible. To accomplish this, we transform the voxel data labeling into a high resolution 2D representation. The voxel data is about 1/25th of the size of the full resolution data. We use the voxel centers as seeds to a region growing algorithm that fills in the gaps in the data where possible.

For each surface S_j, the 2D representation is stored as an image I_j which is located at the position of a plane which is fit to the raw points that fall into the voxels in S_j. These points are projected orthographically onto the plane. Additionally, the centers of voxels labeled empty or occluded are also projected onto the plane in the same manner (Figure 3d). The resolution of I_j is chosen such that the pixels are approximately the same spatial resolution as the raw points in the occupied regions. The remaining pixels in I_j are assigned a new "unknown" label (U).

The goal of the region growing algorithm is to infer the labels of unknown pixels in the empty and occluded regions. The unknown pixels in the occupied regions will be filled later (in the final step). The algorithm is initialized with $I_j^0 = I_j$ (the superscript t indicates the iteration number) and proceeds by iteratively applying the following rules to all pixels in I_j^t labeled U (Figure 3e). For clarity, we omit the j subscript.

$$I^{t+1} = O \text{ if } I^t(x,y) = U \text{ and } d_{min}(I^t(x,y), \mathbb{I}_E^t) > \alpha$$
$$\text{and } d_{min}(I^t(x,y), \mathbb{I}_O^t) < \beta \tag{2}$$

$$I^{t+1} = E \text{ if } I^t(x,y) = U \text{ and } d_{min}(I^t(x,y), \mathbb{I}_O^t) > \alpha$$
$$\text{and } d_{min}(I^t(x,y), \mathbb{I}_E^t) < \beta, \tag{3}$$

where $\mathbb{I}_E^t$ and $\mathbb{I}_O^t$ are the sets of pixels in I^t labeled E and O respectively and $d_{min}(a, \mathbb{B})$ is the minimum Euclidean distance between the pixel a and any pixel in the set $\mathbb{B}$. The thresholds α and β control the radius of influence of occluded and empty pixels. The algorithm terminates when no pixels are changed in an iteration.

C. Opening Detection

Detecting the boundaries of openings in a wall is difficult because an opening may be partially occluded, and some regions in an opening may be labeled as occupied (e.g., a window-mounted air conditioner). Our algorithm, which detects rectangular openings, uses features derived from labels and depth edges in the data to learn a model of the size, shape, and location of openings using a Support Vector Machine (SVM) classifier.

We first obtain a range image R_j of the same resolution as I_j by projecting points within a distance d of S_j onto the modeled plane (Figure 4b). Edges are detected using Canny's algorithm, and essential lines are identified using the Hough transform (Figure 4c).

Given a generic opening candidate Θ, represented as a rectangular region in I_j with width w and height h, we define a 14-component feature vector $V = \{v_1, v_2, \ldots v_{14}\}$. We denote the width and height of the surface S by W and H respectively. The components of V are defined as follows: v_1 = area of Θ; $v_2 = w/h$; $v_3 = w/W$, $v_4 = h/H$; $(v_5 \ldots v_8)$ = distances from the sides to the edges of the wall; v_9 = RMS plane fit residual; $(v_{10} \ldots v_{12})$ = percentages of labels (E, F, O); v_{13} = number of interior rectangles; and v_{14} = number of interior inverted U-shapes (i.e., rectangles located at the bottom of the wall). Feature v_{13} is directed at features that may belong to a window frame or individual window panes. Feature v_{14} is an analogous

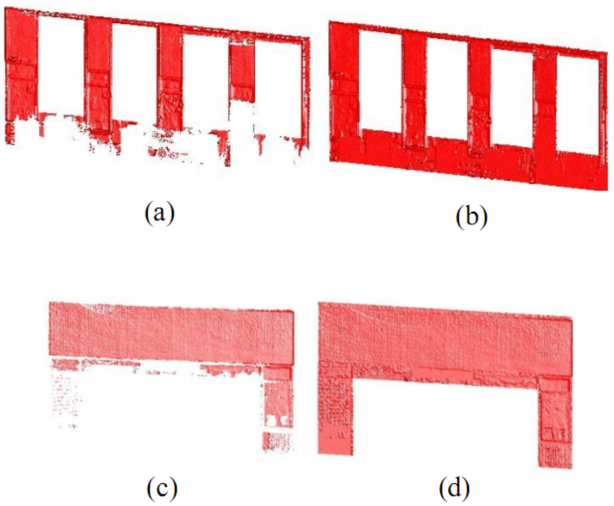

(a)
(b)
(c)
(d)

Figure 5. Occlusion reconstruction. Two examples showing the surface before (a, c) and after (b, d) reconstruction.

case for doors, which often exhibit U-shaped interior frames. We use an SVM with a radial basis function (RBF) kernel to learn a model of openings from these features using manually labeled training data.

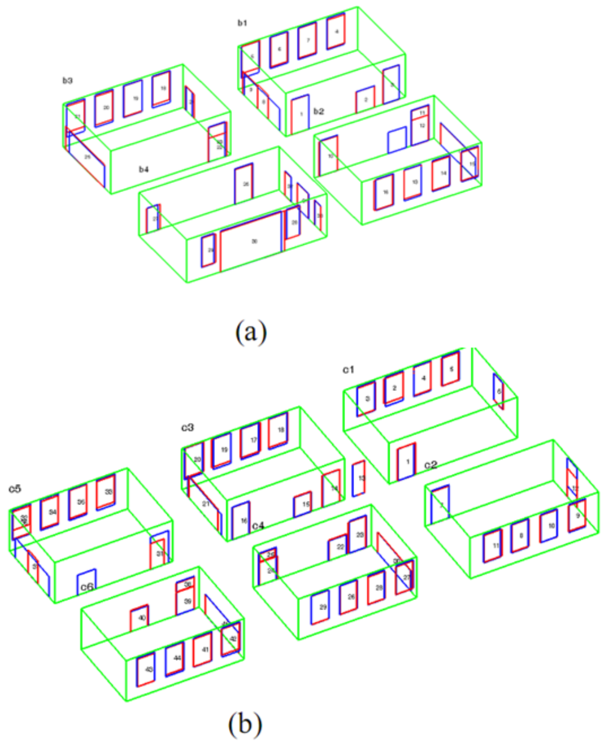
At run time, we enumerate opening candidates from the set of essential lines detected in a surface. Candidates are filtered out if there is not adequate support from the underlying edge image – at least $t_{supp}\%$ of a candidate's boundary must contain detected edges. Often, the classifier will label several overlapping rectangles as openings. We use k-means clustering to group the rectangles, and average the boundary estimates within each cluster to determine the final parameters for each opening.

D. *Occlusion Reconstruction*

Once the openings are determined, the final step is to fill in the wall surface in the occluded region (Figure 5). The regions within openings are excluded from this process. We use the gap-filling technique described by Salamanca et al. [21], which is a 3D extension of the Markov Random Field inpainting algorithm from Roth et al. [20]. Before filling the holes, spikes in the data (caused by mixed pixels in the range data) are filtered using a median filter. For aesthetic purposes, the reconstructed regions are degraded using Gaussian noise of the same magnitude as found in the sensed data.

IV. Experimental Results

We evaluated our algorithm using laser scanner data of a two story building which consists of forty rooms. Scans were obtained using a state of the art laser scanner operated by a professional 3D scanning service provider. The data set contains 225 scans throughout the facility for a total of over three billion points.



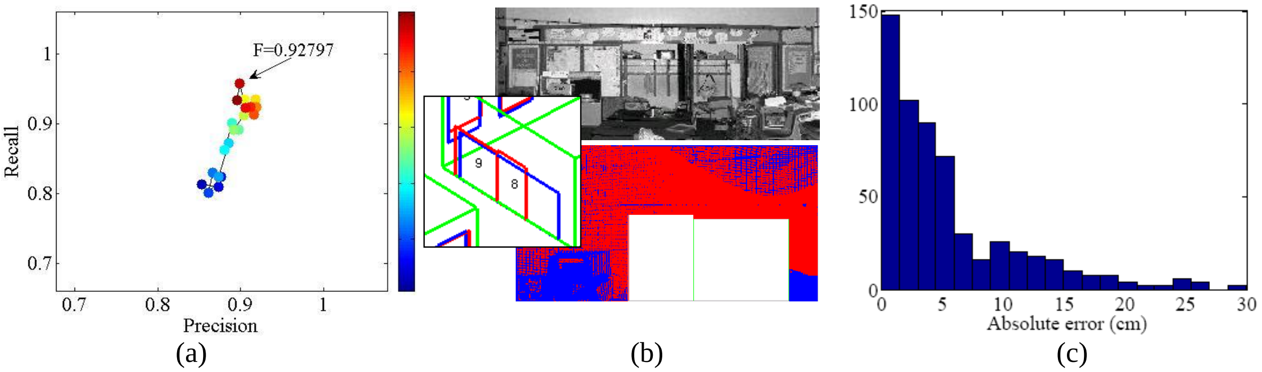
(a)
(b)

Figure 6. Reconstructed model of the facility corresponding to the first floor (a) and second floor rooms (b). The detected openings are shown in red, while the ground truth openings are shown in blue.

In our experiments, we evaluated voxel sizes of 10 and 5 cm. The lower size approached the limit of memory usage in our implementation, but smaller voxel sizes also increase the risk of missing wall sections due to the fact that most walls are not perfectly planar. The resolution of the images I_j was set to 2 cm/pixel, which is the resolution of the final reconstruction.

Our algorithm has a few free parameters, which were set by evaluation on an independent subset of the data. The Hough transform threshold for the number of edge point votes needed for line detection was set to 90, and the edge support threshold t_{supp} was set to 50%. The thresholds α and β were 1 and 2 times the voxel size respectively. The SVM classifier comprising the opening detector was trained on an set of 370 examples (split 50/50 between openings and non-openings). This data was further partitioned into a training set (95%) and validation set (5%) for determining the classifier parameters and preventing overfitting.

The complete algorithm was tested on 10 of the 40 rooms (Figure 6). The wall, floor, and ceiling detection step performed flawlessly on these rooms. These results are encouraging, but the good performance can partially be explained by the simple rectangular structure of the rooms. However, the surfaces were significantly occluded and the



(a) (b) (c)

Figure 7. (a) Precision-recall curves for the opening detector performance. (b) The detector failed on several instances, such as this one, where some doors of the closet were not open during data collection. (c) The histogram of the magnitudes of errors in opening boundary positions.

outside walls are almost entirely filled with windows, which makes the wall detection fairly challenging. On average, 35% of the wall area was occluded, 15% fell within an opening, and the remaining 50% was unoccluded wall surface.

We focused most of our analysis on understanding the performance of the opening detection and modeling steps, since this aspect of the algorithm is considerably less studied than planar wall modeling. We considered two aspects of the performance. First, how reliably can the openings be detected, and second, among the detected openings, how accurately are they modeled? To answer the first question, we compared the detected openings with openings in the ground truth model (i.e., doorways, windows, and closets). The precision-recall curves for the detector, at 5 and 10 cm voxel resolutions, are shown in Figure 7a. We compute the best threshold using the F-measure ($F = 2PR/(P+R)$, where P = precision and R = recall). At this threshold, the algorithm correctly detects 93.3% of the openings (70 out of 75) with 10 cm voxels, and 91.8% with 5 cm voxels. Failed detections mainly occur in regions of severe occlusion and in closets for which the doors were closed during data collection (Figure 7b).

We addressed second question, regarding the accuracy of the reconstructed openings, by comparing the ground truth positions of the sides of each opening with the positions estimated by our algorithm. We define the absolute error for modeling one side of an opening as the absolute magnitude of the difference between the ground truth and modeled position of that edge in the direction normal to the ground truth edge and also within the plane of the modeled surface. The relative error normalizes the absolute error by the ground truth width (for the left and right sides) or height (for the top and bottom) of the opening. The average absolute error was 5.39 cm with a standard deviation of 5.70 cm, whereas the average relative error was 2.56%.

V. Discussion and Future Work

This work on reconstruction of wall surfaces is an exciting first step toward our long-term goal of automated reverse engineering of buildings. Our approach worked very well for basic wall detection and modeling, despite the high levels of occlusion and missing data in the openings. The opening detection performance was good, and the detection failures were reasonable for the given data. The accuracy of the boundaries of the openings is good, but could be improved. Overall modeling accuracy for the AEC domain typically needs to be at least within 2.5 cm of the ground truth. On our evaluation data, 36% of the boundaries fall within this accuracy range. The final reconstructions, with the occluded areas filled in, are visually consistent with the manually created ground truth model. We also evaluated the algorithm on the easier problem of floor and ceiling reconstruction, and no openings were detected in these surfaces (which was the correct result).

In our ongoing work, we are investigating ways to extend our reconstruction algorithms to support more complex room geometries, curved surfaces, and non-rectangular openings. Although such complexities are infrequent, they occur often enough that the situation should be addressed. The opening detection and modeling algorithm may be limited in the boundary accuracy by the simple model of an opening that we are using. Windows and doorways typically have layers of moldings around them and (for windows) crossing them. These patterns can confuse the current detection algorithm because it only has a rudimentary concept of these features. A more sophisticated model of the geometry around an opening could potentially improve the modeling robustness. We are also considering ways to incorporate reflectance information as well as data from different depths relative to the modeled surface (e.g., recessed regions that occur within a window).

Another area that we are looking at is improving the semantics of the reconstructed model by explicitly classifying the openings (e.g., as windows, doors, or closets). Such a classification scheme could also improve the detection performance, since class-specific detectors may be more reliable than the current, generic one.

Finally, we are investigating efficiency improvements, as the current implementation is not optimized for speed. We are looking into ways to improve the ray-tracing performance, for example by using graphics hardware to parallelize the task.

Acknowledgments

This material is based upon work supported, in part, by the National Science Foundation under Grant No. 0856558 and by the Pennsylvania Infrastructure Technology Alliance. We thank Quantapoint, Inc., for providing experimental data. 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.

References

[1] S. Becker. Generation and application of rules for quality dependent facade reconstruction. *Journal of Photogrammetry and Remote Sensing*, 64(6):640–653, 2009. 2

[2] J. Böhm. Facade detail from incomplete range data. In *Proceedings of the ISPRS Congress*, Beijing, China, 2008. 1, 2

[3] J. Böhm, S. Becker, and N. Haala. Model refinement by integrated processing of laser scanning and photogrammetry. In *Proceedings of 3D Virtual Reconstruction and Visualization of Complex Architectures (3D-Arch)*, Zurich, Switzerland, 2007. 2

[4] A. Budroni and J. Böhm. Toward automatic reconstruction of interiors from laser data. In *Proceedings of 3D-ARCH*, February 2005. 2

[5] J. Davis, S. R. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. In *Proceedings of the Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, 2002. 1, 2

[6] P. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proceedings of ACM SIGGRAPH*, pages 11–20, 1996. 2

[7] F. Dell'Acqua and R. Fisher. Reconstruction of planar surfaces behind occlusions in range images. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(4):569–575, 2002. 2

[8] S. El-Hakim. Three-dimensional modeling of complex environments. In *Videometrics and Optical Methods for 3D Shape Measurement (SPIE vol. 4309)*, 2001. 1

[9] S. F. El-Hakim, P. Boulanger, F. Blais, and J.-A. Beraldin. A system for indoor 3D mapping and virtual environments. In *Proceedings of Videometrics V (SPIE v. 3174)*, pages 21–35, 1997. 2

[10] C. Frueh, S. Jain, and A. Zakhor. Data processing algorithms for generating textured 3D building facade meshes from laser scans and camera images. *International Journal of Computer Vision (IJCV*, 61(2):159–184, 2005. 1, 2

[11] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Reconstructing building interiors from images. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 80–87, 2009. 2

[12] GSA. GSA BIM guide for 3D imaging, version 1.0, January 2009. http://www.gsa.gov/bim. 1

[13] D. Hähnel, W. Burgard, and S. Thrun. Learning compact 3D models of indoor and outdoor environments with a mobile robot. *Robotics and Autonomous Systems*, 44(1):15–27, 2003. 2

[14] H. Moravec. Robot spatial perception by stereoscopic vision and 3D evidence grids. Technical Report CMU-RI-TR-96-34, Carnegie Mellon University, September 1996. 3

[15] 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. 2

[16] M. Pauly, N. J. Mitra, J. Giesen, M. Gross, and L. J. Guibas. Example-based 3D scan completion. In *Proceedings of the Third Eurographics Symposium on Geometry Processing*, 2005. 2

[17] M. Pollefeys, D. Nister, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3D reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167, 2008. 2

[18] S. Pu and G. Vosselman. Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(6):575–584, 2009. 1, 2

[19] N. Ripperda and C. Brenner. Application of a formal grammar to facade reconstruction in semiautomatic and automatic environments. In *Proceedings of AGILE Conference on Geographic Information Science*, Hannover, Germany, 2009. 2

[20] S. Roth and M. Black. Fields of experts: A framework for learning image priors. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 860–867, San Diego, California, 2005. 5

[21] S. Salamanca, P. Merchan, E. Perez, A. Adan, and C. Cerrada. Filling holes in 3D meshes using image restoration algorithms. In *Proceedings of the Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, Atlanta, GA, 2008. 2, 5

[22] A. D. Sappa. Improving segmentation results by studying surface continuity. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 2, pages 929–932, 2002. 2

[23] I. Stamos, Y. Gene, G. Wolberg, and S. Zokai. 3D modeling using planar segments and mesh elements. In *Proceedings of 3D Data Processing, Visualization, and Transmission (3DPVT)*, pages 599–606, 2006. 2

[24] S. Thrun, C. Martin, Y. Liu, D. Hähnel, R. Emery-Montemerlo, D. Chakrabarti, and W. Burgard. A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots. *IEEE Transactions on Robotics*, 20(3):433–443, 2004. 2

[25] V. Verma, R. Kumar, and S. Hsu. 3D building detection and modeling from aerial lidar data. In *Computer Vision and Pattern Recognition*, volume 2, pages 2213–2220, 2006. 1

[26] Q. Zheng, A. Sharf, G. Wan, Y. Li, N. J. Mitra, D. Cohen-Or, and B. Chen. Non-local scan consolidation for 3D urban scenes. *ACM Transactions on Graphics*, 29(4), 2010. 2