NSF GRANT # CMMI-0856558
NSF PROGRAM NAME: Automating the Creation of As-built Building Information Models

Methods for Automatically Modeling and Representing As-built Building Information Models

Daniel Huber
Robotics Institute, Carnegie Mellon University

Burcu Akinci
Civil and Environmental Engineering, Carnegie Mellon University

Antonio Adan
Electrical Engineering Department, University of Castilla-La Mancha

Engin Anil
Civil and Environmental Engineering, Carnegie Mellon University

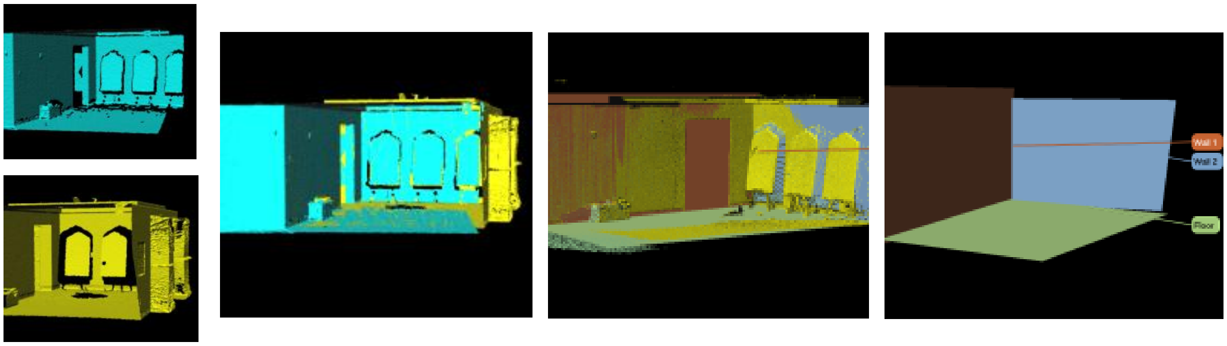
Brian Okorn
Robotics Institute, Carnegie Mellon University

Xuehan Xiong
Robotics Institute, Carnegie Mellon University

Abstract: Laser scanners are increasingly being used to create information rich 3D models of the as-built or as-is conditions of buildings, infrastructure, and other facilities. These "as-built" Building Information Models (BIMs) are created through a time-consuming and error-prone manual process, which is one key barrier to widespread use of as-built BIMs in industry. This paper outlines our research group's progress in developing methods to automate the process of creating as-built BIMs and in creating suitable methods to represent and visualize the information that is unique to as-built BIMs.

1. Introduction. Most of the work on building information models (BIMs) today focuses on representing the condition of a facility as it was designed, rather than its condition as built or as used. While a BIM that represents the as-designed condition of a facility has many potential uses, the actual as-built conditions can differ significantly from the design, and the as-used conditions can change extensively throughout a facility's lifespan. The long-term fidelity and usefulness of a BIM critically depends on the ability to record and represent such changes.

Three dimensional (3D) imaging systems, such as laser scanners, enable the efficient capture of detailed as-built conditions of a facility in the form of sets of 3D point measurements, known as point clouds. Although these point clouds accurately represent the shape of a facility, the fact that points are only low-level, discrete measurements limits the usage of as-built data in this form. Point clouds do not possess any semantic information, such as whether a set of points belongs to a specific wall; nor do they contain high level geometric information, such as surface shape or the boundaries of building components. A high-level, semantic representation of the as-built condition allows analysis and manipulation of the model at the component level (e.g., walls and doors) rather than at the individual point level, which is more natural and efficient. These modeled components have a direct relationship to components in a design model, which enables comparison and analysis between corresponding components. A high-level representation is also more compact, since components can be summarized by a small number of parameters. Representing a wall by a plane equation and a description of its boundary requires much less storage than representing it with a million 3D points. Due to the



Scanning **Registration** **Geometric modeling** **Semantic labeling**

Figure 1. Overview of the points-to-BIM transformation process. Three dimensional data is acquired from fixed locations (Scanning) and aligned in a common coordinate system (Registration). Then, components are manually extracted from the resulting point cloud (Geometric modeling) and finally assigned meaningful labels (Semantic labeling). Notice that the commercial software does not automatically detect the doorway in the wall and that the top edges of the two walls do not align.

numerous advantages of high-level representations of as-built data, the general practice is to convert raw scanned point data into a high-level BIM representation. We call this conversion the *points-to-BIM transformation process* (Figure 1). Unfortunately, the state-of-the-art technique for the points-to-BIM transformation is a costly, time-consuming, and labor-intensive manual process. Furthermore, existing standards and methods for representing BIMs were developed primarily to support models derived from design data (as-designed BIMs), and the requirements for representing as-built BIMs are somewhat different from the representation needs for as-designed BIMs. These two issues – the difficulty in creating as-built BIMs and the limitations in representing as-built BIMs – are two key technological barriers to widespread as-built BIM usage.

The overarching goal of this research is to address these technological barriers to widespread as-built BIM usage by developing algorithms to facilitate the automatic creation of as-built models and by developing novel techniques that support the representation of as-built information in BIMs. In this work, we limit our attention to a subset of possible building components – walls, floors, ceilings, doorways, and windows. These components comprise a significant proportion of the visible elements of the envelope of a typical commercial or residential building. Walls, floors, and ceilings are sufficient to define the basic extents of a room, and these elements alone would be enough to support a variety of BIM usages, such as spatial program validation (i.e., space usage and planning), circulation, and security.

In this paper, we present an overview of our group's recent research on the problem of modeling and representing as-built BIMs. Our work spans four topics: modeling floor-plans of buildings (Section 2), using context to recognize and model building interiors in 3D (Section 3), detailed modeling of walls and other planar surfaces (Section 4), and methods for representing as-built BIMs (Section 5).

2. Algorithms for Modeling Floor-plans of Buildings. Architects and building managers often need blueprints of a facility's as-built or as-is conditions. These conditions may differ from the design blueprints, assuming they still exist at all. We are working on methods to automatically create accurate floor plan models of building interiors using laser scan data.

Our floor plan modeling algorithm is described in detail in Okorn 2010 [1]. In the research, we made three main contributions. First, we designed, implemented, and evaluated a novel method for automatically modeling vertical wall structures from 3D point clouds. Second, we developed several measures for evaluating the accuracy of floor plan modeling algorithms. To our knowledge, there are no accepted methodologies for objectively evaluating such algorithms. Third, we put forward the concept of strategically choosing cross-sections from a 3D model to optimally extract the salient objects (e.g., walls) while being minimally impacted by clutter.

Our approach for floor plan modeling is based on the observation that when the 3D points are projected onto the ground plane, the projected point density is usually highest at wall locations. The process is illustrated in Figure 3. The algorithm begins by automatically identifying and removing the floor and ceiling regions. This is accomplished by projecting the points onto the vertical axis and identifying the bottom-most and top-most local maxima of the resulting histogram. The data points contributing to these maxima correspond to the horizontal surfaces of the ceiling and floor. Next, a 2D histogram is formed by projecting the remaining points onto the ground plane. Linear structures are then extracted from this histogram using a Hough transform.

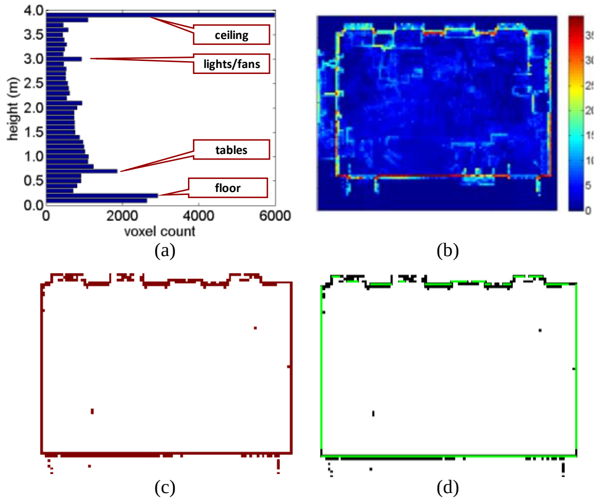


Figure 3. Floor-plan modeling. (a) The height histogram is a projection of the 3D data onto the vertical axis. The large maxima at the top and bottom correspond to the ceiling and floor heights. Variations in data density at other heights are indicative of the degree of clutter at each elevation. (b) Ground plane histograms are formed by projecting voxelized 3D data onto the x-y plane and accumulating the occupied voxel count into a histogram. The dense regions indicate vertical surfaces, which have a high probability of being wall segments. (c) The ground plane histogram is first thresholded to remove low density cells. (d) The Hough transform is then used to detect lines within this thresholded histogram (green detected lines overlaid onto thresholded data).

Finally, line segments may be "snapped" to the dominant orientations found in the facility by rotating them about their centroids if they are sufficiently close to the dominant orientation.

To evaluate the algorithm, we developed an objective measure of the accuracy of a floor plan with respect to ground truth data. The measure strives to correlate well with how a human would subjectively evaluate performance. The resulting evaluation measure consists of two parts. The first part measures line detection capability and is based on an object detection methodology, while the second part measures the modeling accuracy and conciseness.

This floor plan modeling algorithm is simple, straightforward, and works fairly well in practice (Figure 2). One challenge is that extensive clutter in real environments means that the wall structures may not always be readily visible within this histogram projection. We observe that clutter is not necessarily the same at all heights, and we propose strategies for determining the best choice of cross-section location (or locations) to use. Using histograms computed using only heights with low amounts of clutter results in significantly better floor plans.

Going forward, we hope to improve the performance of the algorithm by incorporating full 3D reasoning and detailed recognition of wall surfaces, as described in the next two sections. These more advanced methods should give us the capability of placing window and door openings accurately. In the longer term, we are looking at methods to integrate multi-room reasoning into the approach, which should improve the accuracy of back-to-back wall segments.

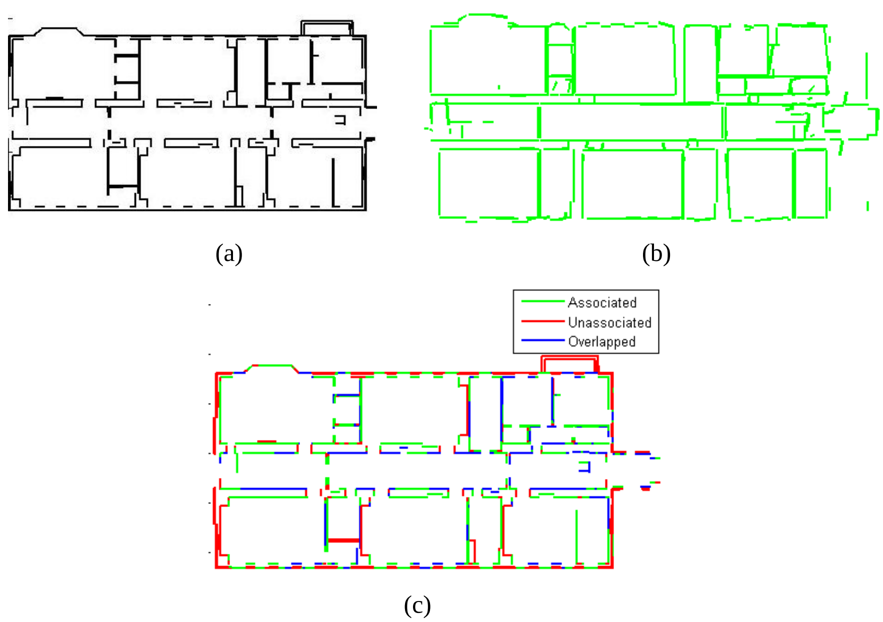


Figure 2. Floor-plan modeling results. (a) Ground truth floor plan of the first floor. (b) Floor plan generated by our algorithm. (c) Results of the evaluation of our model versus ground truth. The main areas that were not modeled are shown in red, and these areas correspond mainly to the exterior sides of exterior walls (which were not included in our input data) and interiors of closets, some of which were not visible in the scans.

3. Algorithms for Using Context to Recognize and Model Building Interiors. In this work, we are developing methods for automatically identifying and modeling the main structural components of building interiors – namely walls, floors, and ceilings. We hypothesize that context – information about the relationships between different components in the facility – is the key to achieving reliable and accurate performance on this problem. For example, if a surface is bounded on the sides by walls and is adjacent to a floor surface on the bottom and a ceiling on the top, it is more likely to be a wall than clutter, independently of the shape or size of that surface. In this way, the interpretation of multiple surfaces can mutually support one another to create a globally consistent labeling.

We have developed and evaluated an algorithm for modeling building interiors using context, and we have experimented with various types of contextual relationships, including adjacency, parallelism, coplanarity, orthogonality, and existence. For classifying the components, we use a machine learning framework based on graphical models and have extended and adapted several existing approaches to work within our problem domain.

The approach that we developed, which is detailed in Xiong 2010 [2], consists of four main steps (Figure 4). First, we encode the input point cloud data into a voxel structure to minimize the variation in point density throughout the data. Next, we detect planar patches by grouping neighboring points together using a region-growing method. We model the patch boundaries using a small number of 2D points on the plane. We use these planar patches, and the relationships between them, as the input to our context-based classification algorithm. The algorithm uses the contextual relationships to label the patches according to structural categories – wall, floor, ceiling, and clutter. Finally, we remove clutter patches from the scene and re-estimate the patch boundaries by intersecting adjacent components.

The key step in the algorithm is the context-based classification method. Our approach uses two types of features derived from the data. Local features encapsulate knowledge about individual patches, such as surface area and orientation. Contextual features

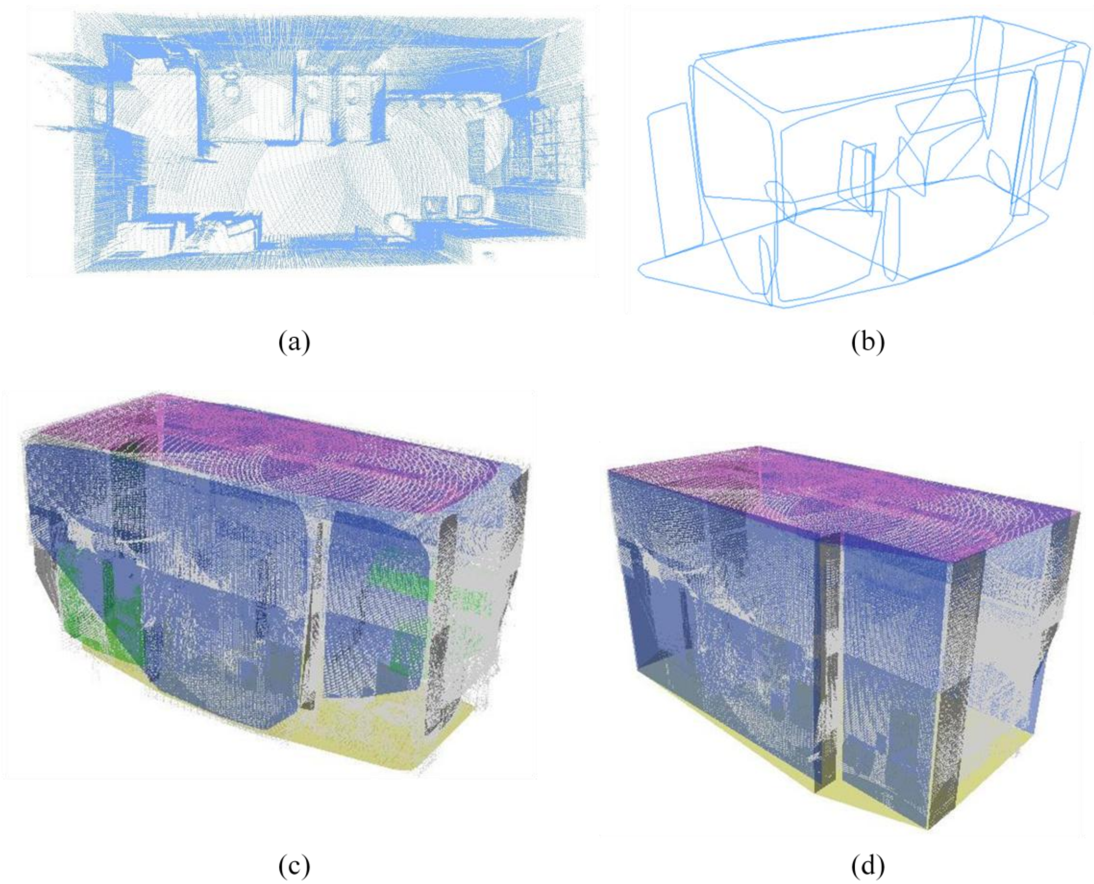
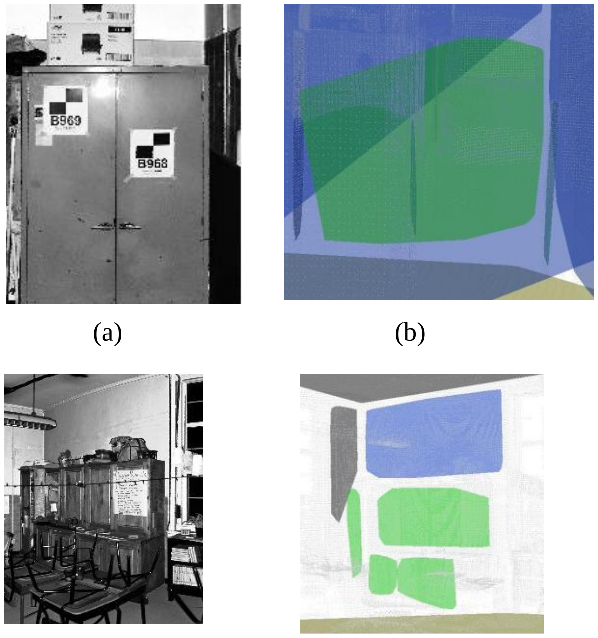


Figure 4. Context-based modeling. (a) First, the input point cloud is encoded in a voxel data structure. (b) Next, planar patches are detected and modeled using a region-growing algorithm. (c) Then, the detected patches are classified using a context-based classification algorithm (magenta = ceilings, yellow = floors, blue = walls, green = clutter). (d) Finally, boundaries are re-estimated and clutter patches are removed.



(a) (b)

(c) (d)

Figure 5. Context-based modeling results on some challenging cases. The algorithm correctly distinguishes items that look significantly like clutter, such as large cabinets (a-b) and bookcases (c-d). Reflectance images are shown in the left column, and the corresponding patches are shown in the right column. Green patches are labeled as clutter, Blue patches are walls, and yellow patches are floors.

encode information about a patch's neighborhood configuration. We investigated various relationships, including orthogonal, parallel, adjacent, and coplanar. We use a machine learning model known as conditional random fields (CRF) to combine these two types of features in an optimization framework. By maximizing the likelihood of the labels assigned to each patch (e.g., wall, ceiling, floor, or clutter) given the local and contextual feature values, the algorithm finds the optimal labeling for the patches, taking into consideration all of the labelings simultaneously. For example, a patch that is coplanar with other patches that are likely to be a wall will be more likely to be a wall as well.

We conducted experiments using data from 26 rooms of a school building that was professionally scanned and modeled. We divided the data into training, validation, and test sets and then evaluated the performance of the algorithm at labeling the detected planar patches. We found that the use of context does improve the performance of the recognition process. We compared the context-based algorithm to one that just uses local features, and we found that the context-based algorithm improved performance from 84% accuracy to 89% (Figure 5). The main failures of the algorithm are in unusual situations, such as the tops of the interiors of short closets (which are considered ceilings in the model). Our approach is effective at distinguishing clutter from surfaces of interest, even in highly cluttered environments. Finally, we found that the coplanar relationship is very helpful for addressing fragmentation of wall surfaces due to occlusions and large window and door openings.

4. Algorithms for detailed modeling of planar surfaces. One of the goals of our research is to explicitly reason about occlusions in the data in order to avoid problems caused by missing data. Most previous work on modeling building interiors focuses on environments with little or no clutter. One reason for this is that clutter causes occlusions, which makes modeling the surfaces of interest more difficult. For automated creation of BIMs to be useful in real-world environments, algorithms need to be able to handle situations with large amount of clutter, along with the resulting occlusions. For example, it is not practical to move all furniture out of a building before scanning it for creating an as-built BIM.

In this research, our goal is to model wall surfaces at a detailed level, to identify and model openings, such as windows and doorways, and to fill occluded surface regions. Our approach utilizes 3D data from a 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 floors and ceilings as well. The method consists of four main steps (Figure 6): 1) *Wall detection* – The approximate planes of the walls, ceiling, and floor are detected using projections into 2D followed by a Hough transform. 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. 3) *Opening detection* – A learning-based method is used to recognize and model openings in the surface based on the occlusion labeling. 4) *Occlusion reconstruction* – Occluded regions not within an opening are reconstructed using a hole-filling algorithm.

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.

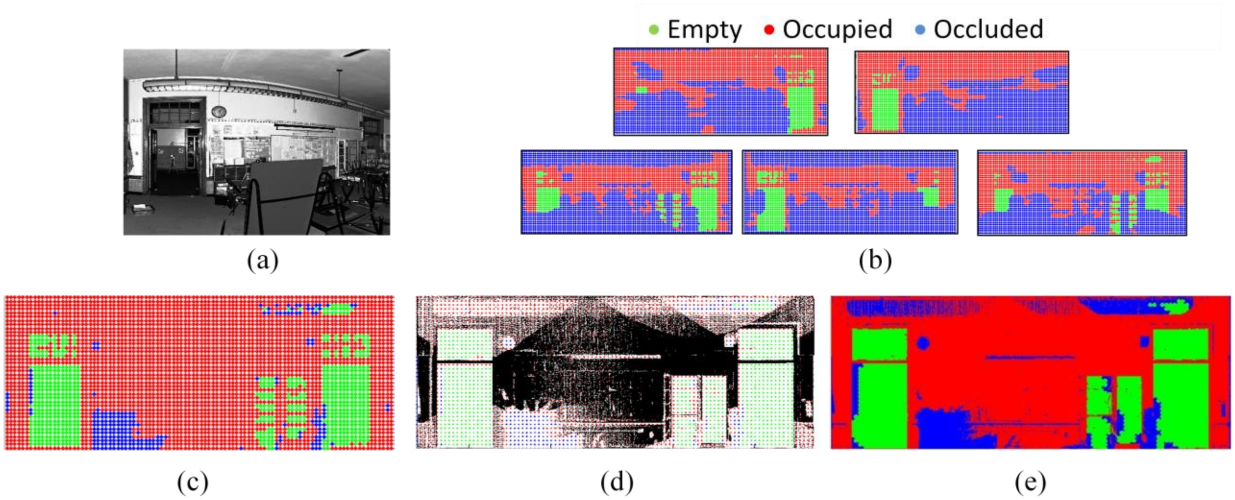


Figure 6. Detailed wall modeling example. (a) Reflectance image from one of five scans used to model the wall. (b) Ray-tracing is used to label occlusion regions on the surface. (c) Labels from multiple laser scans are integrated into a single representation. (d-e) A high resolution labeling (e) is inferred using a region-growing algorithm based on seeds from the low resolution data (d).

We have conducted extensive experiments on the test data of the school building that we described in the previous section. We found that the algorithm for detecting walls, ceilings, and floors work well for large rooms. The method found all the main walls within 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 outside walls are almost entirely filled with windows, which makes the wall detection fairly challenging. On average, 35% of the analyzed area was occluded, 15% fell within an opening, and the remaining 50% was unoccluded wall surface.

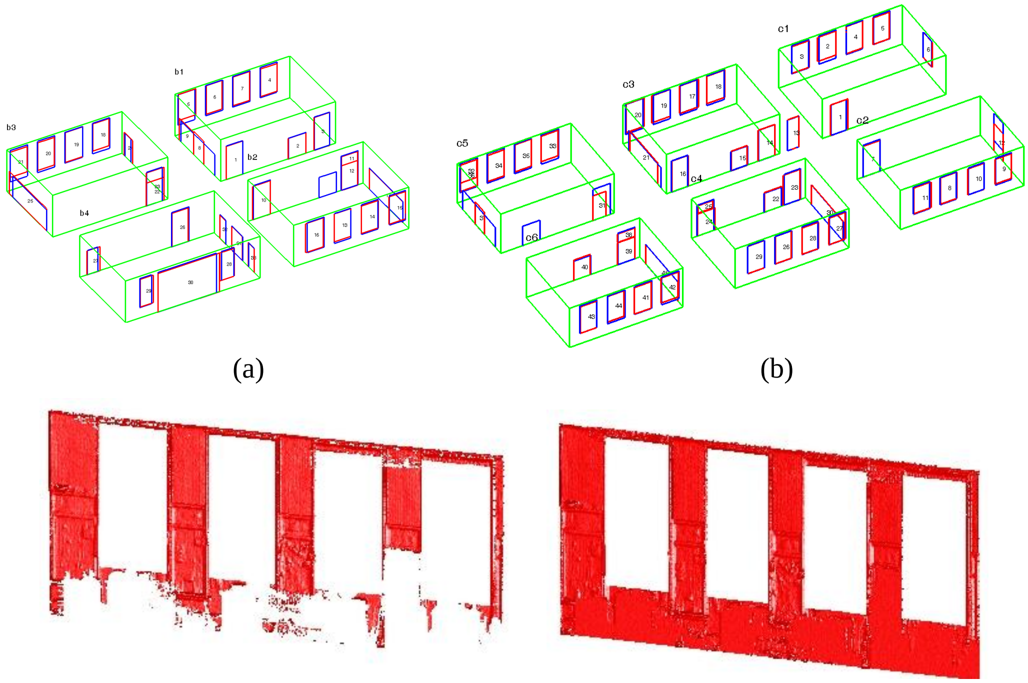
The algorithm for detecting and modeling openings in the walls performed well, achieving an average of 93% accuracy (Figure 7). Failed detections mainly occur in regions of severe occlusion and on closets where the doors were closed during data collection. Evaluation of the accuracy of the modeled openings is still underway, but the early results are promising. Details of the algorithm and the results of our experiments can be found in our technical report [3].

5. Methods for Representing As-built BIMs. Our research on this topic is in its early stages. Up to this point, we have been focusing on exploring the relevant existing capabilities of BIM representation in the existing standards, exploring similar problems in other domains, and identifying the representation requirements for as-built and as-is BIMs.

5.1. Exploring Existing Capabilities of BIM Representations. Initially we are concentrating on various exchange models in the Architecture, Engineering, and Construction / Facility Management (AEC/FM) industry. Most of these models focus on representing as-designed information rather than as-built information. Currently, the focus is on the Industry Foundation Classes (IFC), which is an exchange standard aimed at interoperability of various software utilized in different phases of design of structures. Although, in the development of IFC, use of the model for representing as-built information was not intended, it may be possible to use IFC models for this purpose through various representation mechanisms found in the model.

We are exploring the capabilities of IFC from two perspectives: how to objectify the meta-data and the recognized objects in IFC, and how to represent the data as a consistent whole. We have identified several options for representing occlusions and visible parts of components in IFC, and we are investigating which methods are the most effective and developing ways to visualize the results. Different options include creating separate components to represent the occluded and visible sections of a surface, using sub-components to represent occluded and visible regions, or using virtual elements to provide imaginary boundaries defining occluded regions.

A second question that we are investigating is how to relate the meta-data to the as-built model. IFC has the capability of representing the same object with multiple shape representations. This gives the model the flexibility to the represent the same object with different geometries and different contextual information. Alternatively, we can represent the as-built data by assigning the meta-data and the as-built model into a separate layer in the IFC file. We are currently investigating the tradeoffs between these two approaches.



(a) (b)

(c) (d)

Figure 7. Detailed planar surface modeling results. (a-b) Reconstructed model of the facility cooresponding 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. (c-d) Occlusion region reconstruction shown before (c) and after (d) occluded regions have been filled in.

5.2. Exploring Similar Problems in Other Domains. Laser scanners are being used for capturing large areas for GIS applications. Although the problem in GIS applications is different – there is no as-designed or as-built information, only surface geometry – storing the scan data and surface geometry in GIS databases is also a big challenge in this area. The triangular Irregular Network (TIN) approach gives the flexibility of representing a 3D surface at varying levels of detail. An algorithm determines the number of points required to represent the terrain and reduces the number of points selectively. A less detailed surface can be derived from a detailed surface triangulation for different scale maps. A similar approach can be taken to derive surface representations of as-built models.

5.3. Identifying Representation Requirements for Levels of Detail and for Deviations. Representation requirements for deviations and level of detail may vary depending on the type of analysis that is going to be performed on the data. For example, axis deviations can be represented by only representing the axis of a component (i.e., a wall), whereas for surface deviation analysis, a freeform surface representation may be required.

In the future, this study can be extended to comparing as-designed models to as-built models for extracting defected regions' information. For this purpose, differentiating deviations from defects and understanding what kinds of deviations are considered defects needs to be understood. Every type of deviation may not be considered as a defect, and also, severity of the deviation affects the decision whether a certain type of deviation defines a defect or not. Building performance guidelines are possible sources of information for analyzing representation requirements for defect analysis between as-built and as-designed models.

8. Future Work. Our results, while promising, are preliminary in nature. We are working to extend our results in a number of ways. First, we intend to unify the different modeling and recognition algorithms into a single framework based on our context-based recognition algorithm. This will involve extending the recognition algorithm to accommodate the results of our detailed surface analysis algorithms and adding recognition of windows and doors to the framework. Secondly, we are working on extending the recognition and modeling algorithms to reason about multiple rooms. In their current instantiation, the algorithms operate at the individual room level. We hypothesize that the constraints from adjacent rooms can be exploited to improve the interpretation of difficult to recognize surfaces. Finally, we are working to add a follow-on step that will convert the models produced by our algorithm into volumetric primitives, which are the desired format for most BIM applications.

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

10. References:

[1] 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] 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.

[3] A. Adan and D. Huber, “Reconstruction of Wall Surfaces Under Occlusion and Clutter in 3D Indoor Environments,” Robotics Institute, Carnegie Mellon University, Pittsburgh, PA CMU-RI-TR-10-12, April 2010.