Toward Automated Modeling of Floor Plans

Brian Okorn
Vanderbilt University
2301 Vanderbilt Pl., Nashville, TN 37235
brian.e.okorn@vanderbilt.edu

Xuehan Xiong, Burcu Akinci, and Daniel Huber
Carnegie Mellon University
xiong828@gmail.com, bakinci@cmu.edu,
dhuber@cs.cmu.edu.edu

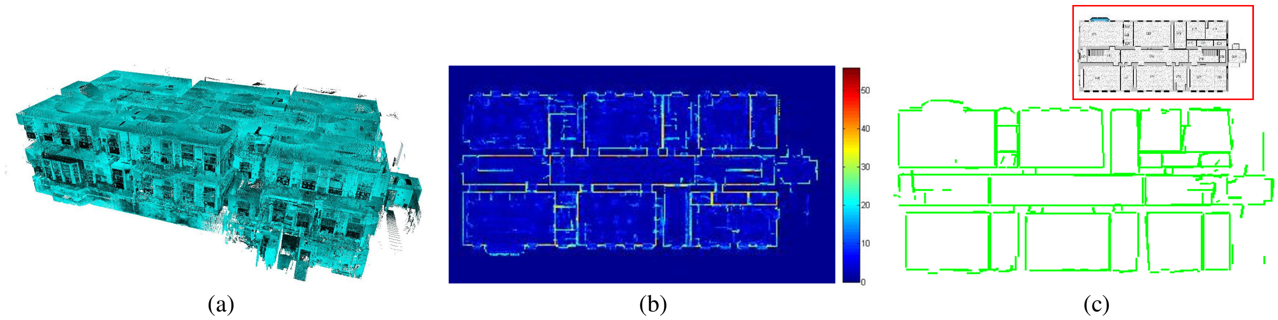


Figure 1. Automated floor plan modeling. Given a 3D point cloud of a facility (a), the algorithm seeks to automatically extract an accurate 2D floor plan model (i.e., top-down view) suitable for use in creating blueprints (e.g., ground truth shown in (c) inset). Our approach uses a Hough transform to extract line segments corresponding to walls from a histogram of point densities in a ground projection (b). The resulting floor plan model is shown in (c).

Abstract

This paper describes an automated method for creating accurate 2D floor plan models of building interiors. Our approach takes as input a 3D point cloud, which is obtained from laser scanners positioned at multiple locations in the facility. We use a histogram of height data to detect floor and ceiling data. We then project the remaining points onto a 2D ground plane and create a histogram of point density, from which line segments corresponding to walls are extracted using a Hough transform. Openings due to doors and windows arise implicitly from the modeling process. Our algorithm operates in unmodified, cluttered environments. We also propose a set of evaluation measures for objectively measuring floor plan modeling algorithm performance. Finally, we propose strategies for mitigating the effect of clutter through strategic selection of cross-sections containing minimal clutter. Our experiments on data from a 40-room school building achieve promising results.

1. Introduction

The creation of blueprints of the existing conditions of buildings and other facilities is a common task in the Architecture, Engineering, and Construction (AEC) domain. Since the actual "as-built" condition of a building can differ from the original plans – due to undocumented renovations, for example – it is often necessary to measure the as-built condition of a facility, even if the original blueprints are available. Predominantly, such as-built facility models are created using surveying methods and manual measurements.

Recently, laser scanners have begun to be utilized for as-built facility modeling, but the current practice for creating such models using laser scanners is still largely a manual process. In this paper, we propose a method for automatically transforming the three dimensional (3D) laser scanner measurements into a top-down floor plan model of a facility (Figure 1).

In current practice, creating a model of a facility using laser scanners involves three steps – data collection, data registration, and modeling. In the first step, laser scanners are placed at strategic locations throughout a facility, and a set of scans is obtained. Each scan consists of a set of 3D points, known as a point cloud. Next, the point clouds are aligned in a common coordinate system, and the data is cleaned up through filtering operations to remove noise, moving objects, or other clutter. Finally, in the modeling step, geometric models corresponding to walls, floors, ceilings, columns, and other structures are extracted from the registered point cloud. The resulting model may be 3D or 2D, depending on project requirements. For floor plan models, it is convenient to perform the modeling using 2D cross-sections of the data. Commercial software packages – either from independent developers or bundled with laser scanner hardware – support this as-built modeling process, but it is a manual and time-consuming operation. Human modelers may take several months to produce an as-built model of an average-sized office building, and the resulting models usually contain geometric errors.

Our goal is to develop tools and algorithms that can automate this modeling process. In this paper, we concentrate on the 2D modeling of building interiors. Specifically, we focus on accurate floor plan modeling of wall structures, as would be needed for blueprints. We assume that the point cloud data for the facility has already been collected and the individual point clouds registered. This registered point cloud serves as the only input to our algorithm. We also assume that the direction of gravity (i.e., the vertical direction) is known, which is generally the case with terrestrial laser scanners.

One of the main challenges to automating the as-built modeling process is handling clutter. Clutter is any 3D data that should not be included in the output model, including furniture, light fixtures, and interior decorations. Any modeling algorithm needs to be capable of functioning in cluttered environments, as it is often impractical to remove furniture and other clutter objects prior to data collection.

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 (Figure 1b). Consequently, we begin by creating a 2D histogram of points projected onto the ground plane. Linear structures are then extracted from this histogram using a Hough transform. This idea is simple, straightforward, and works fairly well in practice. The 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. It should be noted that this method is limited to modeling vertical, planar walls. The extension to non-planar and non-vertical walls is a subject of future work.

The contributions of this paper are threefold. 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.

2. Related Work

Floor plan modeling has been well studied in robotics research, since 2D maps of building interiors are often needed for robotic navigation tasks. Often, the maps are generated by robots, equipped with laser scanners, moving through the environment, which adds the additional challenge of addressing robot localization. Generally, the maps produced by prior work do not aim to be highly accurate or complete, at least not to the degree needed for blueprints. Floor plan mapping methods can be divided into two broad categories: 3D methods and 2D methods. The 3D methods first model walls in 3D and then generate a floor plan from a cross section of the model, whereas the 2D methods only operate in a horizontal 2D plane and usually use a laser line scanner.

A wide variety of methods have been proposed for detecting and modeling planar surfaces in 3D laser scan data. Example approaches include bottom-up region growing using surface normals [3], brute-force plane-sweep search [1, 5], hypothesize and test using the random sample consensus (RANSAC) algorithm [8], Hough transforms [10], and probabilistic methods based on the expectation maximization (EM) algorithm [11]. These methods solve a more complex problem than floor plan modeling and tend to be computationally demanding, due to the extremely large number of 3D points involved.

The 2D methods address this complexity by operating only on a horizontal slice of the environment. Typically, a horizontally mounted laser line scanner is used to obtain measurements, and piecewise-linear models are fit to the resulting data. Various modeling methods have been proposed, including RANSAC, iterative end point fitting [2], split and merge [7], and the Hough transform [4]. Nguyen et al. offers a good comparison of these methods [6]. While less computationally demanding than their 3D counterparts, these methods have the disadvantage that they generally use data from laser scanners mounted close to the ground. Wall data at this height is more likely to be obstructed by furniture and other clutter (see Section 5.3).

Our approach takes the benefits of both the 3D and 2D approaches. Since we use the full 3D laser scan data as input, our method is not constrained to use a single 2D cross section, but since the bulk of our modeling is performed in 2D, it is more computationally efficient. Wulf et al. take an approach similar in spirit to ours [12]. Rather than use ground plane projection density to determine likely walls, they assume the farthest point from the sensor in any vertical scanline is a wall point. This is followed by a sliding window line extraction.

Most evaluation of floor plan maps is qualitative, as in [9], and rarely quantifies the correctness of a line segment map. An evaluation metric was proposed in Nguyen et al. for comparing line sets, but it does not take into account the amount each line segment overlaps another, only that a known segment has a detected segment associated with it [6]. It also does not take into account the over segmentation of lines or penalize overlapping line segments. These limitations motivate the need for a more comprehensive evaluation methodology.

3. Floor Plan Modeling

Our floor plan modeling algorithm involves three steps. First, the ceiling and floor are estimated and the corresponding points are removed from consideration. Next, the remaining points are discretized into a voxel space and projected onto the ground (x-y) plane to form a 2D density histogram. Finally, planar segments are extracted from this density histogram to form the floor plan model. Post-processing operations clean up the model and improve the fit to the data. The next several sub-sections describe these steps in detail.

3.1. Floor and Ceiling Removal

Just as the projection onto the x-y plane is well suited for finding vertical surfaces, the projection onto the vertical (z) axis is useful for finding horizontal surfaces. Observing that that most floors and ceilings are primarily horizontal, we create a height histogram by projecting the 3D measurements onto the z axis (Figure 2). Rather than directly projecting the points, though, we first discretize the points into a 3D voxel space with square voxels (10 cm in our experiments). Each occupied voxel contributes the same increment to the histogram regardless of the number of points that fall into the voxel. This voxel-based method ensures that the histogram measures surface area, rather than the raw number of points. Counting the points directly would bias the histogram toward densely sampled regions lying close to the scanner or where measurements from multiple scans overlap. An alternative approach would be to explicitly estimate the data density at each measured point and weight the point contributions by the inverse density, but the method we adopted is simpler and achieves the same effect.

In the height histogram, the bottom-most and top-most local maxima are identified as the floor and ceiling height respectively. The corresponding points are marked accordingly and are not used for wall detection. This step is not strictly necessary, though, since the contribution of the floor and ceiling is distributed evenly across the entire facility. The floor and ceiling heights are also useful for strategically selecting which cross sections to be used in the floor histogram (Section 5.3). Height histograms can be computed either on a per-room basis or for an entire floor of a facility. The per-room approach has the advantage of being able to handle varying ceiling and floor heights that may exist throughout a building, but it requires assigning scans to rooms.

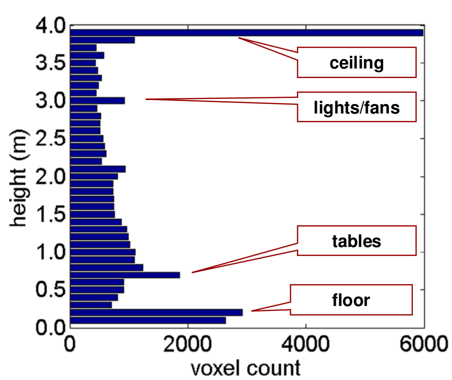


Figure 2. The height histogram is a projection of the 3D data onto the vertical axis. The large maxima at the top and bottom correspond to the ceiling and floor heights. Variations in data density at other heights are indicative of the degree of clutter at each elevation.

3.2. Ground Plane Histogram

The 3D point measurements are projected onto the x-y plane using a similar method to the height histogram generation (Figure 3). The only difference is that the projection has two dimensions instead of one. Just as with the height histogram, the ground plane histogram is generated indirectly by first discretizing the data into a voxel space and then counting the occupied voxels above each 2D ground cell.

3.3. Wall Segment Detection

Walls are detected in the ground plane histogram using a Hough transform approach (Figure 4b) [4]. The Hough transform is a well known algorithm in computer vision that has been shown to be useful for detecting lines and line segments in cluttered data. Briefly, a Hough transform line detector works by detecting peaks in a configuration space, which is two dimensional for lines (orientation and distance from the origin). Each pixel in the input data (here, the ground plane histogram) could be part of an infinite number of lines that pass through that point. This set of lines corresponds to a set of points in configuration space. In this manner, each point in the input data votes for a set of points

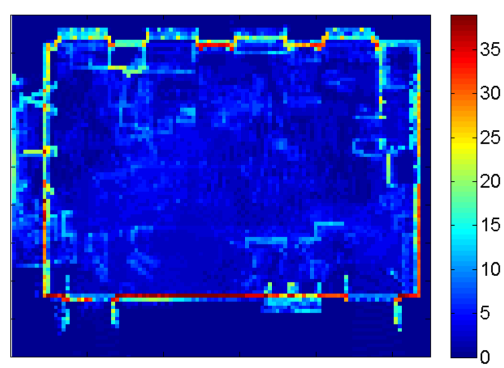
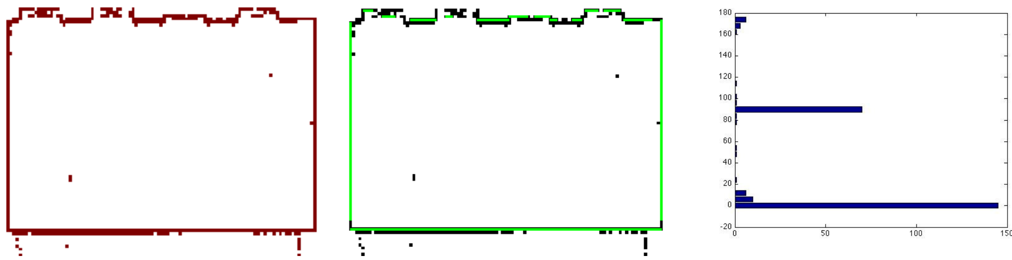


Figure 3. Ground plane histograms are formed by projecting voxelized 3D data onto the x-y plane and accumulating the occupied voxel count into a histogram. The dense regions indicate vertical surfaces, which have a high probability of being wall segments. A histogram for a single room is shown here. Figure 1b shows an entire floor.



(a) (b) (c)

Figure 4. Wall segment detection. (a) The ground plane histogram is first thresholded to remove low density cells. (b) The Hough transform is then used to detect lines within this thresholded histogram (green detected lines overlaid onto thresholded data). (c) The dominant orientations can be determined from peaks in the distribution of all wall orientations relative to the vector [0, 1].

in configuration space. A peak in configuration space corresponds to a line in the input data, since each pixel in that line votes for the same point in configuration space. Analysis of the line's region of support in the data is used to determine the endpoints of the line segment.

There are two common variations of Hough transform line detection. The first approach is to detect all line segments at the same time by detecting the largest peaks in configuration space (parallel approach). The second approach is to incrementally detect lines by first finding the strongest peak, estimating the corresponding line segment, and then removing the underlying data from the ground plane histogram and from the configuration space histogram. The process is then repeated until a stopping criterion, such as the number of lines or a threshold on line strength, is reached. This iterative approach is somewhat slower, but has the potential to perform better because small line segments are sometimes influenced by larger nearby line segments. We investigated these variations, as well as the effects of the various parameters of the Hough transform algorithm, in our experiments (Section 5).

We found, experimentally, that the Hough transform algorithm performs better if the ground plane histogram is filtered to remove cells that are not very dense (Figure 4a). In our implementation, the threshold is set at two standard deviations (σ) above the mean histogram density.

3.4. Dominant Orientation Processing

The walls of many facilities lie along dominant orientations that are perpendicular to one another. These dominant orientations can be automatically detected and then used to improve the floor plan models. The dominant orientations are estimated from the peaks in the distribution of wall orientations found by the wall detection algorithm (Figure 4c). One dominant orientation is located at the largest peak, and the secondary orientation is the largest peak near the perpendicular to the dominant orientation. Detected lines with an orientation within a threshold angle of one of these orientations can be "snapped" to the dominant orientation by rotating them about their centroids.

4. Evaluating Floor Plan Modeling Performance

We developed a method to objectively evaluate floor plan models with respect to ground truth data. We would like a performance measure that correlates well with how a human would subjectively judge performance. Specifically, a high performing algorithm should detect walls where walls truly exist and not detect walls elsewhere. In the places where walls are detected, the segments should conform to the linear segments of the true walls – i.e., they should have the same position, orientation, and endpoints. The detected walls should not overlap one another unnecessarily, and they should not be split into smaller segments than necessary.

Our evaluation measures, which were developed in accordance with these goals, consist 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.

Object detection in computer vision is often quantified using precision-recall (P-R) curves, in which the precision ($P = tp / (tp + fp)$) is plotted versus the recall ($R = tp / (tp + fn)$) as a function of some tunable parameter of the algorithm, where tp, fp, and fn are the number of true positives, false positives, and false negatives respectively.

Given a ground truth floor plan of a facility, we must first determine the data association between wall segments detected by our algorithm and the ground truth wall segments. This step is more difficult than in many other object detection problems, like face detection, because it is not always obvious which ground truth segment a given detected segment is intending to model. We accomplish the data association by matching each detected segment with the closest ground truth segment whose orientation does not differ by more than a given amount. This orientation threshold prevents detected segments from being associated with almost perpendicular ground truth segments. The detected segment is then projected onto the ground truth segment along the ground truth segment's normal direction.

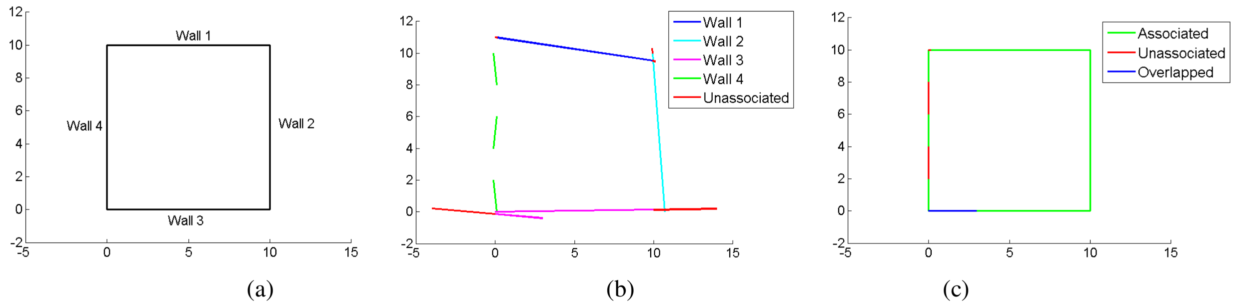


Figure 5. Illustration of the evaluation metric on a synthetic example of a simple room. (a) Ground truth walls. (b) Assignment of detected walls to ground truth walls. (c) Classification of detection accuracy of ground truth walls.

The overlapping regions of the two segments are associated with one another, while any overhanging sections are kept as unassociated, since they may be associated with other segments of the ground truth model. This process continues until all detected segments are associated or are found to have no association. A simple example of this data association is shown in Figure 5.

Once the data association is complete, the number of true positives, false positives, and false negatives can be computed. Rather than counting each segment equally, we weight the contribution by the length of the segment. Otherwise, the evaluation would more heavily penalize short segments. Detected segments that correctly associate with a ground truth wall segment count as true positives, whereas detected segments that have no association are labeled false positives. Segments in the ground truth model with no associated detected segment are false negatives. Using this method, the true positive count is the total length of correctly modeled walls, the false positive count is the total length of hallucinated wall segments, and the false negative count is the total length of undetected wall segments. On the precision-recall curves, precision describes the fraction of the detected wall length that is actually a wall, while the recall describes the fraction of the ground truth wall length that was correctly modeled.

The precision and recall only provide one aspect of the performance of a floor plan modeling algorithm. An algorithm can have high precision and recall but still produce a relatively poor quality model of the floor plan. The second part of the evaluation criteria addresses this issue by measuring the accuracy and conciseness of the correctly modeled segments. The accuracy of the modeled segments is computed in terms of the distance and orientation error, while the conciseness is summarized by measures of over-segmentation error and overlapping segment error. The distance error ($E_d(s_i)$) for a correctly detected segment (s_i) is the average distance between the detected segment and its associated ground truth segment. The overall distance error (E_D) is computed as

$$E_D = \sum_{i=1}^{N}\left[\text{length}(s_i)E_d(s_i)\right] / \sum_{i=1}^{N}\text{length}(s_i), \quad (1)$$

where N is the number of correctly detected segments. Weighting by length prevents smaller lines from having a larger effect on the error than an equivalent length of longer lines. The orientation error ($E_\theta(s_i)$) for a single segment is the angle between the detected and ground truth segments. The overall orientation error (E_Θ) is computed as

$$E_\Theta = \sum_{i=1}^{N}\left[\text{length}(s_i)E_\theta(s_i)\right] / \sum_{i=1}^{N}\text{length}(s_i). \quad (2)$$

All other things being equal, a model should contain the minimum number of wall segments with minimum redundancy. These criteria are captured by the over-segmentation error and overlap error measures. Over-segmentation error (E_{OS}) is computed by counting the number of additional segments the detected model contains beyond the minimum needed. It is computed as the

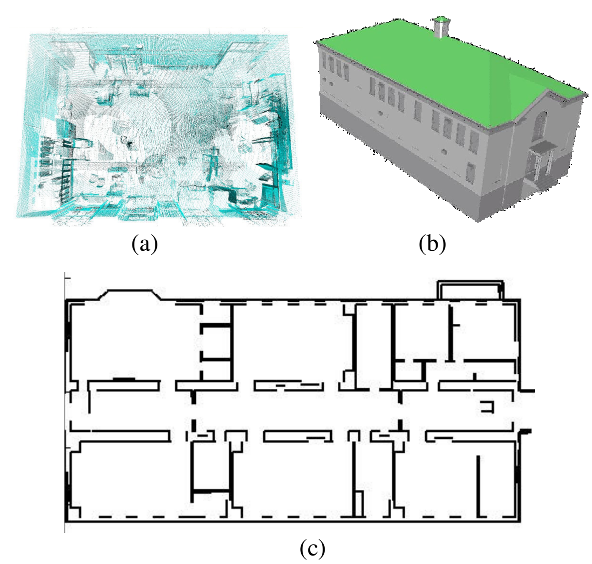


Figure 6. Experimental data. (a) Example data for a single room. (b) The manually created 3D model of the facility. (c) The ground truth floor plan for the first floor.

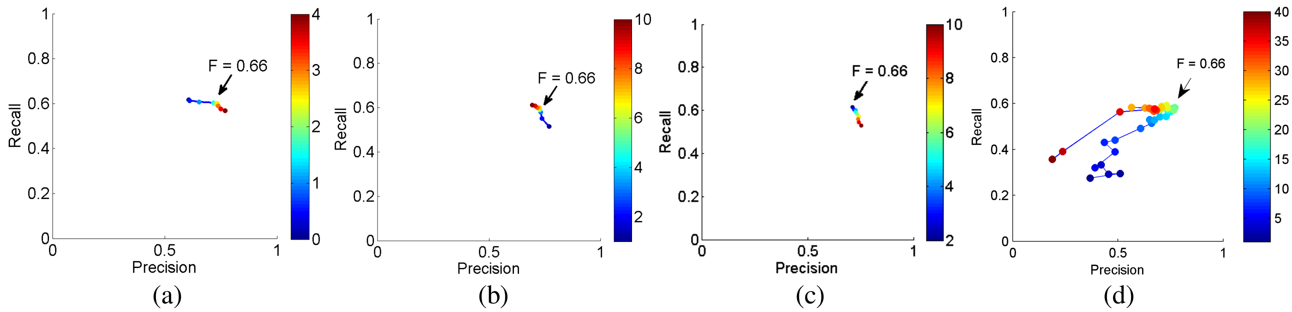


Figure 7. Precision-recall curves for parameter selection experiments: (a) ground plane histogram threshold; (b) Hough transform fill gap; (c) minimum line length; (d) slice number.

ratio of the number of detected segments to the number of modeled segments. The overlap error (E_{OL}) is computed as the total length of ground truth wall surface that is associated with more than one detected wall segment normalized by the total length of detected walls.

5. Experiments

Our experiments use data from a 3D model of a building that was manually modeled by a professional laser scanning service provider. The facility is a two story school house with a basement (Figure 6). It contains a large amount of clutter in the form of desks, tables, chairs, bookshelves, and wall-mounted objects, making this an unusually challenging data set. The building contains 40 rooms – 4 in the basement, 23 on the first floor, and 13 on the second.

The facility was scanned using a laser scanner developed by the service provider, which has state-of-the-art performance characteristics. Scans from 225 locations within the building were obtained, each containing approximately 14 million points. Each room contains between 1 and 11 scanning locations (average of 5.6). The data was registered using surveyed fiducial markers placed in the environment, which is the standard method in the industry. For our experiments, in order to handle the large data sets, each scan was sub-sampled by a factor of 13, giving approximately 83,000 points per scan.

The service provider manually created a 3D model of the facility using the methods described in Section 1. Floor plans derived from the manually created model serve as ground truth for evaluation of our algorithm (Figure 6c).

We conducted three types of experiments to evaluate our floor plan modeling algorithm. The first set of experiments (Section 5.1) focused on understanding the effects that the various parameters of the algorithm have on overall performance. The second set of experiments (Section 5.2) compared the batch and incremental variations of the Hough transform. The final set of experiments (Section 5.3) investigated different strategies for selectively choosing which data to use in the ground plane histogram.

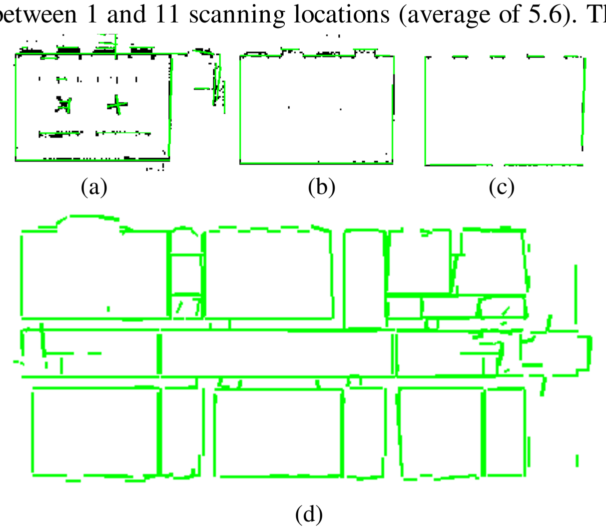


Figure 8. Parameter selection. Variations of the ground plane histogram threshold for a single room – low, with excess detections (a), optimal (b), and high, with missing detections (c). (d) Results for the entire 1st floor at optimal settings ($F = 0.6642$).

5.1. Parameter Selection

Our algorithm has several parameters that can be tuned. In addition to the previously mentioned threshold on the density histogram filter, the Hough transform has a few adjustable parameters: the fill gap, which is the minimum distance between points that can be included in the same detected line, and the minimum line length, which limits short line segment detection. We conducted a set of experiments to determine the effect of each parameter on

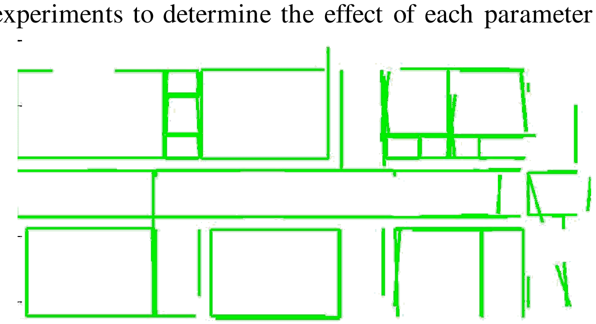


Figure 9. The batch Hough transform method is both visually and statistically worse than the incremental results in Figure 8 ($F = 0.6263$).

(a) (b) (c)

Figure 10. Selective choice of cross-section heights. (a) Precision, recall, and voxel concentration as a function of cross-section height (0.1 m slices) shows a relationship between voxel concentration and performance. (b) Using data from cross-sections at a 30 cm height results in poor floor plans. (c) Choosing heights with minimal concentrations (green circles) improves performance.

algorithm performance. Figure 7a, b, and c highlight the precision-recall curves for these experiments. The best parameter value is computed using the F-measure ($F = 2PR/(P+R)$). Based on these experiments, we set the parameters as follows: ground plane histogram threshold = 2.5 σ, fill gap = 0.7 m, and minimum line length = 0.2 m (Figure 8). Walls within 5° of the dominant orientation were snapped to the dominant orientation.

5.2. Batch versus Incremental Hough Transform

Using the methodology described in Section 3, we determined the best parameters for the batch Hough transform method as well. Figure 9 shows the comparison of the batch and incremental methods for one of the parameters. The incremental method outperforms the batch method, and this trend was found to be true broadly across all of our experiments.

5.3. Selective Ground Plane Histograms

The algorithm described in Section 3 computes the ground plane histograms using data from every height, barring the floor and ceiling data. However, Figure 2 shows that the number of points varies significantly as a function of height. Since the number of wall points should be fairly constant throughout the range, this variation indicates that the amount of clutter changes with height. This conclusion is intuitive, since some objects, such as tables, occur primarily at lower heights in a room, while other objects, such as ceiling fans, occur primarily at higher heights. Indeed, Figure 10a shows an inverse correlation between the number of occupied voxels in a cross-section and the precision and recall of that cross-section.

Based on these observations, we conducted experiments to determine whether performance can be improved by selectively choosing the height ranges used in estimating the ground plane histograms. The first experiment analyzes performance as a function of height, while the second experiment considers an alternate strategy of selecting heights with minimal density in the height histogram.

For our height experiment, we created ground plane histograms using data from a small height interval (0.1 m cross-sections) and varied the slice height in 0.1 m increments. The results (Figure 10a) indicate that slices between 1 and 2.5 meters above the floor perform significantly better than those nearer the floor or ceiling. At the height of a typical mobile robot sensor, the clutter is significant and results in poor floor plans (Figure 10b).

While some heights are better for modeling than others, it is also possible that combinations of heights may provide even more benefit because regions where clutter occurs at one height may be uncluttered at a different height. Our second experiment investigates this possibility by creating ground plane histograms from multiple – not necessarily adjacent – cross-sections. In the second experiment, we ranked cross-section heights in increasing order based on the height histogram magnitudes (Figure 10c). We then computed the algorithm's performance using the first N cross-sections from this list and then varied the value of N. The best results were achieved with $N = 9$.

5.4. Discussion

Figure 11 shows the labeling – analogous to Figure 5c –

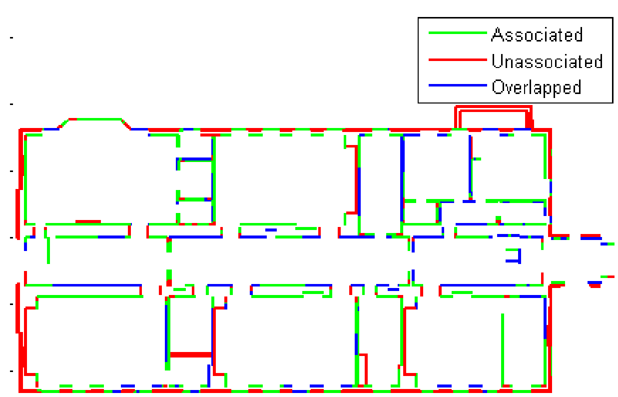


Figure 11. Labeling of ground truth wall segments according to detected wall segment positions. Green and blue segments were modeled by the algorithm, while red segments were missed.

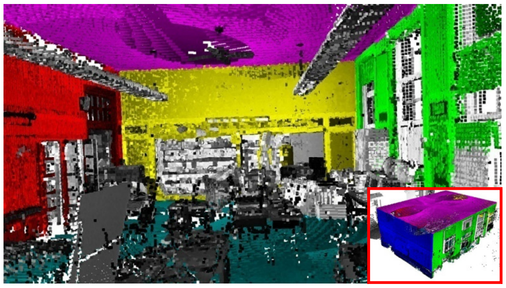


Figure 12. Reprojection of detected walls, floor, and ceiling onto the original 3D data, each shown in a different color. Clutter objects (tables, bookshelves, etc.) are shown in grey.

of the ground truth wall segments according to the detected wall segment positions for the optimal parameter settings. Computing our evaluation measure (using a 36° orientation threshold) resulted in $E_D = 0.049$ m, $E_\Theta = 0.705°$, $E_{OS} = 0.59$, and $E_{OL} = 0.265$. The results indicate that the majority of the walls in large rooms were correctly modeled by the algorithm. The walls that were not detected fall into three categories: 1) external walls, which were not observed by the sensor and are therefore not modeled; 2) small storage rooms, which were not scanned entirely and are also significantly more cluttered than the large rooms; and 3) built in closets (large shallow intrusions on the left sides of the three lower rooms). This third category is difficult to model, since the doors of the back walls of the closets were mostly occluded by objects stored in the closets. The stairwells also were not modeled as accurately, mainly due to the confusion between railings and walls.

6. Summary and Future Work

We have demonstrated an algorithm for estimating the floor plan of a facility in terms of the locations of its walls, even in the face of substantial clutter. Strategic selection of heights for use in the ground plane histogram contributed substantially to the improvement in the model quality. We also proposed a set of quantitative measures for evaluating the quality of a floor plan model with respect to ground truth. This quantitative approach could be used by other floor plan modeling algorithms to gauge their performance improvements and to gauge the difficulty of other data sets.

In the future, we intend to extend this algorithm to explicitly find and model windows and doors. Also there are several improvements that can be made to the wall finding algorithm. For example, the threshold filter can be replaced with a modified Hough transform to take into account uncertainty. Also a final pass filter could be used to improve the quality of the line set by connecting close lines and normalizing angles.

In the longer term, we are investigating alternative methods for floor plan modeling that operate on the data in 3D, rather than in the 2D projection spaces used in this paper. The algorithms in this paper can provide an initial hypothesis for such 3D modeling methods. For example, the detected walls, floors, and ceilings can be used to give an initial label to each 3D point, as shown in Figure 12.

Acknowledgements

This research was funded, in part, by the National Science Foundation (CMMI-0856558) and by the Pennsylvania Infrastructure Technology Alliance. We thank Quantapoint, Inc., for providing experimental data.

References

[1] A. Budroni and J. Böhm, "Toward automatic reconstruction of interiors from laser data," in *Proceedings of 3D-ARCH*, Trento, Italy, February 2009.

[2] R. Duda, P. Hart, and D. Stork, *Pattern Classification (Second Edition)*: John Wiley & Sons, 2001.

[3] D. Hähnel, W. Burgard, and S. Thrun, "Learning compact 3D models of indoor and outdoor environments with a mobile robot," *Robotics and Autonomous Systems,* vol. 44, no. 1, pp. 15–27, July 2003.

[4] P. V. C. Hough, "Machine Analysis of Bubble Chamber Pictures," in *Proceedings of High Energy Accelerators and Instrumentation*, 1959.

[5] M. Johnston and A. Zakhor, "Estimating building floor-plans from exterior using laser scanners," in *Proceedings of Three-Dimensional Image Capture and Applications*, San Jose, CA, January, 2008.

[6] V. Nguyen, S. Gächter, A. Martinelli, et al., "A comparison of line extraction algorithms using 2D range data for indoor mobile robotics," *Autonomous Robots,* vol. 23, no. 2, pp. 97–111, August 2007.

[7] T. Pavlidis and S. Horowitz, "Segmentation of plane curves," *IEEE Trans. on Computers,* vol. 23, no. 8, pp. 860-870, August 1974.

[8] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for Point-Cloud Shape Detection," *Computer Graphics Forum,* vol. 26, no. 2, pp. 214-226, June 2007.

[9] A. Scott, L. Parker, and C. Touzet, "Quantitative and qualitative comparison of three laser-range mapping algorithms using two types of laser scanner data," in *Proceedings of Systems, Man, and Cybernetics*, pp. 1422–27.

[10] F. Tarsha-Kurdi, T. Landes, and P. Grussenmeyer, "Hough-transform and extended RANSAC algorithms for automatic detection of 3d building roof planes from lidar data," in *Proceedings of the ISPRS Workshop on Laser Scanning*, 2007, pp. 407-412.

[11] S. Thrun, C. Martin, Y. Liu, et al., "A real-time expectation-maximization algorithm for acquiring multi-planar maps of indoor environments with mobile robots," *IEEE Transactions on Robotics,* vol. 20, no. 3, pp. 433-443, 2004.

[12] O. Wulf, K. O. Arras, H. I. Christensen, et al., "2D Mapping of Cluttered Indoor Environments by Means of 3D Perception," *International Conference on Robotics & Automation*, New Orleans, Louisiana, April, 2004.