Estimating building floor-plans from exterior using laser scanners

Matthew Johnston and Avideh Zakhor *

Berkeley, CA

ABSTRACT

The knowledge of the interior structure of buildings is important in many civilian and military applications. While it is possible to obtain interior maps by exhaustively inspecting every room in a building, in many situations this is infeasible due to security or safety reasons. In this paper, we develop a method to generate interior building floor plans from the exterior of the structure. We use a laser scanner to measure the range of hundreds of thousands of points on interior walls of the building, exploiting the fact that the laser can go through unobstructed windows. We develop an algorithm to fit planes to the point cloud resulting from the laser data. To accomplish this, the optimal locations of horizontal planes are found such that they model the ceiling and the floors; subsequently, vertical planes are placed perpendicular to those and aligned to fit the data. Once these planes are found and localized, floor plans for each floor are extracted. We show that our proposed method is effective in recovering partial floor plans for three separate building examples.

Keywords: Exterior Modeling, Plane Fitting, Laser Scanners, Floor Plan Estimation, Wall Detection

1. INTRODUCTION

The knowledge of the interior structure of buildings is important in many civilian and military applications. While the floor plans of buildings are readily available from architectural drawings, in many situations they need to be estimated. One way to arrive at a reasonable interior map is to examine every floor, and measure every room. This can be prohibitively time consuming; furthermore, in many scenarios it is infeasible to enter the building and conduct such an extensive analysis of the interior. This paper presents a solution to this problem by developing a method to obtain a floor plan from the exterior of the building under study.

There has been a great deal of research on interior mapping of buildings with laser scanners. However, all the work in this area is focused on sanners that are situated inside the buildings. Morrison et al. place a laser range scanner on a tripod inside the structure, and conduct a scan to build a 3D model.[1] While these scans are generally of high quality and yield accurate models, the scanner must be manually moved to each room, and scans are stiched together manually via a user interface. Thrun et al. use a mobile robot to scan the interior of buildings in order to create a 3D model.[2] They make a further assumption that the environment being scanned is smooth, since each location is only scanned once. Jensen et al. use a mobile robot equipped with a rotating laser scanner to construct 3D map of indoor environments.[3] Jo et al. also use a mobile robot, but here the robot data is used to construct a 2D map from which a 3D map is created.[4] While all of these approaches yield accurate models of the buildings interior, they do so from inside the building. As such, in this paper, we focus on the problem of floor plan recovery from the exterior of the structure.

Our approach can be summarized as follows. We use a range laser measurement tool to obtain a point cloud from the exterior of the building representing data on the walls and ceilings. This is contingent upon the existence of unobstructed windows made of glass for which the laser light passes through, and is bounced off the back and side walls of the rooms. After acquiring this data, the problem reduces to one of plane fitting. We have developed a plane fitting algorithm for the point cloud to approximate the walls and floors of the building. Once all or most of the data in the point cloud are used to construct walls, a floor plan is extracted by computing the intersections between the floor planes and the wall planes.

The outline of this paper is as follows: Section 2 provides an overview of the data collection process. Section 3 describes the plane fitting algorithm for the point cloud. Section 4 describes the process of extracting floor plans from the results of the plane fitting. Finally, Section 5 presents our experimental results and Section 6 discusses these results.

*This work was sponsored by DARPA Visi-Building project.

2. DATA COLLECTION PROCEDURE

We use a range laser measurement system situated outside the building to collect a point cloud for the interior of the building. This is contingent upon having unobstructed windows for which the laser light passes through to be bounced off the interior side walls and back walls and the ceiling. An example of such a point cloud for one room in Berkeley Civic Center with two unobstructed windows is shown in Figure 1. The side wall, ceiling, and the back walls are all discernible in Figure 1.

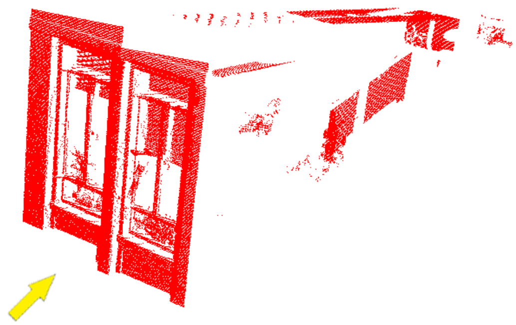


Figure 1. View from the front of the Berkeley Civic Center point cloud obtained from the laser scanning process. The yellow arrow represents the location of the laser scanner. Notice the point clusters to the right of the two windows are measured from the laser going through the open windows.

The particular laser scanner used here is the Leica ScanStation. This unit is capable of scanning up to four thousands data points a second at a minimum of 1.2 mm minimum spacing, and a range of 300 meters. The laser used by this system is green, which is approximately 530 nm.

From the above example, it is clear that if too many windows in a given building are obstructed, fewer internal data points are obtained, and consequently a less complete floor plan is likely to be constructed. To ensure completeness, it is desirable to collect data from all sides of the building, rather than one. The Leica ScanStation allows for registration of data obtained from multiple sites. Figure 2 shows an example of the 3D point cloud resulting from the above scanning process for one side of a large three story building. The laser scan made 630,550 measurements for this structure.

3. DETECTING WALLS IN A POINT CLOUD

Having obtained a three-dimensional point cloud representing points lying on the walls and ceilings of the building, planes need to be found that best characterize those surfaces. To facilitate this process, several assumptions are made about their characteristics. The first assumption is that the ceiling and floors are parallel to the ground, despite slanted ceilings being occasionally found on buildings. The second assumption is that the walls are orthogonal to the ceiling; this is generally valid given that the previous assumption is held. Throughout this paper, planes are represented in three dimensional space by:

$$ax + by + cz + d = 0 \tag{1}$$

where a, b, c, and d are plane-specific values. However, this equation can be simplified in accordance with the assumptions we have made for different types of planes. For the horizontal planes forming the roof and the floors, $a = b = 0$, and for the vertical planes forming the walls, $c = 0$.

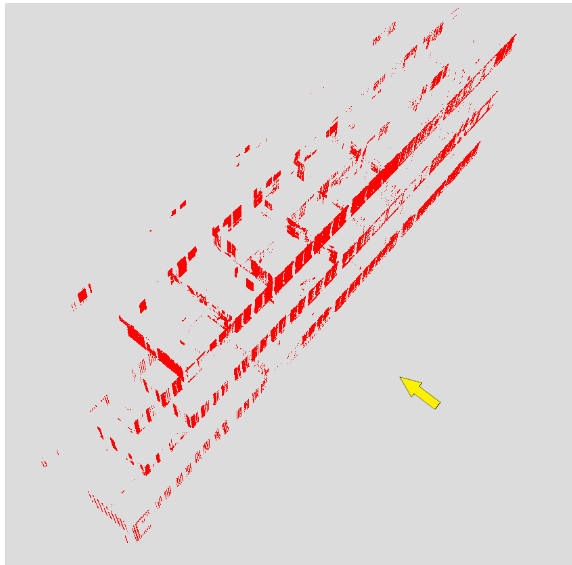


Figure 2. View from above of Building A point cloud obtained from the laser scanner. The yellow arrow represents the location of the laser scanner.

3.1 Finding the ceiling

The first step in the plane fitting process is to localize the ceiling. To do so, a plane is created to fit to the top portion of the point cloud. As described above, this plane is assumed to be parallel to the ground, and is referred to as the cross-section plane. Changing d in Equation 1 moves the plane up and down along the z-axis. Therefore, to determine the location of the ceiling, the cross-section plane starts above the highest point in the point cloud. Then the value of d is decremented by a small amount causing the plane to move downwards, and with each step the algorithm checks the number of points the plane matches. Once a certain percentage of points are matched, the location of the cross-section plane corresponds to the ceiling. The percentage threshold check is in place because the laser may have picked up points on an antenna or other objects on the ceiling; since these do not correspond to the ceiling, it is desirable to pass those points and wait until a significant number of points are matched before the location of the ceiling is specified.

To determine whether a point matches the plane or not, the shortest distance between the point and the plane is computed. If that distance is small, than the point is declared to be included in the plane. This approach of matching points within a small error range is taken in finding all of the planes in our algorithm.

3.2 Finding the boundaries between the stories

In order to deal with buildings with multiple stories, planes are needed to separate different floors from each other. All of the points in the interior of the building are recorded by lasers going through the glass windows. Since the windows are above the laser scanner, which rests on the ground, the points on the interior side or back walls are either at the window level or above it. For each floor, we observe a dense point cloud for the points that are above the windows, followed by a region void of points representing the area below the windows on the floor. Thus, if the point cloud consists of two point clusters vertically separated by a gap of no points, the lower cluster must correspond to walls on a lower floor, and so the lower boundary of that story is placed at the top of the lower point cluster.

To find the boundary of the story below the current one, the cross-section plane is moved downwards from the current plane. For each step, the number of points matching the cross-section plane is recorded. When the number of matched points drops to zero and then increases by a relative threshold, the location of the cross section plane corresponds to the ceiling of a new story. The relative threshold here refers to a fraction of the total number of points. This repeats until all the story boundaries have been found. Figure 3 shows an example of the results of locating the boundaries of the three stories in a building. Notice that each boundary lies where a vertical segment with little to no points meets a vertical segment with many points.

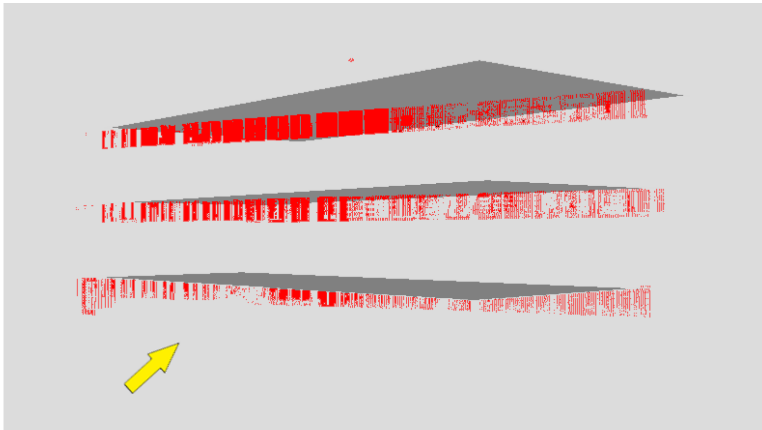


Figure 3. View from side of Building A point cloud after the ceilings of all the stories have been fitted with planes. The yellow arrow represents the location of the laser scanner.

3.3 Finding the interior walls

One way to define a plane is to find a point and a normal vector to the plane. To find the point for an interior wall plane, the cross-section plane introduced in Section 3.1 is used again. That plane starts at the ceiling's location found in Section 3.1, and continually moves downwards beneath it. At each step, the number of points on the cross-section plane that has not previously been matched to a wall is computed. If that number is greater than a relative threshold based on the amount of unmatched points left, then one of those unmatched points is selected at random to be part of the desired wall plane. We define the coordinates of this point as $(x\prime, y\prime, z\prime)$. This is illustrated by the red point in Figure 4(a).

The other component needed to construct an interior wall plane is the direction of the normal vector to the plane. Since all of the planes used as walls are vertical by the earlier assumptions, their normal vectors must be parallel to the ground. This leaves only two degrees of freedom to be determined for the direction of the normal vector. A candidate normal vector is shown in blue in Figure 4(b). To determine the remaining degrees of freedom for the normal vector direction resulting in optimal location of the vertical walls, we rotate the blue normal vector in Figure 4(b) incrementally, and at each angle we create a potential wall plane with that normal vector, as shown in Figure 4(c). We compute the number of points matching each plane corresponding to each angle and choose the angle resulting in a plane with the most matching points as the direction of the normal vector of the wall plane.

Mathematically, the normal vector is parallel to the ground, implying that the vector has no z component. This vector, which we will refer to as $\vec{v}$, is defined to be

$$\vec{v} = [cos(\theta), sin(\theta), 0] \tag{2}$$

where θ is the angle $\vec{v}$ makes with the x-axis. The process of rotating the normal vector to find its optimal direction corresponds to finding the optimal θ. For each normal vector direction in Equation (2) and the point located at $(x\prime, y\prime, z\prime)$ found earlier, a plane P can be described by the following equation:

$$x * cos(\theta) + y * sin(\theta) = x\prime * cos(\theta) + y\prime * sin(\theta) \tag{3}$$

The algorithm iterates through integer values of θ from 0°to 180°in order to find all possible wall planes for the randomly chosen point at $(x\prime, y\prime, z\prime)$. For each angle, the number of points matched by a plane with a normal

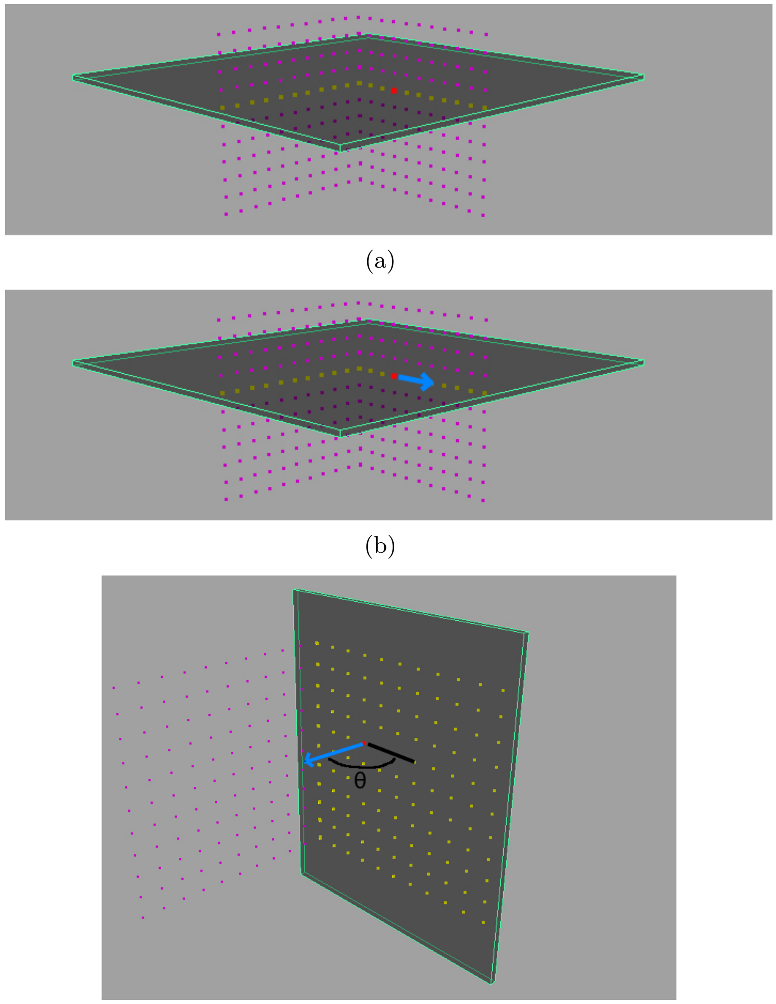
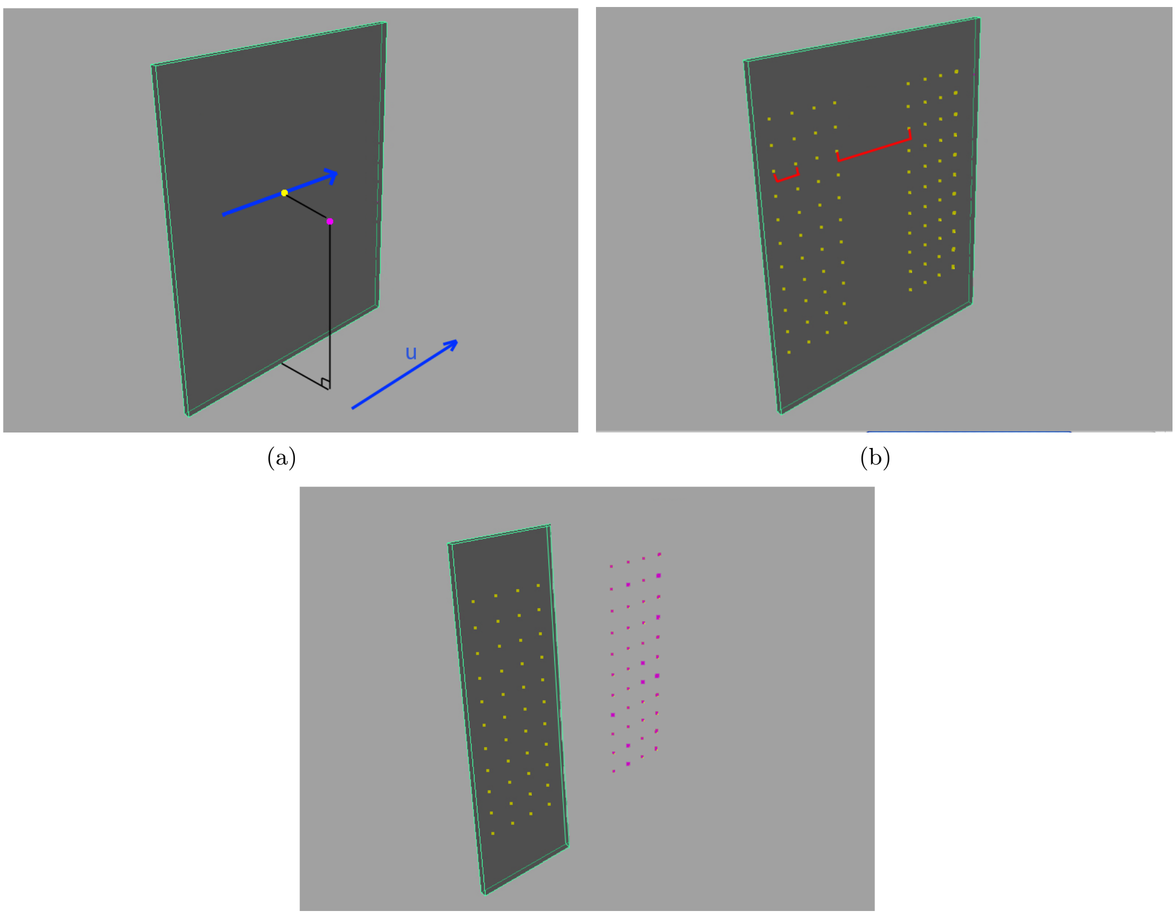


Figure 4. Illustration of plane fitting process; (a) the yellow points represent the points matched by the gray cross-section plane, the red point is the randomly chosen point ($x\prime$,$y\prime$,$z\prime$); (b) to compute the blue normal vector, it is rotated around the red point; (c) after all the angles have been visited, the angle yielding the wall with the most matches is chosen.

vector at that angle is calculated; once all angles have been traversed, the optimal angle θ that results in the maximum number of points being matched is chosen. When counting the number of points that match each wall, a point is only counted if it is within a certain distance of the point at $(x\prime, y\prime, z\prime)$ used to construct the plane. By only matching local points, the chance of counting points that belong to other walls is severely reduced.

Due to the fact that a plane has an infinite surface area and a wall does not, planes must be sized so that they only cover the area that the actual wall covers. The method for sizing the wall is to go through each of the points that are included in the wall and record their coordinates, noting the maximum and minimum values for each of the three coordinates of the point. The wall is defined as being the intersection of the plane and the range defined by the maximum and minimum coordinates. However, we should only declare walls in places where there is evidence to support placing one there. For example, if we have two point clouds separated by a large area of empty space, a wall should not pass through the empty space connecting the two clusters as this may represent a gap between two walls. For the z coordinate, this is not an issue and the maximum and minimum values of that coordinate can be recorded immediately. This is because in actual buildings, walls are typically attached to a ceiling and a floor. However, an extra calculation is required to delimit the extent of the walls in the x and y dimensions.

To deal with this, the points matching the candidate wall are projected onto the plane that defines that wall as shown in Figure 5(a). The projection is carried out by taking the dot product of each point with a vector lying in the wall plane, parallel to the ground, and with the same z-value as the point. In Figure 5(a), the point is shown in purple, and its projection onto the blue vector is shown in yellow. The direction of all such projection vectors in this plane is referred to as u, as shown in Figure 5(a). The projected points on the plane are sorted along the u direction on the wall plane. Once this is done, the distance between every two neighboring points with respect to u is calculated. These intervals are measured by a one dimensional distance, neglecting the change in z, since in our application, walls often contain horizontal gaps but rarely contain vertical ones. As shown in Figure 5(b), the space between every two consecutive points is measured, and the wall region is grown by including each point until we find two points that are separated by more than 30% of the distance between the first and last point as sorted on the vector. This prevents the algorithm from including points that are not intended to be part of the wall, and ensures a somewhat constant point density. This is shown in Figure 5(c). All the points that are considered to be part of the wall after this step are then removed from the point cloud.



(a) (b)

(c)

Figure 5. Illustration of wall sizing process; (a) the 3D purple point is projected onto the blue vector lying in the plane of the wall to result in the yellow point; (b) the distance between every two consecutive points along the u direction is measured; (c) the wall includes only those points that are close together, while the other cluster of points will eventually be classified as a wall through another iteration of the wall sizing process.

After finding a plane to represent a newly found vertical wall, this process is repeated so as to find a new vertical wall. Namely, another point on the horizontal cross-section plane is chosen randomly, and the algorithm tries to find a new wall containing that point. Whenever the randomly selected point does not yield a wall that matches enough points, or the number of unmatched points on the cross section plane falls below a certain threshold, the plane is stepped downwards and the process is repeated. This continues until the z-value of the cross-section plane has moved past the point in the data set with the lowest z-value. At this point, all the walls are represented by planes, providing a basic knowledge of the structure of the building.

Figure 6 shows the result of applying the plane fitting algorithm to the point cloud of the three story building shown in Figures 2 and 3. Thirteen planes have been fitted to the point cloud covering 95.3% of the points. The ceiling planes are removed for ease of viewing. The remaining points lie in between the walls, in clusters that are not dense enough to be chosen by our algorithm. An example of these points can be seen at the top of Figure 6 between the large purple plane on the right of the figure, and the green plane in the center.

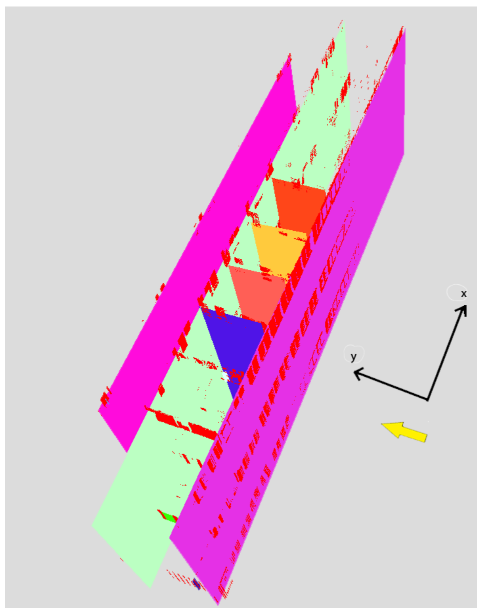


Figure 6. View from above of Building A after plane fitting. The yellow arrow represents the location of the laser scanner.

4. GENERATING FLOOR PLANS

Once the locations of all the walls are determined in the plane fitting process, floor plans can be created for the different stories of the building. Since the location of the boundaries between different stories have already been determined, the next step is to find where each of the walls intersect them. The first task is to determine the range of the x and y coordinates for the floor plan, i.e. the coordinates parallel to the ground, from the entire point cloud. That range is then converted into a range of pixels that can be represented in a bitmap for ease of display. We use a predefined ratio γ to determine how many pixels should lie between two integer values of x and y. The ratio is chosen so that the resulting bitmap has a reasonable size and is visually pleasing. For point clouds with a much smaller x and y range, γ can be increased.

Our algorithm creates as many floor plans as the number of boundaries that have been computed in Section 3.2. For each pixel in the floor plan, the corresponding x and y coordinates (x_0,y_0) are calculated by dividing the pixel values by the chosen ratio γ. These coordinates are part of a 3D point (x_0,y_0,z_0), where z_0 is the z value of the story ceiling boundary for which the floor plan is being computed, i.e. the height of one of the horizontal boundary planes in Figure 3. Once these coordinates are computed, all the walls found in Section 3.3 are searched to determine whether any of them contain (x_0,y_0,z_0). To be included in a wall, a point (x_0,y_0,z_0) needs to (a) be a very small distance away from that wall, (b) be within the x and y boundaries of that wall as computed in Section 3.3 and illustrated in Figure 5, and (c) have z_0 in between the maximum or minimum z for the wall as computed in Section 3.3. Even though in actual buildings, vertical walls are typically attached to a ceiling and a floor, by choosing z_0 to be the height of the ceiling boundary planes, our algorithm only checks for an intersection with the ceiling; this is because the floors of each story cannot be detected due to the geometry of the laser scanning process. In general, more error in the z dimension is allowed than the x and y directions in order to ensure walls pass through the ceiling of the current story so a floor plan can be accurately created. In essence, condition (c) ensures that only walls that are tall enough to be close to story boundaries are considered in the floor plan generation process. If (x_0,y_0,z_0) is found to be matched by at least one vertical wall, the corresponding pixel is turned on, i.e. it is colored black, otherwise it is turned off, i.e. it remains white. After all the pixels have been traversed, the floor plan accurately conveys all the information from the wall detection algorithm.

Figure 7 shows the result of applying this method of extracting floor plans to the three story building shown in Figures 2, 3, and 6. Since the building consists of three similar floors, we have only shown the results for floor 2 here.

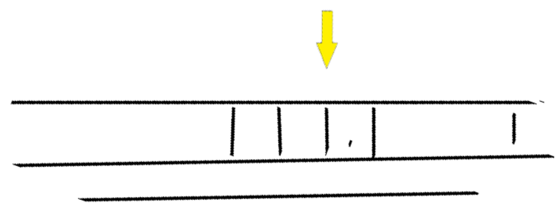


Figure 7. Resulting Floor Plan for the second floor of Building A. The yellow arrow corresponds to the location of the laser scanner.

5. RESULTS

We have run this algorithm on several different buildings, with different degrees of complexity. The first one, Building A, is a three story office building. The results from this building have been shown in Figures 2, 3, 6, and 7. In order to gauge the accuracy of this floor plan, it is superimposed upon the part of the actual floor plan corresponding to the area scanned on the second floor, as shown in Figure 8. As seen, the floor plan generated by our algorithm provides a reasonable estimation of some of the walls in the actual building. However, due to obstructed windows, not all off the walls could be estimated.

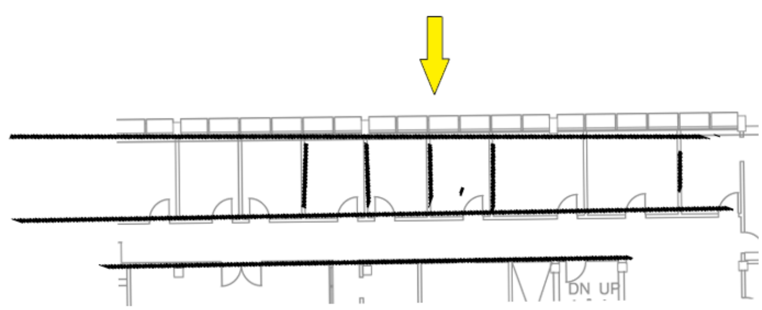


Figure 8. Estimated Building A floor plan for the second floor superimposed upon the actual floor plan. The yellow arrow represents the location of the laser scanner for this data set.

The second building, which we refer to as Building B, is a one story structure comprised of a hallway running down the center. The laser scanner is positioned at one end of the hallway. Figure 9 shows the point cloud resulting from the laser scans, and Figure 10 shows the fitted planes on that point cloud. The point cloud resulting from the laser scanner consists of 77,333 points, and 77,331 of them are matched with two walls and one ceiling. The floor plan derived from these walls is shown in Figure 11.

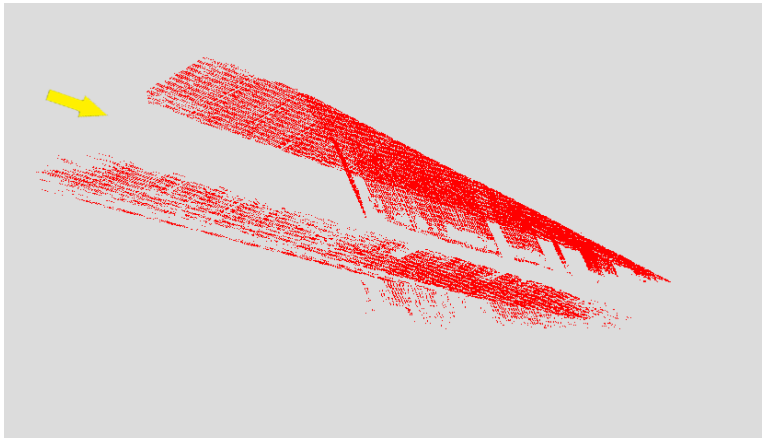


Figure 9. View from below of Building B point cloud obtained from the laser scanner.

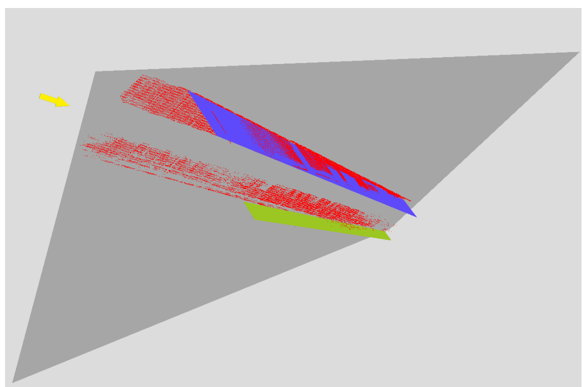


Figure 10. View from below of the Building B planes fit to the point cloud.

Figure 11. Estimated Building B floor plan. The yellow area represents the location of the scanner.

The last building that our algorithm has been applied to is Building C. On this building, the laser scanner is positioned at several different places around the building, leading to a more complete scan of the interior surfaces. The Leica ScanStation allows the point clouds from different viewpoints to be registered with respect to each other. The scanner makes 2,475,973 measurements and the plane fitting algorithm fits 2,326,245, i.e. 94% of the points with walls and ceilings. The raw points from this laser scan are shown in Figure 12. A total of 16 walls have been found in the plane fitting process as shown in Figure 13(a), in addition to the ceiling and the extra floor plane on top of the first floor shown in Figure 13.

Lastly, the results of the plane fitting are used to infer a floor plan. In Figure 14 the results from the floor

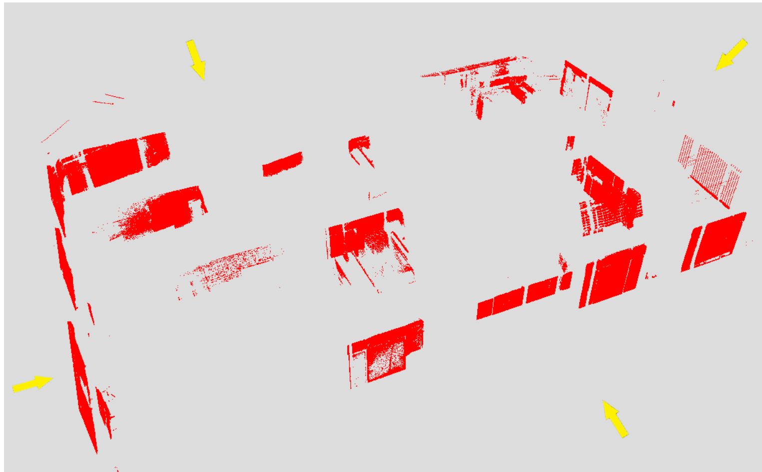


Figure 12. Building C point cloud resulting from the laser scanner, shown from above. The scanner was positioned at each side of the building, and the scans were registered and fused.

plan generation algorithm for the second floor of building C are superimposed on top of the original floor plan to illustrate the accuracy of the floor plan generation method. As seen here, the floor plan again provides a reasonable estimate for some of the walls in the building; however, the interior walls that have not been exposed to the outside in some way are not detected by our approach.

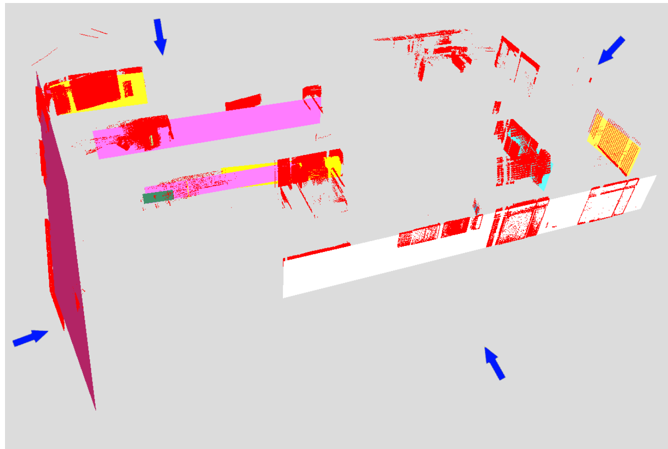
6. DISCUSSION

The quality of the floor plans generated by our proposed method is dependent on the quality of the laser scan process. Factors that affect the laser scan adversely, also cause the floor plans to be less accurate. For example, when there are obstructed windows or many objects behind the windows, the floor plans become less accurate. Also, taking laser scans from multiple points outside of the building offers a more complete scan for a more accurate floor plan. Most buildings have walls that cannot be seen from looking through windows on one side of the building, and it becomes necessary to move to other windows to detect these.

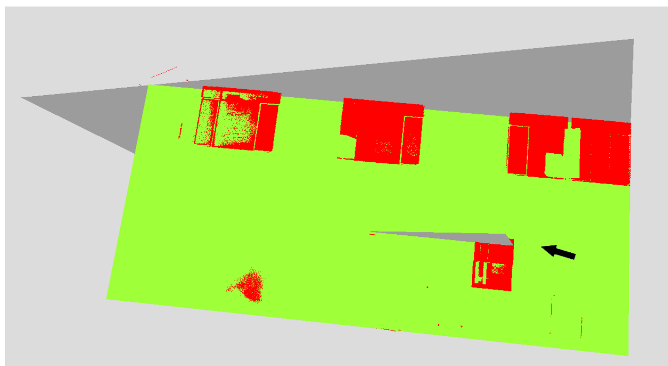
We have shown that our method provides a reasonable estimate at the floor plans of buildings, without the requirement of entering the structure. Because of the dependence on the laser being able to get a clear scan of the building, the completeness of the floor plans is limited. Therefore, expecting this algorithm to yield a floor plan very similar to the ground-truth floor plan is unreasonable. Nonetheless, our method can be used to provide a reasonable estimate of the floor plan in many applications where room by room interior inspection is infeasible.

REFERENCES

1. R. A. Morrison, J. T. Turner, M. Barwick, and G. M. Hardaway, "Urban reconnaissance with an ultra high resolution ground vehicle mounted laser radar," *Proceedings of SPIE, Laser Radar Technology and Applications X*, vol. 5791, pp. 89–96, May 2005.
2. S. Thrun, C. Martin, Y. Liu, D. Hahnel, 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 and Automation*, vol. 20, no. 3, pp. 433–443, June 2004.
3. B. Jensen, J. Weingarten, S. Kolski, and R. Siegwart, "Laser range imaging using mobile robots: From pose estimation to 3d-models," *Proc. 1st Range Imaging Research Day*, pp. 129–144, 2005.
4. S. Jo, Q. M. Shahab, Y. Kwon, and S. C. Ahn, "Indoor modeling for interactive robot service," *International Joint Conference SICE-ICASE*, pp. 3531–3536, October 2006.



(a)



(b)

Figure 13. Building C results after plane fitting; (a) shown from above without ceiling or floors for ease of viewing; (b) shown from the front. In this view, the green plane is the front wall, the gray planes are the ceilings, and the red corresponds to the raw data points. Note that the plane serving as the boundary between the two floors is small, due to a limited number of points on that floor. It is at the location specified by the black arrow.

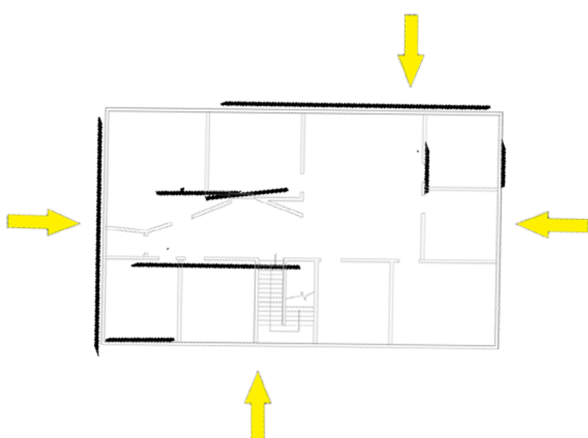


Figure 14. Estimated floor plan of Building C (dark) superimposed on top of original floor plan (light) for the second floor