Two Stage View Planning for Large-Scale Site Modeling

Paul S. Blaer Peter K. Allen

Department of Computer Science, Columbia University
New York, NY 10027

Abstract

We present a systematic view planning method to assist construction of 3-D models of large outdoor sites using a mobile robot platform with mounted scanner. In the first stage, we begin with a 2-D site footprint and the planner generates a minimal set of sufficient covering views. These views which incorporate constraints on the scanner, including field of view, minimum and maximum scanning distance, and grazing angle, serve as the initial set of scans which yields an approximate 3-D model of the site. In the second stage, we update this model by using a voxel-based occupancy procedure to plan and acquire the next best view. The algorithm continues to update the model sequentially until an accurate and complete 3-D model is obtained. Results are shown for a segment of the Columbia University campus. The system can also be used as a planning tool for manual construction of 3-D site models.

1 Introduction

Accurate three-dimensional models of large outdoor structures, such as buildings and their surroundings, have many uses. These models can provide an educational walk around a structure that is thousands of miles away. They can allow engineers to analyze the stability of a structure and then test possible corrections without endangering the original. They can allow us to preserve historical sites that are in danger of destruction, and they can allow us to preserve archaeological sites at various stages of an excavation. In all of these cases, it is important to have an accurate computer based 3-D model of the large scale outdoor structure.

Methods for acquiring such models have progressively increased in accuracy and have evolved from manual methods to more automated methods. Today there are a number of laser range scanners on the market that will quickly generate a dense point cloud of measurements. With a sufficient number of scans around the object being measured, one can generate models of centimeter or better accuracy.

Although the models obtained by laser scanning are now very accurate and the acquisition process is fast and automated, there is still a major human component involved. The scanning sensor must be physically moved from location to location. In addition, a plan must be laid out to determine where to take each individual scan. This requires choosing efficient views that will cover the entire surface area of the structure without occlusions from other objects and without self occlusions from the target structure itself. This is the essence of the so–called view planning problem. Manually choosing the views can be time consuming in itself. In addition, it may require some trial and error which increases the number of needed scans. Since taking a single scan can take up to hour, unneeded scans are a large problem.

We propose to automate this view planning task. Our view planning method has two basic stages. In section 3, we describe the initial static modeling phase. The system computes a set of initial views of the target region. Once these scans have been acquired, we need to refine our model. In section 4, we describe the second stage in which a voxel-based method is used to choose the next best view from information in the initial model. This phase now becomes dynamic as each new scan updates our model and provides new information for the next best view. The entire method can produce an accurate and complete 3-D model of large complicated structures.

We can further automate this process by integrating the scanner and view planning software with a mobile robot. We can mount the scanning equipment on our mobile robot platform, AVENUE [1] (see Fig. 3(a)), which is capable of localizing and navigating itself through an urban environment. The view planning software for model construction is then integrated into the robot's navigation system. This combined system can now plan the views for model construction and then automatically acquire the views.

2 Related Work

Currently there are a number of other research projects attempting to construct three-dimensional models of urban scenes and outdoor structures. These projects include the 3-D city model construction project at Berkeley [8], the outdoor map building project at the University of Tsukuba [17], the MIT City Scanning Project [27], and the volumetric robotic mapping project by Thrun et al [28]. The last of these projects initially focused on the mapping of mines but has recently been extended to outdoor scenes. For the most part, however, these methods leave the actual planning component to a human operator.

The view planning problem can be described as the task of finding a set of sensor configurations which efficiently and accurately fulfill a modeling or inspection task (see [24] and [21]). The literature can be divided into three separate categories. The first two deal with model-based and non-model-based methods. The third describes methods applicable to view planning for a mobile robot.

The model-based methods are the inspection methods in which the system is given some initial model of the scene. Early research focused on planning for 2-D camera-based systems. Included in this are works by Cowan and Kovesi [5] and by Tarabanis et al [25]. Later, these methods were extended to the 3-D domain in works by Tarbox and Gottschlich [26] and by Scott et al [20]. We can also include the art gallery problems in this category. In two dimensions, these problems can be approached with traditional geometric solutions such as in Xie el al [30] and with randomized methods such as in González-Baños et al [9]. The art gallery approach has also been applied to 3-D problems by Danner and Kavraki [7].

The non-model-based methods seek to generate models with no prior information. These include volumetric methods such as Connolly [4], Banta et al [2], Massios and Fisher [14], and Soucy et al [23]. There are also surface-based methods which include Maver and Bajcsy [15], Pito [18], Reed and Allen [19], and Sequeira et al ([13, 22]). A statistical approach is taken by Whaite and Ferrie [29].

View planning for 2-D map construction with a mobile robot is addressed by González-Baños et al [10] and Grabowski et al [12]. View planning for 3-D scenes with a mobile robot is addressed by Nüchter et al [16].

3 Constructing the Initial Model

In the first stage of our modeling process, we wish to compute an initial model of the target region. This model will be based on limited information about the site and will most likely have gaps in the data which must be filled in during the later stage of the algorithm. The data acquired in this initial stage will serve as a seed for the boostrapping method used to complete the modeling process.

3.1 Planning the Initial Views

The procedure for planning the initial views makes use of a two-dimensional ground map of the region to plan a series of environment views for our scanning system to acquire. Maps such as these are commonly available for large scale sites. All scanning locations in this initial phase are planned in advance, before any data acquisition occurs.

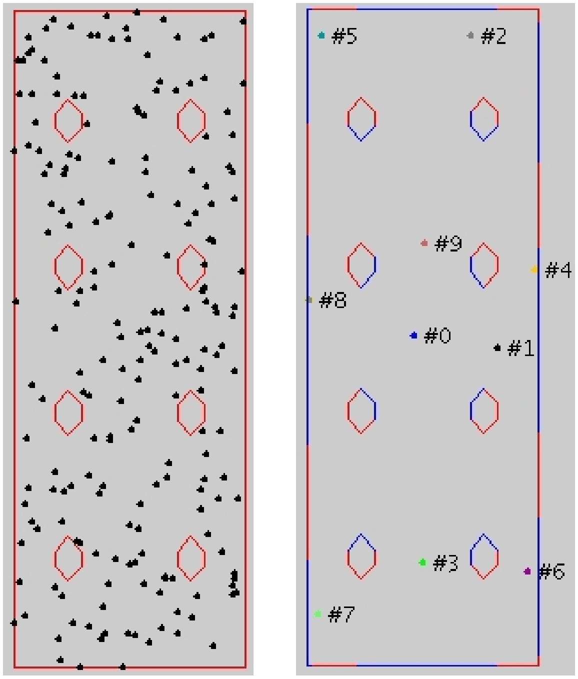
Planning these locations resembles the classical art gallery problem, which asks where to optimally place guards such that all walls of the art gallery can be seen by the set of guards. Solutions to this well-known problem can be applied to our initial planning task. We wish to find a set of positions for our scanner such that it can image all of the known walls in our 2-D map of the environment. The entire view planning problem can be solved by making the simplifying assumption that if we can see the 2-D footprint of a wall then we can see the entire 3-D wall. In practice, this is never the case, because a 3-D part of a building facade or other wall that is not visible in the 2-D map might obstruct a different part of the scene. However, for an initial model of the scene to be used later for view refinement, this assumption should give us enough coverage to be sufficient.

The traditional art gallery problem assumes that the guards can see all the way around their location, that is, they have a 360 degree field of view. It also assumes that the guards have an unlimited distance of vision and that the guards can view a wall at any grazing angle. None of these assumptions are true for most laser scanning systems, so the traditional methods do not apply exactly to our problem. González-Baños et al [9] proposed a randomized method for approximating solutions to the art gallery problems. We have chosen to extend their randomized algorithm to include the visibility constraints of our sensor, such as minimum and maximum range, field of view, and grazing angle.

In our version of the randomized algorithm, a set of initial scanning locations are randomly distributed throughout the free space of the region to be imaged. The visibility polygon of each of these points is computed based on the constraints of our sensor. Finally, an approximation for the optimal number of viewpoints needed to cover the boundaries of the free space is computed from this set of initial locations.

For our initial test of this algorithm, we used a simulated environment. The region (see Fig. 1) represents a long hallway with eight hexagonal pillars evenly spaced and located not far from the walls. In this test region, we chose to use an initial set of 200 random scanning locations (see Fig. 1(a)).

Next, the visibility of each of the viewpoints was computed. We used the ray-sweep algorithm [11] to compute the visibility polygon, which has two types of edges. The first contains the obstacle edges that are on the boundary of the region's free space. The second contains intermediate edges which lie in the interior of the free space. We then discarded the intermediate edges so that the only remaining edges of the visibility polygon were on the boundary of



(a)

(b)

Figure 1: (a) The initial simulated test region for the first phase of our view planning algorithm together with an initial set of 200 viewpoints randomly distributed throughout the free space of the region. (b) The final set of 10 scanning locations chosen for our simulated test region. The clipped obstacle edges for scanning location #0 are indicated in blue. (All the figures in this paper should be viewed in color.)

the free space. Each of these edges was clipped such that the remaining edges satisfied the constraints inherent to our scanning device. This gave a set of obstacle edges on the boundary that a viewpoint at a given location could actually image (see Fig. 1(b)).

For the range constraints, we set a maximum and minimum range for the scanner. To apply the constraint, we first used the maximum range of the scanner to create a circle around our device location. We then clipped all of the currently visible obstacle edges with this circle. In doing this, there are three cases to consider. (1) If both end points of the edge are inside the circle, then the edge is completely visible and we keep it entirely. (2) If only one end point is inside the circle, then we replace this line segment with a clipped segment that goes from the point already inside the circle to the point at which the line intersects the circle. (3) If both end points are outside of the circle, we must determine whether or not the line intersects the circle at all. If the line does intersect the circle, we replace the original end points with the two intersection points; otherwise, we simply discard the line. For the minimum scanner range, we used a similar procedure. A circle whose radius is the minimum range was used and the parts of line segments that fell inside it were dropped.

We also constrained the grazing angle. Our sensor loses accuracy at grazing angles larger than 70^o. Consider figure 2, in which our camera is located at point C and the edge that we are attempting to clip is E. The known distance between the camera and the edge is D, and the grazing angle for an arbitrary point on the edge is θ at distance R from point C. We find the subsegment of E for which θ is no greater than some fixed value (in our case 70^o) for all points on the subsegment.

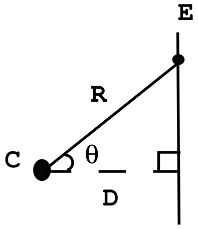


Figure 2: The grazing angle.

We can additionally constrain our sensor to have a limited field of view. Given the field of view and the heading of the sensor, we compute the angle between the heading and the line between the center of the camera system and each end point of the currently visible obstacle edges. For obstacle edges in which one or both end points are at an angle outside the field of view, we find a subsegment for which both end points are within the field of view (if one exists). We then replace that obstacle edge with its visible subsegment. To make use of this constraint, we must also generate random headings along with each potential view location in the previous step of the algorithm. However, no field of view constraint was actually necessary with our current scanner and its 360^o field of view, but it does allow the method to be extended to other sensors.

Finally, we utilized a greedy cover algorithm to select an approximation for the minimum number of viewpoints needed to cover the entire scene. We first select the viewpoint which sees the largest amount of the boundary, and we then remove that section of the boundary from the coverage needed from the remaining potential viewpoints. We repeat this process until either the entire boundary has been covered or until adding additional viewpoints adds too small an amount of new information to warrant continued scanning. In the simulated example, our algorithm usually returns between eight and ten scanning locations (see Fig. 1(b)) with 100% coverage of the region's obstacle boundary. The 2-D planner is summarized in algorithm 1.

Algorithm 1 The 2-D View Planning Algorithm. It must be given an initial 2-D map M to plan the initial views.

```
procedure 2DPLAN($M$)
    Randomly distribute candidate views $V$ in the map
    for all views $v$ in $V$ do
        Compute the visibility polygon $p$ of $v$
        Clip $p$ to sensor range
        Clip $p$ to maximum grazing angle
        Clip $p$ to the field of view
    end for
    Find a greedy cover, $G$, from the potential views $V$
    return $G$                                  ▷ the planned views
end procedure
```

3.2 Initial Modeling of the Campus

As our first actual test of the initial view planning phase, we chose to model the northern end of the Columbia University campus shown in figure 3(b). For our test, we set the threshold such that the algorithm terminated if additional scans added less than 2% of the total boundaries of the target region. Our algorithm (see Alg. 1) typically returned between eight and ten scanning locations for our test area (see Fig. 3(c)) giving us a coverage of 95% of the region's obstacle boundary.

Once the initial set of viewpoints were chosen, the scans needed to be acquired. In this paper, we are only testing the view planning algorithm, and therefore all scans were acquired manually and registered by manually surveying the location of each individual scan. In this particular run of the algorithm, we generated 9 scanning locations. We took scans at each of the nine locations chosen by the planner and registered them. Figure 4 shows the model being constructed from the 2-D plan. The resulting model from this algorithm consists of registered sets of point clouds. These point clouds can then be meshed, by one of many methods, for use in other applications.

4 Three Dimensional Modeling

After the initial modeling phase has been completed, we have a preliminary model of the environment. The model will have holes in it, many caused by originally undetectable occlusions. We now implement a 3-D view planning system that makes use of this initial three-dimensional model to plan efficiently for further views. This subsequent modeling phase does not plan all of its views at once. Instead, it takes the initial model and plans a single next best view that will acquire what we estimate to be the largest amount of new information possible, given the known state of the world. Once this scan has been acquired, the new data are integrated into the model of the world and the next best view is planned.

4.1 Voxel Space

Our method for the 3-D modeling stage requires a different data representation than just the simple point cloud that we were using in the initial stage. We need a way to tell what parts of the scene have been imaged and what parts have not. To do this, we maintain a second representation of the world which keeps track of seen-empty, seen-occupied, and unseen portions of the region to be imaged. This can most easily be represented by a voxel map [6]. Because this is a large scale imaging problem, the voxels can be made rather large and still satisfy their purpose. A voxel size of one meter cubed is sufficient to allow for computing occlusions in our views. Although the planner uses a reduced resolution model of the data, the full data are still used in constructing the final model.

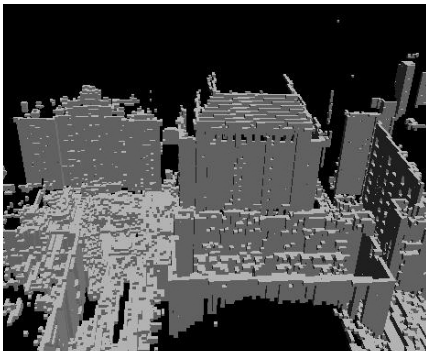


Figure 5: A representation of the seen-occupied cells of the voxel grid. It was constructed from the point cloud acquired in section 3.2.

The voxel representation (see Fig. 5) is generated from the point cloud. Before any data have been acquired, all voxels in the grid are labeled as unseen. When a new scan is taken, voxels that contain at least one data point from that scan are marked as seen-occupied (even if a previous scan had labeled a voxel as seen-empty). For any data point to have been acquired, there must have been an unoccluded line of sight between the scanner position and that data point. A ray is then traced from each data point back to the scanner position. Each unseen voxel that it crosses is marked as seen-empty. If the ray passes through a voxel that had already been labeled as seen-occupied, it means that the voxel itself may already have been filled by a previous scan or another part of the current scan. This means that the voxel itself is only partially occupied and we allow the ray to pass through it without modifying its status as seen-occupied. Using this method, our initial model is inserted into the voxel space and we subsequently update this space.

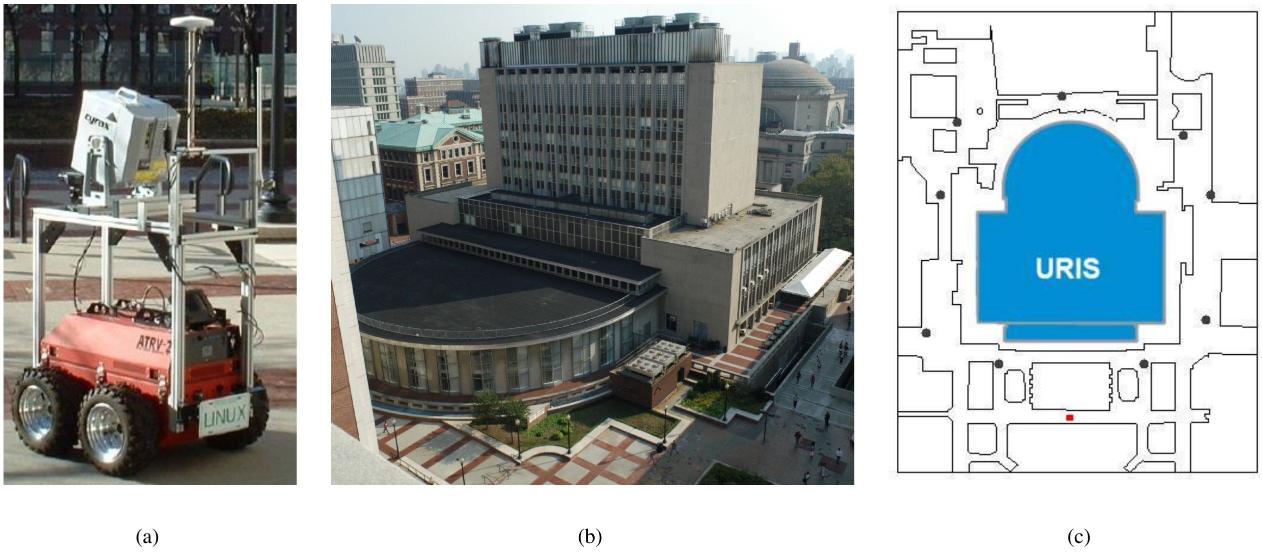


Figure 3: (a) The ATRV-2 Based AVENUE Mobile Robot. (b) A photograph of the center of our test region, taken from the roof of a neighboring building (picture courtesy of Alejandro Troccoli). (c) The 2-D map of the building footprints on the northern portion of the Columbia campus. Also shown are the 9 scan locations (shown as black dots) determined by the initial two-dimensional view planner. The views from these locations cover 95% of the 2-D outline of Uris Hall. The first NBV computed by the 3-D planner is shown as a red square.

4.2 Next Best View

Our approach to this final modeling phase takes its cue from Pito's work [18], in which a grid of cells called the "positional space" is arranged around the object to be modeled. In Pito's work, the objects being imaged are small and the sensor is free to move anywhere around the object. He considers only patches of unknown information at the boundary of his current mesh and projects rays back from them into the positional space. Pito chooses the cell in the positional space that views the largest number of these unknown patches as the next view.

We extend this idea to a voxel–based representation. In our case, we are restricted to operating on the ground plane with our mobile robot. We can exploit the fact that we have a reasonable two-dimensional map of the region. This 2-D map gives us the footprints of the buildings as well as a good estimate of the free space on the ground plane in which we can operate. We mark the voxels which intersect this ground plane within the free space defined by our 2-D map as being candidate views. We can then use these marked voxels on the ground plane as our version of the "positional space."

We wish to choose a location on this ground plane grid that maximizes the number of unseen voxels that can be viewed from a single scan. Considering every unseen voxel in this procedure is unnecessarily expensive and should be avoided. At the end of the first stage of our method, much of the environment has already been imaged and many of the "unseen" voxels will actually be regions in the interior of buildings. Instead, we need to focus on those unseen voxels that are most likely to provide us with useful information about the facades of the buildings. These useful voxels are the ones that fall on the boundaries between seen-empty regions and unseen regions. These boundary regions are most likely to contain previously occluded structures and, in addition, are likely to be viewable by the scanner. If an unseen voxel is completely surrounded by seen-occupied voxels or even by other unseen voxels, then there is a good chance that it may never be visible by any scan. We therefore choose to consider only unseen voxels that are adjacent to at least one seen-empty voxel. Such unseen voxels will be labeled as boundary unseen voxels.

Now that we have a set of appropriate unseen voxels to consider, we proceed with the optimization. As possible positions for the next best view, we use the centers of the voxels which intersect the ground plane within the region's free space. At each such position, we keep a tally of the number of boundary unseen voxels that can be seen from that position. Each position's voxel tally starts at 0 and is incremented for every boundary unseen voxel which that position can view.

To determine whether a boundary unseen voxel can be viewed, we trace rays from its center to the center of each voxel on the ground plane. If the ray intersects any voxel that is seen-occupied, we discard the ray because it may

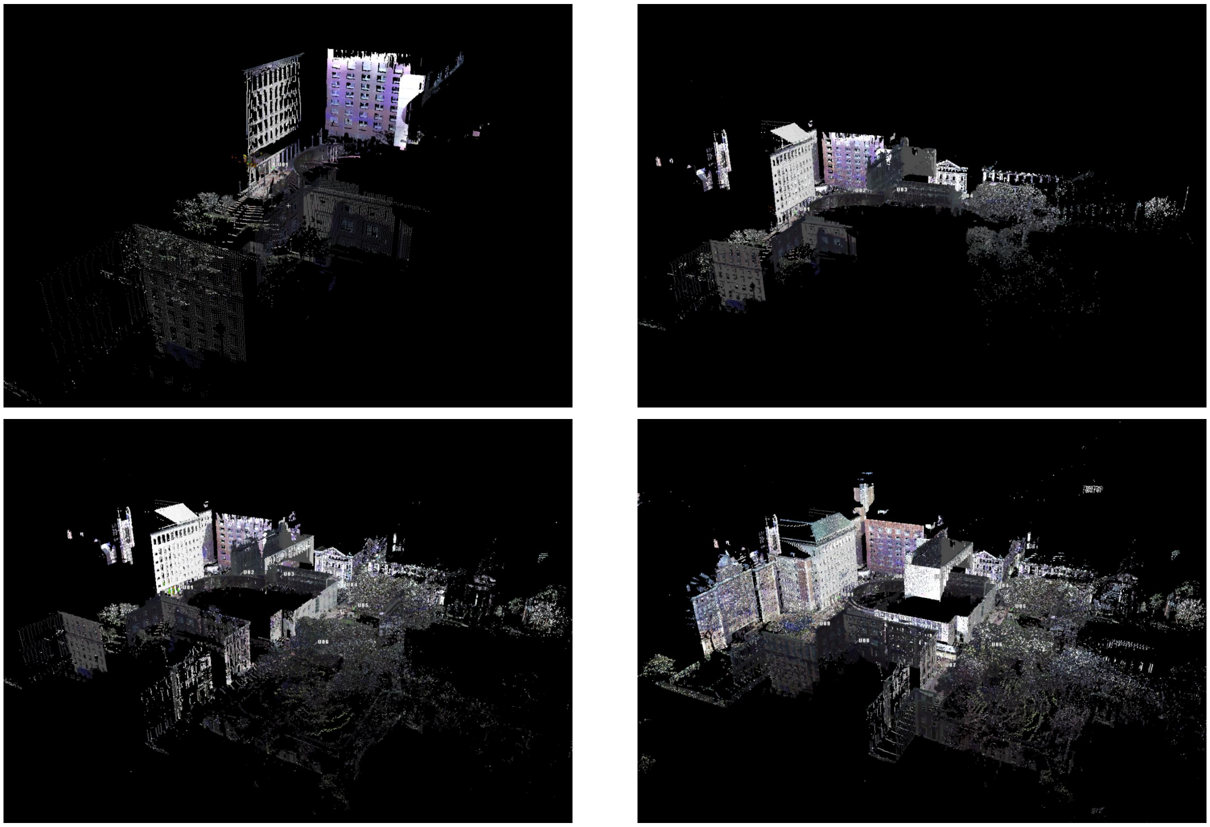


Figure 4: This figure shows the initial campus model being constructed sequentially from the 2-D plan. From the top left, going clockwise: The first scan. Registered scans 1 through 4. Registered scans 1 through 6. The complete initial model with all 9 scans. The scans are texture mapped with images taken from the scanner's built-in camera. (These images are best viewed in color.)

be occluded by the contents of that occupied voxel. If the ray intersects any voxel that is unseen, we discard the ray because we are uncertain of the contents of that voxel and it is still possible that it will be occluded. We must also consider the minimum and maximum range of the scanner. If the length of the ray is outside the scanner's range, then we discard the ray.

We also want to consider the grazing angle condition. If the grazing angle between a ray and the normal to the surface that we expect at a boundary unseen voxel is larger than the maximum angle allowed by our sensor, we discard the ray. Since, by definition, these unseen voxels are unknown, we do not have a good idea of what the surface normal at that location would be. To estimate this, we can look at the local area of voxels surrounding the unseen voxel in question. We can construct a plane that on average divides the unseen voxels in that local region from the seen-empty voxels. This plane is an estimate of the surface that divides the unseen region from the seen-empty region, and its normal can be used to compute an estimated grazing angle.

If a ray has not been discarded by the occlusion, range, or grazing angle condition, we can safely increment the ground plane position that the ray intersects. At the end of this calculation, the ground plane position with the highest tally is chosen as the robot's location to take the next scan. The robot then plans a Voronoi-based path [3] and navigates to the chosen position. It triggers a new scan once it has arrived, and that scan is integrated into both the point cloud model and the voxel representation. This entire second stage is then repeated until we reach a sufficient level of coverage of the site. To decide when to terminate the algorithm, we look at the number of boundary unseen voxels that would be resolved by the next iteration. If that number falls below some small threshold value, then the algorithm terminates; otherwise, it continues. The 3-D planner is summarized in algorithm 2.

4.3 Refining the Initial Campus Model

In section 3.2 we generated an initial viewing plan by using algorithm 1. We acquired and registered each of these planned views with our scanner in order to build an initial

Algorithm 2 The 3-D View Planning Algorithm. It must be given an initial model of the world C, a set of possible scanning locations P, and a threshold value for the number of acceptable unseen voxels as a stopping condition.

```
procedure 3DPLAN($C$,$P$,$threshold$)
    Initialize voxel space from $C$
    for all unseen voxels, $u$, in the voxel space do
        if $u$ has a seen empty neighbor then
            add $u$ to $U$
        end if
    end for
    loop
        for all potential views $p$ in $P$ do
            Count members of $U$ that are visible
        end for
        Acquire scan at $p$ with largest count
        if $count(p) < threshold$ then
            break
        end if
        Register and merge new scan into $C$
        Update voxel space with $C$
        Recompute $U$ from the new voxel space
    end loop
    return $C$                                  ▷ the updated model
end procedure
```

model of the world. Now we wish to refine our model using our 3-D planning algorithm (see Alg. 2).

Our voxel space was initialized such that all voxels were labeled unseen. The space was successively updated. Figure 6 summarizes the number of labeled voxels at each iteration of the algorithm.

First, the initial model was converted to voxels and inserted into the voxel space. Voxels that contained data points from the initial model were labeled as seen-occupied. We then carved the seen-empty voxels by ray tracing from the positions of the scanning system. Our grid had a total of approximately 15.5 million voxels. After the initial model, 76,941 voxels were seen occupied and 114,372 voxels were labeled as unseen, with the rest labeled seen empty. Next, we computed the set of boundary unseen voxels. It turned out that only a small number of the unseen voxels were in this set (21,756). This operation pruned the number of voxels whose visibility we needed to consider by an order of magnitude. At this point, we computed the potential viewing location that could see the largest number of those boundary unseen voxels. This location is shown as the red square in figure 3(c).

This view looked directly at the facade of the central building of our target region. The 2-D planner had computed views that saw this facade from the corners. The building, however, has two parts to it, a wide base and a thin tower. The 2-D planner only took into account the footprint of the building, and as a result the scans from the corners were unable to image much of the tower section of the building. The 3-D planner's first choice for the next best view, was a head-on scan of the building that filled in the large unseen region of the tower's facade (see Fig. 7).

For this experiment, we chose a threshold of 200 boundary unknown voxels as the cutoff for our algorithm. When it was estimated that the next best view would net fewer than this threshold number of voxels, we decided that it would not be worth the effort to continue scanning. In our test, we reached this threshold after the third NBV. Although there were still a sizeable number of boundary unseen voxels (15,156) in the scene, we were restricted by where we could place the robot. Many of these unseen voxels were in locations on the roofs of buildings, and we could not image them from a ground location.

At the end of:	Seen Occupied	Total Unseen	Boundary Unseen	Visible to NBV
2-D Plan	76,941	114,372	21,765	1,025
NBV 1	78,212	111,216	18,540	972
NBV 2	79,005	109,571	15,327	322
NBV 3	79,243	109,011	15,156	122

Figure 6: A summary of the number of labeled voxels after each iteration of the 3-D planning algorithm. The last column indicates the number of boundary unseen voxels that are actually visible to the next best view.

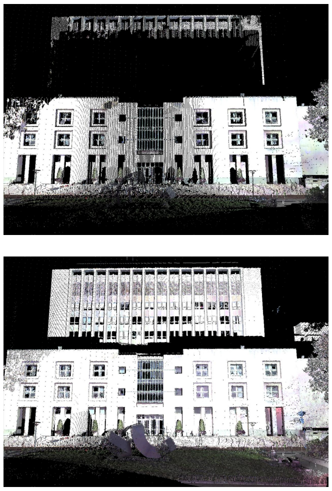


Figure 7: The top is the facade of Uris Hall, scanned from the views generated by the 2-D planner. The bottom is the facade of the same building after acquiring the first NBV.

5 Conclusion

We have presented a systematic method to automatically plan views for constructing 3-D models of large outdoor structures. The procedure starts from a 2-D ground map of the environment, which defines the open space available to the robot, and then progresses through two distinct stages. A preliminary 3-D model is constructed by having the robot compute and acquire a set of initial scans using our 2-D view planning algorithm. This initial model is then refined by a dynamic modeling phase which makes use of a voxel representation of the environment and which successively plans next best views for the robot to acquire a complete model.

The method was successfully tested on the northern half of the Columbia University campus. With our two stage procedure, we were able to construct a detailed 3-D model of the structures in that part of the campus.

For sites substantially larger than the one we tested, one encounters an additional difficulty. The number of unseen voxels which must be considered for the NBV algorithm becomes extremely large even though we limit consideration to the boundary unseen voxels. Our future research will address this important problem associated with very large sites such as New York City's Governors Island and the Romanesque churches of the Bourbonnais region of France.

Acknowledgments

Funded by NSF grants IIS-0121239 and ANI-00-99184.

References

[1] P. K. Allen, I. Stamos, A. Gueorguiev, E. Gold, and P. Blaer. Avenue: Automated site modeling in urban environments. In *3DIM*, pages 357–364, 2001.

[2] J. E. Banta, Y. Zhien, X. Z. Wang, G. Zhang, M. T. Smith, and M. A. Abidi. A "best-next-view" algorithm for three-dimensional scene reconstruction using range images. *SPIE*, 2588:418–429, October 1995.

[3] P. Blaer and P. K. Allen. Topbot: automated network topology detection with a mobile robot. In *IEEE ICRA*, pages 1582–1587, 2003.

[4] C. I. Connolly. The determination of next best views. In *IEEE ICRA*, pages 432–435, 1985.

[5] C. K. Cowan and P. D. Kovesi. Automatic sensor placement from vision task requirements. *IEEE PAMI*, 10(3):407–416, May 1988.

[6] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, pages 303–312, 1996.

[7] T. Danner and L. E. Kavraki. Randomized planning for short inspection paths. In *IEEE ICRA*, pages 971–978, 2000.

[8] C. Frueh and A. Zakhor. Constructing 3d city models by merging ground-based and airborne views. In *IEEE CVPR*, pages 562–569, 2003.

[9] H. González-Banos and J. C. Latombe. A randomized art-gallery algorithm for sensor placement. In *Proc. of the 17th annual symposium on Computational Geometry*, pages 232–240, 2001.

[10] H. González-Banos, E. Mao, J. C. Latombe, T. M. Murali, and A. Efrat. Planning robot motion strategies for efficient model construction. In *International Symposium of Robotics Research*, pages 345–352, 1999.

[11] J. E. Goodman and J. O'Rourke. *Handbook of discrete and computational geometry*. CRC Press, Inc., 1997.

[12] R. Grabowski, P. Khosla, and H. Choset. Autonomous exploration via regions of interest. In *IEEE IROS*, pages 27–31, 2003.

[13] K. Klein and V. Sequeira. View planning for the 3d modelling of real world scenes. In *IEEE/RSJ IROS*, 2000.

[14] N. A. Massios and R. B. Fisher. A best next view selection algorithm incorporating a quality criterion. In *British Machine Vision Conf.*, 1998.

[15] J. Maver and R. Bajcsy. Occlusions as a guide for planning the next view. *IEEE PAMI*, 15(5):417–433, May 1993.

[16] A. Nüchter, H. Surmann, and J. Hertzberg. Planning robot motion for 3d digitalization of indoor environments. In *ICAR*, pages 222–227, 2003.

[17] K. Ohno, T. Tsubouchi, and S. Yuta. Outdoor map building based on odometry and rtk-gps position fusion. In *IEEE ICRA*, pages 684–690, 2004.

[18] R. Pito. A solution to the next best view problem for automated surface acquisition. *IEEE PAMI*, 21(10):1016–1030, October 1999.

[19] M. K. Reed and P. K. Allen. Constraint-based sensor planning for scene modeling. *IEEE PAMI*, 22(12):1460–1467, December 2000.

[20] W. R. Scott, G. Roth, and J.-F. Rivest. View planning for multi-stage object reconstruction. In *International Conference on Vision Interface*, 2001.

[21] W. R. Scott, G. Roth, and J.-F. Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Computing Surveys*, 35(1):64–96, 2003.

[22] V. Sequeira and J. Gonçalves. 3d reality modelling: Photo-realistic 3d models of real world scenes. In *3DPVT*, 2002.

[23] G. Soucy, F. G. Callari, and F. P. Ferrie. Uniform and complete surface coverage with a robot-mounted laser rangefinder. In *IEEE/RSJ IROS*, pages 1682–1688, 1998.

[24] K. A. Tarabanis, P. K. Allen, and R. Y. Tsai. A survey of sensor planning in computer vision. *IEEE Transactions on Robotics and Automation*, 11(1):86–104, February 1995.

[25] K. A. Tarabanis, R. Y. Tsai, and P. K. Allen. The mvp sensor planning system for robotic vision tasks. *IEEE Transactions on Robotics and Automation*, 11(1):72–85, February 1995.

[26] G. H. Tarbox and S. N. Gottschlich. Planning for complete sensor coverage in inspection. *Computer Vision and Image Understanding*, 61(1):84–110, January 1995.

[27] S. Teller. Automatic acquisition of hierarchical, textured 3d geometric models of urban environments: Project plan. In *Proc. of the Image Understanding Workshop*, 1997.

[28] S. Thrun, D. Hähnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker. A system for volumetric robotic mapping of abandoned mines. In *IEEE ICRA*, 2003.

[29] P. Whaite and F. P. Ferrie. Autonomous exploration: Driven by uncertainty. *IEEE PAMI*, 19(3):417–433, March 1997.

[30] S. Xie, T. W. Calvert, and B. K. Bhatacharya. Planning views for the incremental construction of body models. In *ICPR*, pages 154–157, 1986.