View Planning for Automated Site Modeling

Paul S. Blaer and Peter K. Allen

Department of Computer Science, Columbia University

{pblaer,allen}@cs.columbia.edu

Abstract— **We present a systematic method for constructing 3-D models of large outdoor sites. The method is designed for a mobile robot platform and incorporates automated acquisition of scanned data as well as automated view planning and model construction. In our modeling process, we first use a preliminary view or set of preplanned views to yield an initial, approximate, 3-D model of the target structure. Then, we update this model by using a voxel-based procedure to plan and acquire the next best view. This updating is repeated sequentially until an accurate and complete 3-D model is finally obtained. The method was successfully tested on a portion of the Columbia University campus.**

I. 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. At the simpler end of the spectrum, one can send a team of surveyors to take measurements by hand and then have a designer put together a model from those measurements. More sophisticated tools do exist. 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, and each scanning operation itself can take up to one hour depending on the type of sensor and the density of the scan. 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.

We propose to automate this view planning task by mounting the scanning equipment on a mobile robot platform,

AVENUE [1] (see Fig. 1), that is capable of localizing and navigating itself through an urban environment. View planning software for model construction is then added to the robot's navigation system. This paper presents our work on the view planning component of this system.

Our method of model building has two basic steps. In an initial static modeling phase, the system acquires a preliminary view or set of views of the target region. In the second phase, 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.

II. 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 [2], the outdoor map building project at the University of Tsukuba [3], the MIT City Scanning Project [4], and the volumetric robotic mapping project by Thrun et al [5] which 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 [6] and [7]). 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 [8] and by Tarabanis et al [9]. Later, these methods were extended to the 3-D domain in works by Tarbox and Gottschlich [10] and by Scott et al [11]. 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 [12] and with randomized methods such as in González-Baños et al [13]. The art gallery approach has also been applied to 3-D problems by Danner and Kavraki [14].

The non-model-based methods seek to generate models with no prior information. These include volumetric methods

Fig. 1. On the left is the ATRV-2 based AVENUE Mobile Robot. In the center is a photograph of our test case, Uris Hall, taken from the roof of a neighboring building (picture courtesy of Alejandro Troccoli). On the right is a 2-D map of the building footprints on the northern portion of the Columbia campus. Also shown on the right are the 9 scan locations (shown as black dots) determined by the initial two-dimensional view planner of method B. The views from these locations cover 95% of the 2-D outline of Uris Hall.

such as Connolly [15], Banta et al [16], Massios and Fisher [17], and Soucy et al [18]. There are also surface-based methods which include Maver and Bajcsy [19], Pito [20], Reed and Allen [21], and Sequeira et al ([22], [23]). A statistical approach is taken by Whaite and Ferrie [24].

View planning for 2-D map construction with a mobile robot is addressed by González-Baños et al [25] and Grabowski et al [26]. View planning for 3-D scenes with a mobile robot is addressed by Nüchter et al [27].

III. PLATFORM AND ENVIRONMENT

The platform for this project is the AVENUE mobile robot ([28], [1]). This robot has as its base unit the ATRV-2 model originally manufactured by RWI (see Fig. 1). To this base unit, we have added additional sensors including a differential GPS unit, a laser range scanner, two cameras (including one omnidirectional camera [29]), a digital compass, and wireless Ethernet. The system is currently capable of localizing itself with its sensor suite, planning a path to a specified location, and navigating itself to that location.

Our initial tests of this system were performed on the northern half of the Columbia University campus. We will continue to use the campus and its surrounding area as a test bed, but as our system matures we will move to larger sites such as New York City's Governors Island. The site modeling procedure described in this paper was specifically tested on Columbia's Business School Building, Uris Hall (see Fig. 1).

IV. CONSTRUCTING THE INITIAL MODEL: PHASE 1

In the first stage of the 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. The first stage has two possible approaches.

A. Utilizing Existing 3-D Information

Method A for constructing an initial model involves utilizing existing 3-D data about the scene. We have access to 3-D data derived from stereo photogrammetry taken from overflights of the Columbia campus. These data, along with an initial scan taken from the ground, form the bases of our initial model.

The photogrammetry provides sparse 3-D data that outlines the rooftops of the buildings that were imaged. By adding interpolated data to this point cloud, we simulate the higher density of the scans generated by our ground-based sensor. This sampling operation was necessary because the second stage 3-D planning method makes use of a voxel representation of the model and densely sampled point data are required for our voxelization procedure.

In addition to these aerial data, we take a single initial scan of the target building from the ground (see Fig. 2) to fill in the model before passing it on to the second stage of our method. This scan is taken from an arbitrary position in the target region. We must then register the aerial data with the initial ground scan. In our final robotic system this will be done utilizing the localization features of our robot. In this paper, we are only testing the view planning algorithm, and therefore the scans are registered by manually surveying the location of each individual scan. All further scans are registered in a similar manner.

B. Planning the Initial Views

Method B for constructing an initial model makes use of the two-dimensional ground map of the region (see Fig. 1) to plan a series of environment views for our robotic scanning system. All scanning locations in this initial phase are planned in advance, before the robot does any data acquisition.
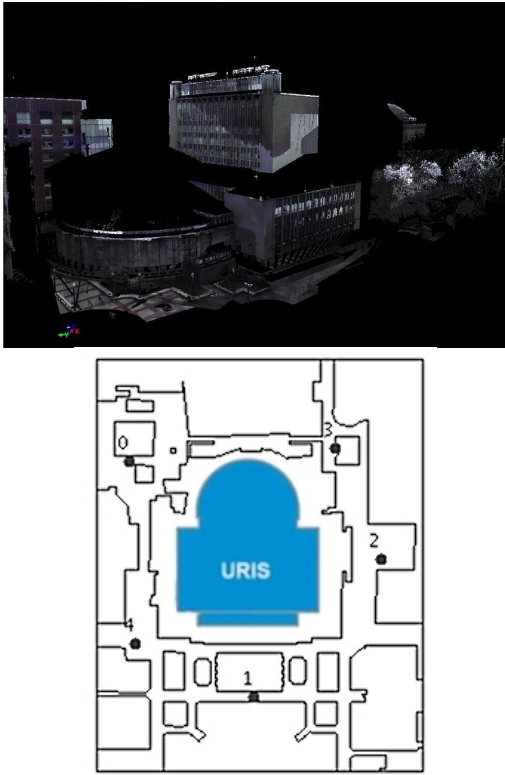
Fig. 2. On the top is the initial ground scan of the target building (with the scanner's automatic texture mapping enabled), best viewed in color. In the map on the bottom the black dots represent the five scan locations used in our 3-D NBV algorithm experiment. Scan 0 is the scan which provided the initial model from phase one, and the remaining scans from phase two are numbered in the order that they were computed by the algorithm.

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 [13] proposed a randomized method for approximating solutions to 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 as well as 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. In our test region of Uris Hall, we chose to use an initial set of 200 random scanning locations. Next, the visibility of each of the viewpoints is computed. We use the ray-sweep algorithm [30] 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 discard the intermediate edges so that the only remaining edges of the visibility polygon are on the boundary of the free space.

Each of these edges is clipped such that the remaining edges satisfy the constraints inherent to our scanning device. For the range constraints, we set a minimum and maximum range for the scanner (with our scanner, between 1 meter and 100 meters is the best range of distances for data acquisition). We also take into account grazing angle, since our sensor loses accuracy at grazing angles larger than 70°. We could additionally constrain our sensor to have a limited field of view, however this is not relevant for our current scanner and its 360° field of view (in two dimensions). This entire procedure gives a set of obstacle edges on the boundary which a viewpoint at a given location can actually image.

Finally, we utilize a greedy cover algorithm to select an approximation for the optimal 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. For our test, we set the threshold such that the algorithm terminates if additional scans add less than 2% of the total boundary of the target object. Our algorithm typically returns between eight and ten scanning locations for our test region (see Fig. 1) giving us a coverage of 95% of the region's obstacle boundary.

Choosing an efficient tour of our chosen viewpoints can be formulated as a standard traveling salesman problem (TSP). Although this problem is NP hard, good approximation algorithms exist to compute a nearly optimal tour of the observation locations. The cost of traveling between two observation points can be computed by finding the shortest unobstructed path between them and then using the length of that path as the cost. The existing path planning module computes the shortest path between two points, and we can use that method to calculate edge costs for our TSP implementation.

Once the initial set of viewpoints and the efficient tour have been chosen, the robot needs to take scan data at each site. The AVENUE system has existing path planning together with localization and navigation modules [31], all of which can be integrated into our view planner.

V. Three Dimensional Modeling: Phase 2

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 model to plan efficiently for further views. In fact, this planning method can be used to fill in more than just small holes, it only needs to be given some starting data from which it can then generate more refined views. In our tests of the method, we started with an initial arbitrarily positioned ground scan together with aerial scan data.

This modeling phase does not plan all of its views at once. Instead, it takes the initial three-dimensional model of the environment 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.

A. 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 would most easily be represented by a voxel map. Because this is a large scale imaging problem, the voxels could be made rather big and would still satisfy their purpose. A voxel size of one meter cubed is sufficient to allow for computing occlusions in our views.

The voxel representation 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 a 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 subsequent scans update this space.

B. Next Best View

Our approach to this final modeling phase takes its cue from Pito's work [20], 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 wheeled 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 cells on this ground plane which are 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. We need to focus on those unseen voxels that are most likely to provide us with useful information about the faces 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. 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.

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 unseen voxels that can be seen from that position. Each position's voxel tally starts at 0 and is incremented by 1 for every unseen voxel that it can view.

To determine whether an 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 could 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 should also consider the grazing angle condition. If the grazing angle between a ray and the surface that we expect at an unseen voxel is larger than the maximum angle allowed by our sensor, we should 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 get

around 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 path 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 unknown 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.

VI. TEST RESULTS

In this section, we report on our test of the view planning algorithm in modeling Uris Hall. For the initial phase of our two stage algorithm, we have chosen to use preexisting 3-D aerial data and an initial ground scan of the target region as described in section IV-A.

We initialized our voxel space such that all cells were labeled unseen. First, the initial model was converted to voxels and inserted into the voxel space. Voxels that contained data points from the first scan were labeled as seen-occupied. We then carved the seen-empty voxels by ray tracing from the origin of the scanning system. Since the overhead data was taken from a great height, we assumed an orthographic projection for the voxels that originated from the aerial data. For initial model data that originated from a ground scan, we carved out the additional seen-empty voxels by ray tracing using a perspective projection from the origin of that scanning system. The resulting model can be seen in figure 3. Red voxels are unseen and gray voxels are seen-occupied.

We performed our next best view computation as described in section V-B. The NBV algorithm chose a view directly behind the building which was a region for which we had no preexisting scan information (see Fig. 3). We then moved the scanner to that location, acquired data, and incorporated those data into our voxel space. This process was repeated until we reached our termination threshold. The results of the first 2 NBV calculations can be seen in figure 3. There were approximately two million voxels in our voxel space. We set our termination threshold to 100 voxels. Our NBV algorithm reached this threshold after four iterations. The four scan locations plus the original ground scan location from the initial modeling phase can be seen in figure 2.

The resulting final model is shown in figure 4. Most of the site was accurately modeled, but there remained three noticeable areas with holes still present. The first problem
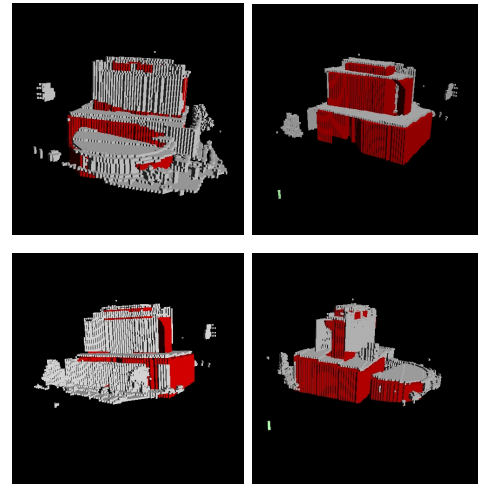

Fig. 3. On the top left is a view of the voxelized initial model generated by the first phase of our algorithm. Voxels in gray are labeled seen-occupied, voxels in red are labeled unseen. On the top right is another view of the initial model. The location of the first NBV computed by the second phase of our algorithm is indicated by the green column. On the bottom left is a view of the model after a scan from the first NBV has been incorporated into the voxel space. On the bottom right is another view of the model in the previous image with the second NBV indicated by the green column.

areas were regions on the front facade of the tower portion of the building. These were areas that were occluded by the lower portion of the building. Since our scanner was restricted to scanning from the ground level and was also restricted in how far away from the target building it could move (because of surrounding buildings), these regions were not actually resolvable with our robotic system. This problem can be seen in the front view of the building on the left side of figure 4. The other two problem regions were on the back side of the model (seen on the right side of Fig. 4). There was a small region near the center of the building that was occluded by a sculpture about which we had no prior knowledge. There was also a region at the corner of the building for which a tree blocked our view. For all these three problems, fewer than 100 voxels would actually have been resolvable from any given NBV, and as a result the algorithm terminated.
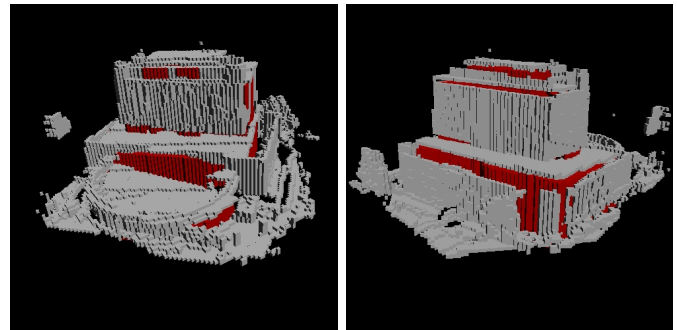

Fig. 4. The final voxel space after all NBV computed scans have been acquired and imported. On the left is a front view, on the right is a back view. These figures are best displayed in color.

VII. Conclusion

We have presented a systematic method for constructing 3-D models of large outdoor structures. The method is designed for a mobile robot platform and incorporates automated acquisition of scanned data as well as view planning and model construction. 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 either by utilizing existing fragmentary 3-D information or by having the robot compute 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 a next best view for the robot to acquire a complete model.

The method was tested on the Columbia University campus and was used to construct an accurate 3-D model of Uris Hall. For this particular test, the initial model was constructed using 3-D aerial data and a single arbitrary ground scan as described in section IV-A. This initial model had only a small fraction of the detailed structure of the targeted building. As a result, the subsequent NBV algorithm had to fill in much of the scene. Nevertheless, the number of additional scans that the algorithm required was small. Therefore, the total distance the robot would need to travel would not be excessive even though the robot would have to retrace its path several times in order to reach the algorithm's next best views.

For sites substantially larger than the one we tested, this path retracing will present a significant problem as the number of required NBV scans becomes large. To minimize this problem, the static view planning method described in section IV-B would be a very useful alternative for obtaining the initial model. This procedure would actually require more initial scans; however, they would all be precomputed, so that an efficient tour of the views could be arranged. With more initial coverage of the scene, the subsequent 3-D planning method would require fewer iterations and smaller travel distances to obtain the complete model. For very large scenes, one encounters an additional difficulty. The number of unseen voxels which must be considered for the NBV algorithm will become extremely large even though we limit consideration of such voxels to the boundaries of the open space. Our future research will address these important problems associated with very large sites.

References

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

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

[3] K. Ohno, T. Tsubouchi, and S. Yuta, "Outdoor map building based on odomoetry and rtk-gps position fusion," in *IEEE ICRA*, 2004, pp. 684–690.

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

[5] 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.

[6] K. A. Tarabanis, P. K. Allen, and R. Y. Tsai, "A survey of sensor planning in computer vision," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 1, pp. 86–104, February 1995.

[7] W. R. Scott, G. Roth, and J.-F. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Computing Surveys*, vol. 35, no. 1, pp. 64–96, 2003.

[8] C. K. Cowan and P. D. Kovesi, "Automatic sensor placement from vision task requirements," *IEEE PAMI*, vol. 10, no. 3, pp. 407–416, May 1988.

[9] 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*, vol. 11, no. 1, pp. 72–85, February 1995.

[10] G. H. Tarbox and S. N. Gottschlich, "Planning for complete sensor coverage in inspection," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 84–110, January 1995.

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

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

[13] 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*, 2001, pp. 232–240.

[14] T. Danner and L. E. Kavraki, "Randomized planning for short inspection paths," in *IEEE ICRA*, 2000, pp. 971–978.

[15] C. I. Connolly, "The determination of next best views," in *IEEE ICRA*, 1985, pp. 432–435.

[16] 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*, vol. 2588, pp. 418–429, October 1995.

[17] N. A. Massios and R. B. Fisher, "A best next view selection algorithm incorporating a quality criterion," in *Brit. Machine Vision Conf.*, 1998.

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

[19] J. Maver and R. Bajcsy, "Occlusions as a guide for planning the next view," *IEEE PAMI*, vol. 15, no. 5, pp. 417–433, May 1993.

[20] R. Pito, "A solution to the next best view problem for automated surface acquisition," *IEEE PAMI*, vol. 21, no. 10, pp. 1016–1030, October 1999.

[21] M. K. Reed and P. K. Allen, "Constraint-based sensor planning for scene modeling," *IEEE PAMI*, vol. 22, no. 12, pp. 1460–1467, December 2000.

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

[23] V. Sequeira and J. G. M. Gonçalves, "3d reality modelling: Photo-realistic 3d models of real world scenes," in *First International Symposium on 3D Data Processing Visualization and Transmission*, 2002.

[24] P. Whaite and F. P. Ferrie, "Autonomous exploration: Driven by uncertainty," *IEEE PAMI*, vol. 19, no. 3, pp. 417–433, March 1997.

[25] 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*, 1999, pp. 345–352.

[26] R. Grabowski, P. Khosla, and H. Choset, "Autonomous exploration via regions of interest," in *IEEE IROS*, 2003, pp. 27–31.

[27] A. Nüchter, H. Surmann, and J. Hertzberg, "Planning robot motion for 3d digitalization of indoor environments," in *ICAR*, 2003, pp. 222–227.

[28] A. Gueorguiev, P. K. Allen, E. Gold, and P. Blaer, "Design, architecture and control of a mobile site modeling robot," in *IEEE ICRA*, 2000, pp. 3266–3271.

[29] S. Nayar, "Omnidirectional video camera," in *DARPA Image Understanding Workshop*, May 12-14 1997, pp. 235–242.

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

[31] A. Georgiev and P. K. Allen, "Localization methods for a mobile robot in urban environments," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 851–864, October 2004.