

## Constraint-Based Sensor Planning for Scene Modeling

Michael K. Reed and Peter K. Allen, *Member, IEEE*

**Abstract**—We describe an automated scene modeling system that consists of two components operating in an interleaved fashion: an incremental modeler that builds solid models from range imagery and a sensor planner that analyzes the resulting model and computes the next sensor position. This planning component is target-driven and computes sensor positions using model information about the imaged surfaces and the unexplored space in a scene. The method is shape-independent and uses a continuous-space representation that preserves the accuracy of sensed data. It is able to completely acquire a scene by repeatedly planning sensor positions, utilizing a partial model to determine volumes of visibility for contiguous areas of unexplored scene. These visibility volumes are combined with sensor placement constraints to compute sets of occlusion-free sensor positions that are guaranteed to improve the quality of the model. We show results for the acquisition of a scene that includes multiple, distinct objects with high occlusion.

**Index Terms**—3D scene reconstruction, model acquisition, sensor planning, active vision.

### 1 INTRODUCTION

To build accurate and complete 3D models of unknown objects or scenes, two fundamental issues must be addressed. The first is how to acquire, model, and integrate noisy and incomplete sensor data into an accurate model. The second issue is the problem of planning the next sensor position and orientation during the modeling process so as to acquire as much new scene information as possible. We have developed a system that creates accurate solid models from multiple range scans. Further, we are able to plan each successive viewpoint to reduce the number of scans needed to create a model. This planning problem has been called the *Next Best View* (NBV) problem and arises naturally in systems that autonomously investigate and model their surroundings [19]. The application area we are focused on is recovery of buildings using range scans in cluttered, high occlusion urban areas, but the methods we describe here are applicable to other modeling tasks as well.

Previous work in 3D model acquisition from range imagery [6], [18], [3], [7], [21], whether with small indoor objects or large outdoor structures such as buildings, has often neglected this planning component in favor of a large number of images and the assumption of complete scene sampling. In cluttered environments such as outdoor, urban scenes, occlusion, and sensor positioning, costs can be high enough to prohibit exhaustive sensing. Our method plans the next sensor viewpoint so that each additional sensing operation recovers object surface that has not yet been modeled and attempts to reduce the number of sensing operations to recover a model. Systems without planning tend to use human interaction or overly large data sets with significant overlap between them. Given large data set sizes and long image acquisition times, reducing the number of views while providing

full coverage of the scene is a major goal. It is assumed that in these environments this reduction will correspond to either improved model quality or a reduction in total model acquisition time.

Previous solutions to the NBV problem differ in their representations of unexplored space, their method of utilizing this representation, or both. Connolly [5] used octrees to model the workspace and differentiated between sensed and occluded octree nodes. Planning was either by ray-casting the model from discrete sensor positions or by using a histogram of normals for unexplored octree node faces. Surface normal histograms were also used by Maver and Bajcsy [10] in a method that used polygonal surfaces to represent regions occluded from the sensor and, therefore, was not affected by the discretization issues associated with octrees. Whaitte and Ferrie [20] used an uncertainty metric for models composed of superellipsoids: Ray-casting operations on the model from discrete sensor positions were used to find the viewpoint that will maximize the reduction in uncertainty. Banta et al. [2] described an algorithm that determines a set of viewpoints from an occupancy grid model by examining regions that have a high degree of curvature. Again, a ray-casting procedure is used to choose the best candidate viewpoint. Pito [9] used a mesh surface model of imaged scene features on which the border elements have been extended a short distance in the viewing direction, in a sense modeling limited parts of the scene occlusion. In an inversion of the ray-casting process, these border elements were used to weight sensor positions, thus avoiding unnecessary ray-casting operations. Work by Kutulakos [8] obtained a hypothesis of how an object surface behaves at the extremal boundary of the model and then observed its deformation during a change in viewpoint. Sobh et al. [13] used coarse and fine sensors and domain knowledge about the scene to determine a hierarchical sensing plan.

Our previous work in sensor planning computed unoccluded sensor viewpoints for a set of known features given a model of a scene, a sensor model, and a set of sensing constraints [1], [17]. However, these methods are only effective for planning to acquire distinct features on known objects. In this paper, we apply these techniques to the problem of acquiring an unknown object. We are able to do this by completely modeling all scene occlusion as well as the sensed surfaces and determining occlusion-free viewpoints for portions of the occlusion boundary. For a survey of the sensor planning literature, see [16].

### 2 OVERVIEW OF MODEL ACQUISITION

We briefly review our model acquisition system which uses range images to build accurate 3D models of an object or scene [12]. The method is an incremental one in which modeling operations may be interleaved with the planning method that determines the next sensor position. For each range scan, a mesh surface is formed and “swept” in space to create a solid volume representing both the imaged object surfaces and the occluded volume (Fig. 1). This is done by applying an extrusion operator to each triangular mesh element, sweeping it along the vector of the rangefinder’s sensing axis, until it comes in contact with the boundary of the workspace. The result is a five-sided triangular prism. A regularized union operation is applied to the set of prisms, which produces a polyhedral solid consisting of three sets of surfaces: a mesh-like surface from the acquired range data, a number of lateral faces equal to the number of vertices on the boundary of the mesh derived from the sweeping operation, and a bounding surface that caps one end. In this method, the occlusion boundary is explicitly realized as a surface: This feature is important because it causes each model to be a closed, bounded set and allows application of the robust analysis and modification functions of solid modeling techniques. Each surface of this model is tagged as “imaged” or

• M.K. Reed is with Blue Sky Studios, 44 S. Broadway, White Plains, NY 10601. E-mail: reed@blueskystudios.com.

• P.K. Allen is with the Computer Science Department, Columbia University, 1214 Amsterdam Ave., New York, NY 10027. E-mail: allen@cs.columbia.edu.

Manuscript received 22 Apr. 1999; revised 12 July 2000; accepted 18 July 2000.

Recommended for acceptance by L.B. Wolff.

For information on obtaining reprints of this article, please send e-mail to: [tpami@computer.org](mailto:tpami@computer.org), and reference IEEECS Log Number 109671.

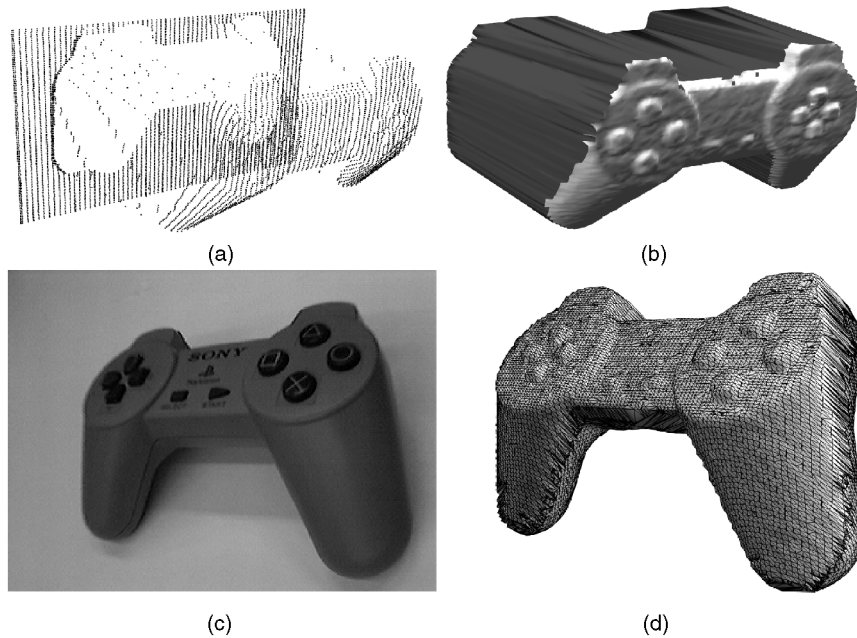


Fig. 1. (a) Data points from range scan of video game controller and (b) solid model created from scan showing imaged (light) and occlusion (dark) surfaces. (c) Image of actual controller and (d) reconstruction of controller from the model at top right and models from two additional viewpoints.

“occlusion,” depending on the following constraint: when the angle between the sensing direction and the surface normal is greater than the “breakdown angle” (the maximum inclination of the sensor with respect to a surface) for the sensor, that surface is labelled as an occlusion surface. It is these surfaces that drive the planning process.

Each successive sensing operation results in new information that must be merged with the current model being built, called the *composite model*, by performing a regularized set intersection operation between the two. The intersection operation propagates the surface-type tags from surfaces in the individual models through to the composite model. These tags are key to our algorithms for planning the next view, as they denote surfaces that are used as targets in the planning process. The system also has other important benefits. It places no topological constraints on the object or scene—there may be multiple objects, objects with holes, etc. The resulting models are guaranteed to be “water-tight” 3D solids at each step in the process and the method supports incremental improvement as new views are integrated into the model.

### 3 STRATEGIES FOR VIEWPOINT PLANNING

A “rough” model is acquired first, e.g., from four distinct views that can then be used to plan the NBV. Deciding the number of initial views depends on the scene and the task and it may be best to have user interaction specify this quantity. Once a partial model is acquired, the planner operates by considering the entire workspace as the potential location for the next sensor placement and then constraining this volume until a solution for the next viewpoint is found. The constraints are represented as volumes in continuous space which may be combined to form a plan via set operators. The object of the planning is to maximize the surface area of the occlusion boundary in the scene that is imaged in each sensing operation.

There are three constraints that the planner considers. **Sensor imaging constraints** are limitations on the imaging of a surface in the scene due to the sensor’s modality or implementation. For example, if the sensor must be within a certain angle of inclination with respect to the surface, this produces a constraint on the

representative volume. **Scene occlusion constraints** are limitations due to the fact that parts of the current composite model block some locations in space from viewing the target surface. Finally, **sensor placement constraints** limit the range of positions in which the sensor may be placed. Each of these constraints may be represented as a volume, called  $V_{\text{imaging}}$ ,  $V_{\text{occlusion}}$ , and  $V_{\text{placement}}$ , respectively.

An overview of the process is shown in Fig. 2. The left column shows the entire model acquisition process, which is an acquire-model-integrate-plan loop. The right column shows the steps in planning. The first step is to select one or more targets. For each target, the sensor imaging constraint is computed to yield  $V_{\text{imaging}}$ . For each target, potential self-occlusions are determined by testing scene model surfaces for intersection with any  $V_{\text{imaging}}$ . Scene occlusion constraints,  $V_{\text{occlusion}}$ , are then computed for each surface that may occlude a target. Next, **visibility volumes**,  $V_{\text{target}}$ , which are the set of sensor positions that may properly image the target, yet are not occluded by parts of the model, are found. These are intersected with the sensor placement constraints to find the final viable positions of the sensor. Usually, more than one target is being considered, so that there is the possibility of imaging multiple targets in one sensing operation. In this case, computation of sensor imaging constraints and sensor occlusion constraints is done for each target surface from a set of eligible surfaces. A running example is described next and depicted in Fig. 3, demonstrating the planning process for a model consisting of three surfaces: a single target and two occluding surfaces.

#### 3.1 Computing Sensor Imaging Constraints

$V_{\text{imaging}}$  describes the locations from which a sensor can effectively image one of the target surfaces. The factors that contribute to the imaging constraints are the modality of the sensor and the geometric parameters describing its ability to acquire images, such as breakdown angle  $\alpha$ , depth of field, standoff (which describes the closest the sensor may be to the target), and its range and resolution. Each of these parameters affect the shape of a volume representing this constraint. For common sensors, the volume  $V_{\text{imaging}}$  may be quickly generated from a polygonal target surface by performing an extrusion operator on the surface, in the direction of the surface normal, with a draft angle equal to the

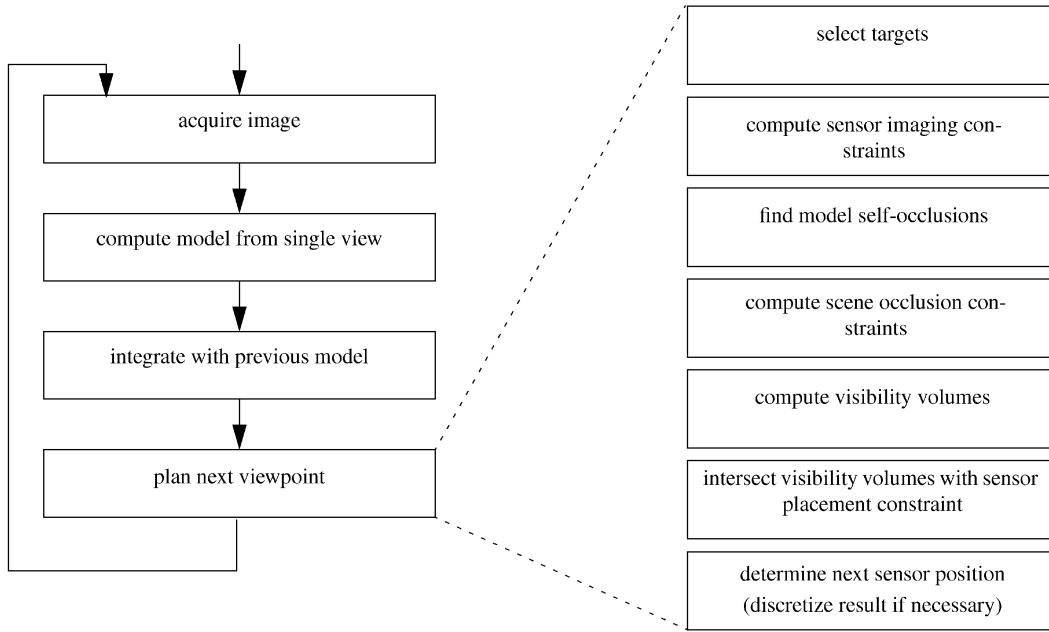


Fig. 2. Overview of the sensor planning process.

breakdown angle  $\alpha$ . Fig. 3a shows the three surfaces and  $V_{\text{imaging}}$  for the target surface assuming a sensor with a 45 degree breakdown angle.

### 3.2 Computing Occlusion Constraints

The occlusion constraints further restrict the positioning of the sensor for a specified target by disallowing all positions in space from which the target is occluded by any part of the environment. The surfaces in the current composite model are analyzed to see if they occlude the sensor from the target. For each surface  $i$  that potentially occludes the target, a volume  $O_i$  is constructed that represents the space that is disallowed for the sensor. To describe the entire space from which it is not possible to see the target, it suffices to compute:

$$V_{\text{occlusion}} = \bigcup_{\forall i \neq \text{target}} O_i \quad (1)$$

that is, the union of these volumes over all surfaces in the composite model comprises the space from which it is not possible to see the target and, hence, captures the occlusion constraints. In the running example shown in Fig. 3b,  $V_{\text{occlusion}}$  is shown for the two occluding model surfaces. Fig. 3c shows  $V_{\text{target}}$  for this scene, the computation of which is described below. The computation of  $O_i$  for a specific target and model surface  $i$  utilizes an algorithm earlier developed for polygonal feature visibility described in [17], [15]. It derives a geometric decomposition of space into volumes from which a specified model feature either can or cannot be fully imaged by an ideal sensor. If a target surface—that is, an “occlusion” surface in the model—is used as a feature, this algorithm may be used to produce valid viewpoints for any target-and-occluding-surface pair. The resulting occlusion volumes are then united (via a union operation) as shown above. The polygons need to be convex which may require a convex decomposition.

### 3.3 Computing Sensor Placement Constraints

Sensor placement constraints describe the physical locations in which the sensor may be placed and are typically derived from a description of the manipulator used to position the sensor. This may be a six-degree-of-freedom manipulator, in which case the sensor placement constraint may be represented by a sphere of

finite radius. Many systems built around laser rangefinders use a turntable to rotate the object and constrain the sensor to one degree of freedom, usually along a linear path, and this produces a cylindrically-shaped sensor placement constraint. We have previously included the resolution constraint for the case of a 2D camera in [14]. Fig. 3d shows  $V_{\text{placement}}$  for a model of a sensor attached to a cartesian manipulator.

### 3.4 Constraint Integration

The planning process constructs a *visibility volume*,

$$V_{\text{target}} = V_{\text{imaging}} - V_{\text{occlusion}}, \quad (2)$$

that describes the set of all sensor positions that have an unoccluded view of the target for a specified model as shown by the volume  $V_{\text{target}}$  in Fig. 3c. Once this visibility volume has been computed, it is only necessary to include the constraint represented by  $V_{\text{placement}}$  to determine the plan:

$$V_{\text{plan}} = V_{\text{target}} \cap V_{\text{placement}}. \quad (3)$$

This integration is shown in Fig. 3d, in which  $V_{\text{target}}$  is shown along with  $V_{\text{placement}}$ , which, in this case, models a sensor attached to a cartesian manipulator.

Each of these constraints is represented volumetrically by a set in three-dimensional space and a solution may be found by applying set operators. The final result is a set of points, lines, surfaces, or volumes that represent admissible viewpoints and may be the empty set if there is no solution. Fig. 3e shows the final  $V_{\text{plan}}$ , formed by the set intersection in (3). This plan represents the accessible, unoccluded positions from which the sensor can properly acquire the target surface.

## 4 EXAMPLE: CITY SCENE

We now show a planning example of the scene shown in Fig. 4, which is composed of multiple objects and has extremely high self-occlusion. These environments are typified by large structures that encompass a wide range of geometric shapes with significant occlusion. Our experimental setup, common in laser rangefinder systems because it maximizes the stability and repeatability of the rangefinder, allows the manipulator to move the sensor only in the

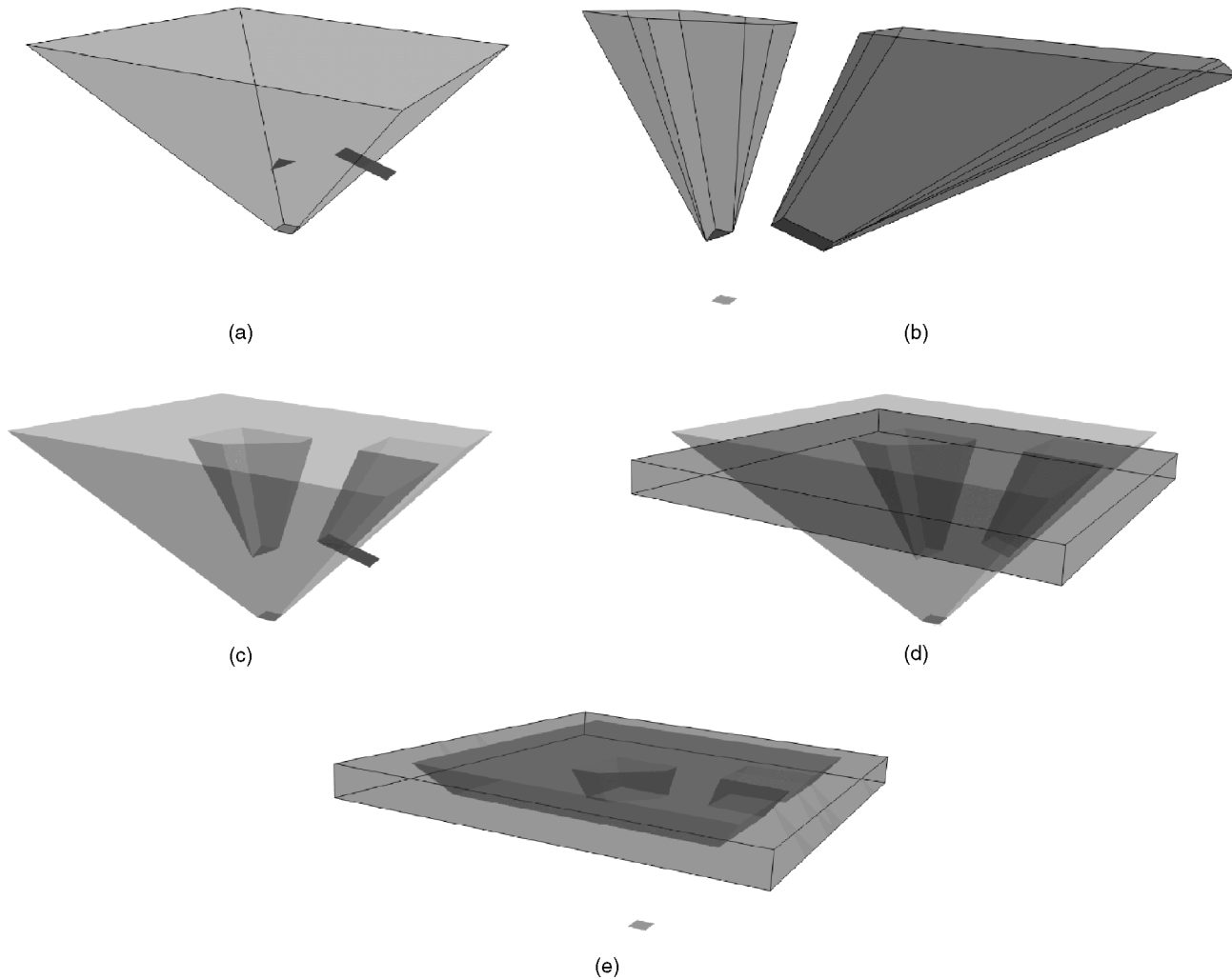


Fig. 3. Constraint integration (a)  $V_{\text{imaging}}$ : visibility volume for unoccluded target. (b)  $V_{\text{occlusion}}$ : occlusion volumes caused by two model surfaces. (c)  $V_{\text{target}}$ : visibility volume formed by  $V_{\text{imaging}} - V_{\text{occlusion}}$ . (d)  $V_{\text{placement}}$ : for a cartesian manipulator, shown with  $V_{\text{target}}$ . (e)  $V_{\text{plan}}$ : (in dark gray) shown with  $V_{\text{placement}}$ .

world  $z$  coordinate, while a turntable rotates the object around the  $z$  axis (cylindrical workspace). The modeling process was initiated by acquiring four range images 90 degrees apart to produce the composite model shown in Fig. 5. In this rendering, "occlusion" surfaces are shown in solid gray, while "imaged" surfaces are shown with their edges visible. Approximately 25 percent of the entire acquirable model surface is, at this point, composed of "occlusion" surface. After decimating these surfaces, the planner selects the 30 largest by area as targets: Thus, 30  $V_{\text{target}}$  volumes will be constructed. In Fig. 6,  $V_{\text{target}}$  is shown for each of these surfaces with the initial model at the center to allow the reader to observe the relative orientations.

These visibility volumes are intersected with  $V_{\text{placement}}$  to compute the sets of occlusion-free sensor positions for the targets, as shown at the left in Fig. 6. Each intersection, shown as a surface on the cylindrical  $V_{\text{placement}}$ , represents a set of occlusion-free sensor positions for a specific target: Overlapping regions represent visibility for multiple targets. In this example, a solution for the next sensor position is found by testing the continuous-space plans for intersection with the vertical path of the sensor at incremental angle  $\epsilon$  ( $\epsilon = 2$  degrees in this example). The results of this process are shown at the right of Fig. 6 in a "planning histogram," where the height of each gray bar represents the area of target surfaces visible from that sensor location. Thus, higher bars denote desirable sensor locations, lower ones less desirable.

The sensor position is found by selecting the peak in the planning histogram. After the next range image is acquired, modeled, and integrated, the planning process is restarted with the next model. In Fig. 7, the continuous and discrete plans are shown for the next seven views, resulting in a total of twelve images automatically acquired, modeled, and integrated. This final model is shown texture-mapped in Fig. 8. Note that the arches and pillars are correctly recovered despite the high occlusion.

## 5 ANALYSIS: MODEL CITY

Table 1 lists measurements of the model during the acquisition process, which can be used to assess the performance and accuracy of the system. These measurements include: **total volume** of the model ( $\text{cm}^3$ ), **total model surface area** ( $\text{cm}^2$ ), **occluded area** ( $\text{cm}^2$ )—the total area of all occlusion surfaces, **planned area** ( $\text{cm}^2$ )—the total surface area of the targets for which plans have been generated, and **percent target area planned**—the surface area of planned-for targets, as a percentage of the total occlusion surface area.

As shown in Table 1, the first four views were acquired without any planning (view zero is just the entire workspace). The total model volume decreases over time, as, indeed, it must for a system that uses set intersection for integration and has not duplicated any sensor viewpoints. The total surface area does not strictly decrease,

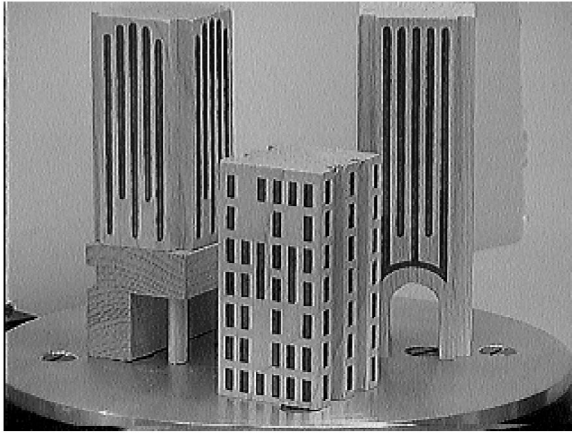


Fig. 4. The city scene consisting of three toy buildings. Note the archway under the rightmost building and one of three pillars visible on the leftmost building.

due to an increase in the scene complexity with the integration of new images. Of particular import is the data in the final column. Because the plans are computed using a fixed number of surfaces at each iteration, it is interesting to see what percentage of the total available target area is being planned for. Clearly, if every target surface were considered, this would be 100 percent each time. Even though only 30 of the largest targets by area are planned for, the percent of the planned area never drops below 10 percent of the total area and in most cases is over 20 percent. This shows that the considerable computational cost saved by selecting a subset of the targets to plan for is a viable strategy. The actual volume of the city scene has been estimated as  $362 \text{ cm}^3$ , whereas the calculated model volume is  $370 \text{ cm}^3$ .

The peaks in the planning histogram are very steep. In some cases, a difference of only a few degrees in sensor placement causes the viewpoint to be an occluded view and prevents it from acquiring new information, reflecting the fact that the scene has high occlusion and viewpoint planning becomes more important. There is at least one feature that is only visible for a nine degree region of the sensor placement constraint. Thus, at least 41 images at equal turntable rotations are needed to guarantee acquisition of that feature if no view planning is performed, as opposed to 12 images needed with sensor planning. Note, in Fig. 8, the recovery of the occluded areas behind the columns and arches which have been recovered correctly with view planning.

The planning component has not yet been optimized and the time to plan a new sensor viewpoint is approximately 120 minutes for each of the planned city scenes. This is a factor of 20 greater than the time required for an unplanned view, which is approximately six minutes. Although the minimum of 41 unplanned images necessary to acquire the scene would have been faster in this situation, it is easy to imagine a situation where this would not be the case: The break-even point is 29 minutes for planning, at which point the two methods take the same amount of time. However, because the viewpoint planning greatly reduces the number of views (from 41 to 12), it also reduces the amount of data storage necessary.

## 6 OTHER CONSIDERATIONS

Planning can be one of the more computationally expensive aspects of the model acquisition process. It is unclear whether the planning algorithm, as currently implemented, is as fast as a brute force ray-casting method. However, in the long run this method will be more beneficial than ray-casting, particularly since it works in continuous instead of discrete space. The cost of our method may be drastically reduced if the computation of the constraints is done in serial fashion and if information from one constraint is used to reduce the

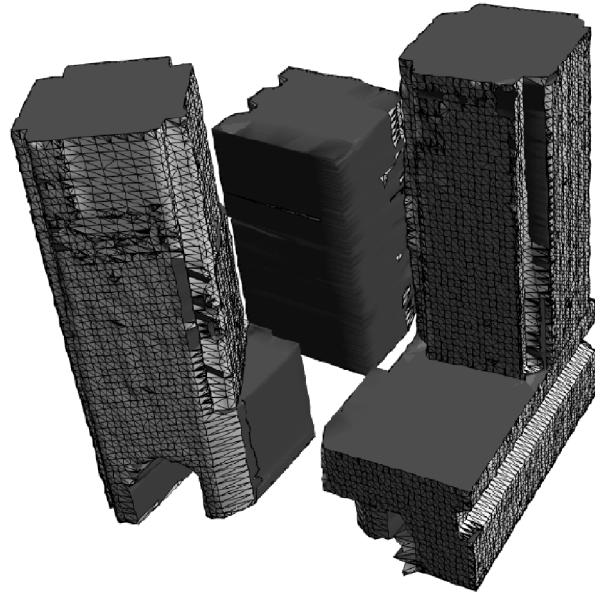


Fig. 5. Composite model of city scene from four sensing operations.

amount of calculation in the more computationally intensive calculations. To illustrate this, consider the current costs of calculating the constraint volumes for a target surface  $f$ :

- Sensor imaging constraint  $V_{\text{imaging}} : O(m)$ , where  $m$  = the number of edges of  $f$ .
- Occlusion constraint  $V_{\text{occlusion}} : O(n^2)$ , where  $n$  = the number of occluding surfaces. This calculation is dominated by the cost of unioning the  $O_i$  volumes.
- Sensor placement constraint  $V_{\text{placement}} : O(1)$ , as this is independent of the surfaces in the model and may be computed off-line.

It is clear that for any real-world situation the most computationally expensive computation is that of the occlusion constraint. However, it can be seen that only those surfaces that intersect the volume described by the sensor imaging constraint need to be considered<sup>1</sup> because no model surfaces outside of that volume may come between the sensor and the target surface. The sensor imaging constraint can be calculated very rapidly, because the vast majority of surfaces in our model are triangular, the rest being simple polygons with a small number of edges. Thus, if  $V_{\text{imaging}}$  is calculated first and, then used to determine the candidate model surfaces for the occlusion constraint, a considerable amount of calculation is avoided. In particular, consider that many of the target surfaces have no model surfaces that might possibly block the sensor from them and so the occlusion calculation is avoided entirely. Candidate model surfaces may be evaluated for possible occlusion simply by testing if they intersect  $V_{\text{imaging}}$ . This ordering of the application of constraints in no way affects the outcome of the planning process.

However, in the interest of further reducing the amount of time spent in the planning process, there are two other optimizations that do reduce the accuracy of the computed plan: decimating the surface of the composite model and discretizing the space representing the sensor placement constraint. These methods reduce the accuracy because they use approximations to either the model or the continuous-space plan and, hence, are not exact solutions.

1. Standoff should not be included here because it may prevent the consideration of occluding surfaces that are closer than the standoff distance.

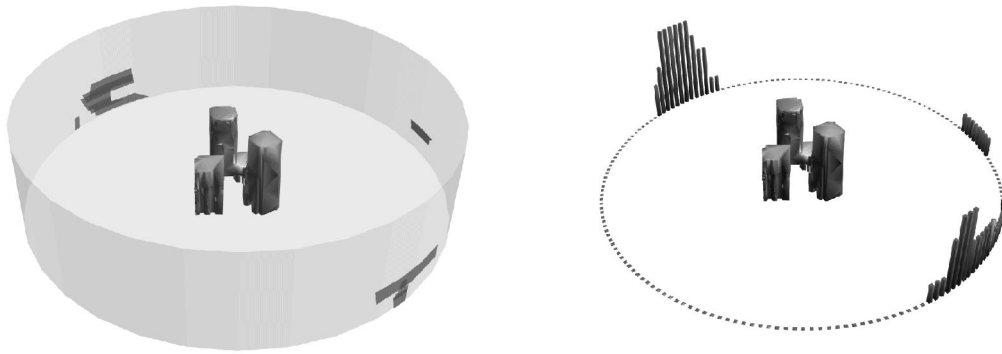


Fig. 6. Continuous-space plans for targets (left) and discretization (right) to area-weighted sensor positions.

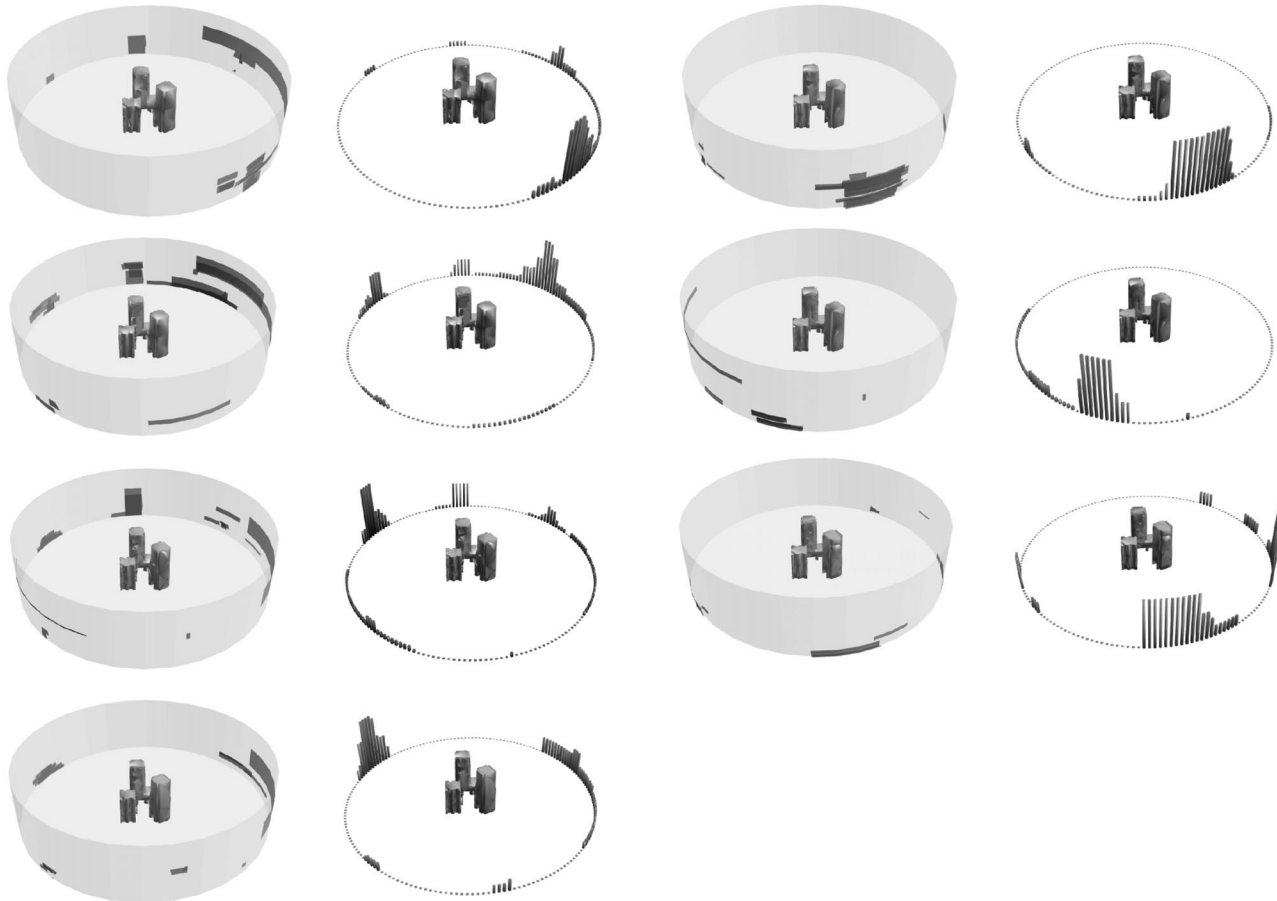


Fig. 7. Plan generation for views 6-9 (left two columns) and 10-12 (right two columns). Each plan is shown as  $V_{plan}$  in continuous space (to left) and planning histogram

Sensor space discretization has been shown previously in the examples. The decimation of the composite model is performed by separating the occlusion boundary into targets consisting of coplanar, convex surfaces via a variant of the Simplification Envelopes (SE) algorithm [4]. SE is a method that generalizes offset surfaces to determine an interior and exterior boundary between which the resulting decimated model must lie and so has the desirable property of having an absolute bound for the distance between the original surface and the simplified one. The distance from these boundaries to the original surface is given as a single real number input to the algorithm. However, in its original form SE produces models whose representative sets may be subsets of the original model and, therefore, may miss some occlusion situations. To prohibit this effect, in applying the SE method it is necessary to allow only the use of an exterior offset surface, so that the resulting simplified model is always a superset of the model from which it is

derived. In addition to this, we have modified the algorithm so that it retains the surface-type tags which describe the surface elements as imaged or occluded and disallows any merging process between them. This is done to ensure that the resulting surfaces are composed of only one type of modeled surface: It is not clear what the resulting surface describes if it is due to the merging of an imaged and an occluded surface. The computational cost of the decimation is  $O(n)$  and, therefore, is fast enough to provide a substantial benefit without a quantifiable loss of fidelity.

A primary consideration for a system that plans dynamically is that of determining when planning is no longer needed and the modeling process is complete. While there are many ways to compute termination criteria, in essence it is a function of the application and its needs. A benefit of our method is that at each step the number and area of occluded and imaged surfaces is known due to the tagging process and this can be used as the

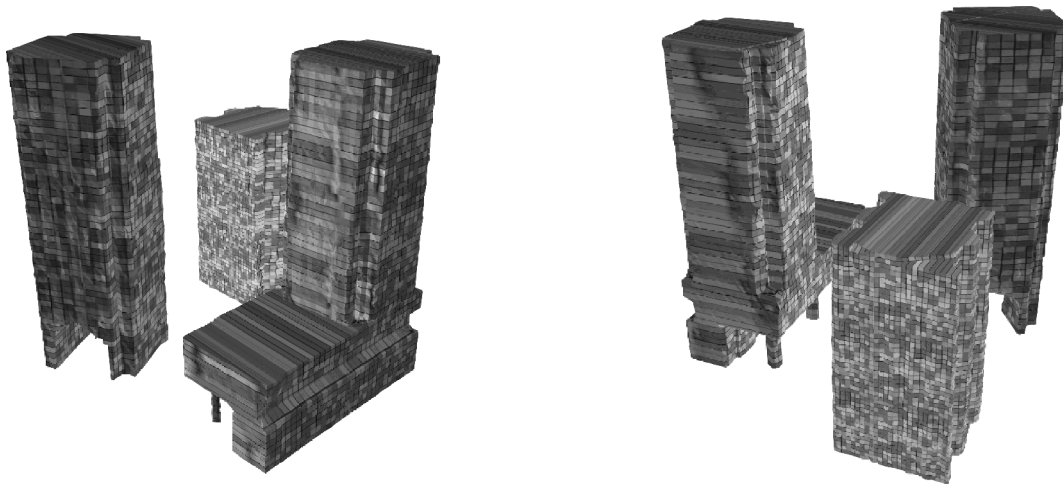


Fig. 8. Final texture-mapped city model.

TABLE 1  
Model Measurements During Acquisition

View #	Volume	Surface Area	Occluded Area	Planned Area	% Target Area Planned
0	4712	1571	1571	–	–
1	1840	1317	942	–	–
2	1052	1151	590	–	–
3	506	733	200	–	–
4	432	658	140	–	–
5	416	656	121	61	50%
6	404	659	104	28	27%
7	391	657	90	12	13%
8	386	647	84	8	10%
9	382	644	75	15	20%
10	380	651	62	7	11%
11	374	622	53	16	30%
12	370	604	36	9	25%

termination criterion. This provides a measure that can be used to quickly create models at reduced resolution or build a more detailed and accurate model, depending on user needs.

When acquiring an image that contains a particular surface, much of the surrounding surfaces are also acquired. Therefore, it makes sense to consider as targets, those parts of the model with a high density of occluded surface. “Occluded” model surface area is not strictly related to actual unimaged object surface area, as in the case of a long cylinder, where the scanner cannot image much of the interior walls, yet is able to acquire the hole and properly model the object’s topology. Because “occluded” surfaces usually lie close to the boundaries of true surfaces, or at worst delimit unexplored volume in the workspace, using them to guide exploration is a sound strategy.

For the modeling process, highly reflective surfaces in the scene can be problematic. In the current implementation, we assume these regions are relatively small in the image and that their artifacts may be removed in image preprocessing. Another limitation of this system is that it considers only total visibility for each target. It is possible that some solutions are missed where partial visibility would allow a better result. “Total visibility” in this context means that every position in the plan can *completely* image the target surface. For example, when planning for two targets, their visibility volumes may be disjoint, meaning that there are no positions that totally image both targets. However, there may be positions that partially image both targets and whose total imaged area is greater than either target individually. A similar

limitation is that, as currently implemented, the system requires full images of the scenes. This is purely an implementation issue of the modeling component of the system: The planning process does not require such images and would operate similarly if the modeling system were modified to use an alternative method to set intersection for integration.

## 7 CONCLUSION

The automated model acquisition method presented in this paper consists of two components that operate in an interleaved fashion: An incremental modeler and a sensor planner that analyzes the resulting model and computes the next sensor position. This planning component utilizes a partial model to determine volumes of visibility for contiguous areas of unexplored scene. The visibility volumes are combined with sensor placement constraints to compute sets of occlusion-free sensor positions that are guaranteed to improve the quality of the model. These sets may be intersected to determine a single best region for the next sensor position, or discretized if a continuous solution is not necessary. We are currently equipping a mobile robot base with sensors (both range and photometric) to automatically acquire models of real buildings using this system. The system will acquire a partial model from a small number of viewpoints which will be used both to plan the next viewpoint and to navigate the mobile sensor base to this position.

## REFERENCES

- [1] S. Abrams, P.K. Allen, and K.A. Tarabanis, "Computing Camera Viewpoints in an Active Robot Workcell," *Int'l J. Robotics Research*, vol. 18, no. 3, pp. 267-285, Mar. 1999.
- [2] J.E. Banta, Y. Zhen, X.Z. Wang, G. Zhang, M.T. Smith, and M.A. Abidi, "A "Best-Next-View" Algorithm for Three-Dimensional Scene Reconstruction Using Range Cameras," *Proc. Intelligent Systems and Advanced Manufacturing Symp.*, 1995.
- [3] Y. Chen and G. Medioni, "Object Modeling by Registration of Multiple Range Images," *Proc. IEEE Int'l Conf Robotics and Automation*, pp. 2,724-2,729, Apr. 1991.
- [4] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Webber, P. Agarwal, F. Brooks, and W. Wright, "Simplification Envelopes," *Proc. SIGGRAPH*, pp. 119-128, 1996.
- [5] C. Connolly, "The Determination of Next Best Views," *Proc. IEEE Int'l Conf. Robotics and Automation*, pp. 432-435, 1985.
- [6] B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images," *Proc. SIGGRAPH*, pp. 303-312, 1996.
- [7] A. Hoover, D. Goldgof, and K. Bowyer, "Building a B-Rep from a Segmented Range Image," *Proc. Second CAD-Based Vision Workshop*, pp. 74-81, Feb. 1994.
- [8] K.N. Kutulakos, "Exploring Three-Dimensional Objects by Controlling the Point of Observation," PhD thesis, Computer Sciences Dept., Univ. of Wisconsin, 1994.
- [9] R. Pito and R. Bajcsy, "A Solution to the Next Best View Problem for Automated CAD Model Acquisition of Free-form Objects using Range Cameras," *Proc. SPIE Symp. Intelligent Systems and Advanced Manufacturing*, 1995.
- [10] J. Maver and R. Bajcsy, "How to Decide from the First View Where to Look Next," *Proc. DARPA Image Understanding Workshop*, pp. 482-496, 1990.
- [11] M. Reed, P.K. Allen, and I. Stamos, "Automated Model Acquisition from Range Images with View Planning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997.
- [12] M. Reed and P.K. Allen, "3D Modeling from Range Imagery," *Image and Vision Computing*, vol. 17, no. 2, pp. 99-111, Feb. 1999.
- [13] T.M. Sobh, J. Owen, C. Jaynes, M. Dekhil, and T.C. Henderson, "Industrial Inspection and Reverse Engineering," *Computer Vision and Image Understanding*, vol. 61, no. 3, pp. 468-474, May 1995.
- [14] I. Stamos and P.K. Allen, "Interactive Sensor Planning," *Conf. Computer Vision and Pattern Recognition*, pp. 489-495, June 1998.
- [15] K. Tarabanis, R.Y. Tsai, and P.K. Allen, "Satisfying the Resolution Constraint in the MVP Machine Vision Planning System," *Proc. 13th IASTED Int'l Symp. Robotics and Manufacturing, (also in Proc. DARPA Image Understanding Workshop)*, 1990.
- [16] K. Tarabanis, P. Allen, and R. Tsai, "Sensor Planning in Computer Vision," *IEEE Trans. Robotics and Automation*, vol. 11, no. 1, pp. 86-105, Feb. 1995.
- [17] K. Tarabanis, R.Y. Tsai, and P.K. Allen, "The MVP Sensor Planning System for Robotic Vision Tasks," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 1, pp. 72-85, Feb. 1995.
- [18] G. Turk and M. Levoy, "Zippered Polygon Meshes from Range Images," *Proc. SIGGRAPH*, pp. 311-318, 1994.
- [19] P. Whaite and F. Ferrie, "Autonomous Exploration: Driven by Uncertainty," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, Mar. 1997.
- [20] P. Whaite and F. Ferrie, "Uncertain Views," *Proc. IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition*, pp. 3-9, 1992.
- [21] M. Wheeler, Y. Sato, and K. Ikeuchi, "Consensus Surfaces for Modeling 3D Objects from Multiple Range Images," *Proc. Sixth Int'l Conf. Computer Vision*, 1998.