3-D Modeling from Range Imagery: An Incremental Method with a Planning Component

Michael K. Reed, Peter K. Allen, and Ioannis Stamos
Computer Science Department, Columbia University, New York, NY 10027

Abstract

In this paper we present a method for automatically constructing a CAD model of an unknown object from range images. The method is an incremental one that interleaves a sensing operation that acquires and merges information into the model with a planning phase to determine the next sensor position or "view". This is accomplished by integrating a system for 3-D model acquisition with a sensor planner. The model acquisition system provides facilities for range image acquisition, solid model construction and model merging: both mesh surface and solid representations are used to build a model of the range data from each view, which is then merged with the model built from previous sensing operations. The planning system utilizes the resulting incomplete model to plan the next sensing operation by finding a sensor viewpoint that will improve the fidelity of the model. Experimental results are presented for a complex part that includes polygonal faces, curved surfaces, and large self-occlusions.

1. Introduction

Automatically constructing 3-D computer models of an object or a scene from range images has recently received increased attention due to the availability of inexpensive, accurate rangefinders and to improvements in reconstruction algorithms. Termed *modeling from observation* [8], this task's inherent difficulty is due to the large scope of the shapes of 3-D objects and the resource-intensive data sets that are acquired. Typically, systems that perform this task model a range image using a surface, volumetric, or parametric model. Because the information from a single range image will not completely describe an object, range images from other viewpoints must be acquired, modeled, and integrated with previously acquired information. The task therefore includes acquisition, modeling and planning components, which make it necessary to address integration issues. Most importantly the modeling process must support incremental integration of new range data, be able to recognize model surfaces that need additional sensing, and must not put restrictions on the topological type of the object to be acquired. The capability of incremental integration in particular is an important one because it allows the reconstruction to progress with each newly acquired range image and therefore permits the use of a sensor planner to determine the next sensing orientation. Other desirable properties are that the system be robust with respect to errors in the range images and that the final model does not have "holes" in its surface, i.e. it is 2-manifold [5].

This paper describes a system that incrementally builds solid models from multiple range images and that exhibits the above-mentioned capabilities. The algorithm consists of two phases that are interleaved during the acquisition process. The first phase acquires a range image, models it as a solid, and merges the solid with any previously acquired information. This phase motivates the generation of a topologically-correct 3-D solid model at each stage of the modeling process, which allows the use of well-defined geometric algorithms to perform the merging task and additionally supports the view planning process. The second phase plans the next sensor orientation so that each additional sensing operation recovers object surface that has not yet been modeled. Using this planning component makes it possible to reduce the number of sensing operations to recover a model: systems without planning typically utilize as many as 70 range images, with significant overlap between them. This concept of reducing the number of scans is important for tasks such as 3-D FAX where the sensing process may add considerable time. In addition, the algorithm presented here avoids the problems associated with discretizing sensor positions by determining sensor visibility for a specific target, and is able to handle object self-occlusion properly. The result is a 3-D CAD model of the object.

2. Background

Recent research on the acquisition, modeling and merging process includes Thompson et al.'s REFAB system, which allows a user to specify approximate locations of machining features on a range image of a part; the system then produces a best fit to the data using previously-identified features and domain-specific knowledge as constraints [17]. The IVIS system of Tarbox and Gottshlich uses an octree to represent the "seen" and "unseen" parts of each of a set of range images and set-theoretic operators to merge the octrees into a final model [16]. Methods that use a mesh surface to model and

0-8186-7943-3/97 $10.00 © 1997 IEEE

integrate each of a set of range images, such as work by Turk and Levoy [21] or by Rutishauser et al. [14], or to model a complete point sampling as by Hoppe [7] or Fua [6] have also proven useful in this task. Both Connolly and Stenstrom [4] and Martin and Aggarwal [11] perform edge detection and projection from intensity images, a concept that is revisited by Laurentini in [10]. Curless and Levoy [5] present a system that uses a mesh in a ray-casting operation to weight voxels in an octree, which is then used as input to an isosurface extraction algorithm. This method has achieved excellent results at a cost of numerous (50 to 70) overlapping sensing operations. In contrast, our method utilizes a planner with the goal of reducing the number of imaging and integration operations.

The planning process presented in this paper operates by reasoning about occlusion, which has been strongly associated with viewpoint planning in the research literature for some time. Kutulakos [9] utilizes changes in the boundary between sensed surface and occlusion with respect to sensor position to recover shape. In Connolly's octree-based work [3], "unseen" space is explicitly represented and used to plan the next view either by ray-casting or by analyzing a histogram of the normals of surfaces of "unseen" space. A similar histogram-based technique is used by Maver and Bajcsy [12] to find the viewing vector that will illuminate the most edge features derived from occluded regions. More closely resembling the work presented in this paper is that of Whaite and Ferrie [23], which uses a sensor model to evaluate the efficacy of the imaging process over a set of discrete orientations by ray-casting: the sensor orientation that would hypothetically best improve the model is selected for the next view. Recent work by Pito [13] removes the need to ray-cast from every possible sensor location by determining a subset of positions that would improve the current model.

3. Model acquisition and merging

The first phase of this system acquires and models range data, and integrates the resulting model into a *composite model* that represents all known information about the object or scene. This is done by representing the data with a mesh surface, which is then extruded in the imaging direction to form a solid. Each model created by our method includes information about the space occluded from the sensor, an important difference from systems that only model sensed surfaces. This *occlusion volume* is a key component of our sensor planning process because it allows the system to reason about what has not been properly sensed. In this section we discuss how a range image from a specific viewpoint is modeled, and how this model is merged into the composite model in an incremental fashion that allows new information to be integrated as it is acquired.

3.1 Acquiring and representing a range image

The acquisition of range data is performed by a robotic system comprised of a Servo-Robot laser rangefinder attached to an IBM SCARA robot, with the object to be imaged being placed on a motorized rotation stage (see Figure 1). The rangefinder acquires a single scan line of data at a time in a plane perpendicular to the robot's z axis. After each scan line has been acquired, the robot steps the rangefinder a small distance in along its z axis. The result of the scanning process is a rectangular range image of the object from a particular viewpoint, the direction of which is controlled by rotating the turntable. A narrow filter is applied to the range image to remove spike noise, after which the point data are used as vertices in a mesh.

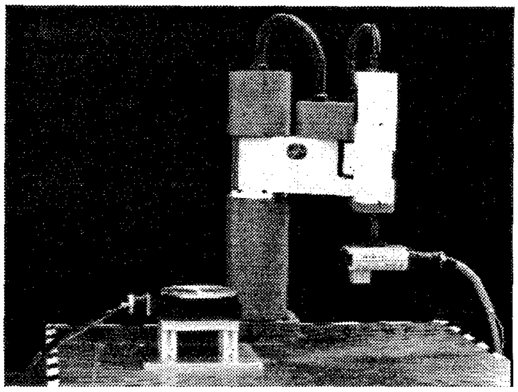

FIGURE 1. Experimental setup showing robot with attached laser rangefinder (to right) and turntable (to left).

A mesh is a piecewise linear surface composed of elements that meet along their edges, which in turn meet at vertices. Meshes are frequently chosen to represent a sampled surface due to their efficiency, their representational flexibility, and the simplicity of mesh algorithms. They find particular application in range imaging where objects are highly oversampled during the sensing process. Mesh surfaces built from these range images may then be efficiently processed to reduce their size [2], fit with more complex surface types [7], or registered to each other [14]. However, since the mesh determined by a single range image is in essence a surface model, it does not contain information that permits spatial addressability (the ability to classify points as inside, on, or outside the model) which is necessary for many tasks and is inherent in solid models. Although a mesh that completely covers an object may be used to determine a solid model, in most incremental modeling techniques the mesh can not be closed until the end of the scanning process. This precludes the use of a planning method or any other procedure that requires a solid model.

A solution to this problem is to build a solid model from each scanning operation that incorporates both the information about the model's sensed surfaces and the occlusion information in the form of the occlusion volume. When building the mesh that will be used to represent a surface from a range image, it is necessary to determine what the mesh connectivity will be. In this regard our work differs from other mesh-based methods such as mesh zippering [21] and other similar re-meshing techniques [14] which retain only elements that lie directly on an imaged surface by removing elements that contain *occlusion edges*. These edges are discernible in the mesh because their lengths exceed some threshold (Figure 2). Our system

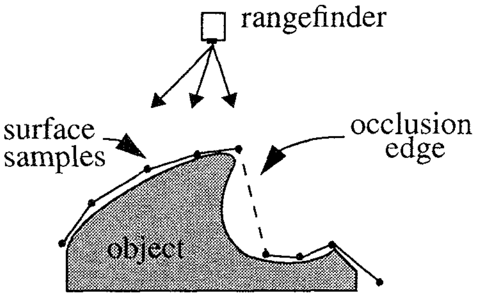

FIGURE 2. Example of edges between sampled vertices on a surface.

retains these elements, since they denote parts of the surface that are occluded from the sensor and need further imaging, and therefore are useful in the planning process. In addition, retaining these elements prevents holes from appearing in the model after the surface is extruded.

As an example of this process, consider the hypothetical object shown at the top of Figure 3. A range image is sampled from the CAD model using the shown sensing direction. The surface model shown in the middle of Figure 3 is typical of mesh-based methods; no occlusion edges are represented, and although it is possible to attach a low "confidence" values to the edges of the two surfaces it is not possible to determine occupancy information in the space between them. In contrast, the mesh shown at the bottom of Figure 3 represents both the imaged surfaces of the object and the occluded regions between the imaged surfaces.

3.2 Sweeping the mesh into a solid

This mesh surface is "swept" to form a solid model S of both the imaged object surfaces and the occluded volume. The algorithm may be stated concisely as:

$$S = \bigcup_{\forall i} extrude(M_i)$$

An extrusion operator is applied to each triangular mesh element M_i, orthographically along the vector of the rangefinder's sensing axis, until it comes in contact with a far bounding plane. The result is the 5-sided solid of a triangular prism (Figure 4). A 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

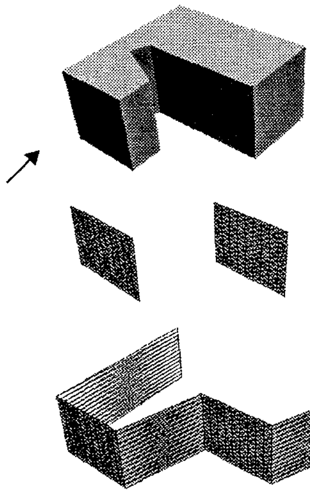

FIGURE 3. Top: Rendering of CAD model of a typical 2-1/2 D part, shown with a sensing direction. Middle: Surface mesh from synthetic range data of CAD part. This mesh does not include any elements that contain occlusion edges. Bottom: Surface mesh generated from synthetic range data, including elements composed of occlusion edges.

derived from the sweeping operation, and a bounding surface that caps one end.

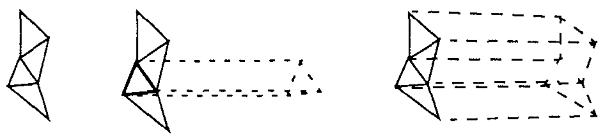

FIGURE 4. Example of a mesh sweep operation. (left to right) Mesh surface, mesh surface with one element swept, and mesh surface with all elements swept and unioned. The sensing direction is from the left.

Due to the large number (> 20k) and small size of the mesh elements involved, this algorithm has some implementation issues that merit discussion. Most importantly, the triangular elements to be swept and unioned are selected using the mesh's connected components and scan-line ordering. This is important because in practice union operations are faster and more robust when there is only one surface of intersection between the two unioned solids: selecting mesh elements at random results in many unnecessary multiple-surface intersections as the solids created from nearby mesh elements are "joined" by the solid from a mesh element between them. We maintain a list of those elements *not yet swept* from which we choose an element that is adjacent to one already swept and unioned, or at random if there are none that satisfy that condition. In addition, because the number of surfaces on an object directly affects the amount of time spent in the union operation, we perform the union operation in a tree-like fashion. The leaves of the tree are the solids derived from each mesh element, the first level is composed of the nodes representing the union of a few hundred solids each, the second level represents the union of tens of solids from the first level, up to the root which is the complete solid.

It is important to be able to differentiate between these surfaces during later model analysis and sensor planning. To do this we attach tags to each surface in the model based on which of the above sets the surface belongs to. All surface elements in the model that were present in the mesh before sweeping and that are composed of edges shorter than a threshold distance should be tagged as "imaged surface". These elements describe surfaces of the object that were imaged properly and do not need to be imaged again. All the remaining surfaces should be tagged as "occluded surface" so that they may be used to drive a later planning process. It should be noted that this tagging procedure must be done to a model from a single sensor position: large faces often get split into smaller ones during the merging process, and will not be differentiable by their edge lengths alone. After the tagging process the solid may be merged with models from other sensor positions, or it may first be used as input to a mesh optimization routine to reduce the number of elements.

As an example of the sweeping process, consider again the hypothetical part shown at the top of Figure 3. Sweeping its mesh (shown at the bottom of Figure 3) results in the solid shown in Figure 5, its surfaces tagged according to the process described above.

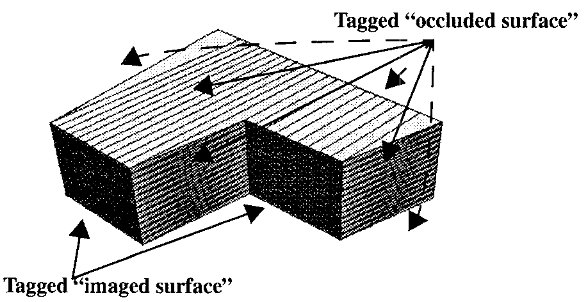

FIGURE 5. **Solid formed by sweeping the mesh shown at bottom of Figure 3 in the sensing direction. Tags for hidden surfaces are shown with dotted arcs.**

3.3 Merging single-view models

Each successive sensing operation will result in new information that must be merged with the current model being built, called the *composite model*. Merging of mesh-based surface models has been done using clipping and re-triangulation methods [21] [14]. These methods are necessary because these meshes are not closed, so specialized techniques to operate on non-manifold surfaces of approximately continuous vertex density are needed. In our method we generate a solid from each viewpoint which allows us to use a merging method based on set intersection. Many CAD systems include highly robust algorithms for set operations on solids, and our algorithm takes advantage of this. This is of critical importance in this application for the following reasons: the high density of the range images (and therefore the small size of many of the mesh elements), the many long and thin lateral surfaces, and most importantly the fact that many of these models will have overlapping surfaces that are extremely close to each other. Finally, because the single-view and merged models should be 2-manifold, it is necessary to use set operations that are able to handle regularized intersection.

The merging process itself starts by initializing the composite model to be the entire bounded space of our modeling system. The information determined by a newly acquired model from a single viewpoint is incorporated into the composite model by performing a regularized set intersection operation between the two. The intersection operation must be able to correctly propagate the surface-type tags from surfaces in the models through to the composite model. Because a surface element on the boundary of the result of a volumetric intersection will be present on the boundary of either one or both of the intersected volumes, there are two cases to consider. In the case that the surface on the boundary of the result is found in only one of the two intersected volumes, the surface-type tag may be directly copied from the original volume to which the surface belonged. In the case where the two volumes have overlapping surfaces, we employ a heuristic to decide what the tag for the surface on the result volume will be: if the tags for the two overlapping surfaces are the same, then that tag is copied to the result surface. If they are different then the tag 'imaged surface' is given priority, since it must be true that the surface was imaged in one of the two volumes.

4. The planning process

As described above, occlusion is an important scene attribute useful to the planning process. More specifically, the occlusion volume has previously been used in one of two ways, both of which assume that occlusions are explicitly represented in the model. In the first, ray casting is applied to the model to find how much occluded volume will be imaged for every sensor position: the sensor position that images the most occlusions is selected [3] [23]. This requires tessellating a viewing sphere to discretize the sensing positions and computing a ray-cast image from each of them, with the disadvantage of high computational cost and the fact that some solutions will be missed. The second method collects a histogram of normals of the surfaces that comprise the occlusions, scaled by surface area [3] [12]. The peak in the histogram denotes the normal of the most area of occluded surface, and an anti-parallel vector is then selected for the sensing direction. This technique is not sufficient because it does not take into account known self-occlusion of the model's surfaces, and therefore may result in a sensor position that acquires no new information. What is desired is a method that takes known self-occlusions into account, and yet does not need to discretize the sensing positions and compute an image for each of them. In the experiments that follow we show that by selecting a specific *target* to be imaged, and from this target and the associated model planning the appropriate sensing direction, that the above problems are avoided.

The planning component presented here is based on previous work on the sensor planning problem in our laboratory [1][19]. The sensor planning problem is that of computing a set of sensor locations for viewing a target given a model of an object or scene, a sensor model, and a set of sensing constraints [20]. The planner used in this work is able to reason about occlusion to compute valid, occlusion-free viewpoints given a specific surface on the model. Once an unoccluded sensor position for the specified surface has been determined, it may then be sensed, modeled, and integrated with the composite model. Thus, the method presented here is target-driven and performed in continuous space. As the incremental modeling process proceeds, regions that require additional sensing can be guaranteed of having an occlusion free view from the sensor if one exists. Other viewing constraints may also be included in the sensor planning such as sensor field of view, resolution, and standoff distance, as will be shown below.

The planning process relies on the construction of a *visibility volume* V_{target} for the target that specifies the set of all sensor positions that have an unoccluded view of the target for a specified model. This can computed by determining $V_{unoccluded}$, the visibility volume for the case where there are no occlusions, and subtracting O_i, the volume containing the set of sensor positions occluded from the target by model surface i, for each surface of the model:

$$V_{\text{target}} = V_{unoccluded} - \bigcup_{i \neq t} O_i$$

The volume described by $V_{unoccluded}$ is a half-space whose defining plane is coincident with the target's face, with the half-space's interior being in the direction of the target's surface normal. O_i, the volume of the set of sensor positions that model surface i occludes from the target, is similar in concept and construction to the occlusion volume discussed earlier, and is generated via a geometric algorithm based on space decomposition that determines the space that the element blocks from viewing the entire target [18]. We illustrate by computing V_{target} for a 2-D target in Figure 6, in which we have also incorporated a resolution constraint so that the sensor must be within a fixed distance from the target's center, and thus in 2-D $V_{unoccluded}$ is a half-circle. Once the visibility volume is computed, viewing parameters that are specific to the real sensor are included to further constrain the visibility volume. Finally, a transform is applied to bring the sensor into the visibility volume for the target, and the model acquisition process repeats.

5. Experimental results

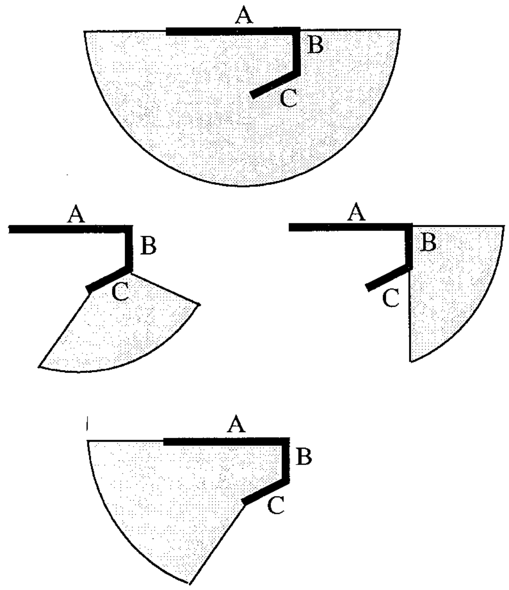

FIGURE 6. Planning for model consisting of three surfaces A (the target), B, and C. Top: the model with $V_{unoccluded}$ for A shown in grey. Middle: the occlusion due to O_C (left) and O_B (right). Bottom: the final visibility volume V determined by $V_{unoccluded} - (O_A \cup O_B)$, i.e. with occlusions taken into account: a point in the grey area has an unobstructed view of the target A.

The capabilities of this system are demonstrated by building a CAD model from distinct views of the object shown in Figure 7, which is a strut-like part. The planning for the sensor orientation is done by the algorithm above during the acquisition process, with the goal of determining a small number of views that will accurately reconstruct the object. This part has both curved and polygonal surfaces, and includes holes that are very difficult to image. The first two images are automatically acquired with a turntable rotation of 90 degrees between them. These two range images, the models acquired from them, and their respective composite model are shown in Figure 8 and Figure 9. The current composite model is shown in the third column of these rows; the shape of the part is already quite evident in the composite model of the second row. A target is designated on this composite model by user interaction from one of the surfaces tagged "occluded surface", and the planning algorithm constructs the plan shown in Figure 10. This plan is executed by rotating the turntable to place the sensor within the grey visibility volume, in this case and additional 83 degrees, which produces the image and model shown in Figure 11. Again, a target is designated and a plan produced, which is shown in Figure 12. The turntable is rotated 134 degrees to move the sensor into this visibility volume, and the image and model from Figure 13 results. This final model is shown rendered and as a mesh in Figure 14. As can be seen, there are "boundaries" where the intersection of the solids from two overlapping sensing operations causes an increase in the density of mesh elements. At this level of resolution the model would be a prime candidate for a decimation algorithm such as the one presented in [2]. We now have a very reasonable 3-D solid in a CAD format that may be used for rapid prototyping, robotics tasks or analysis. Refinement of this part may be accomplished using standard CAD primitives. For example, the holes on the sides of the part which were not completely imaged could be introduced by using a through-hole operator present in most CAD packages. Because this is an incremental method, additional scans may be also taken to improve the quality of the model.

6. Conclusions and future work

We have described a system that builds a 3-D CAD model of an unknown object incrementally from multiple range images. The method relies on the specification of targets on the incomplete model of the object to drive the planning process. This permits a static sensor planner to be used to compute occlusion-free viewpoints of the target and thereby allow each sensing operation to improve the accuracy of the model. The advantages of this technique are that, unlike prior methods, it both avoids discretization of sensor positions and is able to take object or scene self-occlusions into account, with the potential for more accurate sensor positioning. In addition, we presented an algorithm that constructs a solid model from a mesh surface, and allows identification of the occluded and imaged surfaces, using modeling techniques from both mesh surface and solid representations. By combining these two we retain the benefits of mesh surfaces, such as representational flexibility and conceptual simplicity, while still allowing the use of well-defined set-theoretic merging operations inherent to solid modelers. Experimental results of the reconstruction of a complex object using the planner integrated with the model acquisition system were shown.

One area that requires future work is the target selection problem. A solution that may be effective is to select a target based on a density function that determines how much model surface tagged "occluded surface" is nearby the target. This would take advantage of the fact that nearby surfaces are often imaged during the same sensing operation, while avoiding the computational complexity necessary for a optimal solution.

7. References

[1] Steven Abrams, Peter K. Allen, and Konstantinos A. Tarabanis. Dynamic sensor planning. In *Int. Conf. on Intelligent Autonomous Systems*, pages 206–215, Pittsburgh, PA, 1993.

[2] J Cohen at al., Simplification Envelopes. In *Proc. of SIGGRAPH*, pp.119-128, 1996.

[3] C. Connolly. The determination of next best views. In *Proceedings 1985 IEEE Int. Conf. on Robotics and Automation*, pages 432–435, 1985.

[4] C. Connolly and J. Stenstrom. 3d scene reconstruction from multiple intensity images. 1989.

[5] B. Curless and M. Levoy. *A Volumetric Method for Building Complex Models from Range Images*. In *Proc. of SIGGRAPH*, 1996.

[6] P. Fua and P. Sander. Reconstructing Surfaces from Unorganized Points. In *Proc. of Image Understanding Workshop*, 1992.

[7] H. Hoppe. *Surface Reconstruction from Unorganized Points*. Ph.D. thesis, Dept. of Com. Science and Engineering, U. of Washington, 1994.

[8] K. Ikeuchi and P.J. Flynn, Editorial: Recent progress in CAD-based vision, Computer Vision and Image Understanding, 61(3), 1995.

[9] K.N. Kutulakos. *Exploring Three-Dimensional Objects by Controlling the Point of Observation*. Ph.D. thesis, Computer Sciences Department, University of Wisconsin, 1994.

[10] A. Laurentini. Inferring the Shape of the Real Object from the Object Reconstructed by Volume Intersection. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 1993.

[11] W. M. Martin and J. K. Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158, March 1983.

[12] J. Maver and R. Bajcsy. How to decide from the first view where to look next. In *Proc. 1990 DARPA Image Understanding Workshop*, pages 482–496, 1990.

[13] 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, in *Proc. SPIE Symposium on Intelligent Systems and Advanced Manufacturing*, Phila. PA, 1995.

[14] M. Rutishauser, M. Stricker, and M. Trobina. Merging range images of arbitrarily shaped objects. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 573–580, 1994.

[15] J. Stenstrom and C. I. Connolly. Constructing object models from multiple images. *Int. Journal of Computer Vision*, 9(3):185–212, 1992.

[16] G. Tarbox and S.N. Gottshlich. Ivis: An integrated volumetric inspection system. *Computer Vision and Image Understanding*, 61(3):430–444, may 1995.

[17] W. Thompson, H.J. de St. Germain, T.C. Henderson, and J.C. Owen. Constructing high-precision geometric models from sensed position data. In *Proc. 1996 ARPA Image Understanding Workshop*, pages 987–994, 1996.

[18] K. Tarabanis, R.Y. Tsai, and A. Kaul. Computing Occlusion-Free Viewpoints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3):279-292, March 1996.

[19] K. Tarabanis, R. Tsai, P.K. Allen. The MVP sensor planning system for robotic tasks. In *IEEE Trans. on Robotics and Automation*, 11(1), February 1995.

[20] K. Tarabanis P.K. Allen, and R. Y. Tsai. A survey of sensor planning in computer vision. In *IEEE Trans. on Robotics and Automation*, 11(1), February 1995.

[21] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proc. of SIGGRAPH*, pp. 311–318, 1994.

[22] K. Wu and M. Levine. Recovering parametric geons from multiview range data. In *Proc. of IEEE Comp. Society Conf. on Comp. Vision and Pattern Recognition*, pp. 156–166, 1994.

[23] P. Whaite and F. Ferrie. Uncertain views. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3–9, 1992.

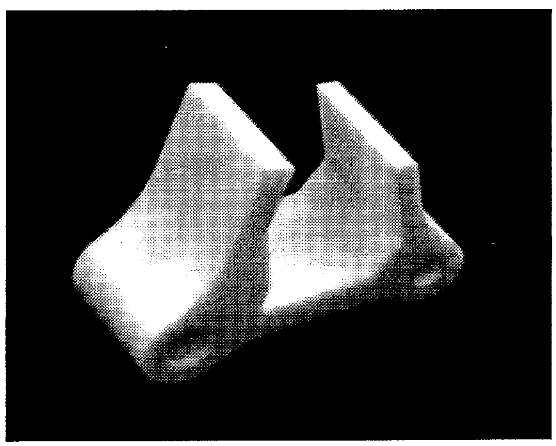

FIGURE 7. Photograph of strut-like part.

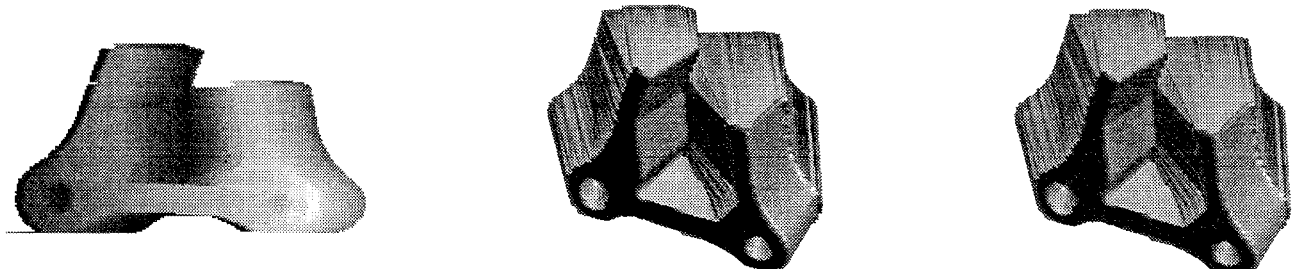

FIGURE 8. The first range image, the solid model constructed by sweeping its mesh surface, and the composite model after one view (left to right). As this is the first view, the composite model is identical to the model constructed from this range image.

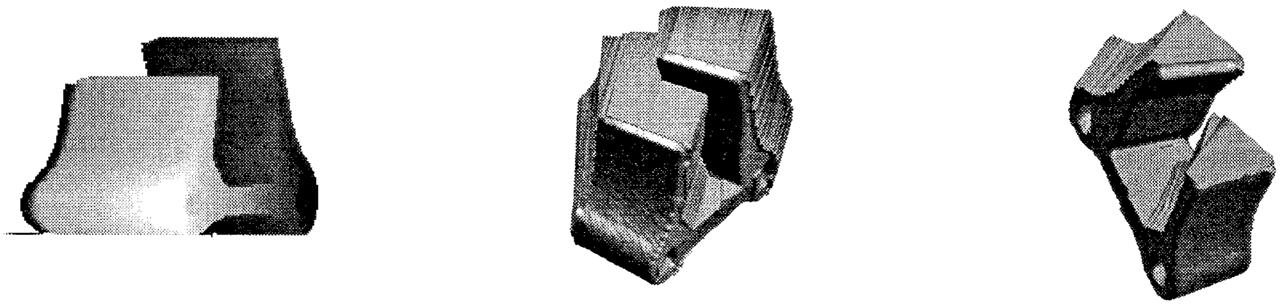

FIGURE 9. The second range image, acquired after a rotation of 90 degrees, the solid model derived from the range image, and the composite mode after two views (left to right). Note that the composite mode now has the overall shape of the object.

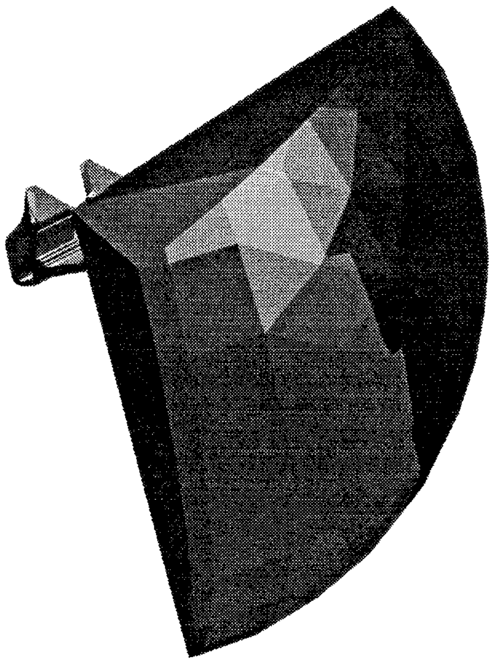

FIGURE 10. Occlusion computation for a target on the composite model: The entire black and grey volume represents $V_{unoccluded}$ for a target from the composite model's "occlusion surface". The black region is $\bigcup O_i$, the union of sensor positions occluded from the target by model surface i. The grey region is the visibility volume $V_{target} = V_{unoccluded} - \bigcup O_i$, i.e. the valid positions for a sensor viewing the target.

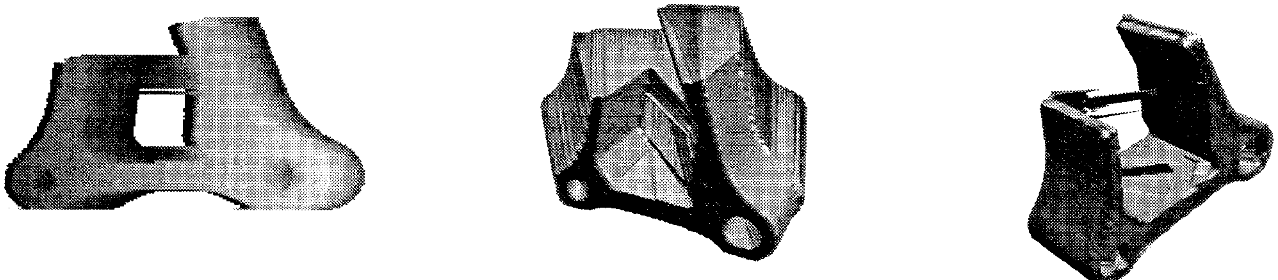

FIGURE 11. The third range image (acquired by use of the plan above), its solid representation, and the composite model after three views. The composite model is now very similar to the object, but there are still some occlusion surfaces between the strut's arms.

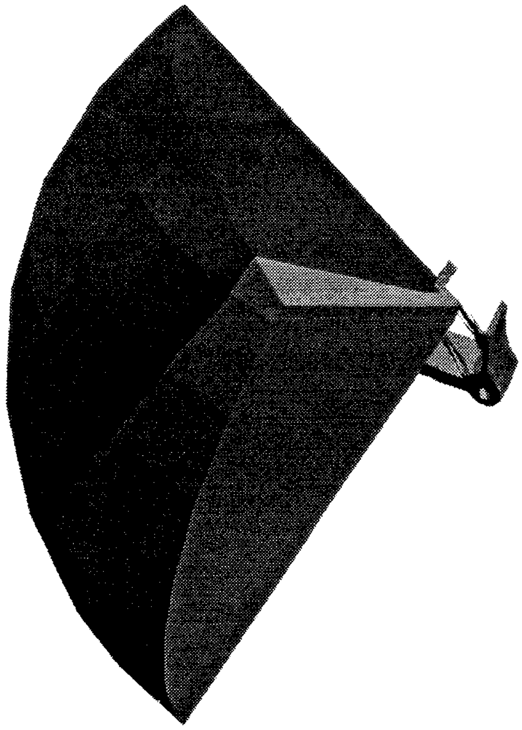

FIGURE 12. Result of sensor planning for a target specified on the "occlusion surface" of the composite model in Figure 11. Again, black volume specifies points that are occluded from seeing the target, grey volume describes the valid sensor positions.

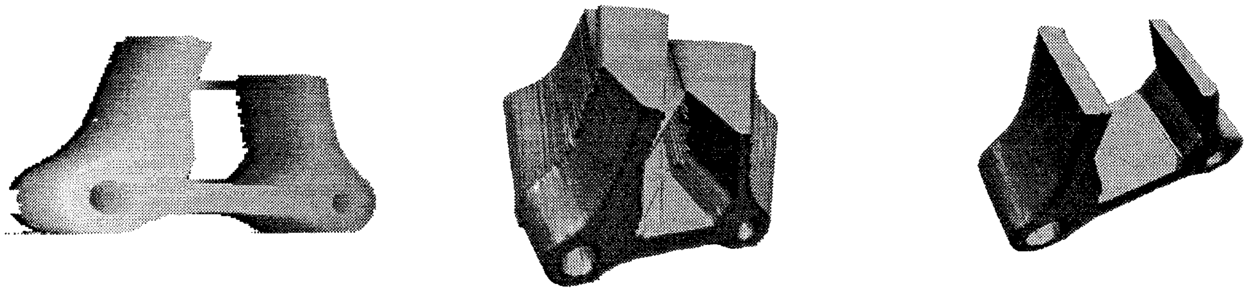

FIGURE 13. Fourth range image of object acquired according to plan in Figure 12, its solid, and the composite model after integration.

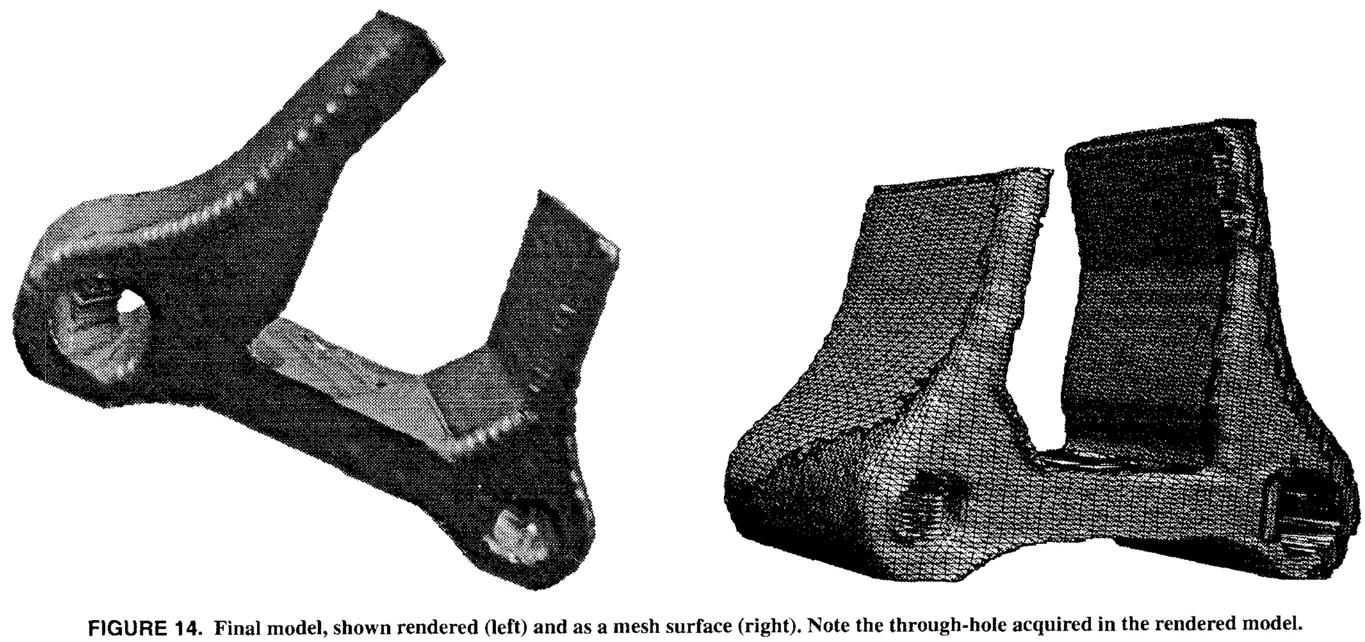

FIGURE 14. Final model, shown rendered (left) and as a mesh surface (right). Note the through-hole acquired in the rendered model.