

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

Michael K. Reed, Peter K. Allen*

Computer Science Department, Columbia University, 450 Computer Science Building, New York, NY 10027, USA

Received 21 July 1997; received in revised form 21 January 1998; accepted 4 February 1998

Abstract

In this article we present a method for automatically constructing a solid (in the CAD sense) model of an unknown object from range images. The model acquisition system provides facilities for range image acquisition, solid model construction and model merging: a solid representation is derived from a mesh surface that models the range data from each view, which is then merged with the model built from previous sensing operations. This modeling system has the benefit of constructing a solid model at each phase of the acquisition process, and is able to model parts that are difficult or impossible using other methods, such as extremely thin parts or those with deep through-holes. We introduce a technique that utilizes the incomplete model resulting from the merging process to plan the next sensing operation by finding a sensor viewpoint that will improve the fidelity of the model. A number of examples are given for the modeling of various objects that include planar and curved surfaces, features such as through-holes, and large self-occlusions. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Solid modeling; Range images; Sensor planning; Surface reconstruction; Laser rangefinder; Sensor modeling

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* [9], 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. As 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 or other irregularities, i.e. it is 'watertight' and conforms to the definition of a 'solid model' [15].

This article 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 surfaces that have not yet been modeled. 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

[☆] Based on "3-D Modeling from Range Imagery: An Incremental Method with a Planning Component" by Michael Reed, Peter Allen and Ioannis Stamos which appeared in Proceedings of 3-D Digital Imaging and Modelling, Ottawa, May 1997; pp. 76–84. © 1999 IEEE.

* Corresponding author.

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.

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 [19]. 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 [18]. Methods that use a mesh surface to model and integrate each of a set of range images, such as work by Turk and Levoy [23] or by Rutishauser et al. [16], or to model a complete point sampling as by Hoppe [8] or Fua and Sander [7] have also proven useful in this task. Both Connolly and Stenstrom [4,17] and Martin and Aggarwal [12] perform edge detection and projection from intensity images, a concept that is revisited by Laurentini [11]. 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 (50–70) 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 [10] 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 [13] 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 [24], 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 and Bajcsy [14] 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. Integration of models from different viewpoints is done via set intersection, similar to that described in [11] for intensity images. 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. We also discuss some of the problems caused by using set intersection methods in the context of sampled data.

3.1. Modeling a surface from a range image

The acquisition of range data is performed by a robotic system similar to many range-scanning configurations: a Servo-Robot laser rangefinder is attached to an IBM SCARA robot, with the object to be imaged being placed on a motorized rotation stage (Fig. 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 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. The points in the range image may then be used as the vertices in a mesh. However, since the mesh

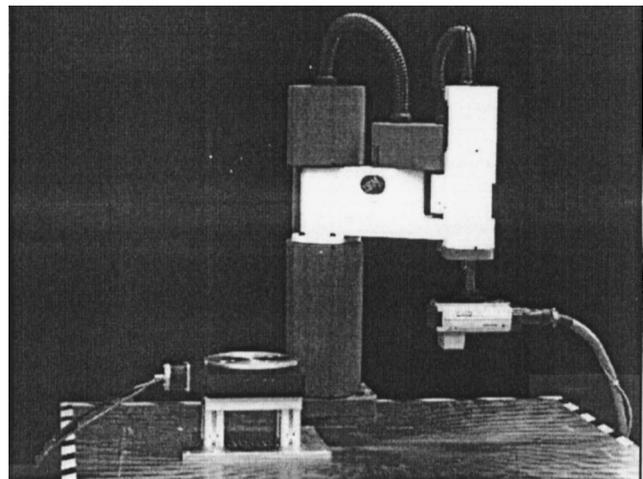


Fig. 1. Experimental set-up showing robot with attached laser rangefinder (to right) and turntable (to left).

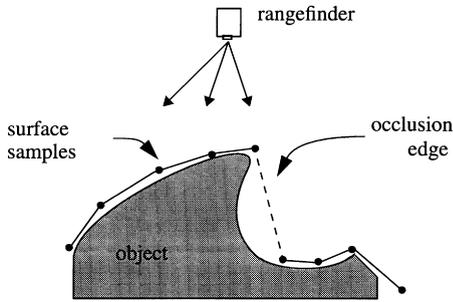


Fig. 2. Example of edges between sampled vertices on a surface.

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 cannot be closed until the entire object has been scanned. Thus, methods that only use the mesh as a surface model require a large number of overlapping scans, will not work with objects whose entire surface is not visible to the sensor, and will preclude the use of a planning method or any other procedure that requires a solid model during the acquisition process.

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

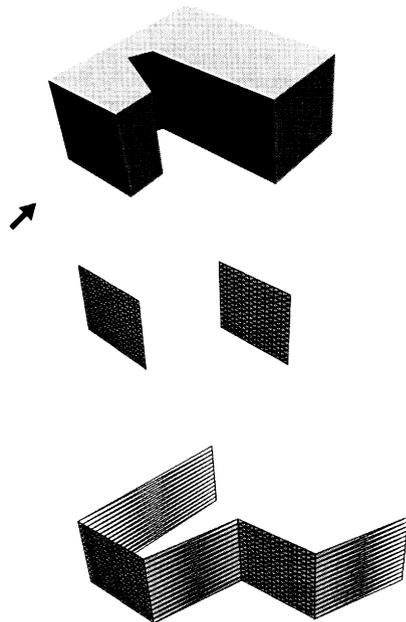


Fig. 3. Top: Rendering of CAD model of a typical 2-1/2 D part, shown with a sensing direction. Middle: Surface mesh from synthetic range data of CAD part. This mesh does not include any elements that contain occlusion edges. Bottom: Surface mesh generated from synthetic range data, including elements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

mesh connectivity will be. In this regard our work differs from other mesh-based methods such as mesh zipping [23] and other similar re-meshing techniques [16] 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 by their orientation or because their lengths exceed some threshold (Fig. 2). Our system retains these elements, since they denote the boundary between surfaces of the object that are imaged by the sensor and the space occluded from the sensor and therefore needs further imaging. (These elements must be handled with care, because they have the potential to violate certain assumptions regarding the model construction process, which we will discuss below.)

As an example of this process, consider the hypothetical object shown at the top of Fig. 3. A range image is sampled from the CAD model using the shown sensing direction. The surface model shown in the middle of Fig. 3 is typical of mesh-based methods; no occlusion edges are represented, and although it is possible to attach 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 Fig. 3 represents both the imaged surfaces of the object and the boundary of occluded space between the imaged surfaces.

3.2. Sweeping the mesh into a solid

This mesh surface M 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:

$$S = \bigcup_{\forall m} \text{extrude}(m), \quad m \in M$$

An extrusion operator is applied to each triangular mesh element m , orthographically along the vector of the range-finder's sensing axis, until it comes in contact with a far bounding plane. The result is the five-sided solid of a triangular prism (Fig. 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 derived from the sweeping operation, and a bounding surface that caps one end.

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



Fig. 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.

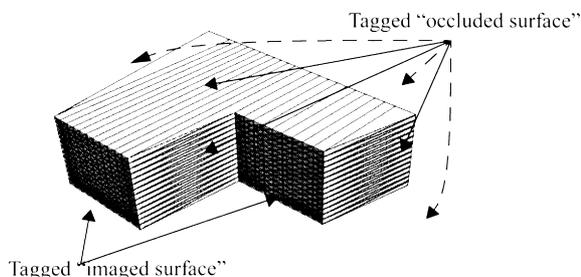


Fig. 5. Solid formed by sweeping the mesh shown at bottom of Fig. 3 in the sensing direction. Tags for hidden surfaces are shown with dotted arcs.

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 each 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 Fig. 3. Sweeping its mesh (shown at the bottom of Fig. 3) results in the solid shown in Fig. 5, its surfaces tagged according to the process described before.

3.3. Merging single-view models

Each successive sensing operation will result in new information that must be registered and merged with the

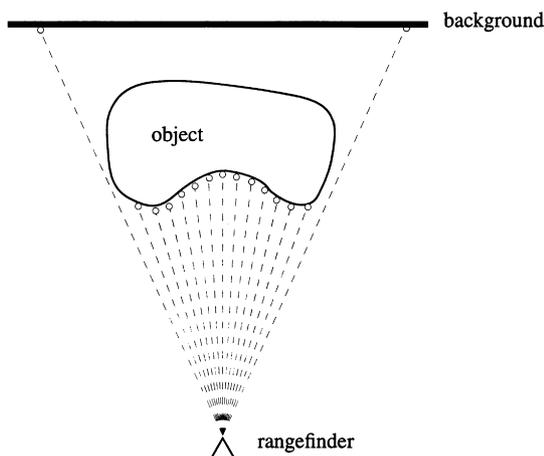


Fig. 6. Sensing behavior of a typical range scanner in 2D. Scanned points are shown as circles, and the sensor's energy emission is shown as dotted lines.

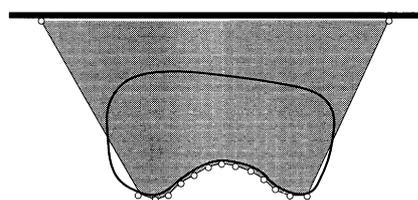


Fig. 7. 2-D example of a mesh surface determined directly from the sensed points (shown as a thin line connecting the sensed points) and the solid formed by sweeping this mesh (shown as the darkened region).

current model being built, called the *composite model*. Registration in our system is done by calibration of the rangefinder, robot and turntable, which produced satisfactory results for our purposes. Automated model registration techniques also exist that may provide a higher degree of registration accuracy [6]. Merging of mesh-based surface models has been done using clipping and re-triangulation methods that also perform some vertex averaging [16,23]. These methods are necessary because the mesh surface models are not closed, so specialized techniques to operate on non-manifold surfaces of approximately continuous vertex density are needed. However, these specialized methods have robustness and accuracy issues that are largely unexplored, and will only result in a closed model when the entire object's surface has been completely imaged, an impossibility for many parts. In our method

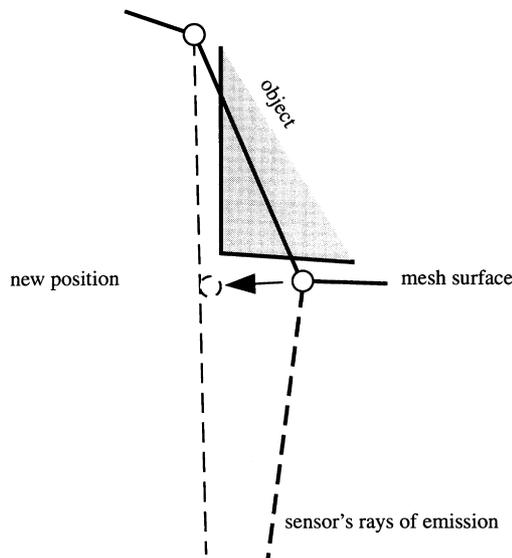


Fig. 8. Detail of a 'missed' object surface leading to the situation in Fig. 7: A remedy is to move the sensed point along the arrow from its original position to the new position.

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.

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 use the following rule 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 solids.

Reconstruction methods based on mesh surface integration and averaging, e.g. [16,23], are often thought to be better than methods based on set intersection because the averaging helps to produce more accurate results. In our

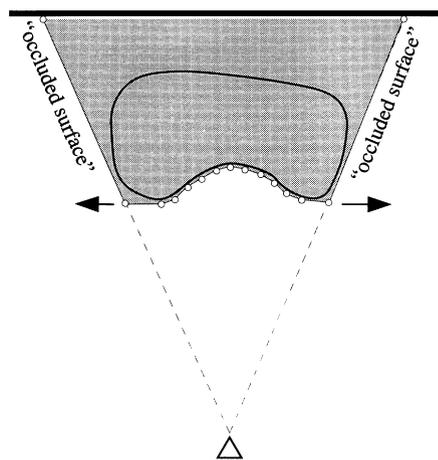


Fig. 9. Effect of nonuniformly dilating mesh on swept solid: The vertices of the (two) occlusion surfaces are moved until they coincide with the rays of the sensor's nearest adjacent sensing ray, in the direction shown by the arrows. The area resulting from the swept surface is shown in grey, and is clearly a superset of the actual object.

method, although intersection is used during the merging phase of the model-building process, during the range image acquisition, averaging methods may still be used to improve the quality of the mesh surface. If, after this has been done, there are still differences in the location of the same surface seen in different views, they are more likely to be the effect of incorrect registration rather than sensing errors. In this case, averaging will produce a model that may or may not be more accurate, in effect blurring the features or surfaces in the vicinity of the vertex. Additionally, registration errors may prevent a closed surface from being formed when using mesh surface integration methods, in contrast to our method which will produce solid models even in the presence of significant registration error. Nevertheless, methods that utilize averaging during integration provide a convenient solution to the artifacts that sometimes affect set intersection-based methods.

3.4. Interaction of sampling sensors and set intersection methods

The construction of single-view and composite models as described above makes two major assumptions. The first is typical for all systems that perform modeling from observation:

Assumption: *Any object surface that adheres to certain orientation constraints will be sampled in detail sufficient for the task.*

In reconstruction methods based on mesh surfaces, mesh elements corresponding to occlusions in the object or scene are identified by their orientation and are removed, as previously described. This allows those modeling systems to assume the converse of the above assumption: that all (remaining) model surfaces correspond to and properly capture actual surfaces in the object. However, our method also constructs surfaces from the boundaries between imaged and occluded space in the scene: as described above these are tagged as ‘occluded surface’. These surfaces clearly do not fall under the above assumption, since they are not part of the object's imaged surface, and in fact may be arbitrarily far from true surfaces in the scene. Moreover, their behavior violates a second assumption that is a requirement for using set intersection as a method of integration, whether it be by intensity, rangefinding, or other sensing methods:

Assumption: *The space bounded by the solid representing any single view of the object must be a superset of the space occupied by the actual imaged object.*

Thus, it must never happen that the solid constructed from a single view does not contain some region or feature that is present in the actual object. If this happens, due to the semantics of set intersection, that region or feature will never be able to be acquired, no matter how many other single-view models properly represent it.

Unfortunately, this assumption is violated if the scene is modeled using the points from the range image directly as

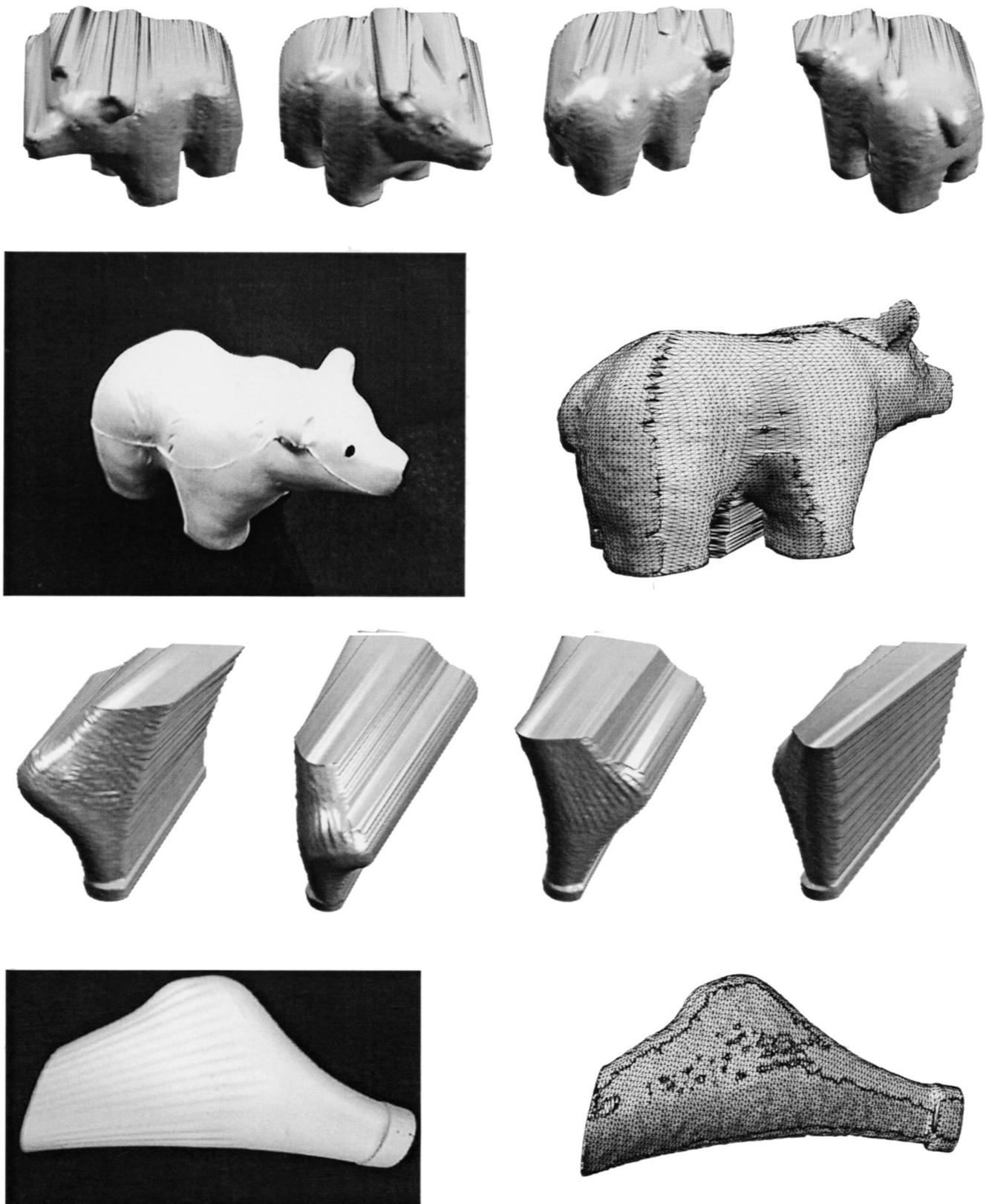


Fig. 10. Models of hip prosthetic and toy bear in four views. The solid models from four distinct range images are shown, as is the wireframe of the composite model next to a photograph of the actual object.

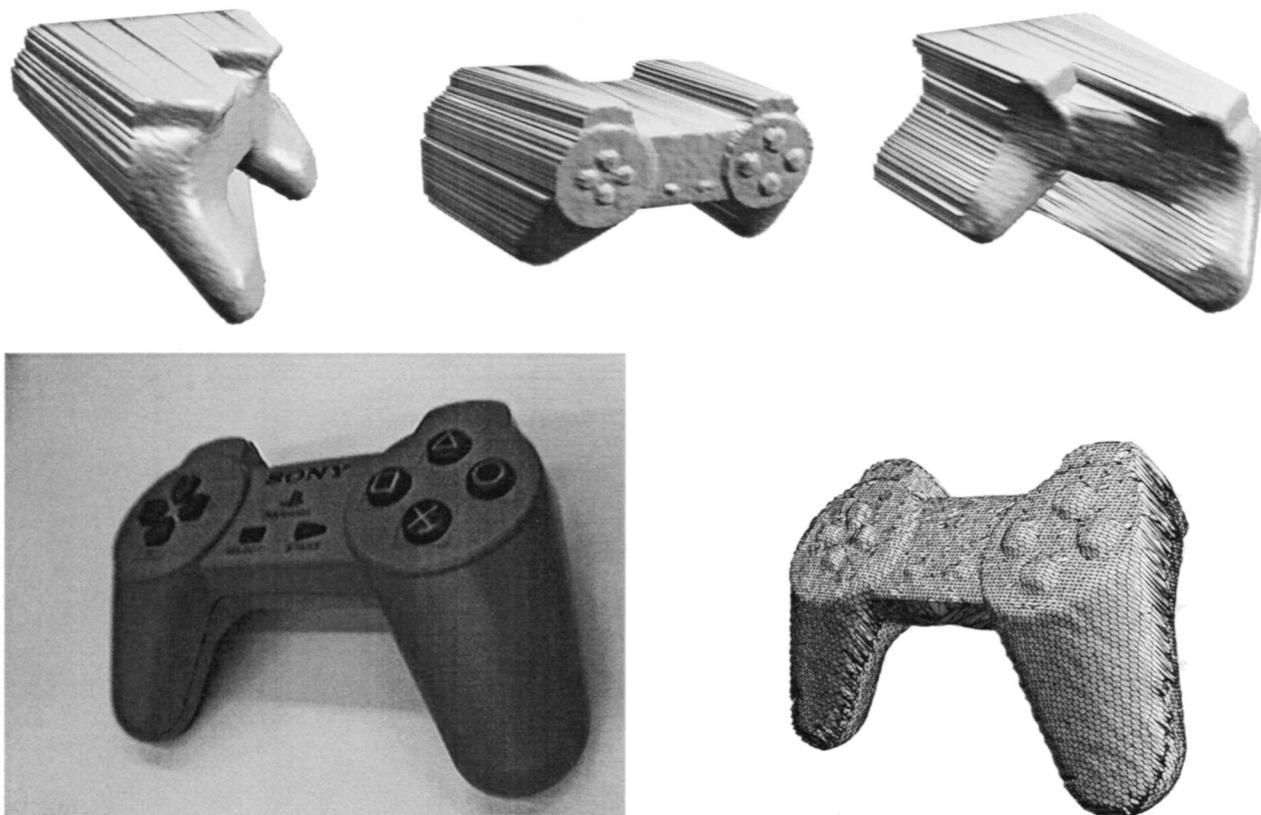


Fig. 11. Model of the video game controller in three views. Solid models from each of three views are shown at top; bottom left is photograph of actual part, bottom right is wireframe of acquired model.

vertices in a mesh surface. This is because the modeled surface error, i.e. the distance by which the mesh surface may deviate from the true surface, is related to the distance between adjacent vertices in the mesh, and thus the sampling interval, angle, and distance of the surface w.r.t. the sensor are all related (Fig. 6). As shown in the example of Fig. 7, a surface constructed from the sensed points may drastically underestimate the surface of the object, and therefore the solid formed by the swept surface may not include some parts of the original object. Again, methods based on set intersection require that the object is never underestimated, as once a part of an object is removed during the intersection process, it may never be recovered.

It is because of this phenomenon that previous model construction techniques using mesh representations discard mesh elements that have a large separation between vertices. However, we propose another solution to this problem, in which the initial mesh is nonuniformly dilated so that it correctly represents the *largest* object that could have produced the point sampling given by the range image. In order to accomplish this, it is necessary to be able to identify those elements of the mesh surface that may be interior to the space occupied by the sensed object. It is precisely those surfaces that represent the boundary between imaged and unimaged surface that may contribute to this problem by ‘missing’ the object surface, and as noted above these elements have already been identified and

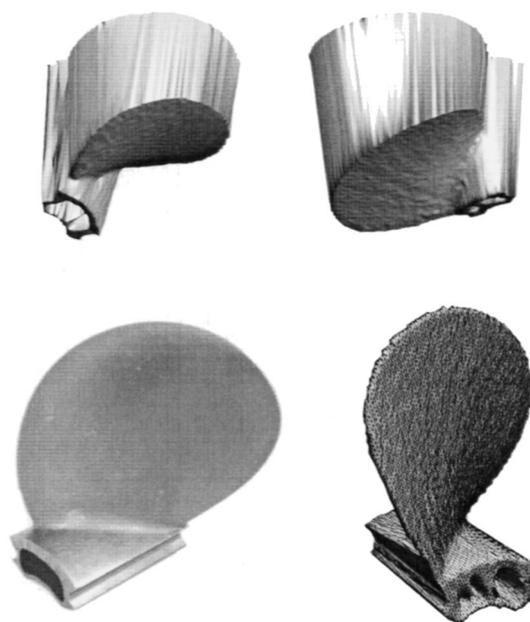


Fig. 12. Model of propeller blade in two views. Solid models from each view are shown at top; bottom left is photograph of actual prop, bottom right is wireframe of acquired model.

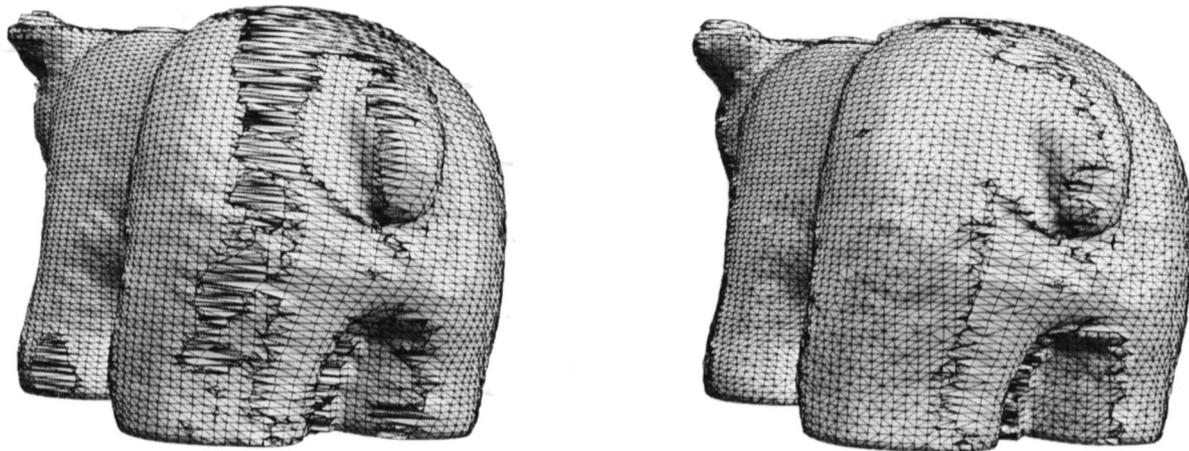


Fig. 13. Bear model detail showing effect of dilation process on model-building. Left: model built using sensed points directly as vertices in mesh from each view. Right: model built using dilation process on mesh of each range image prior to sweep operation.

tagged as ‘occluded surface’. Therefore it is possible to proceed by extending those surfaces so that they are guaranteed to be ‘outside’ the actual object surface. This is done by identifying one or more of the element’s vertices and moving it in the direction of the sensor’s baseline until it approaches the position where the nearest sample would have been. Fig. 8 details such a case: the vertex of an element is moved so that it approaches the next adjacent sensing ray, and therefore must be ‘outside’ any sensed object. Because the sensor’s resolution, and therefore the magnitude of the displacement of the vertex, depends on the distance from the sensor, it is important to have a sensor model from which parameters such as beam angle between adjacent samples may be calculated. An alternative to this technique, pointed out by one of the anonymous reviewers, is to insert a new vertex in the mesh at the appropriate point. This method has the advantage of allowing the new vertex to be tagged for future identification in the merging or planning processes. The effect of the nonuniform dilation process on the 2-D example is shown in Fig. 9, and is shown for real range data in the following section. It should be noted that the magnitudes of these displacements are small (< 1 mm) for the range images in this paper: they are not visible with the naked eye in any of the models from single views, but the effect of this process is readily visible in the final model as we show below.

3.5. Examples of object modeling without sensor planning

A few examples are shown here to illustrate the described process. Shown are models constructed from four, three, and two views taken from equivalent turntable rotations. The first two examples are of the construction of a prosthetic hip model and of a toy bear from four range images (Fig. 10). The solids constructed from each of the four range images are shown, along with a photograph of the object and a wire-frame rendering of the final model. The models have a relatively uniform sampling of their surfaces,

except at the regions where there was overlap between two or more of the single-view models (and therefore overlap of the range images as well). In these places, there tends to be a greater concentration of faces due to the effects of intersecting the surfaces.

In Fig. 11, the model of a hand-held video game controller is modeled in three views. This part consists of polygonal and curved surfaces at varying levels of detail, including buttons on its front surface. As a last example, Fig. 12 shows a propeller blade modeled in two views. This example is particularly interesting because it shows that this method can model objects that are not acquirable using methods that rely on mesh overlap: no mesh overlap of the opposing view of the blade surface is possible caused by the extremely thin surface there, and hence those

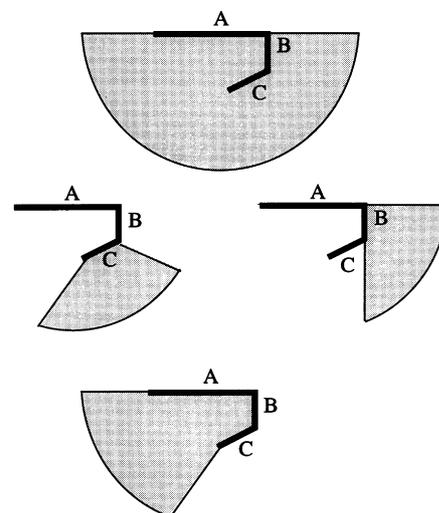


Fig. 14. Planning for model consisting of three surfaces A (the target), B, and C. Top: the model with $V_{\text{unoccluded}}$ for A shown in grey. Middle: the occlusion due to O_C (left) and O_B (right). Bottom: the final visibility volume V determined by $V_{\text{unoccluded}} - (O_B \cup O_C)$, i.e. with occlusions taken into account: a point in the grey area has an unobstructed view of the target A.



Fig. 15. Photograph of strut-like part.

methods that rely on it will fail. This situation will occur whenever a part has a feature whose width approaches the sampling interval, a common event with machine parts such as gears and propellers.

Fig. 13 demonstrates the positive effect of the nonuniform dilation process of Section 3.4 on a model, in this case the previously-shown model of the bear toy. The left image of Fig. 13 shows the backside of the bear model when

no such process is applied to the mesh, and thus the swept surface is computed directly from the sensed points. As can be seen, there are considerable artifacts that not only poorly represent the actual surface of the object but also generate undesirable local geometry. In contrast, the right image in the figure shows the same portion of the model when it was built using the dilation process. The surfaces here have a much more uniform appearance, and also more accurately model the object.

The performance of this model building process on a Silicon Graphics Onyx II using two processors is as follows. The mesh construction, dilation and extrusion to a solid takes approximately 4 min for each 110×128 rectangular range image, slightly less for those images where more background is visible. Merging solids using the set intersection operator takes approximately 2 min per model. The number of surfaces on the resulting models are as follows: propeller: 17 758, bear: 17 335, hip: 17 228, game controller: 20 301.

4. The planning process

The model construction process we have described so far performs no planning of sensor viewpoints; we use equal turntable rotations between range images. In this section, we describe how we may dynamically plan the next viewpoint

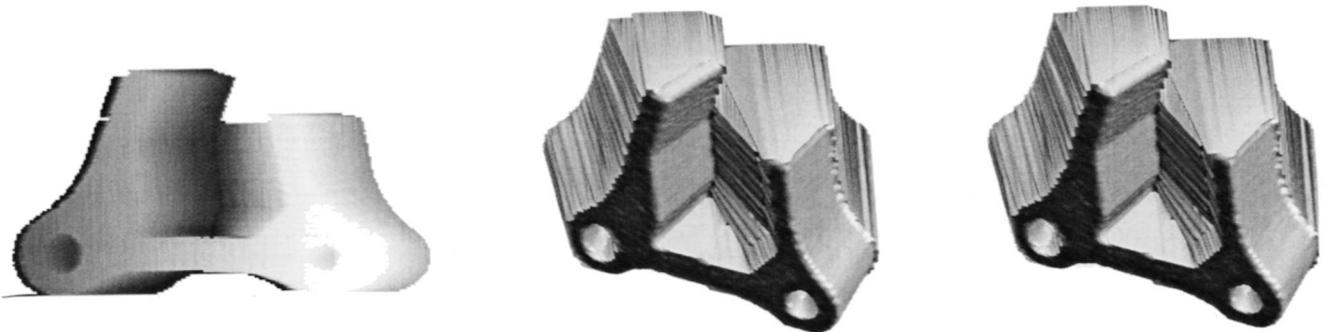


Fig. 16. The first range image, the solid model constructed by sweeping its mesh surface, and the composite model after one view (left to right). As this is the first view, the composite model is identical to the model constructed from this range image.

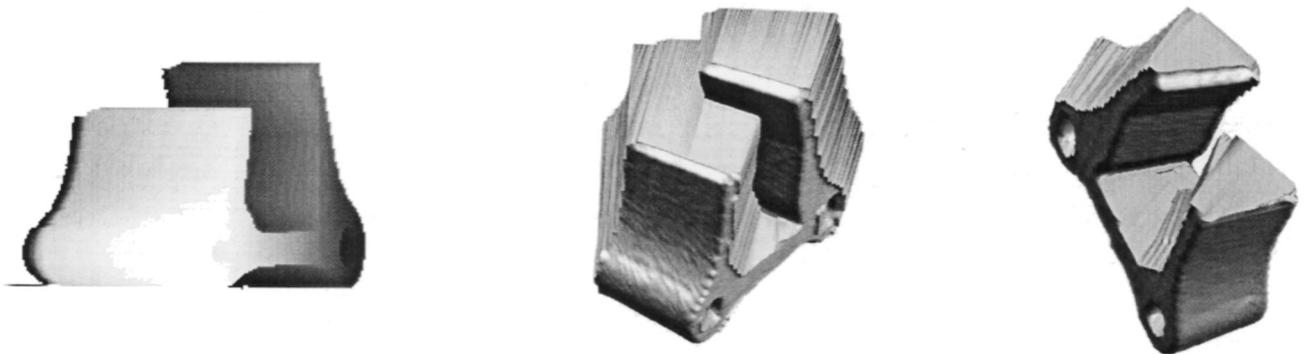


Fig. 17. The second range image, acquired after a rotation of 90 degrees, the solid model derived from the range image, and the composite mode after two views (left to right). Note that the composite mode now has the overall shape of the object.

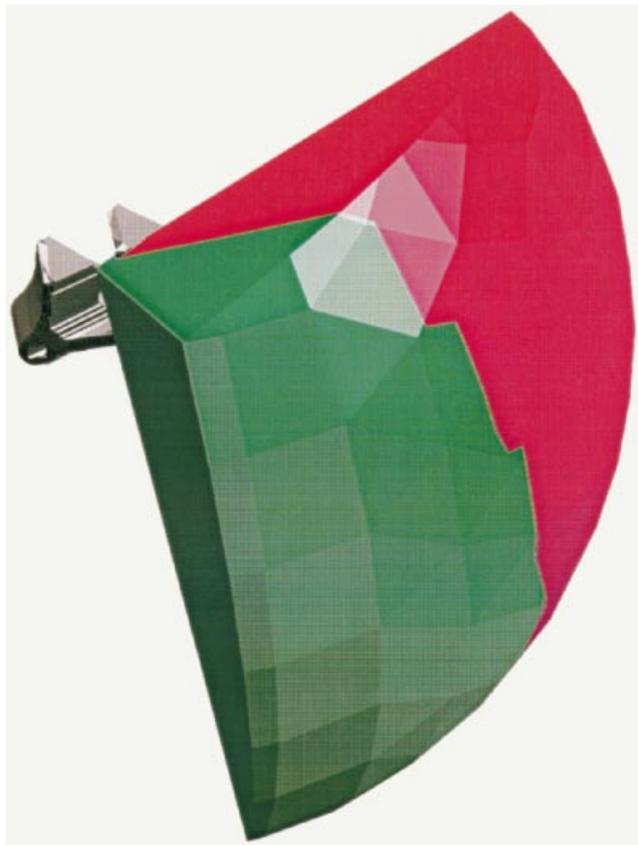


Fig. 18. Occlusion computation for a target on the composite model: The entire green and red volume represents $V_{\text{unoccluded}}$ for a target from the composite model's 'occlusion surface'. The red space is $\cup O_i$, the union of sensor positions occluded from the target by model surface i . The green space is the visibility volume $V_{\text{target}} = V_{\text{unoccluded}} - \cup O_i$, i.e. the valid positions for a sensor viewing the target.

during the model acquisition process. This planning system relies on occlusion information, which as we have noted before is an important scene attribute useful to the planning process. Occlusion was used previously to assist planning 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 [3,24]. 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 [3,13]. 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 previous problems are avoided.

4.1. Planning for unoccluded viewpoints

The planning component presented here is based on previous work on the sensor planning problem in our laboratory [1,21]. 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 [22]. 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 was 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

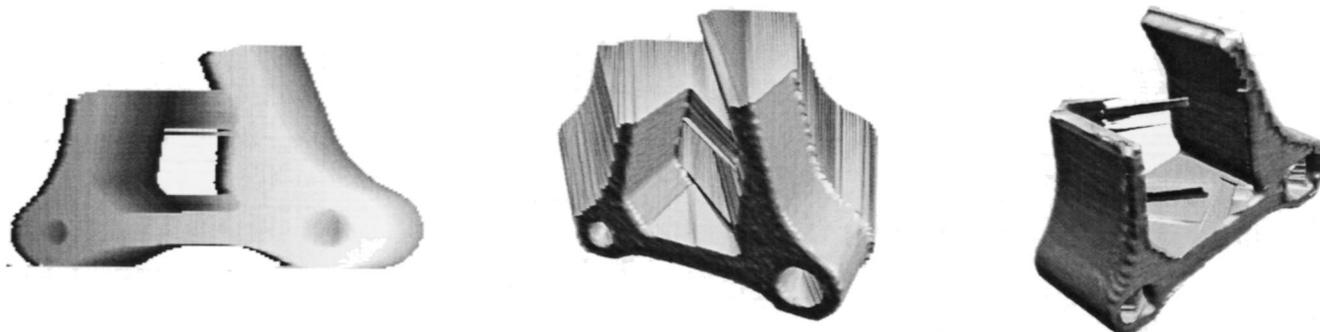


Fig. 19. The third range image (acquired by use of the plan above), its solid representation, and the composite model after three views. The composite model is now very similar to the object, but there are still some occlusion surfaces between the strut's arms.

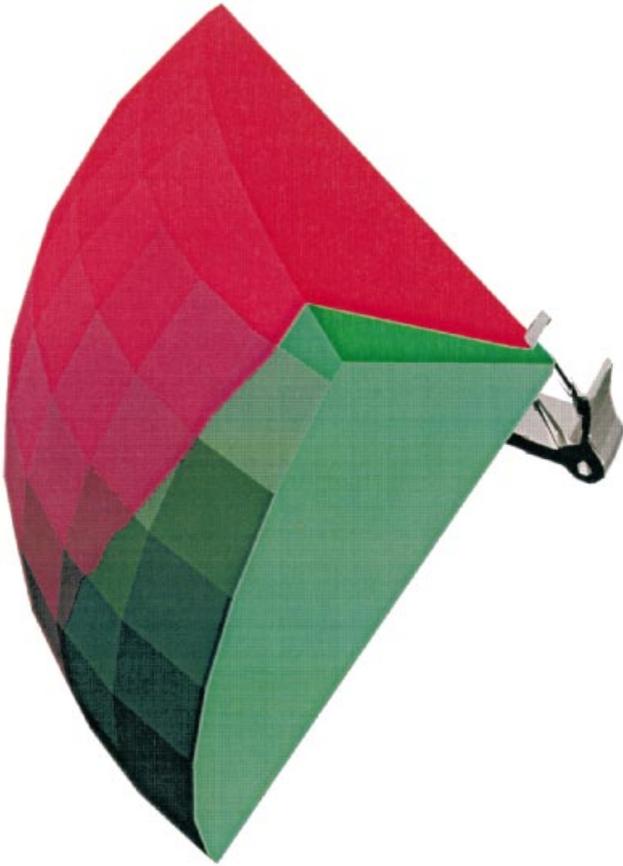


Fig. 20. Result of sensor planning for a target specified on the ‘occlusion surface’ of the composite model in Fig. 19. Again, red volume specifies points that are occluded from seeing the target, green volume describes the valid sensor positions.

of all sensor positions that have an unoccluded view of the target for a specified model. This can be computed by determining $V_{\text{unoccluded}}$, 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:

$$V_{\text{target}} = V_{\text{unoccluded}} - \cup O_i$$

The volume described by $V_{\text{unoccluded}}$ 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 set of sensor positions whose visibility of the target is occluded by model surface i , 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 [20]. We illustrate by computing V_{target} for a 2-D target in Fig. 14, 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, $V_{\text{unoccluded}}$ 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.

4.2. Example of the planning process

The capabilities of this planning process are demonstrated by building a model from distinct views of the object shown in Fig. 15, 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 acquired without using sensor planning: the first image is taken from an arbitrary position, while the second image is acquired after a turntable rotation of 90 degrees. These two range images, the models acquired from them, and their respective composite model are shown in Figs 16 and 17. 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 Fig. 18. This plan is executed by rotating the turntable to place the sensor within the green visibility volume, in this case an additional 83 degrees, which produces the image and model shown in Fig. 19. Again, a target is designated and a plan produced, which is shown in Fig. 20. The turntable is

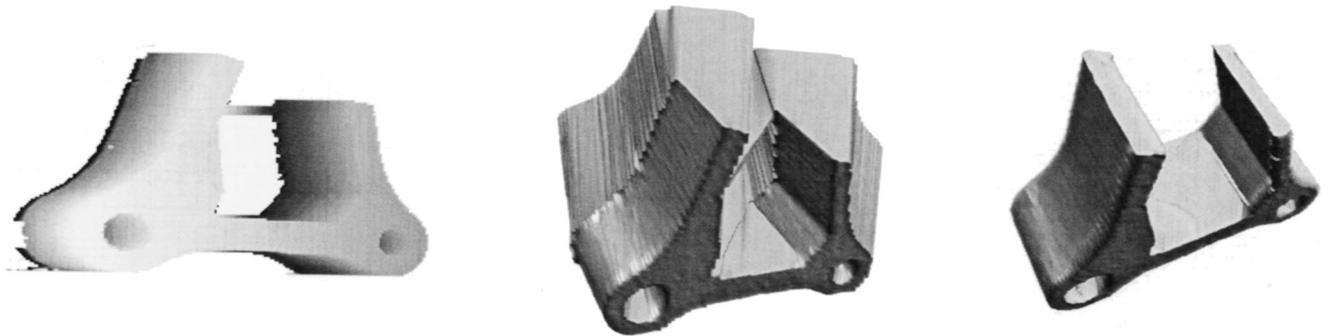


Fig. 21. Fourth range image of object acquired according to plan in Fig. 20, its solid, and the composite model after integration.



Fig. 22. Final model, shown rendered (left) and as a mesh surface (right). Note the through-hole acquired in the rendered model.

rotated 134 degrees to move the sensor into this visibility volume, and the image and model from Fig. 21 result. This final model is shown rendered and as a mesh in Fig. 22. As was present in the earlier models, there are ‘boundaries’ where the intersection of the solids from two overlapping sensing operations causes an increase in the density of mesh elements. As this is an incremental method, additional scans may be also taken to improve the quality of the model.

In the previous case, a plan was generated by considering a single target surface. Although this method will work, it is only local in its scope and does not take global visibility of all possible targets into account. A more sophisticated method would be to plan using multiple targets, and select the sensor position that images the most area. Although in many sensing scenarios this is unnecessary, we show here how multiple targets may be used to plan. In this example, the composite model of the strut after two views is shown in Fig. 23. This model was decimated using a modified version of the Simplification Envelopes method [2], in which the

boundary between surfaces labelled ‘imaged surface’ and those labelled ‘occluded surface’ is retained. After this decimation, the 30 occluded surfaces with the largest area are selected as targets and plans are generated for each of them. These plans (shown at the top left of Fig. 23) are then intersected with the sensor’s reachable space, in this case a cylinder (shown in the top right of Fig. 23). Overlapping plans appear as darker regions and represent sets of positions in sensor space where more than one target is visible. These regions may then be searched for the position that images the most target area.

5. Conclusions

We have described a system that builds a 3-D CAD model of an unknown object incrementally from multiple range images. The method is based on an algorithm that constructs a solid model from a mesh surface, and allows identification of the occluded and imaged surfaces, by 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. Benefits including guaranteed water-tight models and the ability to acquire both very thin objects and features such as through-holes are realized using this method. In addition, we presented a planning method that relies on the specification of targets on the incomplete model of the

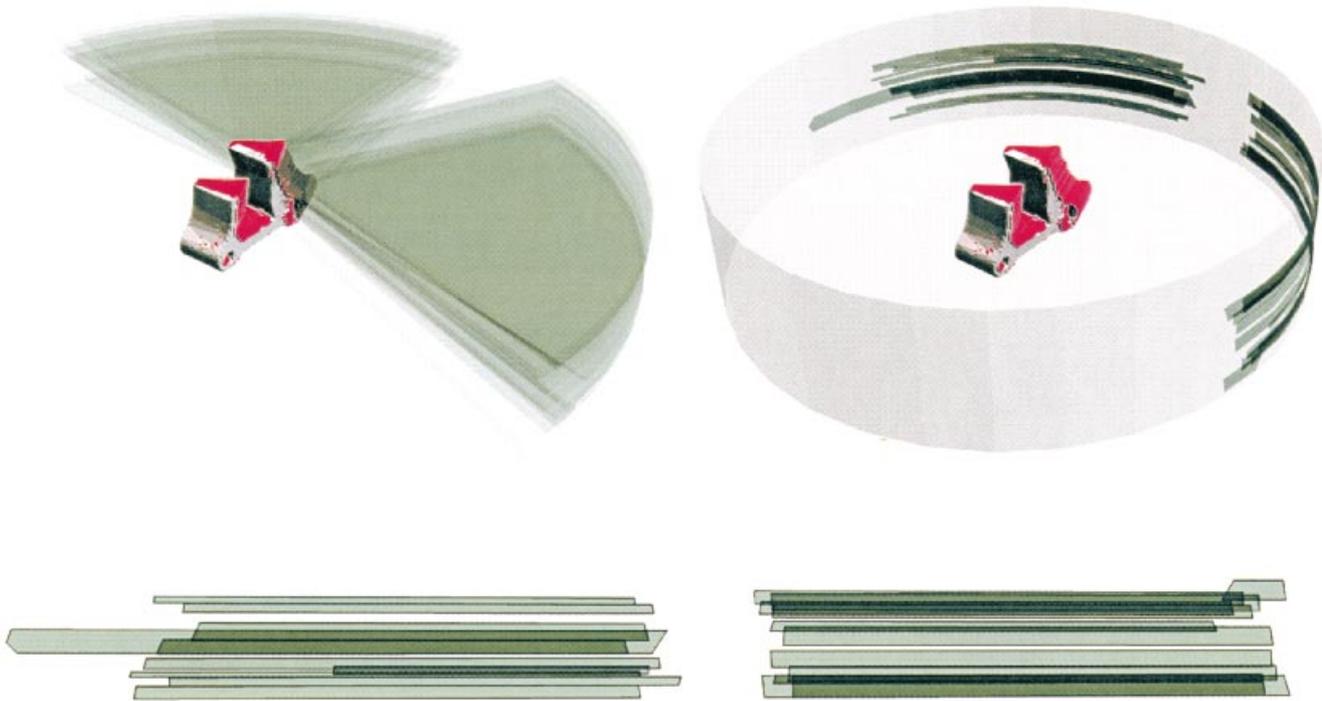


Fig. 23. Planning using multiple targets. Top left: Model after two views with plans for the 30 largest occlusion surfaces. Top right: Model with intersections of plans and sensor’s reachable space (a cylinder). Bottom: Intersections unwrapped from cylinder onto plane to clearly show overlaps.

object to drive the planning process. This permits a static sensor planner to be used to compute occlusion-free view-points 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.

Acknowledgements

We thank Ioannis Stamos for his help in generating the visibility volumes, and the anonymous reviewers for their thoughtful suggestions. This work was supported in part by an ONR MURI Grant, DARPA AASERT awards DAAHO4-93-G-0245 and DAAH04-95-1-0492, and NSF grants CDA-96-25374 and IRI-93-11877.

References

- [1] S. Abrams, P.K. Allen, K.A. Tarabanis, Dynamic sensor planning, in: *International Conference on Intelligent Autonomous Systems*, Pittsburgh, PA, 1993, pp. 206–215.
- [2] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Webber, P. Agarwal, F. Brooks, W. Wright, Simplification Envelopes, in: *Proceedings of SIGGRAPH*, 1996, pp. 119–128.
- [3] C. Connolly, The determination of next best views, in: *Proceedings 1985 IEEE International Conference on Robotics and Automation*, 1985, pp. 432–435.
- [4] C. Connolly, J. Stenstrom, 3D scene reconstruction from multiple intensity images, in: *Proceedings 1989 IEEE International Conference on Robotics and Automation*, pp. 124–130.
- [5] B. Curless, M. Levoy, A volumetric method for building complex models from range images, in: *Proceedings of SIGGRAPH*, 1996.
- [6] C. Dorai, A.K. Jain, Optimal registration of object views using range data, *IEEE Trans. Pattern Anal. Machine Intelligence* 19 (10) (1987) 1131–1138.
- [7] P. Fua, P. Sander, Reconstructing surfaces from unorganized points, in: *Proceedings of ARPA Image Understanding Workshop*, 1992.
- [8] H. Hoppe, Surface reconstruction from unorganized points, Ph.D. thesis, Dept. of Computer Science and Engineering, University of Washington, USA, 1994.
- [9] K. Ikeuchi, P.J. Flynn, Editorial: Recent progress in CAD-based vision, *Comput. Vis. Image Understanding* 61(3), 1995.
- [10] K.N. Kutulakos, Exploring three-dimensional objects by controlling the point of observation, Ph.D. thesis, Computer Sciences Department, University of Wisconsin, USA, 1994.
- [11] A. Laurentini, Inferring the shape of the real object from the object reconstructed by volume intersection, in: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1993.
- [12] W.M. Martin, J.K. Aggarwal, Volumetric descriptions of objects from multiple views, *IEEE Trans. Pattern Anal. Machine Intelligence* 5 (2) (1983) 150–158.
- [13] J. Maver, R. Bajcsy, How to decide from the first view where to look next, in: *Proceedings 1990 DARPA Image Understanding Workshop*, 1990, pp. 482–496.
- [14] R. Pito, R. Bajcsy, A solution to the next best view problem for automated CAD model acquisition of free-form objects using range cameras, in: *Proceedings SPIE Symposium on Intelligent Systems and Advanced Manufacturing*, Philadelphia, PA, 1995.
- [15] A.A.G. Requicha, Representations for rigid solids: Theory, methods, and systems, *ACM Comput. Surv.* 12 (4) (1980) 437–466.
- [16] M. Rutishauser, M. Stricker, M. Trobina, Merging range images of arbitrarily shaped objects, in: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994, pp. 573–580.
- [17] J. Stenstrom, C.I. Connolly, Constructing object models from multiple images, *Int. J. Comput. Vis.* 9 (3) (1992) 185–212.
- [18] G. Tarbox, S.N. Gottshlich, Ivis: An integrated volumetric inspection system, *Comput. Vis. Image Understanding* 61 (3) (1995) 430–444.
- [19] W. Thompson, H.J. de St. Germain, T.C. Henderson, J.C. Owen, Constructing high-precision geometric models from sensed position data, in: *Proceedings 1996 ARPA Image Understanding Workshop*, 1996, pp. 987–994.
- [20] K. Tarabanis, R.Y. Tsai, A. Kaul, Computing occlusion-free view-points, *IEEE Trans. Pattern Anal. Machine Intelligence* 18 (3) (1996) 279–292.
- [21] K. Tarabanis, R. Tsai, P.K. Allen, The MVP sensor planning system for robotic tasks, *IEEE Trans. Robotics Autom.* 11(1) (1995).
- [22] K. Tarabanis, P.K. Allen, R.Y. Tsai, A survey of sensor planning in computer vision, *IEEE Trans. Robotics Autom.* 11(1) (1995).
- [23] G. Turk, M. Levoy, Zippered polygon meshes from range images, in: *Proceedings of SIGGRAPH*, 1994, pp. 311–318.
- [24] P. Whaite, F. Ferrie, Uncertain views, in: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1992, pp. 3–9.