Automated Model Acquisition from Range Images with View Planning

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

Abstract

We present an incremental system that builds accurate CAD models of objects from multiple range images. Using a hybrid of surface mesh and volumetric representations, the system creates a "water-tight" 3D model at each step of the modeling process, allowing reasonable models to be built from a small number of views. We also present a method that can be used to plan the next view and reduce the number of scans needed to recover the object. Results are presented for the creation of 3D models of a computer game controller, a hip joint prosthesis, and a mechanical strut.

1. Introduction

Recently there has been much research addressing the problem of automatically creating models from range images. Solving this problem has major implication in the application areas of reverse engineering, virtual reality, and 3D Fax. A number of interesting research problems need to be studied in order to make these systems more functional. Our research and the results described in this paper focus on the following problem areas:

- •Model Fidelity: It is important to recover the correct geometry of an object whose shape is *apriori* unknown.
- •Topological Correctness: Model building systems should create topologically correct models, without holes and surface inconsistencies.
- •Planning the next view: Data redundancy should be minimized while at the same time guaranteeing complete coverage of the object. This is important for applications such as 3D Fax.

This paper describes a system that incrementally builds CAD models from multiple range images with these issues in mind. In our method, a mesh surface is created from a range image, which is then extruded in the imaging direction to form a solid. The creation of the extruded solid produces a topologically-correct 3-D CAD model. A key component of the model-building stage is the tagging of surfaces as to their type: either "properly imaged" or due to occlusion artifacts. We are able to use our previous results in sensor planning to create continuous regions of unoccluded viewing space which also can include sensor-specific constraints. 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 work was supported in part by an ONR MURI Grant, DARPA AASERT awards DAAHO4-93-G-0245, DAAHO4-95-1-0492, and NSF grants CDA-96-25374 and IRI-93-11877.

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 [15]. 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 [14]. Methods that use a mesh surface to model and integrate each of a set of range images, such as work by Turk and Levoy [19] or by Rutishauser *et al.* [12], or to model a complete point sampling as by Hoppe [5] or Fua [4] have also proven useful in this task. Both Stenstrom and Connolly [13] and Martin and Aggarwal [9] perform edge detection and projection from intensity images, a concept that is revisited by Laurentini in [8]. Curless and Levoy [3] present a system that resembles ours in that it uses a surface mesh from each range image as a step towards construction of a solid. The mesh is used 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 [7] utilizes changes in the boundary between sensed surface and occlusion with respect to sensor position to recover shape. In Connolly's octree-based work [2], "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 [10] to find the viewing vector that will illuminate the most edge features derived from occluded regions. Whaite and Ferrie [21] use 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. More closely resembling the work presented in this paper is that of Pito [11], which performs the raycasting operation as well, but only at those regions in the sensor's space that are known to image new surfaces.

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. 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. 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. This is a typical configuration in which 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 rotation stage and the laser system are calibrated so that we may align the range images into a common coordinate system.

The point data are used as vertices in a mesh, but 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. The mesh surface $\boldsymbol{M}$ is "swept" to form a solid model $\boldsymbol{S}$ of both the imaged object surfaces and the occluded volume. The algorithm may be stated concisely as:

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

An extrusion operator is applied to each triangular mesh element $\boldsymbol{m}$, orthographically along 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. 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 which of the above sets the surface belongs to. All surface elements in the model whose surface normals form an angle greater than a threshold when compared with the incident sensor beam should be tagged as "imaged surface". This threshold may be found by determining the device's *breakdown angle* empirically, so that "imaged surface" 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 "unimaged surface" so that they may be used to drive a later planning process. As an example of the sweeping and tagging process, consider the mesh shown at the left in Figure 1. The sweeping operation results in the solid shown at the right of the figure, its surfaces tagged according to the process described above.

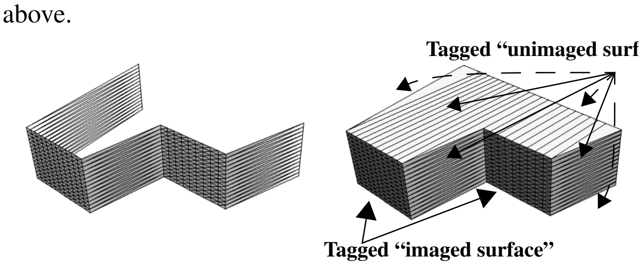


Figure 1. Solid formed by sweeping a mesh (left of figure) in the sensing direction. Tags for hidden surfaces are shown with dotted arcs.

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 [19] [12]. 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

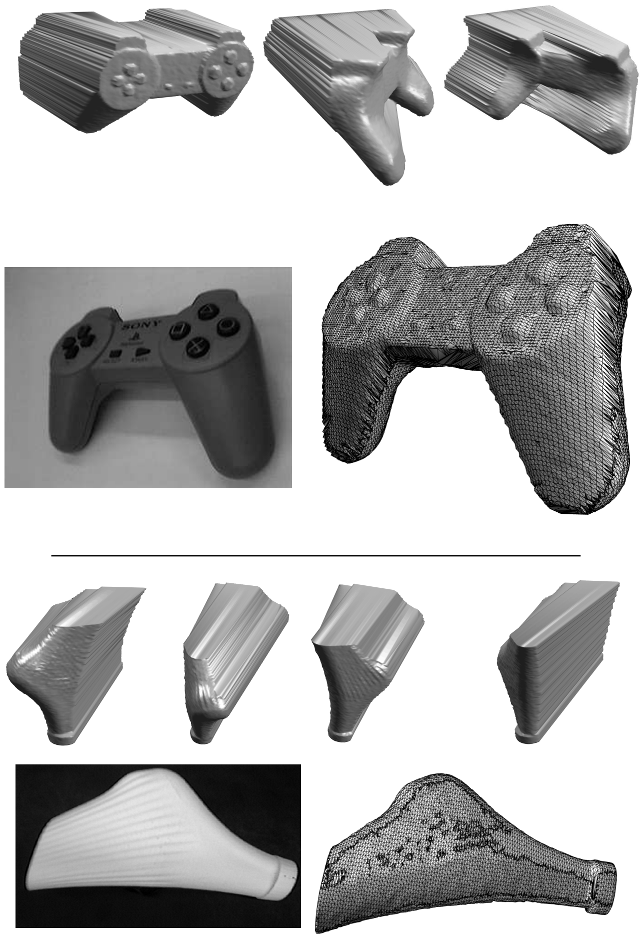


Figure 2. Models for the video game controller and the hip joint prosthesis, showing partial models and the final model next to an image of the actual part.

we generate a solid from each viewpoint which allows us to use a merging method based on set intersection. The merging process 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. To demonstrate the operation of this modeling system, the construction of two models is shown in Figure 2: a controller for a video game and a hip joint prosthesis For each of these, the top row shows the extruded solid models built from each range image (three for the controller, four for the prosthesis), and below them is the final composite model. Using only a small number of scans, the models show large amounts of detail and correctly capture the geometry and topology of each object. Each of these models has been physically built on our rapid prototyping system.

4. The planning process

Occlusion is an important scene attribute useful to the planning process and has previously been used in one of two ways. In the first, a discrete ray casting method 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 [2] [21]. This has 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 [10]. This technique is not sufficient because it does not take into account known self-occlusion of the model's surfaces. 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.

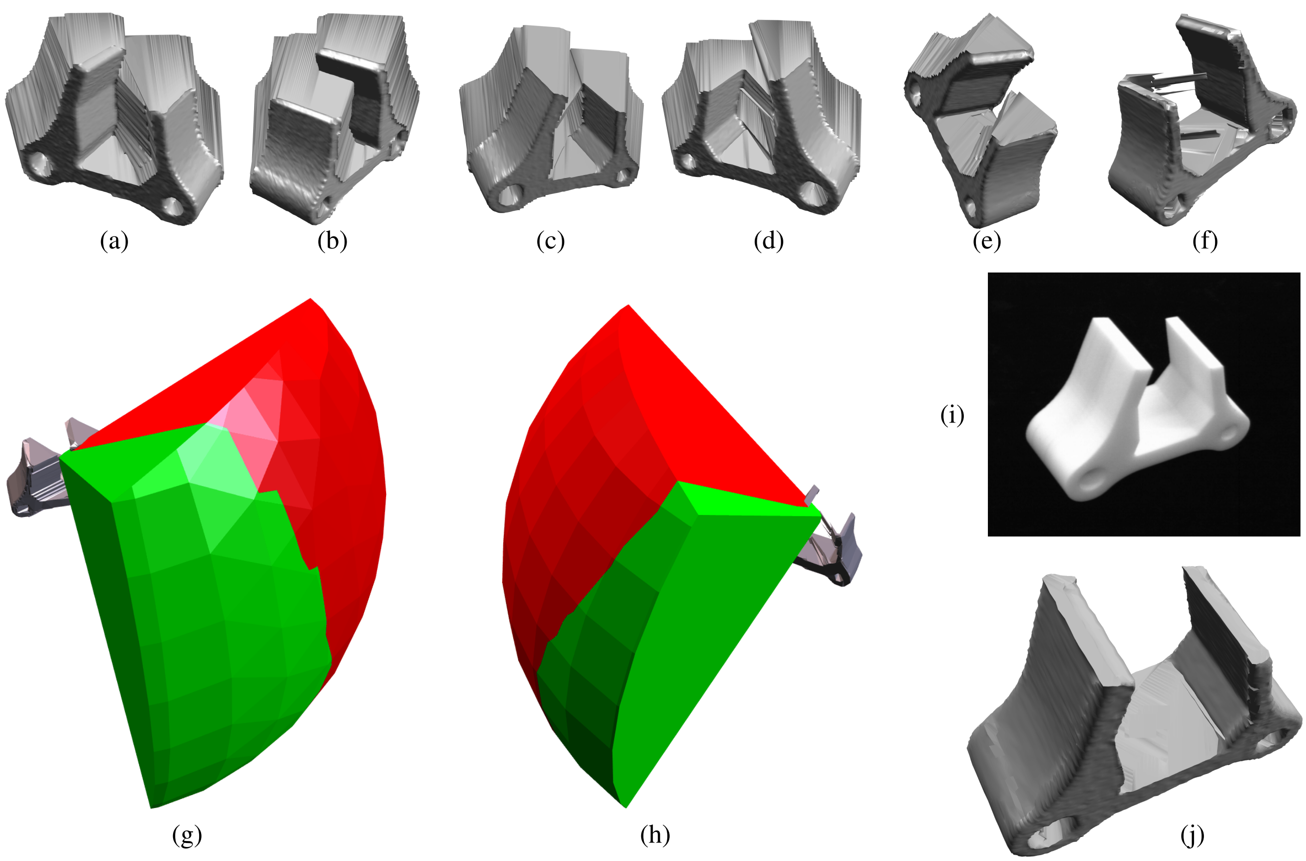
Our planning component is based on previous work on the sensor planning problem [17] [18] and is performed in continuous space. Given a target "unimaged" model surface, the planner constructs a *visibility volume* $\boldsymbol{V}_{\text{target}}$, following [16]. $\boldsymbol{V}_{\text{target}}$ for a target $\boldsymbol{T}$ specifies the set of all sensor positions that have an unoccluded view of the target for a specified model. This can computed in four steps:

1) Compute $\boldsymbol{V}_{\text{unoccluded}}$, the visibility volume for $\boldsymbol{T}$ in the case where there are no occlusions.
2) Compute $\boldsymbol{M}$, the set of occluding model surfaces by including model surface $\boldsymbol{F}$ if $\boldsymbol{F} \cap \boldsymbol{V}_{\text{unoccluded}} \neq \emptyset$.
3) Compute the set $\boldsymbol{O}$ of volumes containing the set of sensor positions occluded from $\boldsymbol{T}$ by each element of $\boldsymbol{M}$.
4) Compute $\boldsymbol{V}_{\text{target}} = \boldsymbol{V}_{\text{unoccluded}} - \bigcup \boldsymbol{o}\ ,\ \forall \boldsymbol{o} \in \boldsymbol{O}$

The volume described by $\boldsymbol{V}_{\text{unoccluded}}$ 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. Each element of $\boldsymbol{O}$ is generated by the decomposition-based occlusion algorithm presented in [16], and describes the set of sensor positions that a single model surface occludes from the target.

It is important to note that this algorithm for determining visibility does not use a sensor model, and in fact part of its attractiveness is that it is sensor-independent. However, for reasons of computational efficiency it makes sense to reduce the number of surfaces in $\boldsymbol{M}$, and therefore the number of surfaces used to calculate $\boldsymbol{O}$. Since $\boldsymbol{M}$ is determined by considering which model surfaces intersect $\boldsymbol{V}_{\text{unoccluded}}$, if $\boldsymbol{V}_{\text{unoccluded}}$ is constrained then in many cases $\boldsymbol{M}$ will be reduced as well. To constrain $\boldsymbol{V}_{\text{unoccluded}}$, we may consider specifics of the sensor, for example that the sensor can only properly image a surface when the surface inclination is within some bounds. If, for example, the sensor's breakdown angle dictates that it must be inclined less than some angle $0 < \theta < 90$ to the surface in order to image it properly, then $\boldsymbol{V}_{\text{unoccluded}}$ may be modeled as a truncated prism, and so the volume $\boldsymbol{V}_{\text{unoccluded}}$, as well as $\boldsymbol{M}$, is reduced in size. In a situation where there is a single target face, all that remains is to compute a transform that will bring the sensor into the visibility volume for the target, and then repeat the model acquisition process.

As an example of such a system, a CAD model is built from distinct views of the object shown in Figure 3(i), 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. A simple sensor model is used that assumes the sensor has six degrees of freedom and an infinite field of view, although it is possible to add more specific constraints as we will show below. This part has both curved and polygonal surfaces, and includes holes that are very difficult to image. Figure 3(a) and 3(b) show two models that were automatically acquired from range images with a turntable rotation of 90° between them. Figure 3(e) shows the integration of these into a composite model. We have manually designated a target on this composite model from those tagged "unimaged surface". Figure 3(g) shows the visibility volume for this target assuming a spherical sensor positioning geometry and a sensor grazing angle of about 45°. This volume is shown with a light and a dark region: the entire volume represents $\boldsymbol{V}_{\text{unoccluded}}$ for the target, while the light region represents $\boldsymbol{V}_{\text{target}}$ and the dark region represents the total occlusion due to model surfaces, i.e $\bigcup \boldsymbol{o}$. This plan is executed by rotating the turntable to place the sensor within the visibility volume, in this case an additional 83°, from which the model in Figure 3(c) is acquired. The composite model at this point is shown in Figure 3(f). Again, a target is designated on the composite model and a plan produced, which is shown in Figure 3(h). The turntable is rotated 134° to move the sensor into this visibility volume, and another model is acquired, shown in Figure 3(d). The final composite model is shown rendered in Figure 3(j), under the image of the actual part.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 3. Strut part reconstruction: (a-d) models created from four distinct range scans. (e) composite model found by intersecting (a) and (b). (f) composite models found by intersecting (c) and (e). (g) visibility volume computed from target on composite model (e). (h) visibility volume computed from target on composite mode (f). (i) image of actual strut part. (j) final reconstruction computed by intersection of (f) and (d).

5. Improving the planning process

The previous example used a manual target selection to find the visibility volume for an unimaged surface, as well as a simplified sensor model. We now discuss an approach to planning the next view that will automate the viewpoint selection process and will also include a stronger set of sensor constraints. The best next viewpoint is the one that will image the most "unimaged" surface elements in the current composite model. We may determine an appropriate next view as follows:

1) Compute ${}^{i}\boldsymbol{V}_{\text{target}}$ for each "unimaged" surface i, keeping track of the target's surface area.
2) Intersect each ${}^{i}\boldsymbol{V}_{\text{target}}$ with the sensor's reachable space.
3) Search this intersection for a point that will image the most surface area associated with each ${}^{i}\boldsymbol{V}_{\text{target}}$.

The first step is the algorithm described in Section 4. Once the visibility volume is computed, the second step determines valid sensor positions via an intersection operation with a surface or volume representing the set of all possible sensor positions. If the sensor can be positioned arbitrarily, then the sensing space is a volume. If the sensor is constrained to a fix offset, as in many laser scanning systems, then the sensing space is a surface. In either case, because we are building these visibility and sensing objects as CAD primitives, we can easily find continuous intersections between them. The result is a decomposition of sensor space into volumes or surfaces from which zero, one, or more target areas are visible. The third step may be accomplished by a variety of methods. Typically, this type of problem is solved by discrete sampling of sensor space, with accumulation of the target area for each containing $\boldsymbol{V}_{\text{target}}$ at each sample point [11].

We show some results in Figure 4 of this algorithm for the strut part. In this case the plans use the model of the strut after two views, and so the model is that shown in Figure 3(e). For reasons of computational efficiency, the model is first decimated to reduce the later computation using a modified version of the fast and robust Simplification Envelopes algorithm [1]. Figure 4(a) and (b) show the results of visibility planning for imaging the largest "unimaged" surfaces in the decimated model. Figure 4(a) shows the visibility volumes, i.e. all ${}^{i}\boldsymbol{V}_{\text{target}}$, placing no restriction on the sensor's ability to orient itself. These volumes are shown truncated for clarity, but actually extend to the limits of the modeled space. A point that is interior to any of these volumes is able to entirely image the corresponding model surface. If a point is not interior to any such volume, none of the planned-for model surfaces is fully visible for that point. Likewise, a point that is interior to more than one vis-

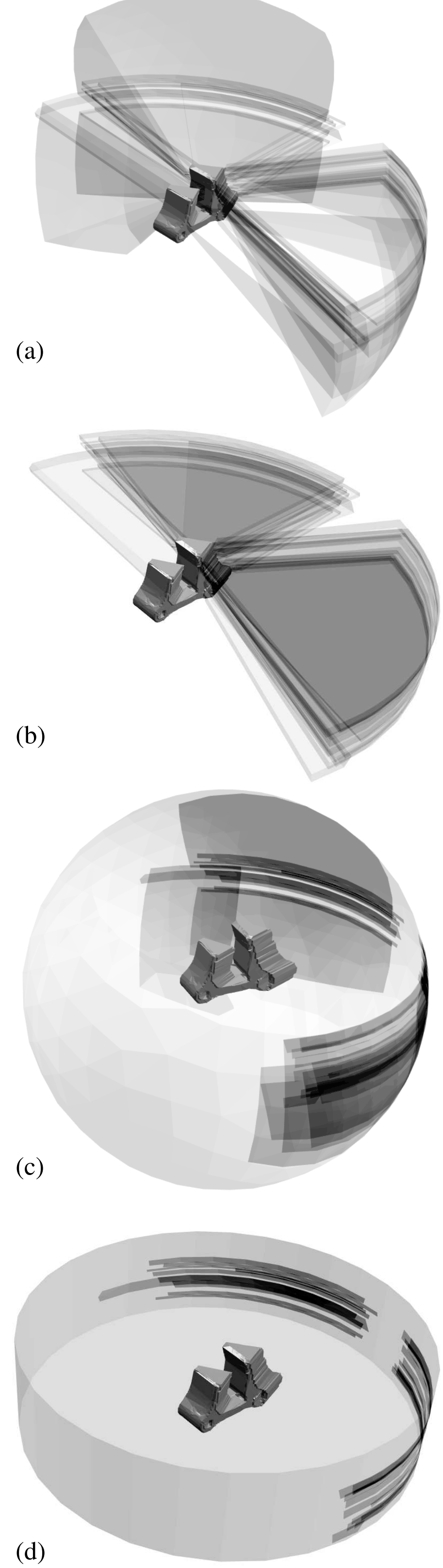


Figure 4. Results of visibility planning.

ibility volume has each of the corresponding surfaces in full visibility: these overlapping volumes show up in the figure as darker regions. Figure 4(b) shows the effect of including a sensor constraint in the planning process. In this case the constraint is that in the sensor operates orthographically in the z direction ("up" in the examples). This is typical of imaging systems such as ours where a light-striping rangefinder moves perpendicular to the striping plane and a turntable controls rotation of the part. This constraint is integrated during the occlusion planning phase by altering the shape of $\boldsymbol{V}_{\text{unoccluded}}$ that is generated for each target.

Figure 4(c) and (d) show the results of intersecting the visibility volumes with sensor space representations. Figure 4(c) is the intersection of a spherical sensing surface with the visibility volumes in Figure 4(a). Figure 4(d) shows the intersection of a cylindrical sensing surface with the visibility volumes from Figure 4(b). Given these representations, we can now search for regions of maximum visibility. These regions appear as darker regions in Figure 4(c) and (d), and signifies where the visibility volumes from more than one target intersect the sensor space representation. Choosing the next imaging position can be done by sampling this representation as described above. The visibility planning has, thus far, been in continuous space, allowing the possibility of a continuous-space solution rather than a discrete sampling. This is useful in situations where high accuracy is necessary for sensor placement, which can cause problems for discrete methods.

6. Conclusion

We have presented a system that creates CAD models from multiple range images. The method incrementally builds models that are solids at each step of the process. It requires a calibrated sensor in order to align the individual models. We have developed a method to reduce the number of scans by using a sensor planner that is able to reason about occlusion and sensor constraints. The result is the computation of a new viewpoint from which to create a new model to be merged with the composite model. We believe this method has promise for building high-fidelity models as well being able to build approximate models from a very small number of scans.There are a number of problems that still need to be addressed in this research. In the model-building phase, these are sensor resolution artifacts at occlusion edges in the scene, which can cause problems when using set intersection methods. In the planning phase, there are granularity issues for the size of the surface elements, and refining the computation of the optimal viewpoint.

7. References

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

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

[3] B. Curless and M. Levoy. *A Volumetric Method for Building Complex Models from Range Images*. *SIGGRAPH*, p. 303-312, 1996.

[4] P. Fua and P. Sander. Reconstructing Surfaces from Unorganized Points. In *Proc. DARPA IUW,* 1992.

[5] H. Hoppe. *Surface Reconstruction from Unorganized Points*. Ph.D. thesis, CS Dept, U. of Washington, 1994.

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

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

[8] A. Laurentini. Inferring the Shape of the Real Object from the Object Reconstructed by Volume Intersection. In *Proc. CVPR*, p. 280-285, 1993.

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

[10] 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.

[11] 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.

[12] 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.

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

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

[15] 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.

[16] 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.

[17] K. Tarabanis, R. Tsai, P.K. Allen. The MVP sensor planning system for robotic tasks. In *IEEE Trans. on Robotics and Automation*, 11(1), p. 72-85, February 1995.

[18] 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), p. 86-104, February 1995.

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

[20] 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.

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