Automated Model Acquisition using Volumes of Occlusion

Michael Reed and Peter K. Allen

Center for Research in Intelligent Systems
Department of Computer Science
Columbia University, New York, NY 10027 *

Abstract

Two primary requirements of any system that relies on active vision to perform modeling from observation are that it be able to use previously acquired data to drive the sensing process and that it can iteratively incorporate newly sensed data. This paper discusses an approach to automating CAD model acquisition by allowing the system to keep track of what parts of the sensed object have yet to be imaged. This is achieved by explicitly representing the volume of occlusion as well as its surfaces of the object obtained from any one sensing operation.

Models built from distinct views of the object, and which include their volume of occlusion, are merged using set operations. The resulting composite model consists of the visible surfaces from each model along with the intersection of the volumes of occlusion and contains the necessary information for planning the next viewpoint.

1 Introduction

Recently much attention has been given to work progressing in the field of *modeling from observation.* The combination of increased research efforts and the arrival of working products based on prior results has done much to drive interest in this field, leading to systems that are able to handle a wider variety of shapes, materials, and scales of objects. One aspect of particular interest is that of automating the process with the eventual goal being a system to which the user presents an object and a set of desired tolerances and from which is received a CAD model of the object within the specified limits.

In this paper we describe an approach to automated model acquisition that combines work in sensor planning, range data acquisition, segmentation and polyhedral model construction. There have been a number of previous efforts in automating the model generation process. In the past reconstruction was often used as the front end to an object recognition system and so was concerned more with determining discriminating features than with building a precise model. Recent work has reached the point where the resulting model is precise enough to be used in manufacturing processes [Tarbox and Gottshlich1995] [Sobh *et al.*1995], or for display in virtual reality environments [Shum *et al.*1994].

*This work was supported in part by DARPA contract DACA-76-92-C-0007, NSF grants CDA-90-24735 and IRI-93-11877, and an ONR MURI Grant

Many of the efforts have focused on creating specific surface or volume modeling primitives. Some representative examples include generalized cylinders [Nevatia and Binford1977], superquadrics [Solina1987], bicubic patches [Potmesil1982], and integrated point sets [Chen and Medioni1991]. A number of recent efforts have attempted to generate Boundary Representations (B-reps) from multiple views, including [Parvin and Medioni1992, Hoover *et al.*1994, Stenstrom and Connolly1992] and our work is in this vein. The B-rep is used by most CAD systems and can efficiently encode the geometry and topology of a wide range of objects.

As CAD models become more central to parts design and manufacture, the ability to automatically generate these models from existing objects becomes paramount. There are still parts which are best designed using the tools of model makers, in materials such as clay or wood. It has been said that everyone would be using CAD systems if they were "as comfortable and easy to use as foam, clay, and pine." [Wohlers1994]. As long as this state of affairs continues, there will be parts for which there are no CAD data. Without CAD data, it is not possible to use rapid prototyping systems to produce additional models, nor is it possible to benefit from any of the advanced analysis, manufacturing, and process planning capabilities of today's CAD/CAM systems. Applications in which 3-D solid or surface data must be acquired from physical models or prototypes include [Diamond and Kreplin1994] model makers in the automotive industry ,Consumer goods manufacturers, Inspection and quality assurance of parts, and reverse engineering.

Much prior work in sensor planning addresses the problem of determining sensing strategies for the purpose of scene reconstruction [Maver and Bajcsy1990] [Xie *et al.*1986] [Connolly1985] [Shmuel and Werman1990] [Whaite and Ferrie1990]. In these systems a model of the scene is incrementally built by successively sensing the unknown world from effective sensor configurations using the information acquired about the world to this point. At each step of the algorithm, new sensor configurations are chosen based on a particular criterion (e.g. choose the sensory configuration that will explore the largest

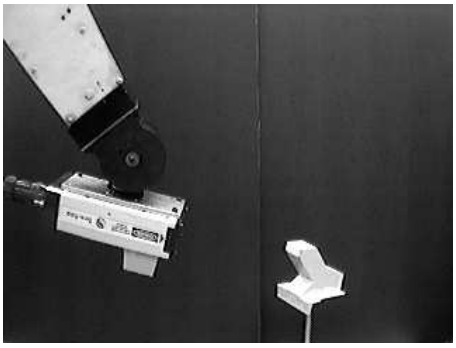

Figure 1: Experimental setup showing robot with attached laser rangefinder scanning a part.

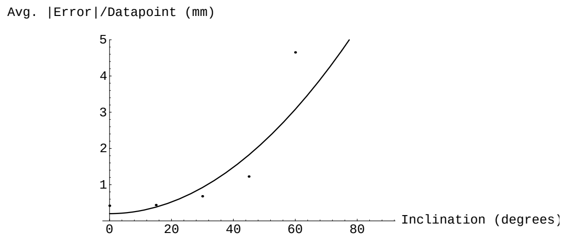

Figure 2: How the error per data point varies with the inclination of the imaged surface. Each plotted point is the average error over 10,000 samples.

area of unknown space). The sensory information acquired at each step is then integrated into a partially constructed model of the scene and new sensor configurations are generated until the entire scene has been explored.

This paper is organized as follows: First, we give an overview of our approach. Second, we discuss the low-level processing applied to range imagery in preparation for building a model of a single view. Third , we motivate our use of BSP trees as an intermediate representation for the model acquired from each viewpoint and discuss how this model is merged with previously acquired data. Lastly we show some initial experimental results and discuss how we may plan the next view.

2 Overview

Our approach consists of a series of steps which are repeated until a satisfactory model is built. Generation of a model from a single image is the basic step in the process of building a complete model of an object. It involves acquiring a single range image by moving the laser scanner to a specific position/orientation and then entering a loop where the laser is commanded to scan a line, then the robot moves the laser in its image plane an increment, and then repeats until the entire image has been acquired (see figure 1). The image is then filtered in some way to remove spike noise and smooth the data somewhat. A segmentation phase then splits the data into regions of data points that are on the same geometric entity. Surfaces are then fit to these entities, and entered into cyclic graph that keeps track of their topological attributes. From this graph a model is built that includes not just the imaged surfaces but also the volume of occlusion.

Each time a single-image model is generated it is added to the composite model using boolean operations. Before being intersected with the composite model, a transformation must be applied that describes the difference in the viewing positions of the rangefinder. In our current system, these transformations are known *a priori*. However, this part of the process may be automated by using the volumes of occlusion and the error associated with the scanned surfaces.

3 Image preprocessing and segmentation

In order to build as complete a model as possible the object must be imaged from different positions and these views merged in some way. The process of scanning an object from one position and generating incomplete model information consists of several phases, taking the data from the scanning (or imaging) phase through the segmentation phase. Here we discuss our major concerns in each of these phases. Later we discuss how views are merged and give examples of the segmentation and merging processes.

Scanning is by no means a perfect process. Laser rangefinders have noise that is affected by the surface material, color, and geometry. Further, the noise is not normally distributed, and is therefore difficult to remove [Besl1988]. Figure 2 shows how the error per data point typically ranges for just one of these contributing factors: the angle of inclination of the surface with respect to the imaging laser. To produce accurate estimates, the data must be smoothed by applying a filter or using an estimation technique. However, any filter that distorts the data, as many do in the vicinity of edges, will have detrimental effects on the later recovery of surfaces near those edges [Besl and Jain1986]. In our work we have minimized the processing of the data before the geometric relationships between the data are analyzed, using one pass of a median filter to remove the spike noise and reduce other anomalies without excessively degrading the data.

Although for simple data such as spheres or cubes it is possible to go directly from the point data to a complete volumetric model [Bajcsy and Solina1987], this method limits the complexity of the model unless combined with other modeling methods such as Constructive Solid Geometry. In addition, the direct methods have a high computational cost when integrating information from multiple views: and entire recalculation of the object's surface is often required after each merge operation. For more complex objects it is necessary to segment the image into a collection of smaller surface patches [Jain and Flynn1993].Most methods rely on some previous knowledge of the object being imaged to improve the performance of the estimation.

Because the data is from polyhedral objects , we have implemented an adaptive planar fit based on that used by [Chen1988] to determine local surface normals. This method uses a combination of region growing and surface normal analysis to facilitate rapid segmentation, while still allowing one to adjust the parameters of the segmentation. These parameters include the minimum area of each segment, the minimum deviation of surface normals between adjacent segments, the maximum error of a point fit to a surface patch and the maximum error of entire surface patch fit.

After segmentation there are pixels that have not been classified as belonging to any surface. In order to classify these pixels, we apply a post-segmentation filtering process [Suk and Bhandarkar1992]. An unclassified pixel that is surrounded by classified pixels may be handled by applying a mask to the image which classifies the pixel according to the majority region in the mask. A set of unclassified pixels in the vicinity of an edge may be split equally between the regions that border them and marked not to be used in computing the geometry of a face; although they may be used to compute a face's boundary, the fact that they were not initially segmented into one of the adjacent surfaces implies that they were noisy or were badly altered in preprocessing. Pixels that were classified into different regions that are actually from the same region can be merged if the regions they belong to have surface equations that are similar.

4 Building single-view models

Constructing a high-level model from the point data acquired during a single scanning process consists of two steps. First, the segmented regions in the range image must be analysed to determine the polygons that will be made to approximate the objects faces visible from that viewpoint. In the second step the volume of occlusion is built to bound the volume that is not determined by that same viewpoint.

4.1 Forming surface polygons from a segmented image

After segmentation, it is necessary to fit polygons to each region. This may be done by traversing the region's boundary to generate vertices, or by using the planes of adjacent regions to compute the intersections. The latter method has the advantage that the calculation of the region's geometry is much more accurate than using image space to calculate the region's vertices. Unfortunately, some of the boundaries between adjacent regions in image space will be step edges, so that the planes for those regions will not intersect at the boundary in the image . This is complicated by the fact that boundaries at roof edges are always a single edge between two regions, while step edge boundaries may be multiple edges. To generate polygons we use a hybrid method that incorporates as much large-scale geometry as possible, while still handling the situations where none is available.

To compute the vertices of a polygon for a region R, we first follow R's contour in the segmented image to

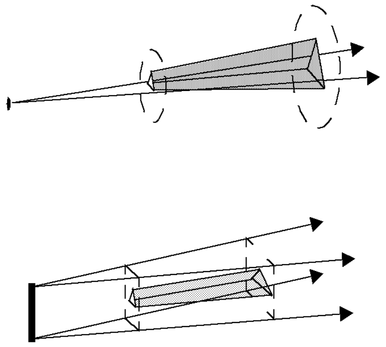

Figure 4: The volume of occlusion for a view of triangular object modeled by the perspective projection umbra (top, as for a point sensor) and by a hybrid orthographic/perspective projection umbra(bottom, as for our sensor).

generate an ordered list of regions that bound R (see figure 3.a, which is detail from the first image in figure 4.a). For each region that bounds R, the pixel where that boundary started and the type of edge (step or roof) are recorded. For step edges, the coordinates of the pixels along the boundary are also recorded, for reasons we will explain.

We then iterate through the list of regions and compute vertices. If the current and next regions bound R by roof edges (as A and B do), then we compute the intersection of their underlying planes and then intersect the resulting line with R's plane. This gives us the vertex at the corner where A and B meet R (figure 3.b). Alternatively, if R's boundary with the current region is a roof edge, but the next is a step edge (as is the situation with regions B and C), we may use pixel P where the regions meet R as the vertex (figure 3.c). We may do this because we know there is a straight edge from the previously found vertex to P, since the boundary between A and R is a roof edge. Lastly, if R's boundary with the current region is a step edge (as with region C) we rely on computing the vertices in image space, using bounded deviation.

The output of this process is an ordered list of the vertices of the polygon that is described by the region R in the segmented image. In all cases the plane of R is used to compute the vertices either by intersection or by computing the Z value of a pixel in the image. This assures us of having all the vertices coplanar in model space.

4.2 Generating the volume of occlusion

It is not enough to generate the polygons for the object's faces that are regions in the segmented range image. By also explicitly generating the faces that complete the volume of occlusion we may retain information that can drive the sensor planning process. The faces that make up the volume of occlusion may also help improve the quality of the final model by filling any gaps due to inaccuracies in the imaging and model-building process.

To generate the volume of occlusion from a viewpoint, we must build the set of faces that project from

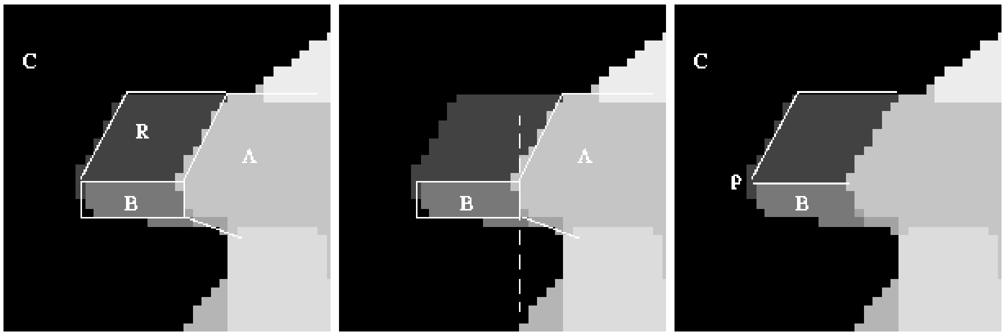

Figure 3: a) Detail of range data highlighting R and its bounding regions A, B and C (detail of first image in figure 4). b) intersecting planes to determine vertex. c) Using vertex in image space on roof edge.

all step edges in the single-view model away from the viewpoint, towards infinity. If the sensor we use can be modeled as a point, the volume of occlusion would be the perspective projection umbra of the object (see figure 4). Due to the fact that our laser scanner acquires data in 2D (a stripe) and that it is moved *linearly* to obtain a range image, the point sensor model does not apply. Instead, we must model the sensor as a perspective projection in 2 dimensions and an orthographic one in the third. Because the sensor is moved linearly in one dimension the data is in this way similar to that used in serial-section reconstruction.

The set of faces that constitute the volume of occlusion include the imaged faces (which are surfaces on the object) and the faces that are made by connecting adjacent surface vertices on step edges along with their projections. Although these faces actually go off to infinity (as does the occlusion volume), we model them as being cut off by some far bounding plane. These faces are also tagged as belonging to the occlusion volume so that we may use them in the planning process.

4.3 Binary space partitioning trees as a modeling primitive

An important aspect of any model acquisition system is the speed with which it can develop a complete model from a set of incomplete views of the object acquired from a sensor. There are a variety of methods used to merge multiple views of an object together, including methods based on Constructive Solid Geometry, volumetric approaches, and surface merging. The desiderata for a representation are that it be effective at modeling the segmented data of each view and be able to *quickly* merge the different views into a single, accurate representation of the data. Such a representation is the BSP tree [Fuchs *et al.*1980, Naylor1981]

A BSP tree is a method by which n dimensional space is partitioned by $n-1$ dimensional hyperplanes. Once a space has been partitioned by a hyperplane, it is represented by two n dimensional half-spaces, one on each side of the partitioning hyperplane. In the 3-D case planes divide space into two half-spaces, one containing the polygons in front of the plane, and one containing the polygons behind the plane. A polyhedral scene may be completely represented if the planar equation of one of the polygons is selected to form the root of the tree, and the front and back half-spaces are recursively partitioned in this manner. We may represent space contained or outside a polyhedra by associating a label at each of the leaves denoting the space represented there as either "in" or "out" [Thibault and Naylor1987].

The BSP tree has unique advantages as an intermediate representation for the different views. Its tree structure allows very efficient algorithms to be developed, it is compact, and it is numerically robust. Of particular interest to us is the speed with which two BSP trees may be merged. This corresponds to the intersection of the two volumes, and is the operation we use to compute the composite model. To obtain the boolean AND of two trees, we first apply a transformation to each node in every tree to bring them into a common coordinate system. These transforms are known from the position and orientation of the laser unit during each scan. It is then necessary to compute which "in" space from the first tree is contained by the "in" space from the second tree, and vice versa. This is easily calculated by filtering the polygons from the first tree through the second tree, splitting them by the hyperplanes at each node in the second tree. Polygons that end up at "in" leaves are contained by both polyhedra. Applying this to both trees and unioning the result produces the space contained by both objects. An analysis of the robustness and complexity of set operations is given in [Naylor *et al.*1990]. To briefly restate here, merging two n-node BSP trees provides a complexity which is worst case optimal $\Theta(n^2)$. However, the expected case for two "good" (i.e. balanced) BSP trees is $\mathcal{O}(n \log n)$.

4.4 Experimental results

We proceed with an example of building a composite model from three distinct views. The real range images presented here were acquired using a Servo-Robot, Inc. laser rangefinder unit attached to a Puma 500 robot arm. The laser uses an oscillating mirror to acquire scan lines in the unit's X-Z plane; the Puma moves the unit in its X-Y plane after each scan has been taken. Three range images of the part in figure 5 are shown in figure 6. Since the object to be modeled is on a stand or hung from a support while

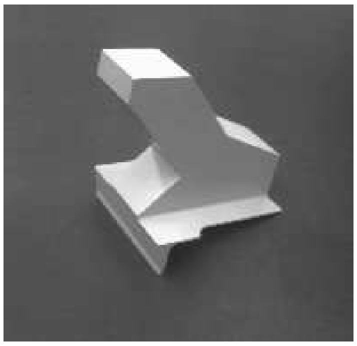

Figure 5: A machined part.

it is being imaged some distant objects often make up the background of the image. The background is either a flat plane (if the object is in front of a wall) or a jumble of miscellaneous shapes. In either case, since we want to deal with the range data of the object exclusively we strip out the background data by examining the histogram of the range image. It is assumed that the object of interest is closer to the rangefinder than background planes or clutter, so we may effectively employ a threshold to remove everything but the object. A median filter is then applied to remove spike noise and smooth the data, and each image is segmented. The segmented images of the real data are shown in figure 7. We then build the single image model for each viewpoint whcih includes the surfaces and the volume of occlusion, shown in figure 8. These models are rendered from different viewpoints than thier range image was acquired so as to show the occlusion volume.The final composite model of the object is shown in 9.

5 Future work and conclusions

Future work on this system is towards the goal of being able to construct CAD models without operator intervention. Our current system still requires the operator to input the transformations for the viewpoints ahead of time. While this may be acceptable for some cases, as the objects get more complicated it will become increasingly important to have the views generated automatically. While there are systems that can generate viewpoints given features on the model, what is desired is a way to assess the current state of the model and use that information to plan the next viewpoint.

The information necessary to achieve the viewpoint planning is contained in our composite model. There are two primary sources of planning information which are associated with the polygons (either surface or from a volume of occlusion) in the composite model. One is a flag which describes if the polygon is from a face or from a volume of occlusion, and the other is an error term that represents the error of fitting the planar surface to its range data if that polygon is from a model surface. Polygons that are from a volume of occlusion have error of this type equal to zero.

The calculation of the next viewpoint may be driven by first building as complete a model as possible, and second by making the model conform to an error metric. In the first case we must continue the imaging process until the composite model has no occluded regions. This may be done by classifying the polygons that represent occlusion into two types. The first are those that unbounded, that is, represent a face of the volume that would continue off to infinity from the model. In a single-image model all the polygons that make up the volume of occlusion are this type. The second are those that are bounded due to the intersection of two models, and hence are part of a composite model. For polygons of the first type, we may plan the next viewpoint by choosing a viewing direction that is perpendicular to the surface of the polygon, causing the next compositing operation to bound that polygon. Following [Maver and Bajcsy1990], a histogram of the surface normals weighted by polygon area should be built so that the direction that bounds the most volume of occlusion will be chosen. For those of the second type, we wish to choose a viewing direction opposite the surface normal and a position so that the polygon will be imaged as fully as possible. However, since convexities will exist on most objects, the polygon may be blocked by self-occlusion by another part of the object. In these cases it would be useful to use the results from sensor planning research [Tarabanis *et al.*1991, Tarabanis *et al.*1994, Tarabanis *et al.*1995] to compute a strategy for scanning.

References

[Bajcsy and Solina1987] Ruzena Bajcsy and Franc Solina. Three dimensional object representation revisited. In *Proceedings International Conference on Computer Vision*, London, June 1987.

[Besl and Jain1986] P. J. Besl and R. C. Jain. Invariant surface characteristics for 3D object. *Computer Vision, Graphics, and Image Processing*, 33:33–80, 1986.

[Besl1988] P. Besl. *Surfaces in Range Image Understanding*. Springer-Verlag, 1988.

[Chen and Medioni1991] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proceedings 1991 IEEE International Conference on Robotics and Automation*, pages 2724–2729, April 1991.

[Chen1988] C. H. Chen. *3D-Poly: A Robot Vision System for Recognizing Objects in Occluded Environments*. PhD thesis, School of Electrical Engineering, Purdue University, December 1988.

[Connolly1985] C. Connolly. The determination of next best views. In *Proceedings 1985 IEEE International Conference on Robotics and Automation*, pages 432–435, 1985.

[Diamond and Kreplin1994] Amit Diamond and Terry Kreplin. 3D laser digitizing for reverse engineering, moldmaking, quality assurance, and rapid prototyping purposes. In *Rapid Prototyping and Manufacturing '94*, Dearborn, Michigan, April 1994. Society of Manufacturing Engineers and the Rapid Prototyping Association. (in Additional Papers Addendum).

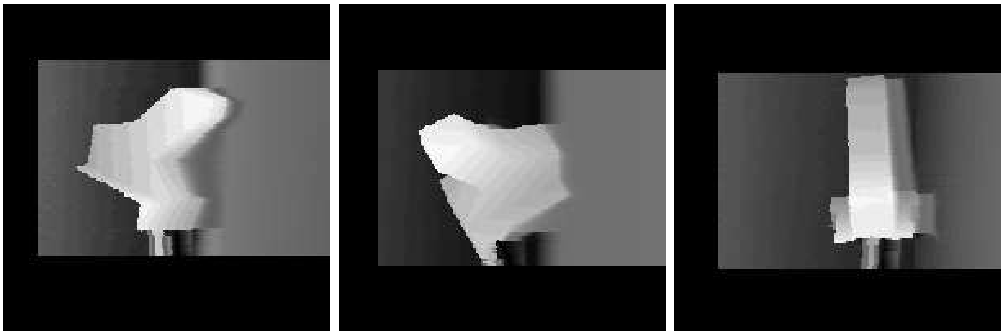

Figure 6: Real range data of part taken from three different views.

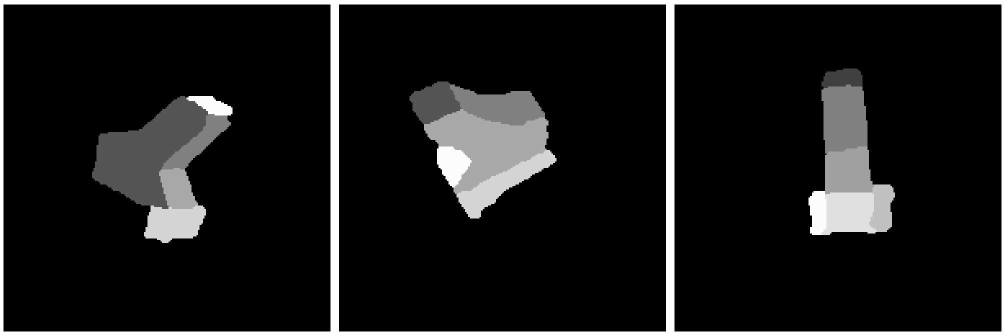

Figure 7: Segmentations of the real range data.

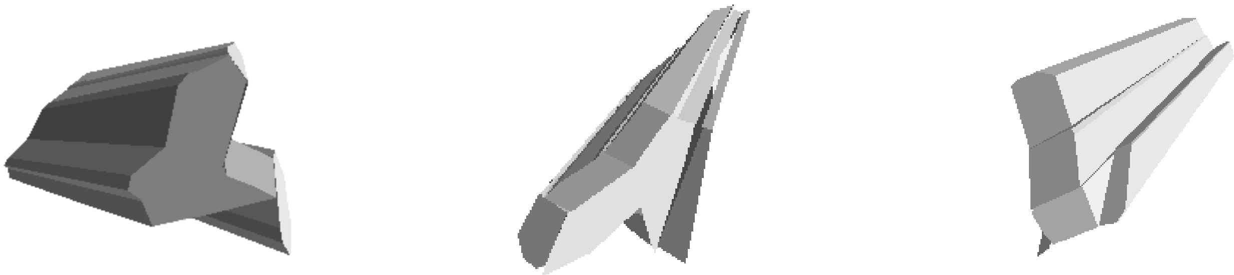

Figure 8: BSP tree models of the segmentations.

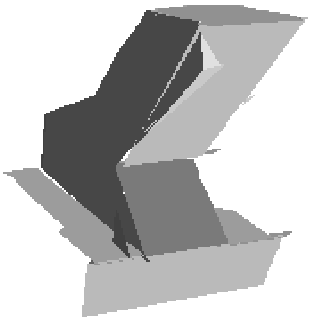

Figure 9: Final model after merging BSP trees.

[Fuchs *et al.*1980] H. Fuchs, Z. Kedem, and B. Naylor. On visible surface generation by a priori tree structures. *Computer Graphics*, 14(3):124–133, June 1980.

[Hoover *et al.*1994] Adam Hoover, Dmitry Goldgof, and Kevin Bowyer. Building a b-rep from a segmented range image. In *Proceedings of the 1994 Second CAD-Based Vision Workshop*, pages 74–81, Champion, PA, February 1994.

[Jain and Flynn1993] Anil K. Jain and Patrick J. Flynn, editors. *Three-Dimensional Object Recognition Systems.* Elsevier, Amsterdam, 1993.

[Maver and Bajcsy1990] J. Maver and R. Bajcsy. How to decide from the first view where to look next. In *Proceedings 1990 DARPA Image Understanding Workshop*, pages 482–496, 1990.

[Naylor *et al.*1990] B. Naylor, J. Amantides, and W. Thibault. Merging BSP trees yields polyhedral set operations. *Computer Graphics*, 24(4):115–124, August 1990.

[Naylor1981] Bruce F. Naylor. *A Priori Based Techniques for Determining Visibility Priority for 3-D Scenes.* PhD thesis, University of Texas at Dallas, May 1981.

[Nevatia and Binford1977] R. Nevatia and T. Binford. Description and recognition of curved objects. *Artificial Intelligence*, 8:77–98, 1977.

[Parvin and Medioni1992] B. Parvin and G. Medioni. B-rep from unregistered multiple range images. In *IEEE International Conference on Robotics & Automation*, pages 1602–1607, Nice, 1992.

[Potmesil1982] Michael Potmesil. Generating three dimensional surface models of solid objects from multiple projections. Technical Report 33, Image Processing Laboratory, RPI, October 1982.

[Shmuel and Werman1990] A. Shmuel and M. Werman. Active vision: 3d from and image sequence. In *Proceedings of the 10th International Conference on Pattern Recognition*, pages 48–54, June 16-21 1990.

[Shum *et al.*1994] H. Shum, K. Ikeuchi, and R. Reddy. Virtual reality modeling from a sequence of range images. In *Proceedings 1994 ARPA Image Understanding Workshop*, pages 1189–1198, November 1994.

[Sobh *et al.*1995] T. M. Sobh, J. Owen, C. Jaynes, M. Dekhil, and T.C. Henderson. Industrial inspection and reverse engineering. *Computer Vision and Image Understanding*, 61(3):468–474, may 1995.

[Solina1987] Franc Solina. *Shape recovery and segmentation with deformable part models.* PhD thesis, Department of Computer Science, University of Pennsylvania, December 1987.

[Stenstrom and Connolly1992] J. R. Stenstrom and C. I. Connolly. Constructing object models from multiple images. *Int. Journal of Computer Vision*, 9(3):185–212, 1992.

[Suk and Bhandarkar1992] Minsoo Suk and Suchendra M. Bhandarkar. *Three-Dimensional Object Recognition from Range Images.* Computer Science Workbench. Springer-Verlag, New York, 1992.

[Tarabanis *et al.*1991] K. Tarabanis, Roger Tsai, and Peter K. Allen. Automated sensor planning for robotic vision tasks. In *IEEE International Conference on Robotics and Automation*, Sacramento, April 9–11 1991.

[Tarabanis *et al.*1994] K. Tarabanis, Roger Tsai, and Peter K. Allen. Analytical characterization of the feature detectability constraints of resolution, focus and field-of-view for vision sensor planning. *Computer Vision, Graphics, and Image Processing*, 59(3):340–358, May 1994.

[Tarabanis *et al.*1995] K. Tarabanis, R. Y. Tsai, and A. Kaul. Computing occlusion-free viewpoints. *IEEE Transactions Pattern Analysis and Machine Intelligence*, to appear 1995.

[Tarbox and Gottshlich1995] G.H. Tarbox and S.N. Gottshlich. Ivis: An integrated volumetric inspection system. *Computer Vision and Image Understanding*, 61(3):430–444, may 1995.

[Thibault and Naylor1987] W. Thibault and B. Naylor. Set operations on polyhedra using binary space partitioning trees. *Computer Graphics*, 21(4), July 1987.

[Whaite and Ferrie1990] P. Whaite and F. Ferrie. From uncertainty to visual exploration. In *Proceedings of the International Conference on Computer Vision*, pages 690–697, 1990.

[Wohlers1994] Terry Wohlers. State of the industry. In *Rapid Prototyping and Manufacturing '94*, Dearborn, Michigan, April 1994. Society of Manufacturing Engineers and the Rapid Prototyping Association.

[Xie *et al.*1986] S. Xie, T.W. Calvert, and B.K. Bhattacharaya. Planning views for the incremental construction of body models. In *Proceedings of the 8th International Conference on Pattern Recognition*, pages 154–157, 1986.