

CAD Model Acquisition using BSP Trees

Michael Reed, Peter K. Allen, and Steven Abrams

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

Abstract

This paper discusses an approach to automating CAD model acquisition by creating Binary Space Partitioning (BSP) trees from laser range data. Distinct views of the object, each represented by a BSP tree, are then merged using set operations. BSP trees have proven their utility in 3-D modeling, graphics and image processing, and their tree structure allows efficient algorithms to be developed that are compact and numerically robust. These properties are of primary importance when considering an intermediate representation between raw sensor data and existing CAD models.

1 Introduction

Automating the acquisition of CAD surface or solid models from laser scan data has been identified as one of the major goals in the field of computer vision. 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." [Wohlens1994]. 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 carve clay models directly from conceptual sketches. CAD models of the clay parts, ranging from a mirror housing to entire car bodies, must be derived from physical models which undergo alterations during wind tunnel or other testing.

- Consumer goods manufacturers hand sculpt models and experiment with different shapes and ergonomic designs, making adjustments and corrections as they go.
- Inspection and quality assurance of parts can be accomplished by digitizing samples and comparing the part shapes against CAD data.
- The area of "reverse engineering" is in need of a fast, simple and effective means of reproducing existing parts quickly. Major advances in the field of rapid prototyping are predicated upon having accurate CAD models of objects to be built.
- Even though an initial object design may be described via a CAD model, during the manufacturing process many changes are made at the physical level that need to be propagated back to the CAD model.

In this paper we describe an approach to automated model acquisition that combines work in range data acquisition, segmentation and polyhedral model construction to address some of the problems cited above. There have been a number of previous efforts in automating the model generation process. Many of these 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.

What this paper motivates is the use of BSP trees as an intermediate data structure that can easily be derived from low-level scanned range data, and from which multiple views can be efficiently merged into a single B-rep description from which a CAD model may be derived. The BSP tree represents volumes by partitioning space with planes, and therefore is limited in that it may only represent polyhedra. It does, however, have other attributes which make it an attractive primitive for modeling 3-D objects, as we shall discuss.

*This work was supported in part by DARPA contract DACA-76-92-C-0007, NSF grants CDA-90-24735, CDA-90-22468 and IRI-93-11877, North American Philips Laboratories, Siemens Corporation and Rockwell International.

This paper is organized as follows: First, we describe our approach to segmenting range images into planar faces. Second, we motivate our use of BSP trees as an intermediate representation for each scan. The planar faces can be used as input to an algorithm that produces a BSP tree for an arbitrary view. Here we also describe a merge algorithm based on BSP tree set operations that allow BSP trees from multiple views to be quickly merged. Lastly we show some initial experimental results using both synthetic and real range data.

2 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]. 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], 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 (a requirement of BSP tree models), we have implemented an adaptive planar fit. This 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 will still be pixels that have not been classified as belonging to a surface. In order to classify these pixels, we apply a process in the manner of [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.

2.1 Forming 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 generate an ordered list of regions that bound R (see figure 1.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 1.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 1.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

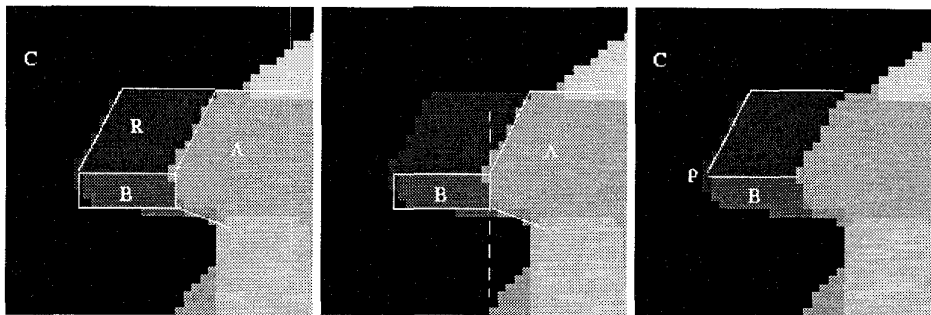


Figure 1: a) Detail of range data highlighting R and its bounding regions A, B and C (detail of first image in figure 4). b) intersecting planes to determine vertex. c) Using vertex in image space on roof edge.

model space.

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. 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

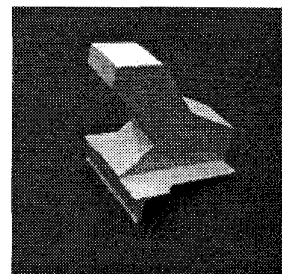


Figure 2: A machined part.

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 $\theta(n \log n)$.

4 Experimental results and conclusions

We have implemented our method and tested it on both synthetic and real polyhedral imagery. First, we generated synthetic range images by simulating range scans of the part shown in figure 2 from a set of hand-chosen views. The output of these scans is shown in figure 3. Each scan is then segmented (figure 4) using the algorithm described above, and converted to polygons which are inserted into a BSP tree. The result is a set of BSP trees that encode the topology and geometry of the faces in each view. In figure 5 we show renderings of the BSP tree representations of the distinct views. Lastly we merge the trees into a single BSP tree that encodes the objects volume. Figure 6 shows the rendered BSP tree of the complete volume. It is a simple matter to then traverse this tree and output the normal B-rep (face, edge, vertex) relationships needed to create a typical CAD model of an object.

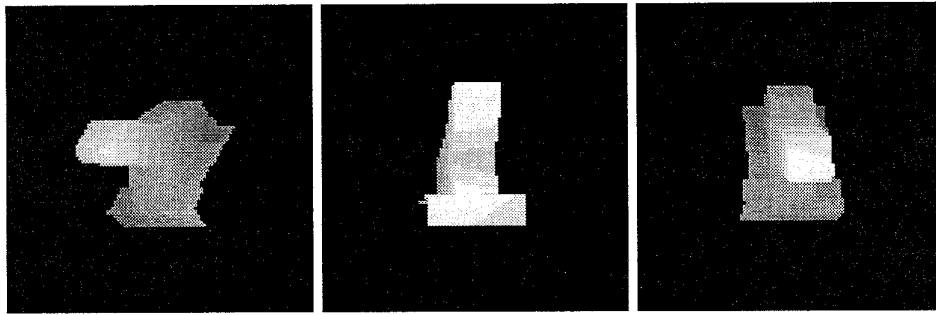


Figure 3: Synthetic range data of part taken from three different views.

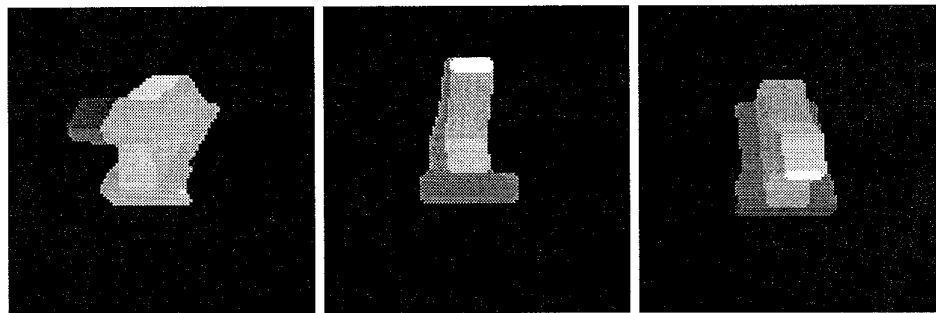


Figure 4: Segmentations of the range data in figure 3.

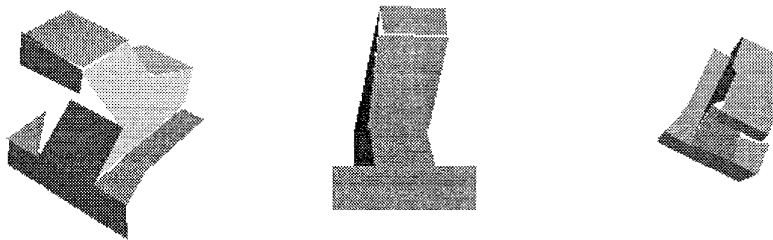


Figure 5: BSP tree models of the segmentations in figure 4, from slightly different angles.

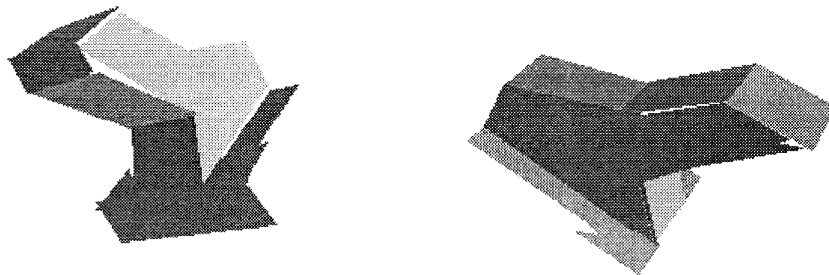


Figure 6: Final model after merging BSP trees.

The real range images were acquired using a Servo-Robot, Inc. laser rangefinder unit attached to a Puma 500 robot. 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 images of the part in figure 2 are shown in figure 7. Since the object is on a stand or hung from a support while 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. This is quite easily done 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 8. As in the case of the synthetic data, we then build the BSP tree for each image, shown in figure 9, and merge them to get the final BSP tree of the object 10.

An eventual goal of this system is to be able to construct CAD models without operator intervention. One approach would involve integrating a system for viewpoint planning [Tarabanis *et al.*1991] to compute a strategy for scanning. This could be done off-line in the case of a part inspection task, or during the scanning process by using the acquired data to maximize new information, reduce the total number of scans, or reduce occlusion.

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 and multiple range images. In *IEEE International Conference on Robotics and Automation*, pages 2724-2729, Sacramento, April 9-11 1991.
- [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. Soc. of Man. Eng. and the Rapid Prototyping Assoc.
- [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.
- [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.
- [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.
- [Thibault and Naylor1987] W. Thibault and B. Naylor. Set operations on polyhedra using binary space partitioning trees. *Computer Graphics*, 21(4), July 1987.
- [Wohlert1994] Terry Wohlert. State of the industry. In *Rapid Prototyping and Manufacturing '94*, Dearborn, Michigan, April 1994. Society of Manufacturing Engineers and the Rapid Prototyping Association.

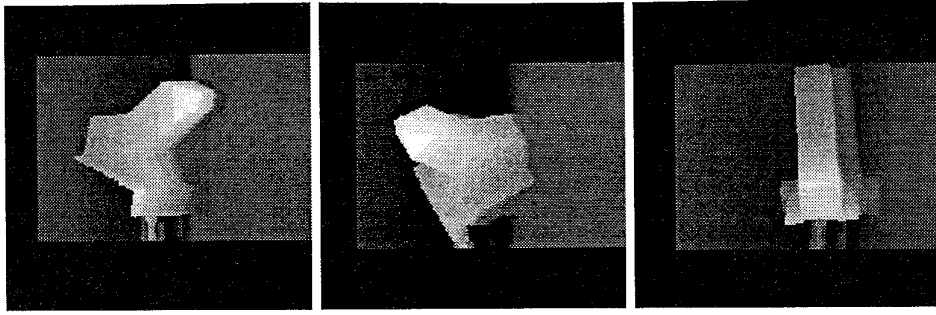


Figure 7: Real range data of part taken from three different views.

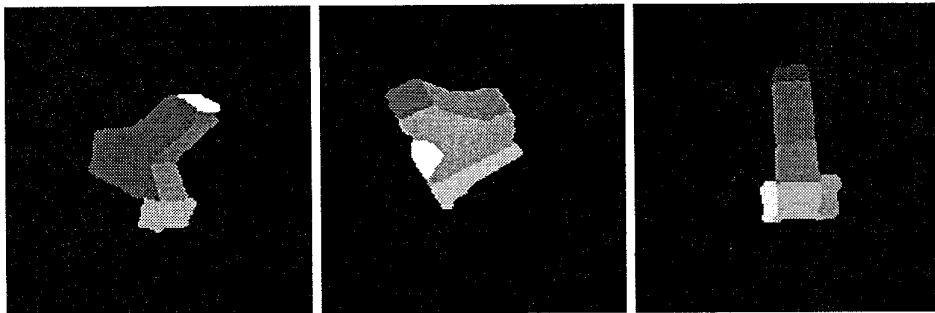


Figure 8: Segmentations of the real range data.

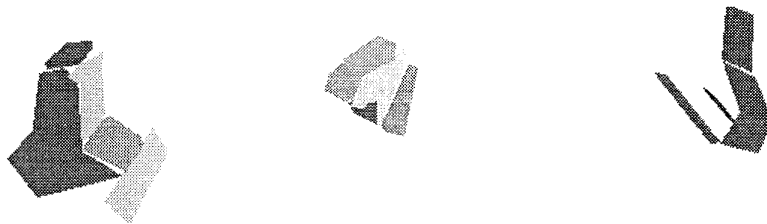


Figure 9: BSP tree models of the segmentations.

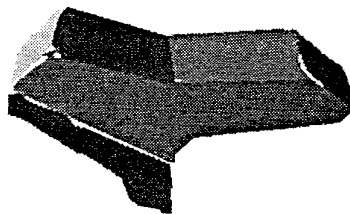


Figure 10: Final model after merging BSP trees.