

Solid Model Acquisition from Range Imagery

Michael K. Reed

**Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Graduate School of Arts and Sciences**

**Columbia University
1998**

© Copyright 1998 Michael K. Reed All Rights Reserved

ABSTRACT

Solid Model Acquisition from Range Imagery

Michael K. Reed

Recently the use of three-dimensional computer models has greatly increased, in part because of the availability of fast, inexpensive graphics hardware and technologies such as VRML-ready Internet browsers. These models are often of existing objects and are typically built by hand using CAD software, an error-prone and labor-intensive process. This thesis investigates methods by which these models may be automatically acquired, processed, and utilized by using range data.

An incremental modeling method is described that builds accurate solid models of objects from multiple range images. A hybrid of surface mesh and volumetric representations is used to create a “water-tight” 3-D model at each step of the modeling process, allowing models to be built from a small number of range images. The method is able to model scenes consisting of multiple, disconnected parts without imposing restrictions on their topology. The resulting models retain information that denotes each surface element as properly acquired or requiring additional sensing.

An important part of the model acquisition process is the determination of the next sensor viewpoint. We introduce a planning method that computes visibility for model surfaces to determine occlusion-free sensor positions that ensure that model fidelity is improved. These sensor positions are computed in continuous space, allowing a more complete solution than planning methods that rely on spatial discretization.

These two processes are combined in a system capable of acquiring models of a wide variety of shapes and scenes. Examples are shown using objects with features such as holes and disconnected parts, as well as scenes with large self-occlusions. Applications of this work include graphics, manufacturing, robot navigation and architectural site modeling.

To my parents.

Acknowledgments

There are many people to whom I am indebted for making graduate school a productive and enjoyable experience. First and foremost, I'd like to thank my family for always being supportive of my endeavors: my siblings David and Jenny, who are my best friends, and my mother and father, who have untiringly shown me guidance, patience, and love. Without them, I never would have been able to get here.

I would like to express my deep gratitude to my advisor, Peter Allen, who arranged for much support and material during my time at Columbia. My thanks also go to my thesis committee: Shree Nayar, John Kender, Tony Renshaw, and Sven Dickinson. Each gave me helpful counseling and needed encouragement long before I drafted them into my committee.

A lot of technical and moral support came from my fellow students. Steve Abrams, Paul Michelman, Ioannis Stamos, Paul Oh, Atanas Gueorguiev, and Brian Leibowitz all helped make things run smoothly by improving my ideas, code, and daily life. Thanks also to Marc Gwadz for sharing in the entire experience, despite his pleas for leniency.

My office mates Andrew Miller and Joshua Gluckman provided many hours of entertainment and refreshing commentary through their enthusiasm for Rubik's Cube, Prairie Dog Town¹, and more academic diversions.

Finally, I'd like to thank all my friends and loved ones outside of school for making the trip worth it.

1. Rubik's Cube and Prairie Dog Town are trademarks of OddzOn Products, Inc. and Binary Arts Corporation respectively.

Table of Contents

Chapter 1	Introduction	1
1.0	An Introduction to the Problem	1
1.1	Applications	2
1.2	An Introduction to a Solution	3
1.3	Desiderata for Model Acquisition Systems	6
1.4	What Makes a Good Model?	7
1.5	What Objects Can We Model?	9
1.6	Summary of Contributions.	12
1.7	Thesis Outline	13
Chapter 2	Related Research	15
2.0	Introduction	15
2.1	Early Work.	16
2.2	Modeling from a Complete Sampling	17
2.3	Modeling from Multiple Views: Incremental Methods.	19
2.3.1	Volumetric Techniques.	19
2.3.2	Techniques Utilizing Surface Models.	20
2.3.3	Techniques using Primitives.	23
2.3.4	Other Techniques and Domains	25

Chapter 3 3-D Modeling from Range Images 27

3.0 Introduction	27
3.1 Modeling using Meshes and Volumes.....	28
3.2 Range Image Acquisition.....	30
3.2.1 Rangefinder Camera Characteristics.....	31
3.3 Early Image Processing Applied to Range Images	37
3.4 Modeling a Surface from a Range Image	41
3.5 Sweeping the Mesh to Construct a Solid.....	43
3.6 Surface Type Identification	46
3.7 Merging Single-View Models	48
3.8 Experimental Results: Modeling from Predetermined Viewpoints..	52
3.8.1 Hip Prosthetic	52
3.8.2 Toy Bear	54
3.8.3 Video Game Controller.....	55
3.8.4 Propeller Blade	55
3.8.5 Analysis.....	57
3.9 Interaction of Sampling Sensors and Set Intersection Methods	58

Chapter 4 Planning Sensor Viewpoints 63

4.0 Introduction	63
4.1 Background	64
4.2 Strategies for Viewpoint Planning.....	68
4.3 Using Constraints to Plan for Occluded Viewpoints.....	70
4.4 Selecting Target Surfaces	73
4.5 Computing Sensor Imaging Constraints	75
4.6 Computing Occlusion Constraints.....	77
4.7 Computing Sensor Placement Constraints.....	80
4.8 Combining Constraints	81
4.9 Computational Considerations.....	82
4.10 Terminating Conditions.....	85
4.11 Example: Single Target, Synthetic Data	87
4.12 Example: Strut Model	93
4.13 Considering Multiple Targets	96
4.14 Example: City Scene	99

4.15 Analysis: Model City	107
4.16 Discussion of Trade-Offs between Sensing and Planning	108

Chapter 5 Conclusions and Future Work 115

5.0 Conclusions	115
5.1 Summary	115
5.2 Contributions	117
5.3 Limitations	118
5.3.1 Limitations of the Modeling Algorithm	118
5.3.2 Limitations of the Planning Algorithm	120
5.4 Future Work	121
5.5 Final Remarks	123

Appendix: Calibration 125

A.0 Introduction	125
A.1 Calibration	125
A.1.1 Rangefinder / Robot System Calibration	126
A.1.2 Determination of Rangefinder Transform	130

Bibliography 133

List of Figures

Figure 1-1.	Overview of automated model acquisition based on active vision.	5
Figure 3-1.	Details related to modeling in the overall model acquisition process .	29
Figure 3-2.	Experimental setup.	30
Figure 3-3.	Rangefinder internals.	31
Figure 3-4.	Rangefinder scanning.	32
Figure 3-5.	Laser rangefinder nominal behavior.	33
Figure 3-6.	Behavior of laser rangefinder with surfaces of high inclination.	33
Figure 3-7.	Behavior of laser rangefinder at sharp depth discontinuities	34
Figure 3-8.	Behavior of laser rangefinder during beam occlusion	35
Figure 3-9.	Effect of point light source illuminating a surface.	35
Figure 3-10.	Linear pushbroom camera model.	36
Figure 3-11.	The volume of occlusion for a view of a triangular face.	37
Figure 3-12.	Filling of unsensed sample due to scene occlusion.	40
Figure 3-13.	Example of occlusion edges between sampled points on a surface.	42
Figure 3-14.	Mesh surface models.	43
Figure 3-15.	Example of a mesh sweep operation.	44
Figure 3-16.	Solid formed by sweeping mesh in the sensing direction.	44
Figure 3-17.	Solid model from a single range image.	45
Figure 3-18.	Tagged single-view model.	47
Figure 3-19.	Result of tagging the single-view bear model.	48
Figure 3-20.	Example of the merging process.	51
Figure 3-21.	Hip prosthetic model, acquired in four views.	53
Figure 3-22.	Models of the toy bear in four views.	54
Figure 3-23.	Model of the video game controller in three views.	55
Figure 3-24.	Model of propeller blade in two views.	56
Figure 3-25.	Sensing behavior of a typical range scanner in 2D.	59
Figure 3-26.	2-D example of a mesh surface determined from the sensed points	59
Figure 3-27.	Dilation of mesh surface.	60
Figure 3-28.	Effect of nonuniformly dilating mesh on swept solid.	62
Figure 3-29.	Bear model detail showing effect of dilation process.	62
Figure 4-1.	Change in area of occluded surface as more views are taken.	70

Figure 4-2.	Overview of the sensor planning process.	71
Figure 4-3.	Unimaged model surface	73
Figure 4-4.	Constraints on sensor imaging due to projected target area.	75
Figure 4-5.	Variation in the volume representing imaging constraints.	76
Figure 4-6.	Occlusion computation by spatial decomposition.	79
Figure 4-7.	Volumetric representation of sensor placement constraints.. . . .	81
Figure 4-8.	Simplification Envelopes.. . . .	84
Figure 4-9.	A situation where it is impossible to image the entire object surface. .	86
Figure 4-10.	Synthetic planning example.	87
Figure 4-11.	Example: sensor imaging constraint.	88
Figure 4-12.	Example: occlusion constraint.. . . .	89
Figure 4-13.	Example: Visibility volume.. . . .	89
Figure 4-14.	Example: including sensor placement constraints.. . . .	91
Figure 4-15.	Example: continuous-space planning solution.. . . .	92
Figure 4-16.	Example: discretized solution.	92
Figure 4-17.	Photograph of strut part.	94
Figure 4-18.	Initial models from strut acquisition.	94
Figure 4-19.	The composite model of the strut after two views.. . . .	94
Figure 4-20.	Occlusion computation for a target on the composite model.	95
Figure 4-21.	The solid acquired from the third range image via the first plan.. . . .	95
Figure 4-22.	Result of sensor planning for a target specified.. . . .	96
Figure 4-23.	Fourth model acquired according to the second plan.	96
Figure 4-24.	Final strut model, shown rendered and as a mesh surface.. . . .	97
Figure 4-25.	Planning for multiple targets.	98
Figure 4-26.	Planning for a sensor constrained to move vertically.	99
Figure 4-27.	The city scene, consisting of three toy buildings.. . . .	100
Figure 4-28.	Preliminary model of city scene, from two perspectives.. . . .	101
Figure 4-29.	Initial planning for city scene.	101
Figure 4-30.	Plans for top 30 targets by area.	102
Figure 4-31.	Results of discretization of sensor space.. . . .	103
Figure 4-32.	Plan generation for views 6, 7, and 8.	104
Figure 4-33.	Plan generation for views 9, 10, and 11.	105
Figure 4-34.	Plan generation for view 12.. . . .	106
Figure 4-35.	Final model of city scene.. . . .	106
Figure 4-36.	Final model of city scene, texture-mapped.	107
Figure A-1.	Transformations between world, gripper, and rangefinder.	127

Chapter 1 Introduction

1.0 An Introduction to the Problem

The use of 3-dimensional computer models has greatly increased recently, in part because of the availability of fast, inexpensive graphics hardware and technologies such as VRML-ready Internet browsers. These models are often of existing objects and are typically built by hand using CAD software, an error-prone and labor-intensive process. This thesis describes methods by which these models may be automatically acquired using range sensors.

The use of range sensors has also been steadily increasing over recent years, and range data are now acquired from domains as far apart in scale as radar imaging of the Martian surface and STM images of atoms in a lattice. More typically, the data are acquired by imaging objects less than a cubic meter in size, because of the limited dynamic range of the currently-available commercial sensors. These images are processed by registration into point sets that, although they contain a great quantity of information because of their sheer size, don't contain much useful structure: they describe a set of samples of the

geometry, but they do not describe the surfaces, boundaries, or topology of the sensed object. Since most tasks require a higher-level model for any planning, analysis, or reasoning, these points must be processed so that such operations as object recognition and CAM may proceed at a conceptual level convenient for humans.

1.1 Applications

In addition to the data-driven need for effective algorithms to process range data, there are also many applications that would benefit from accurate, automatic model acquisition, particularly in tasks where creative freedom might be constrained by typical CAD interfaces. There are still parts which are best designed using the tools of model makers, in materials such as clay or wood, rather than with the mouse- and tablet-based CAD systems. 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” [Wohlert 1994]. Even if every new object was designed using computer modeling tools, there would still be many objects for which there are no appropriate computer models. Without models it is not possible to benefit from any of the advanced analysis, manufacturing, and process planning capabilities of today’s CAD/CAM systems, or to use rapid prototyping systems to produce part replicas. Applications in which 3-D solid or surface data must be acquired from physical models or prototypes include:

- Virtual environment generation – Virtual worlds used in the entertainment and graphics communities will require many models of real-world objects. The more realistic environments will require large numbers of accurate models.

- Inspection and quality assurance – Automated part inspection tasks may benefit by using CAD models rather than simpler metrology techniques. Samples of a manufactured object may be digitized and compared against a nominal CAD model, and a comparison carried out based on automatically or interactively identified part features.
- Reverse engineering – An effective means of reproducing existing parts quickly will allow designers to use hand-sculpted models to experiment with different shapes and designs. Alterations made to the physical or CAD models may then be realized using Rapid Prototyping (RP) technology to produce the part.
- Site modeling – A model of the architectural structures in a specific locale is useful for a variety of tasks, including rover navigation. Rover-based rangefinders, in conjunction with a Global Positioning System receiver, may be used to acquire CAD models from which such tasks as collision avoidance and path planning may be carried out.
- 3D FAX – The ability to “transport” replicas of objects from one location to another opens a host of new application possibilities. To accomplish this, an acquired model of an object is sent via modem to an RP machine in a remote location, where it is physically realized.

1.2 An Introduction to a Solution

One way to find a solution to this problem is to see how it is solved in the natural world, and it is not necessary to look any farther than our own behavior in object modeling. When

people are given objects they have not seen before, their investigative behavior usually follows a particular pattern. First the object is examined from a particular viewpoint, at which instant some representative model of the object has been composed in the person's mind. Of course, this model cannot describe the object completely, since at least one side of the object remains unseen. The person then rotates the object a certain amount, and pauses again. The information from this next view must then be incorporated into the model as well. For certain types of objects, particularly $2^{1/2}$ -D parts (such as keys), a person will only use two views (the two of maximal area), and will execute a very rapid rotation between them. We are, of course, ignoring a detailed account of the variations, as it is well known that the investigative behavior varies with the task. In the general case, however, this analysis holds.

From this description of behavior an analogous algorithm may be determined, as shown in figure 1-1. An initial model is acquired, integrated with any previously-built model, and the next view is planned using information from the new model. This process iterates until a satisfactory model has been built. The underlying structure of this algorithm is that of *active vision*, in which sensing operations are interleaved with operations that plan the next sensor position [Bajcsy 1988] [Ballard 1991].

The key subtasks of modeling from range images are the following:

- acquisition – sensing the object to produce a range image.
- registration – aligning two or more distinct range images so that the object features they describe are in appropriate physical relationships.
- modeling – constructing a high-level representation from a set of range images.

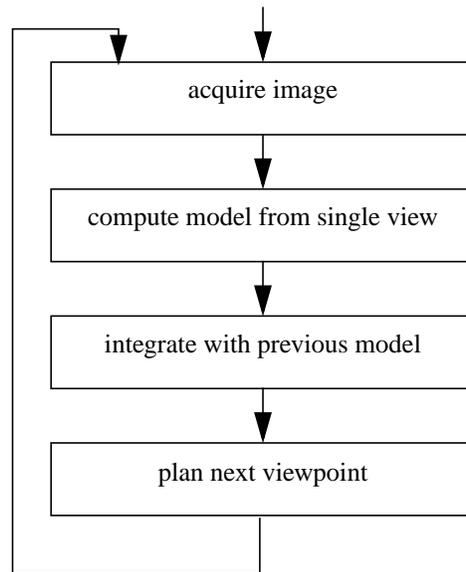


Figure 1-1. Overview of automated model acquisition based on active vision.

- integration – combining the information present in two or more distinct range images, including resolving ambiguities caused by conflicting data.
- planning – determining where to position the sensor to best acquire an image.

Although these necessary tasks have been fully identified, current systems are still a long way from easily being able to automatically build a complete model of an arbitrary object or scene. Much of the problem remains in the modeling process itself: not only are there difficulties in computing a model that describes a single range image well, but combining the models from distinct images taken from different viewpoints presents a major hurdle. Furthermore, solutions to the problem of planning appropriate sensor viewpoints still have significant drawbacks that limit their application.

1.3 Desiderata for Model Acquisition Systems

In [Curless & Levoy 1996] a set of desirable properties for surface reconstruction systems is given. These properties, with the addition of some others, may act as a starting point for evaluating a system:

- No restrictions on topological type – No assumptions should be made as to whether the object is of a particular genus, or even if it consists of a single connected component.
- No holes in the reconstruction – The resulting model should be “watertight”, even in situations where the entire surface of the object has not been imaged. This corresponds to the definition of a *solid model*, which is discussed later in this chapter. In places where the object’s surface has not been imaged, holes in the model should *not* be filled arbitrarily, but in some fashion that takes into account the *reason* it was not imaged, e.g. occlusion.
- Incremental incorporation of new data – The system should be able to update the current model of the object after each sensing operation. This allows the incorporation of planning algorithms to determine an appropriate next sensor position based on the current model.
- Robustness – The system should be robust in its representation and integration of range data, some of which is nearly coincident due to repeated sampling of the same surfaces.

- Time and space efficiency – Because each range image contains upwards of 15k points, and there are many range images to be incorporated, the methods should be efficient in both time and memory usage.
- Order-independent modeling – The order in which a set of images are merged together should not affect the quality of the resulting model, therefore allowing parallelization.
- Utilization of all range data – Any redundant sampling of a surface should be used to increase the accuracy of the model, if possible.
- Representation of model surface quality – The system should be able to distinguish model surfaces that are properly acquired from those that are not, for later analysis and model use.
- Automation of viewpoint planning – In order to be able to acquire complex scenes in varied environments, the system should be able to automatically determine sensor viewpoints during the model acquisition process.

How successfully each of these criteria is met by the methods presented in this thesis is discussed later in later chapters.

1.4 What Makes a Good Model?

We use the word “model” throughout this thesis, and therefore it is necessary to describe more precisely what is meant by that term. It should be noted that most modeling systems are not capable of representing arbitrary shapes, but this in itself is not a detriment in certain situations. In this thesis we consider modeling objects that already exist: we are not

concerned with modeling, say, mathematical functions in two dimensions, or hypothetical objects with non-manifold topologies. Thus we can follow some of the lessons learned by the Computer-Aided Design community in their search for very robust, highly flexible modeling techniques for physical objects, called *solid modeling*¹ [Requicha 1980]. The most important lesson learned was that to be useful, the modeler must be able to represent objects *unambiguously*. That is, a model cannot represent two different physical objects at the same time. This realization was due to shortcomings found in models produced by ambiguous systems based on wireframe or surface modelers. However, just as important to CAD advancements as the understanding of the ambiguity issue was the recognition that the modeler should be *unable* to represent objects that are not physically realizable, for example a cube with one face missing. It was this concept that allowed the modeling systems to drastically improve their robustness and efficiency (for details on this, see [Requicha 1980] or [Mantyla 1988]). These benefits were realized because the modeling system would only have to deal with objects of dimensionality three and finite volume, and could make assumptions about the result of any operations on these objects, most notably through Euler operators [Baumgart 1975]. This capability came at the price of flexibility: many designers like to work with surfaces during design, but these surfaces violated the realizability constraint and therefore could not be constructed in isolation.

The advantages of solid modeling are not just that it permits robust operations: indeed its greatest benefit is that it is informationally complete with in the geometry and the topology of the object represented. This allows algorithms to be written that analyze the

1. The basic premise behind modern solid modeling is to use a manifold boundary surface to model a closed, bounded subset of Euclidean 3-space.

model to answer arbitrary geometric questions. In contrast, wireframe and surface modelers are both unable to answer basic geometric questions, such as point classification, and also are not unambiguous in their modeling of a general object. It is for these reasons that we require our system to produce solid models as a result.

The fact that we require this result adds significant difficulty to our modeling process. As the CAD community learned, if a solid model is desired it is very important to disallow operations that deal with non-solid objects, since once the validity of a solid model is violated it is nearly impossible to repair. Besides the validity requirements, there are non-required quality issues as well. Typical violations of either of these are the following [ViewPoint Datalabs 1995]:

- Collision Errors – more than two surfaces sharing the same edge.
- Seam Errors – only one surface at an edge, i.e a hole in the model.
- Normal errors – a normal that does not point outward from the model.
- Edge errors – any colinear edges that are consecutive.

1.5 What Objects Can We Model?

Although it would be nice to have a system that places no limits on the type of objects that may be modeled, in practice such limits are necessary. They arise from many different sources and may be categorized as *scene domain constraints*, *sensing constraints*, and *modeling constraints*.

Scene domain constraints limit some property of any scene that is imaged. This may include the number of objects, their arrangement, or the type of objects that may be present. For example, the system may be limited to acquiring just one object that is within a certain size or is composed of only smooth surfaces, has no discontinuities, or is polyhedral. Common constraints seen in the literature are that there is only one object in the image and that it is of topological genus zero – that is, it has no holes. In our system, we put no such constraints on the object: it may have flat or curved surface, holes, or thin features, or be composed of disconnected parts. It may, in fact, be a scene composed of many objects, though in the rest of this thesis we shall use the terms “object” and “scene” interchangeably. We do, however, impose the constraint that the object or scene we are modeling may be fully imaged from each sensor location. This is possible only for scenes that have features viewable from the “outside” – i.e., from a spherical shell on which an imaginary sensor might travel. This constraint rules out scenes that are modeled from the “inside”, for example, a room that is imaged by multiple scans from the same central location.

Sensing constraints describe limits of the sensor such as depth of field, resolution, and minimum standoff or distance to the surface. In addition, the resolution of the images provided by the sensor limits the size of the smallest features that are recoverable. These issues are discussed in Chapter 3 along with other topics related to the sensing modality, but it should be stated that the current system can resolve features as small as 2mm.

Finally, modeling constraints describe what can or cannot be represented with the modeling system being used. For example, some modeling methods cannot model objects

with holes, or cannot model arbitrary mathematical functions. Because we are interested in modeling only objects that already exist, we are not concerned with the latter drawback. Instead, we utilize boundary-representation solid modeling as described above.

Now that the types of constraints are understood, we can more adequately describe the capabilities of our system: it is capable of modeling a scene of one or many objects, which may or may not overlap, each with curved or polygonal surfaces, through-holes, or thin features. It will produce a polyhedral solid model that represents all object surfaces that have been imaged by the sensor. Where occlusion situations in the scene have prohibited sensing, the resulting model surfaces are labelled as such so that they may be clearly distinguished from properly acquired surfaces.

It is just as important to describe the situations in which the system will *not* be able to produce a model as those in which it can. As mentioned above, the algorithms presented here are only applicable in situations where the sensor has access to the exterior of the scene or object; it is not capable of modeling an object from its interior, such as a room from its center position. Because we are using non-contact sensors that have a minimum sensing distance, called *standoff*, nothing can be reconstructed in the model that is not farther than that distance from the sensor. Thus there are many objects with concavities that may not be fully acquired, e.g. a block with spirally-twisting through-hole. This constraint applies to all such systems, and amounts to saying that all features that are not visible from the “outside” of the object cannot be reconstructed. The level of the ability (or lack thereof) to image features on a surface is sensor-dependent, but the techniques presented here are generally applicable and independent of the sensor type: they will work

with laser rangefinders, stereo cameras, and contact-based digitizing. One additional limitation on our system is that no part of the object may move relative to the world coordinate frame.

1.6 Summary of Contributions

As will be discussed in the next chapter, there have been many systems that solve various aspects of the model acquisition task. The techniques developed in this thesis contribute to the advancement of the state-of-the-art in the following ways:

- The method can acquire models of complex scenes including free-form and multi-part objects of arbitrary topological type.
- The final model is that of a “water-tight” solid, even if the entire object surface has not been sensed. Thus solid models may be produced in very few sensing operations.
- Surfaces in the model are identified as properly-acquired object surfaces or as those that require further sensing.
- The method allows incremental improvement of the model through additional sensing.
- Sensor viewpoints are planned during the model acquisition process by computing solutions to visibility queries for model surfaces that require additional imaging.
- A full implementation of the method has been completed, and a number of objects and scenes have been acquired to test the system.

1.7 Thesis Outline

This thesis describes both the problems involved in autonomously constructing models of objects and a system that addresses these problems. The layout of the remainder of the text is as follows:

- Chapter 2 discusses the previous work in modeling from observation.
- Chapter 3 discusses the problem of modeling from range imagery; it covers the first three stages shown in figure 1-1: range image acquisition, modeling, and integration.
- Chapter 4 describes the sensor planning issues and presents a solution to planning the next view. This is the fourth stage in figure 1-1.
- Chapter 5 presents our conclusions and caveats, and describes future research on this problem.
- The Appendix describes camera calibration methods for our sensor.

Chapter 2 Related Research

2.0 Introduction

Although there has been considerable work done on various types of modeling from a series of images, much of the work has been in the acquisition of models solely for the purpose of solving the recognition task. The work surveyed here considers the problem of building a physical model that mirrors the topological and geometric relationships of a sensed object. Thus, it includes all the systems that build objects for manufacturing tasks, but excludes (because of limited utility) work that determines a non-geometric representation. For example, a system that builds only feature index structures for recognition does not create a generally applicable model. The data may be acquired by various sensors, including cameras, rangefinders, or Coordinate Measuring Machines (CMM).

2.1 Early Work

Some of the earliest research into this problem, even though it does not provide a physical model, is included for historical reasons. Much of this work relied on edge detection and segmentation of intensity images, instead of the use of range data, because range sensors of good quality were unavailable. The idea of modeling by observation was under discussion at least as early as [Winston 1970], the purpose of which was not to acquire a physical model of an object but to determine a high-level structural description of the physical relationships among object parts. [Underwood & Coates 1975] also relied on the intensity image for reconstruction and could recover convex polyhedra by edge detection and reasoning about topological information – that is, information about the relationships of the surfaces (planar faces) and edges. Models created from additional views were merged by a matching process that recognized the surfaces already incorporated in the model and added only the unmodeled surfaces. [Potmesil 1982] was one of the first works to use range data to construct physical models from individual views and to address the associated registration and integration issues. This method used bicubic patches to fit areas of surface data acquired by a structured light technique: these patches were then inserted into a quadtree to allow a hierarchical representation that permits analysis at different resolutions. The quadtrees from different views are registered by minimizing a distance function that depends on the similar shapes in each quadtree. The transformed surfaces are then merged using raycasting to determine a single integrated surface.

The majority of the systems that followed the earliest work either assume a complete sampling or use multiple views. In the first case, the method is concerned with constructing a surface that fits data from a single image, or that fits data from multiple

images that have already been combined into a single cloud of points. The multi-view methods work by processing a series of images and integrating them one at a time into the model, and are therefore able to be used with sensor planning systems. A technique that is a hybrid between the two uses parametric shape models to represent recognizable shapes in the image.

2.2 Modeling from a Complete Sampling

The systems that perform modeling from a complete sampling benefit from the assumption that occlusions in the scene have been already been dealt with. Still, the task is far from simple, and a significant issue is assuring that the part topology is properly reconstructed. In one such system, [Hoppe 1994] describes a method that generates a mesh from a set of surface samples on the object to be modeled. The surface samples are assumed to cover the entire surface of the object, and are used as input to an algorithm that determines an octree representation. Isosurface extraction is then performed on the octree, which produces a mesh as an approximation to the object surface. The mesh is dense, though not as dense as the sampling. This initial mesh is then decimated to reduce its complexity by minimization of an energy function. The resulting set of mesh elements is segmented by examining the angle between adjacent elements, and finally piecewise smooth surfaces are fitted to the segments. A similar system was proposed by [Fua & Sander 1992].

Another system that uses octrees and a complete surface sampling is [Wheeler 1996]. In this work, registered mesh surfaces are again used to populate an octree, this time by a

constructing a consensus surface that combines the data from different views. This surface is then used to compute the signed distance values in the octree. Marching cubes is then used to extract the triangular mesh model. As with many other methods that use consensus surfaces (see [Turk & Levoy 1994] for example), the main issue is identifying criteria from which to construct the “optimal” consensus surface.

In the REFAB system [Thompson *et al.* 1996], multiple scans of a 2-1/2 D part typical of manufacturing are registered using a hypothesize-and-test technique between possible face matches in distinct images. The data is transformed into a common coordinate frame, and the user then specifies the approximate location of the features of the part. The system then finds the best fit for these features both in relation to the local data and among themselves. These constraints result in a model that, although it is less likely to fit the data as well as possible, is more likely to model a real part appropriately.

There have been many systems that use a deformable mesh surface that is iteratively fit to a complete point sampling. [McInerny & Terzopolous 1993] describe a system that uses a triangular mesh to represent a tension spline surface. This mesh is fit to 3-D point data from range and CT images by applying a deforming force based on each mesh node's distance to the nearest 3-D sample point. The technique in [Chen & Medioni 1994] similarly uses a triangular mesh to represent a “balloon”, with spring models used to keep the surface smooth. This model is then “inflated” inside the point sampling until the mesh surfaces contact come in contact with range data. More recently, [Shum *et al.* 1997] propose the use of a weighted least-squares algorithm to integrate resampled range images simultaneously, thus determining an optimal global registration. Mesh models that are

topologically homeomorphic to a sphere represent each range image, with integration being formulated in terms of principle component analysis. One problem with each of these systems is that the model is limited to be a single mesh surface of genus zero.

2.3 Modeling from Multiple Views: Incremental Methods

The systems that use multiple views have additional complexity to be concerned with, in particular the data registration and merging process. In the multi-view reconstruction literature, there are two main approaches. In the first, a volumetric model of the workspace acts as the initial model of the object; each subsequent sensing operation modifies this representation to account for space sensed as occupied or as empty. The other approach is to model the data from each sensing operation with a surface, in which case the integration method must be able to combine the surfaces from different views.

2.3.1 Volumetric Techniques

The methods that utilize a volumetric model of the workspace often use the *visual hull* of an object's silhouette in an image (either intensity or range) to bound the space which a model may occupy. [Martin & Aggarwal 1983] used the silhouette contour from a single intensity image to create a "volume segment" representation of each of the first two views. Additional views are then integrated by clipping the volume segment representation against them. The concept of the visual hull may be extended to utilizing range data, which provides the additional information of what the empty space is between the sensor and the imaged object. Other work that relies on this includes the following: [Potmesil

1986], which uses an octree representation to model both the objects in the scene and unacquired space; [Connolly & Stenstrom 1989], which applies edge detection and orthographic extrusion to build solid representations of intensity images; [Tarbox & Gottschlich 1995], which presents a method for automated visual inspection based on volumetric comparisons of acquired octree models; [Hoover *et al.* 1994], which illustrates well the problems associated with attempting to infer the topology and compute the geometry from segmented regions. A discussion of the properties of these techniques is presented in [Laurentini 1993].

2.3.2 Techniques Utilizing Surface Models

Methods that reconstruct the object by using surface models have recently received attention because of their non-discretized representation. Based on polygons, mesh surfaces, splines, and other continuous-space primitives, these techniques also tend to generate models that display more attractively. The 3-D Mosaic project [Herman *et al.* 1984] [Herman 1985] models urban environments by analyzing a sequence of stereo pairs. Junctions at edges in the image were matched to determine correspondence resulting in a 3D wire-frame and finally a polygonal representation. [Vemuri & Aggarwal 1987] represent a set of points (registered using the method in [Vemuri & Aggarwal 1986]) by adaptively fitting tension splines to fixed-area patches of the surface, and assigning them a classification based on their curvature. A merging phase then takes place based on this curvature. [Parvin & Medioni 1992] extract planar and quadratic regions from a series of range images and use them to build an adjacency graph. A match based on adjacency constraints determines the transform between images and the representations are merged

using boolean operations depending on the types of the surface primitives being merged: plane-plane, plane-quadratic, and quadratic-quadratic merges are all handled separately. [Wang & Wang 1994] utilize Kalman filtering as a means to build a hierarchical B-spline model from multiple range images. Imaged data is treated as a set of external measurements, which the Kalman filter uses to determine the internal state of the model. The internal state is the set of control points for the B-spline patches that best model the data.

[Seales & Faugeras 1994] uses a series of trinocular intensity images to determine a set of occluding contours as well as a set of fixed edges of the object. The occluding contours are used to recover the surface, since they determine the surface normals at all points on the silhouette. The fixed edges are used to determine the transformation from image to image. The points on the contours are fit with splines, which are then sliced using a plane orthogonal to a major axis of the splines at regular intervals. Mesh elements are then generated between slices, to complete a surface mesh of the entire object.

[Soucy & Laurendeau 1992] take a point-based approach to the merging problem, but remove redundant data to improve the quality of the model. In this work, range images are merged using point-by-point intersections after applying a known inter-view transform. Data points in two images are labelled as the same (and merged to one data point) if they are close to each other (after applying the transformation) and they are both visible in the same view. A quality metric is assigned to each data point that is related to the point's deviation in 3D position between all views in which it is present. When merging data points, this value is used to weight the data point's position so that when it is merged more

importance is given to more reliable information. The final point set is Delaunay triangulated, with multi-resolution models being available by merging adjacent triangles of the highest resolution model.

One of the surface-based methods of particular interest is that of surface meshes, because of their ability to flexibly represent many real-world objects at varying resolutions. One such method is that of [Rutishauser *et al.* 1994]. In this system, simple triangulation is built on each of several range images taken from different viewpoints. These triangulations only approximate the surfaces imaged, and not any discontinuity in the range data. The merging process consists of transforming the triangulations into a common coordinate and then re-triangulating the areas where there is overlap. The former is done by a process analogous to the Iterative Closest Point (ICP) scheme [Besl & McKay 1992]. Retriangulation is performed using a nearest-neighbor approach: Each edge from one triangulation is compared to nearby points from the other triangulation. One of the points is selected to build a triangle with the edge, and then the process is repeated until no more triangles can be built. Another mesh-based system is the Zipper system [Turk & Levoy 1994], a variant of which is used for the Cyberware scanner, a commercial system for acquiring models of small parts (and more recently the producers of a whole-body scanning system). The scanned object's surfaces in the range images are triangulated at the resolution of the sensor to produce one or more disjoint meshes, which are then aligned using an ICP variant. The aligned meshes are then merged by "zippering" the overlapping edges, which removes the redundant surfaces and clips the meshes to each other. This is effective provided that there is overlap between the images. Similar methods are presented in [Pito 1996], which uses knowledge of the sensor position to select the best mesh

elements from two images and a hole-filling procedure to ensure gaps are not left in the surface.

Recently [Curless & Levoy 1996] presented a system which combines the volumetric- and surface-based techniques. A surface mesh constructed on the initial range image is used in a ray-casting operation to weight voxels in an octree. This octree is then used as input to an isosurface extraction algorithm. This system produces excellent results, but uses large number of scans (e.g. 70) to ensure complete coverage of the object's surface.

2.3.3 Techniques using Primitives

A variant of the surface-based methods are those that build continuous-space volumetric primitives, such as spheres or cylinders. [Ferrie & Levine 1987] uses a two-level approach to determine the inter-frame transforms for a series of intensity images. After a shape-from-shading algorithm has been applied, principle curvatures on the object's surface are calculated. Local features are identified as neighborhoods of local maxima and minima in the principle curvatures. In the first level of this approach, these local features are used to determine the transform to bring the two depth maps into alignment. However, because this method can fail in the presence of large object movement, a global correspondence strategy is also employed. In a related work, the same authors are concerned only with acquiring a simple model that may be used for general grasping or collision avoidance tasks [Ferrie & Levine 1988]. Surface descriptions, which may be cylinders or ellipsoids, are built from 3D point data derived from shape-from-shading techniques. After having a transform applied to bring the primitive into a world coordinate frame, it may then be

combined with other partial models by removing redundant primitives and retaining the unique ones. There has been much work using volumetric primitives that are able to describe a wide range of shapes. These primitive, called *geons*, are a set of shape primitives that may be used (through parametrization) to construct a wide variety of single-part objects, and through composition to build more complex multi-part objects. The OPTICA system [Dickinson *et al.* 1992] operates by using an *aspect hierarchy* that describes the topology of a distinct view (or *aspect*) of one or more of the geon primitives. Each aspect may be a projection of a number of geons, and each component of the hierarchy may be a part of more than one aspect. An associated probability matrix describes the likelihood of a given aspect being a projection of a given geon, and similar matrices describe the relationships in the rest of the hierarchy. A segmented image is used to determine a graph of connected regions, which are then used in conjunction with the aspect hierarchy to determine the most likely primitive. This technique was later applied to range images in [Dickinson *et al.* 1997]. Similar qualitative primitives and view aspect graphs are used in [Raja & Jain 1994] and [Wu & Levine 1994], both of which use range data instead of a single intensity image. Once the complete set of range data has been acquired, and redundant data removed, simulated annealing is used to determine the shape parameters, position, and orientation for each of the 7 geon types. A least-squares error is calculated for each, with the best fitting geon selected as the model. [Whaite & Ferrie 1992] describe a system that acquires data by using views all equidistant from the view sphere, i.e. there are no scaling issues. After each view's data has been acquired (or in this case synthesized), a superquadric model is computed from the data set. Areas of the model that do not have sufficiently many data points within a low error bound are then selected

for more imaging. The use of superquadrics allows in some cases a high quality model to be acquired in just one viewing operation due to high symmetry. This advantage may be offset by the lack of expressive power of this representation. So far only simple shapes have been handled with this method.

2.3.4 Other Techniques and Domains

The use of models that combine information from temporally separate sensing operations, or allow for moving objects, is of much use to automated vehicles, which benefit from having a single map containing the terrain they have passed over, are currently located on, and have sensed but have not yet traversed. [Kweon & Kanade 1990] uses a technique called the locus method to build elevation maps from range images. By using a simple statistical model of the uncertainty of the range data, an uncertainty map is also generated. Feature-based and iconic matching is then used to compute the position and orientation changes so that the maps may be integrated. Overlapping points are handled by applying a function to the points and the local uncertainty. In [Asada *et al.* 1992], the range image is segmented into unexplored, occluded, traversable, and obstacle regions. Traversable regions, which are expected to be relatively large, are matched with previous segmentations and used to compute rotation parameters, while the translation may be estimated via sensors. Region-based matching is then done for the remaining obstacles, and any that are not found are classified as moving. A reliability weight is calculated for each point in the image, and the augmented images are merged to a single terrain map.

Combined use of a CMM and intensity images is an idea that has been used in several systems. This type of integration can be thought of as a manufacturing-specific variation of work bringing together vision and touch for recognition tasks, as in [Allen 1987]. [Sobh *et al.* 1995] uses a segmented intensity image to guide a high accuracy CMM, allowing the control of the delicate CMM probe so that it is introduced and manipulated on the part correctly.

Due to the extreme differences in the data acquired from our sensor and X-RAY or CT modalities, we also do not survey the reconstruction literature from biomedical applications, although many of the techniques described here have been applied to that field. For further discussion of techniques that have been applied to the medical field, see [Kalvin 1992] [Stytz *et al.* 1991].

Chapter 3 3-D Modeling from Range Images

3.0 Introduction

In this chapter, we discuss the issues relevant to the acquisition, modeling, and integration of range images:

- How and under what constraints are the images formed?
- How is an image modeled?
- How are the images or models registered into the same coordinate frame?
- How is the information from different images integrated?

As mentioned previously, desirable properties for a modeling system include:

- No restrictions on topological type
- No holes in the reconstruction
- Incremental incorporation of new data
- Robustness
- Time and space efficiency

- Order independent modeling
- Utilization of all range data
- Representation of model surface quality

This chapter describes in detail the modeling process and how it meets the above criteria. In section 3.1 an overview of our algorithm is given. Each of the following sections discusses a component of our modeling algorithm: section 3.2, range image acquisition; section 3.3, range image preprocessing; section 3.4, surface modeling; section 3.5, solid construction; section 3.6 surface annotation or “tagging”; section 3.7, model integration. Section 3.8 describes a nonuniform dilation technique as a solution to issues particular to our modeling method. The final section shows some results of modeling different objects.

3.1 Modeling using Meshes and Volumes

One of the central components of this thesis is an algorithm that creates a *single-view model* from a range image which represents all imaged surfaces and occlusion information in the current sensor and scene configuration. 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, which produces a *composite model* that represents all known information about the object or scene.

In the following sections we discuss in detail the steps in the model-making algorithm, which are shown in figure 3-1. Model construction consists of the first three stages in the process displayed to the left in the figure, and is shown in expanded detail on the right side. These steps include acquiring and processing a range image from a specific viewpoint, modeling this data with a mesh surface, constructing a solid from this surface, and integrating this model into a *composite model* which describes the scene as imaged by more than one view.

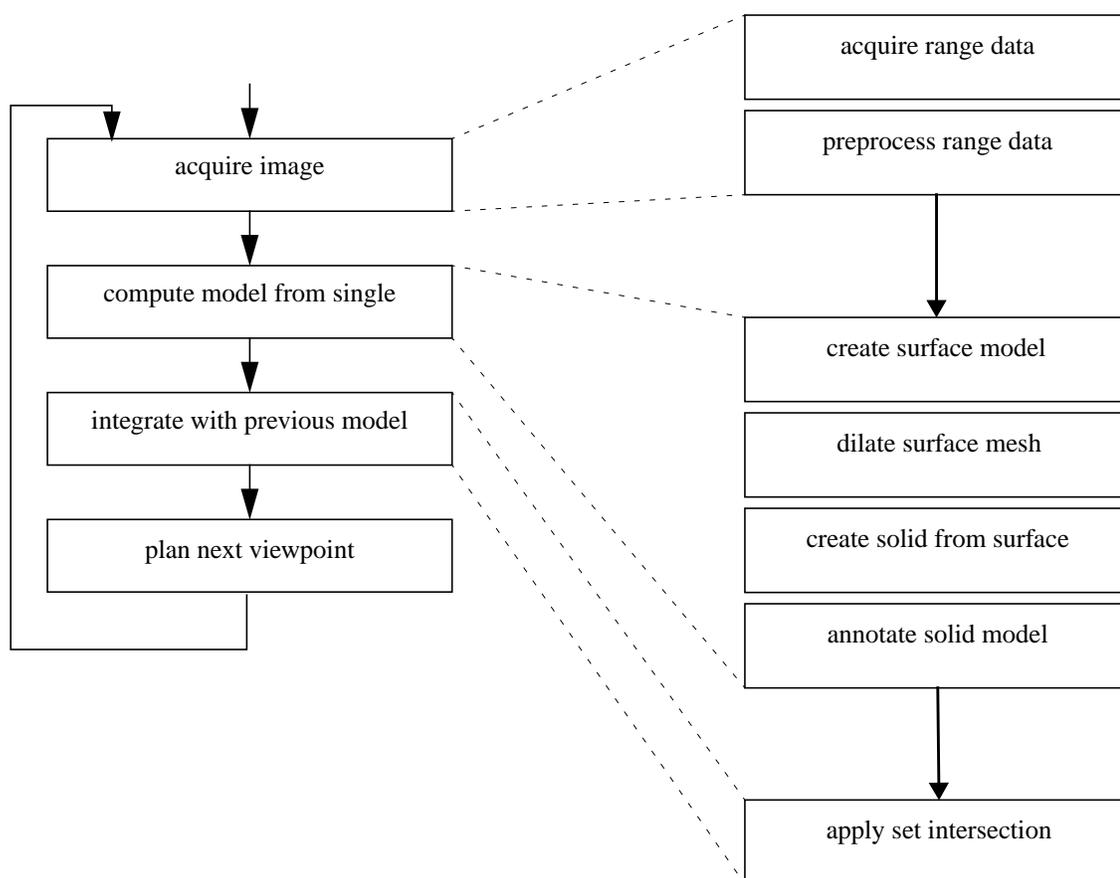


Figure 3-1. Details (right) related to modeling in the overall model acquisition process (left).

3.2 Range Image Acquisition

Range images may be acquired by using a variety of different techniques, such as tactile sensing, stereo cameras, or laser rangefinders based on time-of-flight or triangulation. Each method has its own sensing parameters that affect the speed and density of the sampling and the size and shape of the distribution of error. In this work, a triangulation-based laser rangefinder (shown in figure 3-2) is attached to a robotic arm to acquire a rectangular range image. However, in order to properly interpret the data in these images,

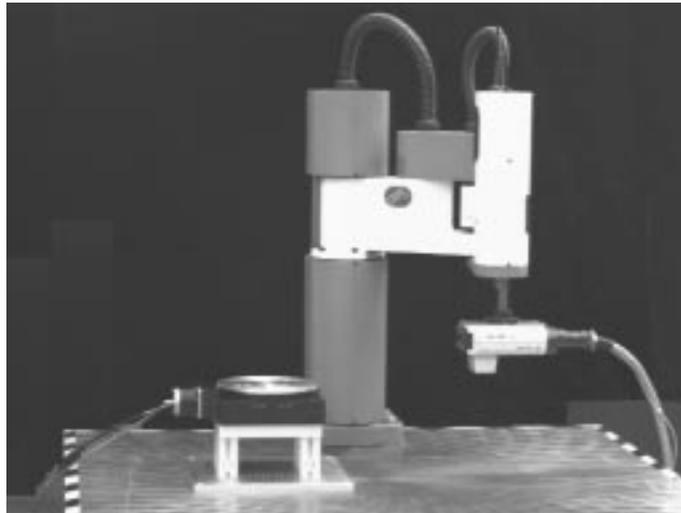


Figure 3-2. Experimental setup showing triangulation-based laser rangefinder attached to SCARA manipulator (at right). The turntable is used to rotate parts to present different orientations to the scanner.

it is necessary to understand its significance and the types of errors that are encountered. This, in turn, requires an understanding of some of the basic workings of laser rangefinders, which are described here. For an in-depth discussion of other range imaging devices, their manufacturers and performance, see [Besl 1988] [Jarvis 1993].

3.2.1 Rangefinder Camera Characteristics

The main issues that need to be considered when analyzing the laser rangefinder are its geometry and its behavior when it is acquiring a single point. These two issues affect both the accuracy of each acquired data point and the interpretation of the final scanned image.

The rangefinder used most often in the reconstruction literature is an active triangulation-based sensor that emits a small-cross-section beam that is detected by a CCD sensor¹. In order to accelerate the image acquisition process, the beam is scanned using a rotating mirror (see figure 3-3). The attributes of the beam such as its wavelength, energy, and

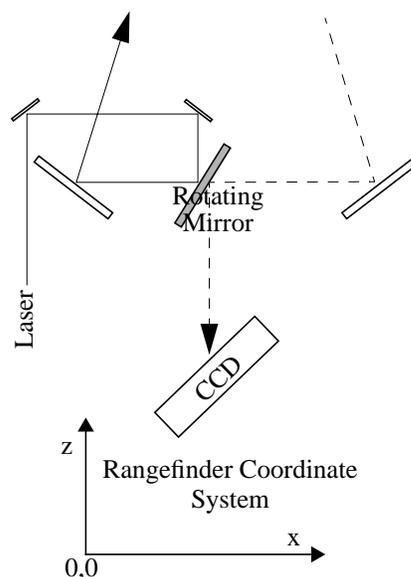


Figure 3-3. Rangefinder internals. The solid line represents the emitted beam, the dashed line represents the returned laser light. There are four fixed mirrors and one rotating mirror (at the center).

1. There are many different implementations of the triangulation-based laser rangefinder, but they all use similar principles. Described here is the “synchronous scanner”, which is also used in the experiments for this thesis. Where differences exist from other types of laser rangefinders they are noted.

emission (mirror) angle θ are used, along with the reflected illuminant's sensed position on the CCD and knowledge about the emitter-detector baseline separation, to determine the sensed surface's distance from the rangefinder and position on the X axis. Thus the data acquired from the rangefinder are a function of the emission angle; that is

$$f(\theta) = (x,z) \quad (\text{EQ 3-1})$$

as shown in figure 3-4. One point to note here is that the (x,z) pairs, which are ordered by θ , may have more than one z value for the same x , and in fact may not be strictly increasing in x at all. This may be evident from the fact that sharp depth discontinuities may cause the beam to be incident to a surface with a smaller x value. However, any points from a scan on a surface that may be represented as a continuous function in the rangefinder's X coordinate will have strictly increasing x values.

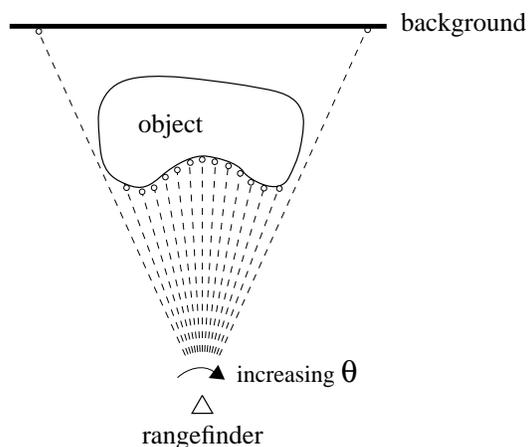


Figure 3-4. Rangefinder scanning: sensed data are ordered according to increasing Θ .

The nominal behavior of the rangefinder as it acquires each point is shown in figure 3-5. The surface is assumed to be matte, and therefore follows a Lambertian reflectance model. The emitted beam, which is continuous in intensity across its diameter, is received at the

detector with lower intensity and a larger cross-sectional diameter. Detector electronics take the discretized signal and estimate the point on the detector of highest intensity. However, this estimation becomes difficult if the angle of the surface is high with respect to the emitter or if there are sharp edges in the scene. In the former case, the emitted beam spreads out across the surface and no longer represents a sample from a small-cross-section beam (see figure 3-6). The response at the detector becomes lower in intensity and

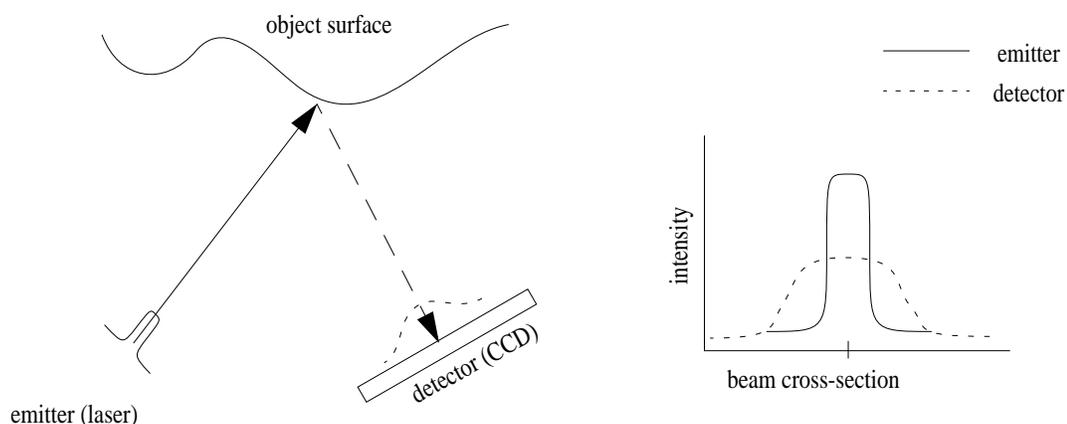


Figure 3-5. Laser rangefinder nominal behavior: acquiring a point on a smooth surface oriented towards the sensor.

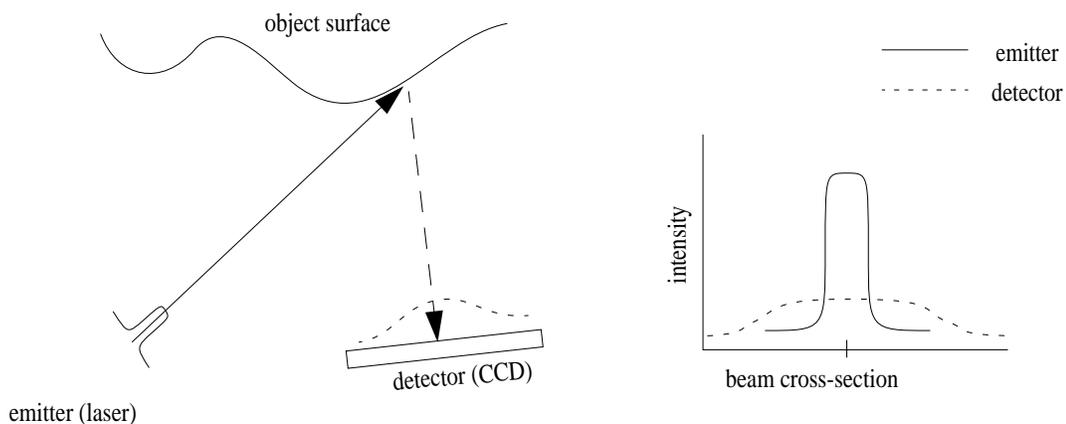


Figure 3-6. Behavior of laser rangefinder with surfaces of high inclination.

more distributed across the surface of the CCD, resulting in poor localization of the point with respect to the sensor's baseline. In the latter case, where there are sharp depth discontinuities in the scene, the beam may be incident to more than one surface due to the finite width of the beam, as in figure 3-7. This will cause two peaks in the signal at the

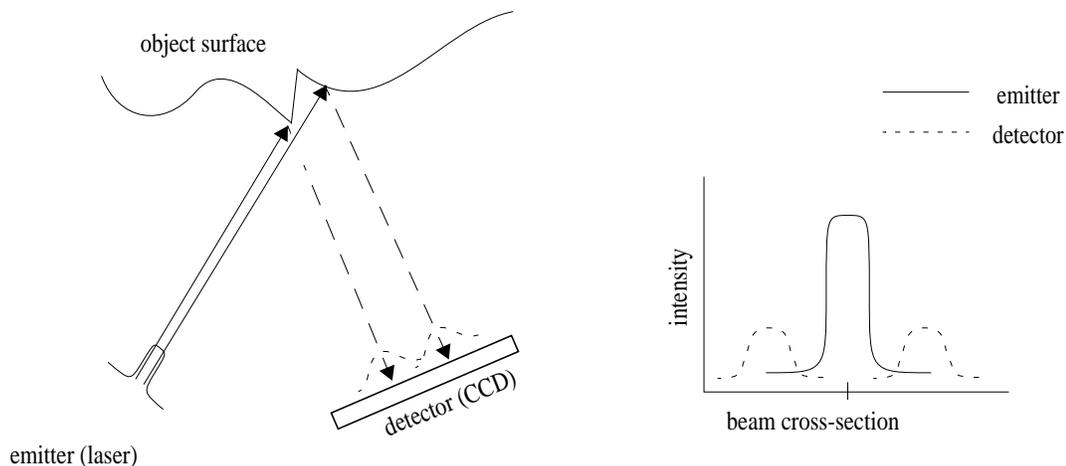


Figure 3-7. Behavior of laser rangefinder at sharp depth discontinuities: emitted laser beam is incident on two surfaces, causing multiple peaks in detected signal.

detector, each representing the reflection from one surface. If the peaks overlap, it may be impossible to distinguish between them, in which case the point reported by the sensor will be on the line defined by the two actual points, halfway between them. These points have been appropriately called “mixed pixels” in [Pito 1997].

One final situation where the sensor does not produce a correct sample from the object surface is if the incident beam is hidden from the detector by scene occlusion, as shown in figure 3-8. As we shall demonstrate later on in this chapter, our system is robust to these types of errors.

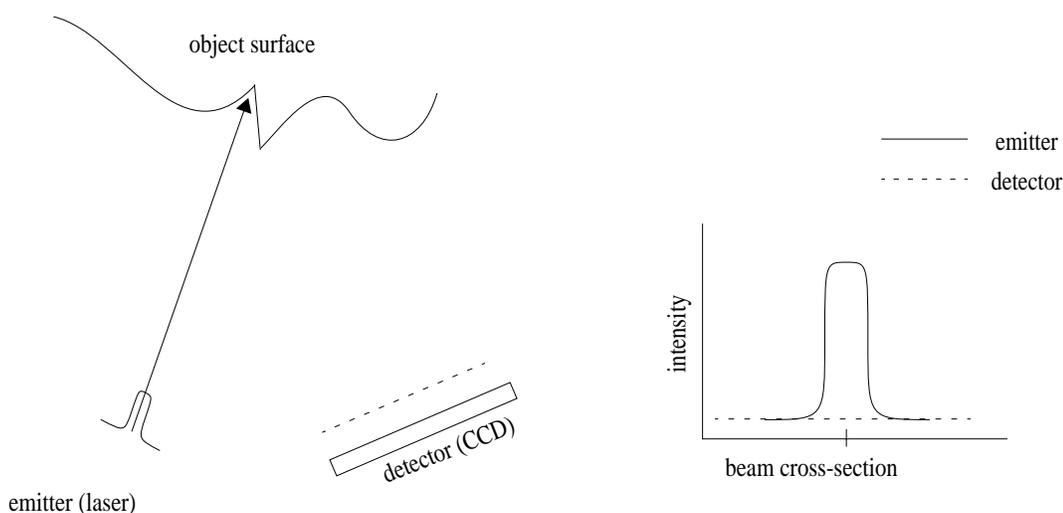


Figure 3-8. Behavior of laser rangefinder during beam occlusion: the point of contact between the emitted laser and the object surface is occluded from the detector.

It is important to be able to characterize the spatial distribution of the points acquired from the sensor. To do this, it is first necessary to discuss the shadowing effect caused by point light sources. A point light source produces from any object it illuminates a shadow called the *umbra* (see figure 3-9). The umbra is found on all the surfaces of the object that are not

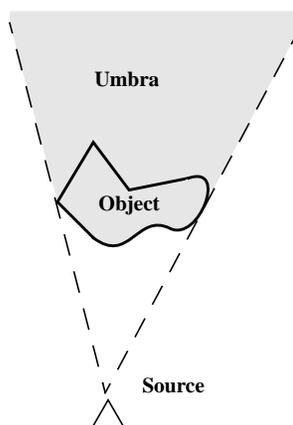


Figure 3-9. Effect of point light source illuminating a surface, showing umbra in grey.

directly illuminated, and also everything behind the object that is occluded from the source by the illuminated surfaces. Because the rangefinder's scanned beam emanates from a single point in the x-z plane, the sensed data describes both the object's illuminated surfaces as well as the umbra of the object in this plane. There can be no data acquired from within the umbra of the sensor/object pair in the scanning plane. This relationship between sensor and data does not, however, continue into the third dimension. Because a robot arm is used to step the rangefinder linearly along its y axis, the resulting sampling is not from the point-source umbra in 3 dimensions. Instead, it more closely resembles data acquired from serial-slice techniques – i.e., the occlusions caused by the linear movement of the sensor in this direction are orthographic. There is a characterization of a similar sensor using a passive camera instead of the active rangefinder called a linear pushbroom camera [Hartley & Gupta 1994], commonly used for satellite imagery such as SPOT's HRV camera (see figure 3-10). This may be compared with our experimental setup, shown in figure 3-2.

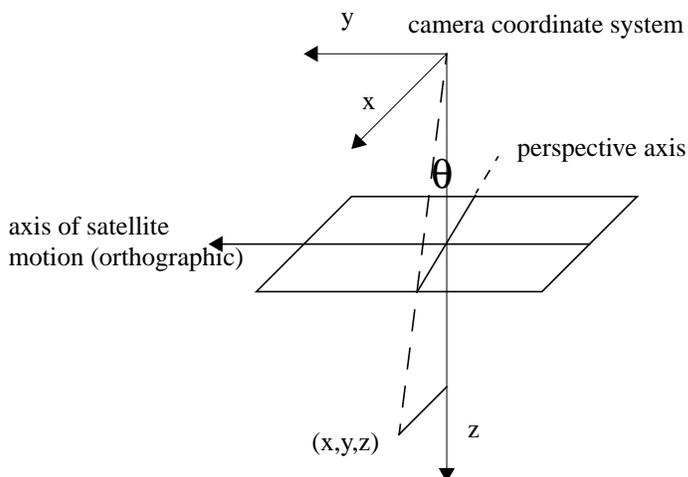


Figure 3-10. Linear pushbroom camera model.

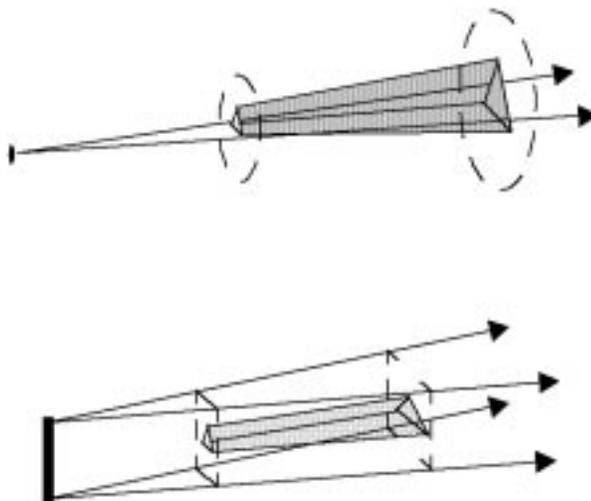


Figure 3-11. The volume of occlusion for a view of a triangular face modeled by the perspective projection umbra (top, as for a point sensor) and by a hybrid orthographic/perspective projection umbra (bottom, as for our sensor).

This has a great effect on the shape of the volume of occlusion in a single image: instead of being a cone formed by a single point (a sensor location) and the extremal boundaries of the imaged object, it is more prism-like in shape (figure 3-11). The sharp edge of the prism is the path of travel of the sensor.

3.3 Early Image Processing Applied to Range Images

In light of the above problems that may appear in range images, it is necessary to consider techniques to improve their quality and usefulness. Range images may be improved through a combination of filtering, averaging between images, and interpolation. Each of these operations makes a different contribution to the quality of the final image: filtering removes spike noise, an averaging process improves accuracy, and interpolation fills in gaps in the sampling due to the detector occlusion situations discussed in section 3.2.

Spike noise may occur in scenes that have surfaces with high reflectance. In these cases, if the orientation of such a surface causes the emitted laser to be reflected with a much higher intensity than the surfaces nearby, the rangefinder will report an erroneous depth value at that point. However, these cases are easily identified because they are typically only a sample in width: the high reflectance of the surface causes this type of noise to be localized. Because the samples which have spike noise can be clearly recognized, it is not necessary to apply a filter to the entire range image to remove them. Instead, selectively modifying only the noisy sample with the result of a median filter to the surrounding samples allows the noise to be removed while retaining the quality of the remainder of the image.

In order to improve the accuracy of the image in the presence of systematic errors caused by quantization and noise, the same image may be acquired multiple times and averaged. This is only possible, of course, in situations where the scene is static and the sensor is able to repeat exactly the geometry of its mirror configuration during sampling. If this is the case, then at each mirror angle θ , the 3-D point X_θ may be computed from N samples ($X_{\theta,1} \dots X_{\theta,N}$) by

$$\bar{X}_\theta = \frac{1}{N} \sum_{i=1}^N X_{\theta,i} \quad (\text{EQ 3-2})$$

The benefit here comes from the reduction in the variance of the noise, which translates to a reduction in the error of the sampled point if a Gaussian error model is assumed¹ and the

1. It is known that rangefinder error is not strictly Gaussian, as it is dependent on features in the scene. However, it holds for much of the acquired data, and for the purposes of examining the error it serves as a useful model.

noise has a mean of zero. If the standard deviation of the error associated with X_θ is σ , then after N samples the standard deviation in the mean becomes¹

$$\sigma_m = \sqrt{\text{Var}(\bar{X}_\theta)} = \frac{\sigma}{\sqrt{N}} \quad (\text{EQ 3-3})$$

Thus the accuracy is proportional to $1/\sqrt{N}$.

As previously described in figure 3-7, if the object surface is not a function of the rangefinder's emission angle θ , i.e., there is a depth discontinuity, there may be "mixed pixels" which appear between two surfaces separated by a sharp discontinuity. However, it is clear that the surface between this point and its neighbors is always *external* to the actual surface of the object. Because of this and the fact that in this work set intersection is used as a method of integrating views, these false surfaces (and therefore the points that determine them) have no ill effect on the modeling process. In fact, because the surfaces constructed on these points may be clearly identified due to surface normal comparisons, they are a key indicator used in planning the next view. In systems that do not use set intersection as an integration method, these surfaces must not be included in the model, a significant disadvantage. One caveat: such surfaces are always adjacent to sharp discontinuities in the range image, which in turn are places particularly vulnerable to sampling errors that adversely affect the final model. A method to detect and account for these errors is presented in section 3.9.

1. $\text{Var}(\bar{X}_\theta) = \text{Var}\left(\frac{1}{N} \sum_{i=1}^N X_{\theta,i}\right) = \frac{1}{N^2} \sum_{i=1}^N \text{Var}(X_{\theta,i}) = \frac{1}{N^2} N \sigma^2 = \frac{\sigma^2}{N}$

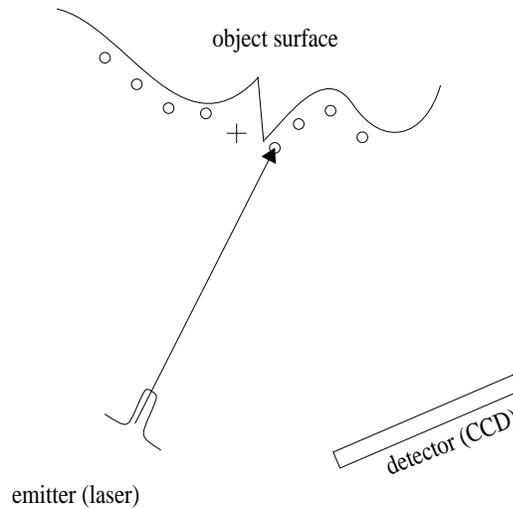


Figure 3-12. Filling of unsensed sample due to scene occlusion: As shown in figure 3-8, the sample (marked as a cross) was not acquired. It is filled in as the convex combination of the samples on either side of it. No object surface can extend beyond this point, because it would then be visible to the detector.

Finally, there is the problem of missing data points – that is, those surfaces from which the emitted beam is occluded from the CCD detector by other surfaces in the scene, as in figure 3-8. First, note that because of the short baseline used in most modern laser rangefinders (see [Besl 1988]), these are relatively rare occurrences. In addition, if a time-of-flight sensor is used rather than one based on triangulation, these situations may be avoided entirely. However, an effective solution is to interpolate a point composed of the convex combination of the (correctly-sampled) points on either side of the occluded sample (see figure 3-12). It is not possible for an object surface to be external to the surfaces created by these points: if there was such a surface, it would not have been occluded and would have been correctly sampled. Again, such occlusions and the surfaces created by these points denote regions of sharp depth discontinuity in the range image, where surfaces have not been properly acquired. These are a key part of the view planning process, and are described later in this chapter.

3.4 Modeling a Surface from a Range Image

After applying the low-level image processing, the points in the range image may then be used as the vertices in a mesh following a simple connectivity. However, since the mesh determined by a single range image is in essence a surface model, it does not contain information that permits spatial addressability (the ability to classify points as inside, on, or outside the model), which is necessary for many tasks and is inherent in solid models. Although a mesh that completely covers an object may be used to determine a solid model, in most incremental modeling techniques the mesh can not be closed until the entire object has been scanned. Thus, methods that only use the mesh as a surface model require a large number of overlapping scans, will work only with objects whose entire surface is 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, one that incorporates both the information about the model's sensed surfaces and the occlusion information in the form of the occlusion volume. When building the mesh that will be used to represent a surface from a range image, it is necessary to determine what the mesh connectivity will be. In this regard our work differs from other mesh-based methods such as mesh zippering [Turk & Levoy 1994] and other re-meshing techniques [Rutishauser *et al.* 1994] [Pito 1996], 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 (figure 3-13). Our system retains these elements, since they denote the boundary between surfaces of the object that have been imaged by the sensor and the space occluded from the sensor (and

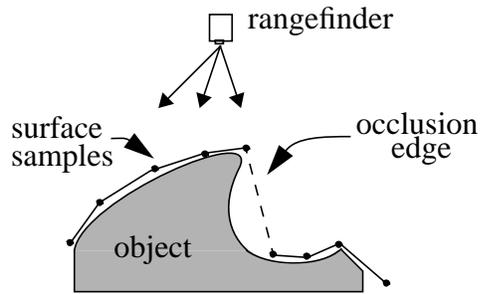


Figure 3-13. Example of occlusion edges between sampled points on a surface.

therefore in need of further imaging). These elements must be handled with care, because they have the potential to violate certain assumptions made during the model construction process, which are discussed in section 3.9.

As an example of this process, consider the hypothetical object shown at the top of figure 3-14. A range image is sampled from the CAD model using the indicated sensing direction. The surface model shown in the middle of figure 3-14 is typical of mesh surface-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. These methods use only these surfaces because it allows the assumption that the resulting mesh is a good approximation to actual surfaces on the object, and therefore simplifies processing. However, it is clear that scene information is represented by the occlusion elements and that discarding them is not desirable. The mesh shown at the bottom of figure 3-14 represents both the imaged surfaces of the object and the boundary of occluded space between the imaged surfaces.

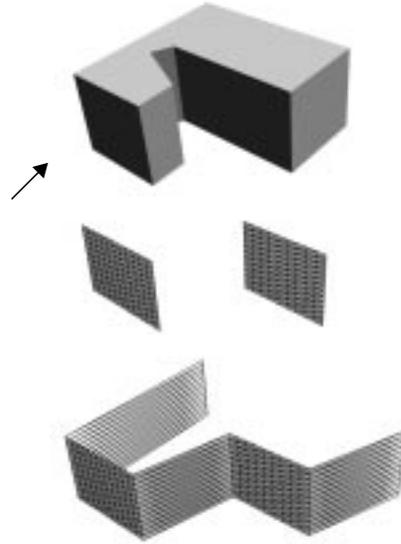


Figure 3-14. Mesh surface models. Rendering of CAD model of a typical 2-1/2 D part, shown with a sensing direction (top). Surface mesh from synthetic range data of CAD part (middle). This mesh does not include any elements that contain occlusion edges. Surface mesh generated from synthetic range data, including elements composed of occlusion edges (bottom).

3.5 Sweeping the Mesh to Construct a Solid

To form a solid model from the sampled range data, the 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:

$$S = \bigcup \text{Extrude}(m), \quad \forall m \in M \quad (\text{EQ 3-4})$$

An extrusion operator is applied to each triangular mesh element m , orthographically along the vector of the rangefinder’s sensing axis, until it comes in contact with a far bounding plane. The result is the 5-sided solid of a triangular prism (figure 3-15). 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

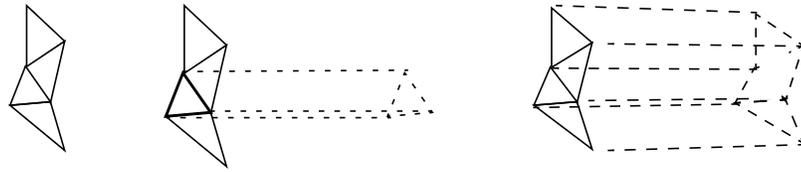


Figure 3-15. Example of a mesh sweep operation (left to right): Mesh surface, mesh surface with one element swept, and mesh surface with all elements swept and unioned. The sensing direction is from the left.

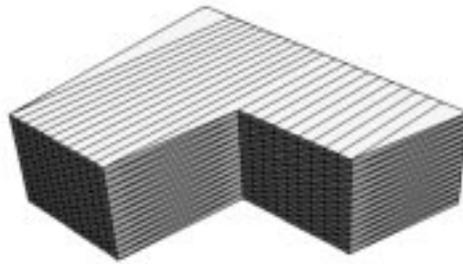


Figure 3-16. Solid formed by sweeping the mesh shown at bottom of figure 3-14 in the sensing direction.

sweeping operation, and a bounding surface that caps one end. We call this solid a *single-view model*.

As an example of the sweeping process, consider again the hypothetical part shown at the top of figure 3-14. Sweeping its mesh (shown at the bottom of figure 3-14) results in the solid shown in figure 3-18. In the following example, shown in figure 3-17, we show results from real range data of a toy bear. The range image shown includes samples of the background, which are removed unless they support the geometry of the surface that bounds the volume of occlusion. The solid is shown with its edges visible to better visualize the shape of the acquired surface; to permit this the example is at one-quarter the resolution typically used.

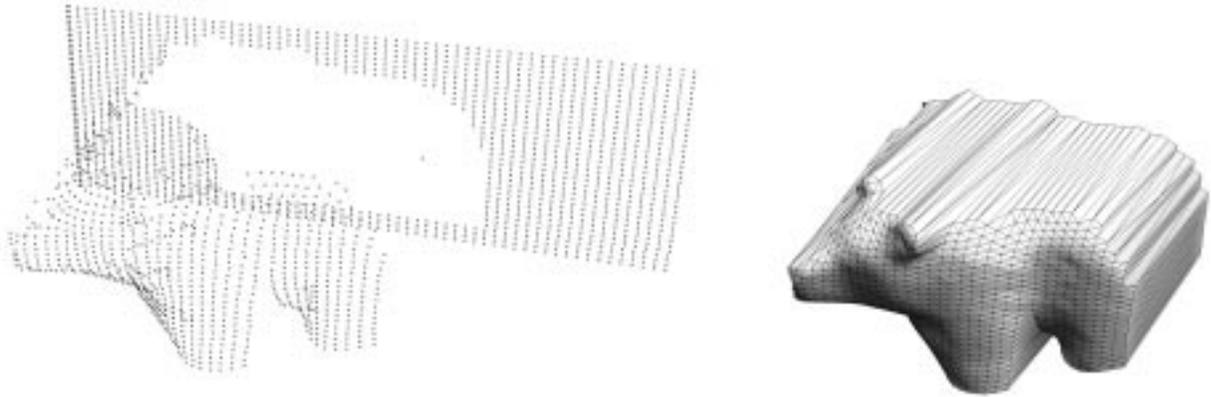


Figure 3-17. Solid model from a single range image. Left: 64x50 range image of toy bear. Right: solid model, showing density of surface elements.

Because of the large number of set operations required to construct a model in this way, it may appear that simpler methods may suffice in the construction of the single-view model. For example, it has been proposed that we use a cutting plane to “cap” the open end of the mesh surface, thus forming a closed solid. However, there are several reasons why many of the simpler methods are not an improvement. First, our method is extremely robust to errors in the input mesh surface. It will still produce a topologically-correct solid model even if there are mesh vertices that are identical (which creates mesh elements of zero area), and also if the mesh is not a 2-manifold surface, i.e. it is self-intersecting. These problems occur commonly in surfaces created from range images, and at least in the latter case may be expensive to detect and remove. Second, this algorithm is highly amenable to parallelization, and in fact its current implementation takes advantage of multiple processors to partition the task and achieve faster processing times. Finally, alternative

techniques introduce their own numerical issues, which, in combination with their use of non-manifold modeling prevents the use of robust modeling techniques, for example, cutting a surface with a planar sheet in an attempt to form a solid.

3.6 Surface Type Identification

It is important to be able to differentiate between the properly imaged surfaces and those due to occlusion or solid construction during later model analysis and sensor planning. To do this we attach tags to each surface in the model based on which of these two sets the surface belongs to: later in the modeling process these tags may be retrieved and used to analyze the model.

All surface elements in the model that were present in the mesh before sweeping and that meet certain geometric criteria are tagged as “imaged surface”. These elements describe surfaces of the object that were imaged properly and do not need to be imaged again. The criteria determine if the surface was properly imaged by the sensor, and consist of a threshold angle and an edge length. The threshold angle compares the normal of the surface element with the direction of the rangefinder’s laser: if these vectors are close to anti-parallel, i.e. the angle between them is 180° , it is likely that the sensor correctly imaged the surface there. However, as the vectors approach orthogonal, the samples tend to spread out and thus the surface is not as well acquired. The edge length serves a similar purpose, and is used in conjunction with the threshold angle to increase the selectivity of the process. In the modeling examples shown at the end of this chapter, typical values for the threshold angle and the edge length are 80° and 6mm.

All the remaining surfaces that were not identified as “imaged surface” should be tagged as “occluded surface” so that they may be used to drive a later planning process. These include surfaces that were not imaged properly by the sensor, as well as those that were due to the solid construction process described in the previous section. For example, the large capping surface that bounds the model’s “far side” (from the viewpoint of the sensor) will be among the surfaces tagged “occluded surface”. An illustrative example of the tagging process is shown in figure 3-18 for the single-view model constructed in figure 3-16. Shown in figure 3-19 is the result of tagging the model in figure 3-17: red surfaces denote those that are tagged as occluded.

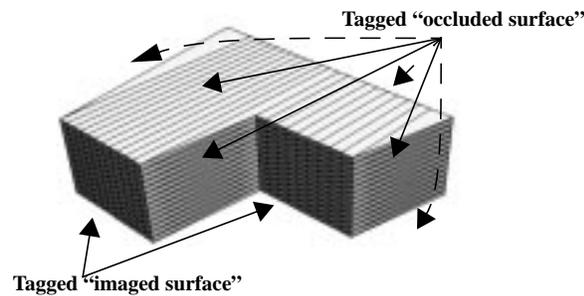


Figure 3-18. Tagged single-view model: tags for hidden surfaces are shown with dotted arcs.

It should be noted that this tagging procedure must be done to each model from a single sensor position individually: large faces often get split into smaller ones during the merging process, and their surface type will not then 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.

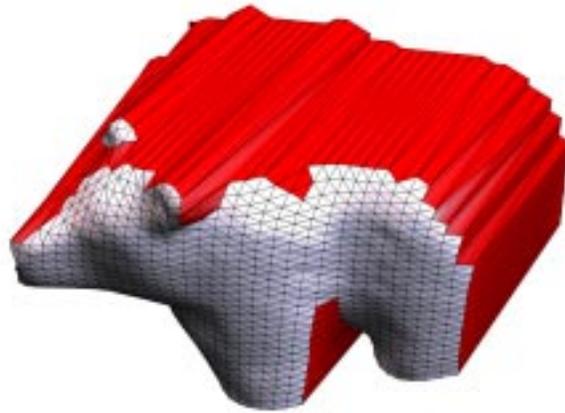


Figure 3-19. Result of tagging the single-view bear model: red surfaces denote those tagged as occluded, white surfaces are those tagged as imaged.

3.7 Merging Single-View Models

Each successive sensing operation will result in new information that must be registered and merged with the 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 [Pito 1997] [Dorai & Jain 1997]. As mentioned previously, merging of mesh-based surface models has been done using clipping and re-triangulation methods that also perform some vertex averaging [Turk & Levoy 1994] [Rutishauser *et al.* 1994]. 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 we generate a solid from each viewpoint which allows us to use a merging method based on set intersection. As described in the first chapter, solid modeling systems allow 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. In the implementation of the modeling component, we use an intersection operator provided by Spatial Technology's geometric modeling system, ACIS.

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.

As an example of the merging process, let’s again consider the toy bear model, but assume that we have two tagged single-view models already constructed, as at the left of figure 3-20. These two models, shown from above, were constructed from range images taken with the bear rotated 90° between them. As before, surfaces tagged “occluded” are shown in red, while “imaged” surfaces are shown with their edges visible. The right of the figure shows the result of set intersection on the two models, i.e. the composite model. Note the model surfaces derived from those tagged “occluded” retain their tags, even if they are split during the intersection process.

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

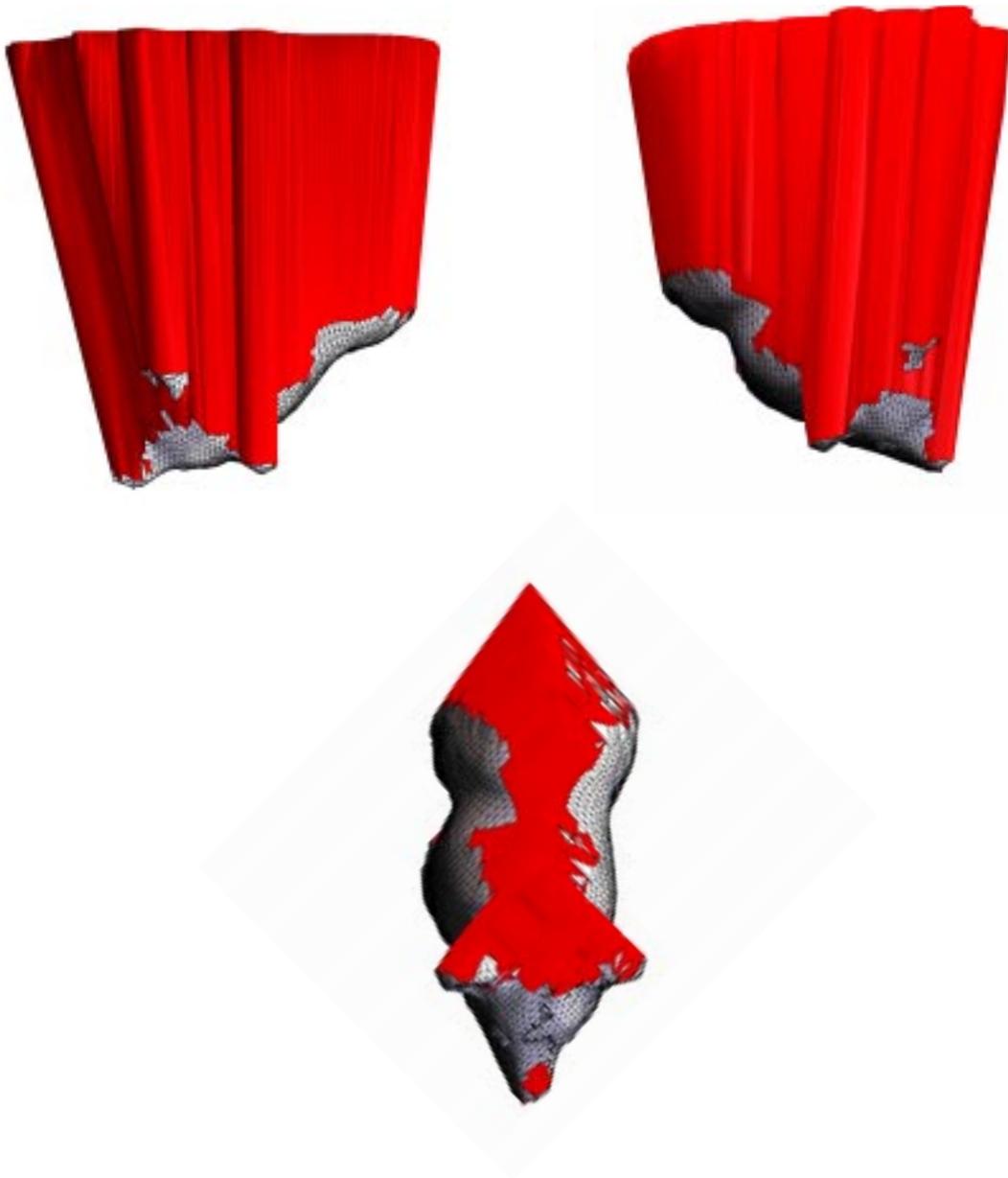


Figure 3-20. Example of the merging process on two single-view models of the toy bear. The top row shows two single-view models, with “occluded” surfaces shown in red and “imaged” surfaces shown with edges visible. Applying set intersection to these two models results in the model shown at the bottom of the figure.

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.

3.8 Experimental Results: Modeling from Predetermined Viewpoints

The examples shown here illustrate the modeling process. Shown are models constructed from four, three, and two views taken from equivalent rotations on a turntable: there is no on-line planning of the sensor positions. 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.

The acquisition of range data is performed by a Servo-Robot laser rangefinder attached to an IBM SCARA robot, with the object to be imaged being placed on a motorized rotation stage (see figure 3-2). The rangefinder acquires a single scan line of data at a time in a plane perpendicular to the robot's z axis. After each scan line has been acquired, the robot steps the rangefinder a small distance in along its z axis. The result of the scanning process is a rectangular range image of the object from a particular viewpoint, the direction of which is controlled by rotating the turntable.

3.8.1 Hip Prosthetic

The first example is of the construction of a prosthetic hip model from four range images, with turntable rotations of 90° (figure 3-21). The solids constructed from each of the four range images are shown, along with a photograph of the prosthetic and a wire-frame rendering of the final model. Of note are the many small parallel ridges in this object

which, although not visible in the final reconstruction, were acquired and can be seen in the third single-view model.

Some features of interest in the final reconstruction are the regions of increased surface element density, which appear in narrow, curved paths on the surface of the model. These regions are due to overlap in the intersection of single-image models, where new boundaries are created during the intersection process to accommodate the surfaces of two models. Also of interest are the regions where there are “patches” of smaller surfaces than the surrounding areas. These occur where two single-image models have surfaces that overlap each other, again causing increased tessellation.

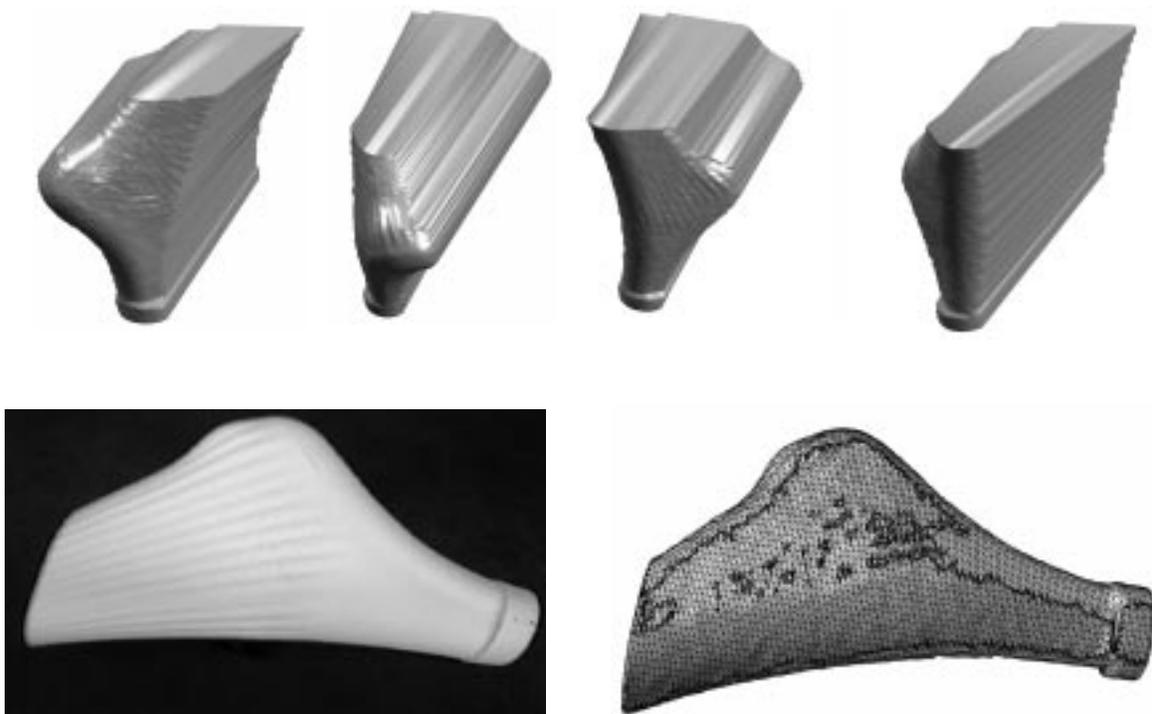


Figure 3-21. Hip prosthetic model, acquired in four views.

3.8.2 Toy Bear

The example shown in figure 3-22 is a reconstruction of a toy bear in four views. Again, 90° turntable rotations were executed between images. The top row shows the four solids constructed from each of the range images, while the bottom row shows a photograph of the object next to the acquired model. In this case, none of the four views were directed perpendicular to the side of the bear, and hence the empty space under the bear was not acquired. This can be seen in the reconstruction as model surfaces that fill in the space between the bear's legs. Again, due to overlapping views, there are some surfaces on the model that are more highly faceted than the surrounding regions. These regions may be acquired either by imaging from more appropriate sensor locations, or by turning the bear on its side to image its stomach, then using a registration algorithm to bring this view into alignment, as demonstrated in [Yang and Allen 1998].



Figure 3-22. Models of the 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.

3.8.3 Video Game Controller

In figure 3-23 we show the model of a hand-held video-game controller modeled in three views, with 120° rotations of the turntable. This part consists of polygonal and curved surfaces at varying levels of detail, including buttons on its front surface that are 2mm in height.



Figure 3-23. 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.

3.8.4 Propeller Blade

The final model demonstrated in this chapter is a propeller blade, modeled in two views, shown in figure 3-24. The propeller blade is very thin along the surface of the blade, and is

topologically of genus 1 (there is a hole through the base of the prop). 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 due to the extremely thin surface there, and hence those 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 or propellers. Finally, the deep through-hole in its base makes it impossible to acquire surface data there, so any method based on acquiring direct surface samples will be unable to produce a solid model. Our method, however, constructs the solid model and properly acquires the topology of the part.

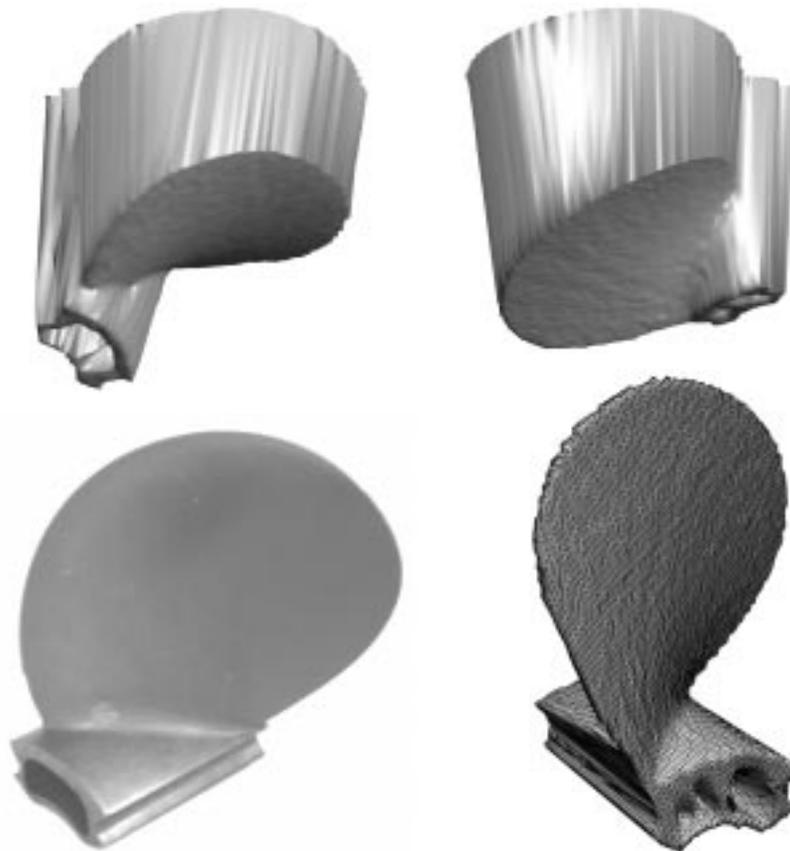


Figure 3-24. Model of propeller blade in two views. Solid models from each view are shown at top, bottom left is photograph of actual prop, bottom right is wireframe of acquired model.

3.8.5 Analysis

The modeling system described is capable of acquiring objects at a very high level of detail. One measure of this is the number of polygonal faces used in the final models. This is shown for each of the above parts in Table 3-1.

TABLE 3-1. Count of polygonal surface elements for each model

Model	# views	# polygons
hip	4	17228
bear	4	17335
game controller	3	20301
prop	2	17758

Because of the subjective nature of comparisons between objects it is difficult to come up with an algorithmic means by which to measure the accuracy of the modeling process. To our experience, the best way is to use the model to produce a RP solid which may be directly compared to the object. Using this method, many modeling problems that are nearly invisible when examining the model graphically become immediately apparent. However, as a step in the quantitative direction, we compute the volumetric difference between the video-game controller and its model. This is a reasonably complex part, with flat and curved surfaces and fine detail. Using a water displacement test, the controller measures approximately 215ml, while the model displaces 210ml. This compares favorably with a model volume, computed using computational geometry techniques¹, of 199ml. The difference between the two displacement measurements is 2%. We note here that volume calculations may be readily performed on solid models, as may calculations of higher-order moments, but not on surface models.

1. See [O'Rourke 1994] for a discussion of this method.

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 four minutes for each 110x128 rectangular range image, slightly less for those images where more background is visible. Merging solids using the set intersection operator takes approximately two minutes per model.

3.9 Interaction of Sampling Sensors and Set Intersection Methods

It is important to note that our method utilizes not only mesh elements from the imaged object surfaces, but also elements that describe the boundary between imaged and occluded space in the scene. These latter surfaces, because they represent occlusion and not physical surface information, may be arbitrarily far from true surfaces in the scene. This fact is a critically important one, because this behavior violates an assumption that is a requirement for using set intersection as a method of view integration, whether it be by intensity, rangefinding, or other sensing methods: that 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 be the case that the solid constructed from a single view does not contain regions or features 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 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 with respect to the sensor are all related (see figure 3-25). As shown in the example of figure 3-26, 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.

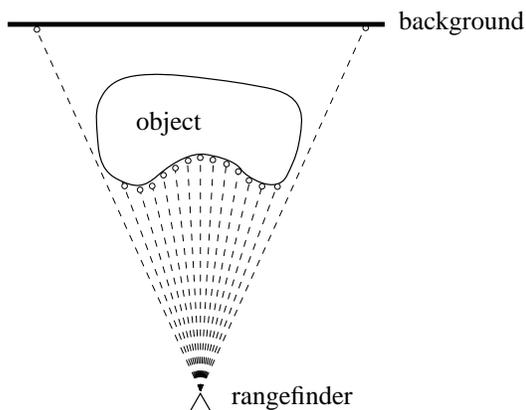


Figure 3-25. Sensing behavior of a typical range scanner in 2D. Scanned points are shown as circles, and the sensor's energy emission is shown as dotted lines.

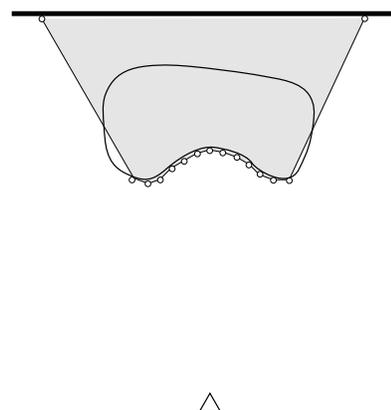


Figure 3-26. 2-D example of a mesh surface determined directly from the sensed points. The surface is shown as thin line connecting the sensed points, the solid formed by sweeping the mesh is shown as the darkened region.

It is due to 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, we must 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 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.

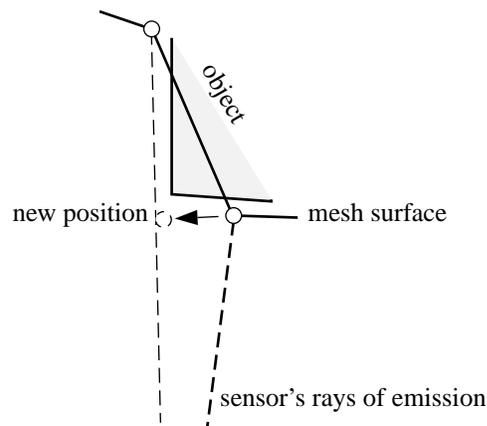


Figure 3-27. Dilation of mesh surface. Shown is detail of a “missed” object surface leading to the situation in figure 3-26: A remedy is to move the sensed point along the arrow from its original position to the new position.

Once the surface element that requires modification has been identified, the direction in which the vertices will be moved must be determined. There are several different ways to decide this; among the choices are:

- In a direction orthogonal to the sensor’s emission angle at that point
- In a direction parallel to the sensor’s baseline
- In a direction found by examining an interpolating function derived from the surrounding data

Each of these methods will behave appropriately in many instances, but will fail under certain conditions. In the system presented in this thesis, we identify one of the element's vertices and move it along the sensor's baseline until it approaches the position where the nearest sample would have been. To differentiate between positive and negative motion on the baseline, the normalized dot product of the surface element's normal with the baseline's positive direction vector is used. An example of this process is shown in figure 3-27: the vertex of an element is moved so that it approaches the next adjacent sensing emission 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 sensed 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 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 figure 3-28, and is shown for real range data in the following section. It should be noted that the magnitudes of these displacements are small ($< 1\text{mm}$) for the range images in this thesis: 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 shown in figure 3-29, a view of the previously-shown model of the bear toy. The left image of figure 3-29 shows the backside of the bear model when no such processing 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

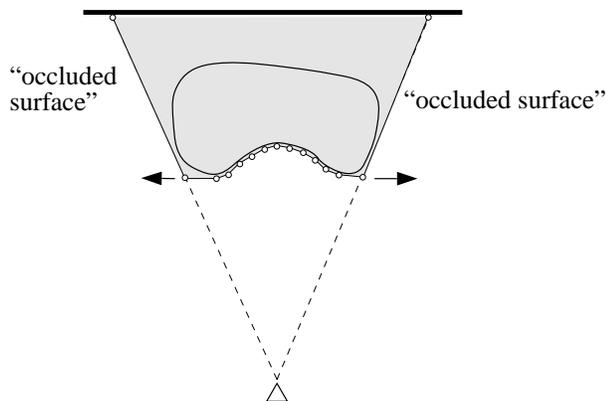


Figure 3-28. Effect of nonuniformly dilating mesh on swept solid. The vertices of the (two) occlusion surfaces are moved until they coincide with the rays of the sensor's nearest adjacent sensing ray, in the direction shown by the arrows. The area resulting from the swept surface is shown in grey and is clearly a superset of the actual object.

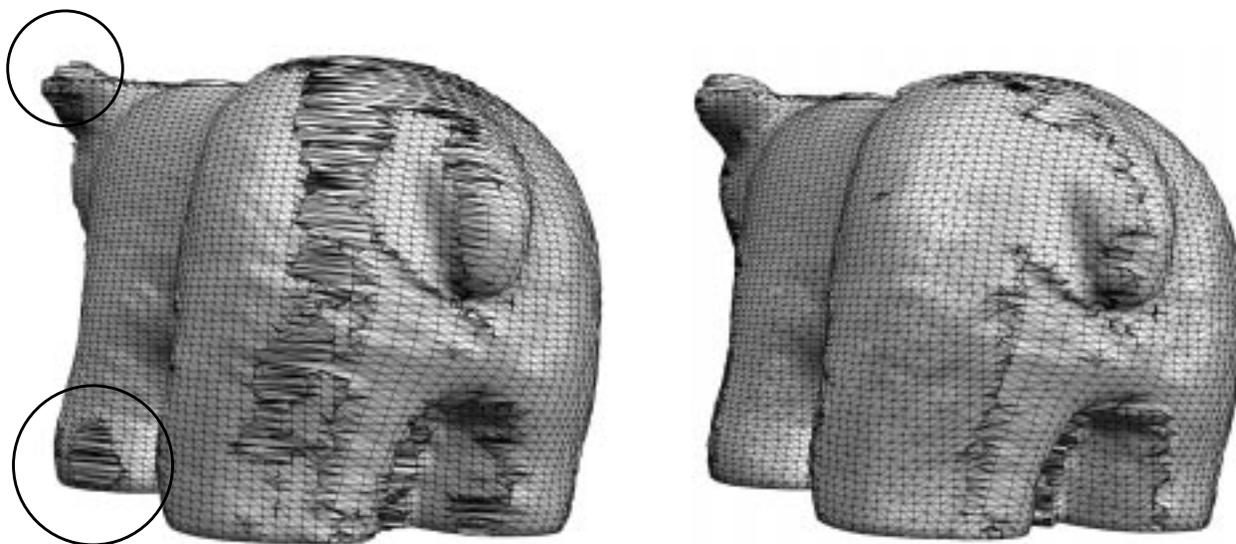


Figure 3-29. Bear model detail showing effect of dilation process on model-building. Left: model built using sensed points directly as vertices in mesh from each view. Right: model built using dilation process on mesh of each range image prior to sweep operation.

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.

Chapter 4 Planning Sensor Viewpoints

4.0 Introduction

The model construction process described so far includes no planning of sensor viewpoints; the turntable supporting the object is rotated equally from one image to the next. This chapter describes how to automatically plan the next viewpoint during the model acquisition process. Planning the next sensor viewpoint is carried out between modeling operations, so that new scene information is always being incorporated into the composite model. The composite model is used to drive the planning process.

The planning process presented here operates by reasoning about occlusion, which has been strongly associated with viewpoint planning in the research literature for some time. By using the occluded volume as a cue to determine the next sensor position, and by also making sure that the sensor is not blocked by any of the object's surfaces, this algorithm guarantees that previously-unsensed scene information will be acquired for the next modeling operation.

4.1 Background

Sensor planning is its own field of study, with a correspondingly large body of work to go with it. Solutions to the sensor planning problem depend on the task, and this is what narrows the amount of previous work referenced here. Traditional sensor planning may be divided into two classes: *Static Sensor Planning* and *Dynamic Sensor Planning*. The Static Sensor Planning problem is that of computing a set of sensor locations for viewing a set of features given a model of a stationary object or scene, a sensor model, and a set of sensing constraints. The Dynamic Sensor Planning problem is more complicated in that it allows the objects to move along known trajectories. For more information on these two problems, see [Abrams 1997] [Abrams *et al.* 1993] [Tarabanis *et al.* 1995b] [Sedas-Gersey 1993].

The planning problem in the object reconstruction task, however, corresponds more to what is called *active vision* than to classical sensor planning. Active vision is based on the idea that alternating between sensing and modification of the sensor's position, orientation, and other parameters will result in increased capabilities for vision systems [Ballard 1991] [Shmuel & Werman 1990]. It is assumed that a model of the sensor is known, along with knowledge of the ways in which that sensor may be positioned. The object or scene is assumed to be stationary, and in practice a maximal bounding volume is often assumed. All planning knowledge is determined by the current state of the scene's model and the known attributes of the sensor.

These methods use such information about the current model as quality of fit, surface normal, and extremal boundary shape, as well as knowledge about what is *not* known

about the model, such as the volume of occlusion. A discussion of the problem and an analysis of strategies for perceptual behavior can be found in [Bajcsy 1988], which categorizes active vision strategies into two classes: bottom-up strategies are driven by the data without a pre-specified task, while top-down strategies rely on a known task to guide the sensing process. The search for information is described in terms of an estimation and control problem, with procedures that determine the number of views required to produce an estimate of a certain quality. Another solution discussed is that of deriving a rule which determines when enough data have been acquired.

There are three basic techniques which provide the basis for the majority of the viewpoint planning methods used. In the first, ray casting is applied to the model to find how much of the occluded surface will be imaged for every sensor position: the sensor position that images the most occlusions is selected. This requires tessellating a viewing sphere to discretize the sensing positions and computing a ray-cast image from each of them, with the disadvantages 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. The peak in the histogram denotes the normal of the greatest 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.

[Connolly 1985] describes two view planning algorithms that take a partial octree model as input. The first, dubbed the *Planetarium Algorithm*, evenly tessellates a viewing sphere

surrounding the object. Recall that an octree consists of nodes labelled empty (the initial state), occupied, and unseen. At each vertex in the tessellation, the area of unseen octree nodes for an origin-centric viewing direction is calculated from a hidden-line view of the model. The vertex of the tessellation with the highest value is used as the next viewing position, since it will eliminate the most unknown area. The second algorithm calculates the number of faces of the octree that have one of the six possible orientations. Only faces that separate unseen and empty nodes are considered; in effect this results in a calculation of the normals of the surfaces making up the occluded volume. The next sensor direction is chosen from the highest number of faces for each of the x, y, and z directions. This method has the advantage of being computable in time proportional to the unseen surface area of the object, which decreases with each view, as opposed to requiring evaluation at every point on a view sphere at each iteration.

[Maver & Bajcsy 1990] use surfaces denoting scene occlusions for planning the next view. After scanning the data once, the occlusions are modeled by polygons orthogonal to the viewing direction. For each polygon, a direction is computed such that the polygon may be completely imaged from that direction, provided there is no self-occlusion there. These direction angles are then inserted into a histogram, with the result that maximal peaks in the histogram represent viewing angles that will image the most occluded areas.

The error of fit of a model to range data from a surface has also been used [Whaite & Ferrie 1990] [Whaite & Ferrie 1992]. In this case, a superellipsoid is fitted to a surface that is assumed to be a segment from a larger data set. An error function is computed for each data point based on the square of the distance from the point to the model surface. The

planning component is uses the fact that maximum improvement in the model can be achieved by performing sensing operations on the parts of the model with the highest error. Therefore, from the errors an estimate can be made of which sensing operation will result in the greatest relative improvement.

[Kutulakos 1994] describes a method of shape recovery that uses only intensity images from a continuously moving camera. The extremal boundary is extracted from each image and its deformation between images is used to construct local surface information. This is done at some point on the extremal boundary by moving the camera in a circle on the plane perpendicular to the boundary and coincident with that surface point, then measuring the change in the shape of the boundary. The curvature of local surface is determined and then this local region is expanded by repeating the process on the points on the edge of this region. This work is unique in that it uses only the silhouette of the object to drive the planning process and also because it is one of the few works that use continuous motion for reconstruction rather than discrete views. his method will have difficulties when applied to objects with certain types of concavities.

As discussed in the previous section, the work presented in [Sobh 1995] integrates vision and CMM data to acquire a CAD model of an object. 2D vision acquires attributes of the object to be sensed, which are then used as input to a sensing agent. The agent is implemented as a finite state machine which monitors the position of the CMM probe with respect to the part. The probe is guided from a distant position to one that is actually on the feature to be sensed, where it then carries out the measuring process at the resolution of the CMM. The ordering of the sensing operation is defined by a recursive definition of

parts that assumes that part features completely enclose one another and do not overlap, an assumption that holds for the machining features of many 2-1/2 D parts.

Recent work by Pito [Pito 1995] removes the need to ray-cast from every possible sensor location by determining a subset of positions that would improve the current model. This is done by extending the border elements of a mesh surface model a short distance in the viewing direction, in a sense modeling limited parts of the scene occlusion. These elements are then used to determine which sensor locations might improve the model. A ray-casting operation is then performed on the model from those sensor locations to determine which provides the best coverage of these elements. This greatly reduces the amount of ray-casting that needs to be done, particularly when most of the object has already been imaged, but it is not clear how well these extended surfaces model scene occlusion at any but the local level.

4.2 Strategies for Viewpoint Planning

Before describing our algorithm for planning in detail, it is important to consider exactly what the benefits from planning are, and how they effect the use of a planner in a modeling system. Certainly two primary benefits of planning are to avoid unnecessary sensing operations and to ensure that certain types of features are acquired. However, there is a cost to planning: it requires analysis of the model and expensive calculations, so it is important to avoid planning views if it is not necessary. Determining precisely when it is necessary is beyond the scope of this work, but it is important enough to merit a cursory examination here.

First, consider what happens when modeling is done without planning: the first few sensing operations from random or pre-set viewpoints will result in acquiring much new information. This is because there is so much unexplored space in the scene that no matter what method is chosen, new viewpoints will usually result in “good” sensor positioning. In addition to this, it is also interesting to note that the area of surfaces tagged “occluded surface” does not always decrease with each additional view. Particularly in the first few sensing operations, the area of these types of surfaces may increase, due to the arrangement of the objects in the scene.

Because set intersection is used to merge the models from different views, the total model volume will always decrease over time and converge at the actual object volume, assuming distinct sensor viewpoints are used. But what about the surface area of the model, in particular the area of surfaces labelled as occluded? The graph in figure 4-1 shows the effect on the total area of “occluded surface” during a typical sensing process. At the beginning of the sensing process, this area may increase or decrease with each additional view. However, as the model surfaces become more accurate, and the remaining “occluded surface” becomes closer to the proper surface, each additional sensing operation will replace “occluded surface” by “imaged surface”, and thus the reduction will tend towards monotonicity.

The underlying principle is that for the initial views it makes no sense to do any planning. Instead, a certain number of sensing operations should be done at equidistant or equiangular positions, so as to acquire an initial model. The number of these initial viewpoints may vary with the task and depend on the amount of scene occlusion. Once a

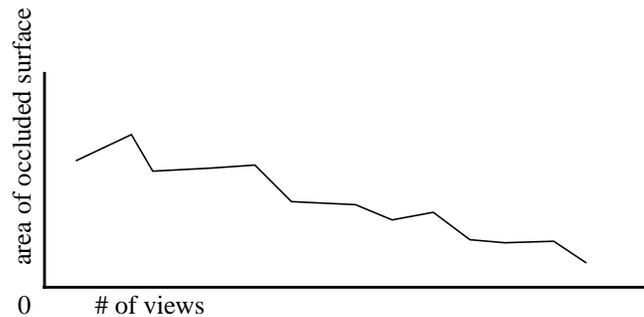


Figure 4-1. Change in area of occluded surface as more views are taken.

preliminary model has been built from these sensing operations, a planning algorithm should be used to determine appropriate viewpoints.

4.3 Using Constraints to Plan for Occluded Viewpoints

The planning component presented here is based on previous work on the sensor planning problem in our laboratory [Tarabanis *et al.* 1995] [Tarabanis *et al.* 1995b]. The planner used in this work is able to reason about occlusion to compute valid, occlusion-free viewpoints for a specific surface on the model. Once an unoccluded sensor position for the specified surface has been determined, the surface may 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 to have an occlusion-free view from the sensor if one exists. Other viewing constraints – such as sensor field of view, resolution, and standoff distance – may also be included in the sensor planning, as is shown below.

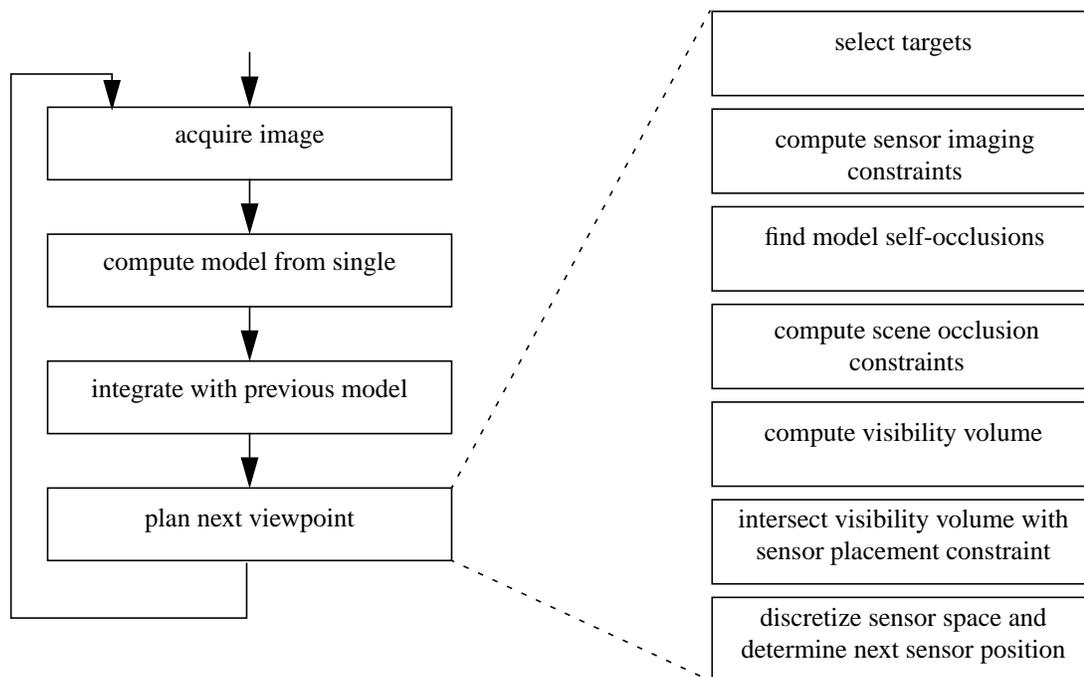


Figure 4-2. Overview of the sensor planning process.

The process operates by considering the entire workspace as the potential location for the next sensor placement, and then constraining this volume until a solution for the next viewpoint is found. The constraints are represented as closed, bounded sets which may be combined to form a plan via set operators. An overview of the process is shown in figure 4-2.

There are three constraints that we shall consider: sensor imaging constraints, scene occlusion constraints, and sensor placement constraints. Sensor imaging constraints are limitations on the imaging process due to the sensor's modality or implementation. For example, if the sensor must be in front of the target in order to image it – as most sensors

must, except penetrating modalities such as x-ray – this will constrain the resulting plan to be in the half-space in front of the surface. Similarly, if the sensor must be within a certain angle of inclination with respect to the surface, this will further constrain the result. Scene occlusion constraints are those due to the fact that parts of the current composite model block some locations in space from viewing the target surface. Finally, sensor placement constraints limit the range of positions in which the sensor may be placed. For sensors attached to 6-degree-of-freedom manipulators, this may be the entire workspace. However, for the majority of implemented systems, and in particular those using laser rangefinders, this constraint is a hemispherical or cylindrical surface. Each of these constraints may be represented as a volume, called V_{imaging} , $V_{\text{occlusion}}$, and $V_{\text{placement}}$ respectively.

The planning process constructs a *visibility volume* V_{target} that describes the set of all sensor positions that have an unoccluded view of the target for a specified model. V_{target} is determined entirely by volumes representing the sensor imaging and scene occlusion constraints. It can be computed by determining V_{imaging} , which may be thought of as the visibility volume for the case where there are no occlusions, and subtracting $V_{\text{occlusion}}$, the volume containing the set of sensor positions occluded from the target by model surfaces. Since $V_{\text{occlusion}}$ is the union of all O_i , the volume that is occluded from the target by model surface i , we have:

$$V_{\text{target}} = V_{\text{imaging}} - \bigcup_{\forall i, i \neq \text{target}} O_i \quad (\text{EQ 4-1})$$

Once this visibility volume has been computed, it is only necessary to include the volume representing the placement constraint to determine the plan:

$$V_{\text{plan}} = V_{\text{target}} \cap V_{\text{placement}} \quad (\text{EQ 4-2})$$

4.4 Selecting Target Surfaces

We must be able to select a subset of the targets for planning purposes if there are more than one, as is the case here. It is computationally undesirable to plan for all possible targets, especially since the iterative nature of this method assures the acquisition of all relevant targets by the time the modeling is completed. This implies that it is better to plan for some targets rather than others, and this is in part due to the contiguous nature of the surfaces tagged “occluded”. These surfaces are on the boundary between imaged and occluded space, and hence tend to have large area. When acquiring an image that contains a particular surface, much of the surrounding surfaces are also acquired. It therefore makes sense to consider those parts of the model with a high density of occluded surface. Of course, “occluded” model surface area is not strictly related to actual unimaged object surface area, as the counter-example of figure 4-3 shows. However, because in the

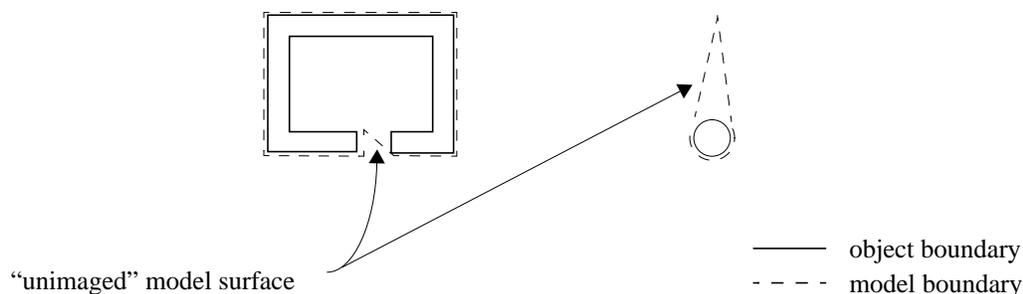


Figure 4-3. Unimaged model surface is not necessarily related to unimaged object surface. On the left, a small unimaged model surface corresponding to a large unimaged object surface. On the right, a large unimaged model surface corresponding to a small unimaged model surface.

majority of cases “occluded” surfaces lie close to the boundaries of true surfaces, or at worst delimit unexplored volume in the workspace, using them to guide exploration is a sound strategy.

A good target is one that has large area and is compact in the sense that its visibility volume is as small as possible. This can be assured if the polygonal target surface is as close to being a regular convex polygon as possible, i.e., with all sides the same length. In the case of the system described here, where the surface elements are triangular, this means that small aspect ratios are favorable. The reason this is important is that elements such as these have visibility volumes that have close to a minimal surface area for the corresponding target surface area, and hence are less likely to be occluded by model features.

For a target to be properly acquired, it needs to satisfy certain detectability conditions. Most notably, it must be large enough so that it is sampled to the effective resolution by the rangefinder. Note that this assumes that the target behaves like a true imaged object surface, even though it is in fact part of an occlusion situation in the scene. By treating it like a true surface, we are in effect saying that by scanning it at the stated resolution, we will either resolve the underlying surface or, if not, at least acquire new information about other occlusions there.

For a target to satisfy this detectability constraint, it must be of such a size and orientation that it can be sampled by the sensor. To guarantee this, it is necessary to ensure that the imaged cross-section of the feature subtends an angle $\phi \geq 2\theta$ (where θ is the beam deflection between samples) when imaged by the sensor, as shown in figure 4-4. This, in

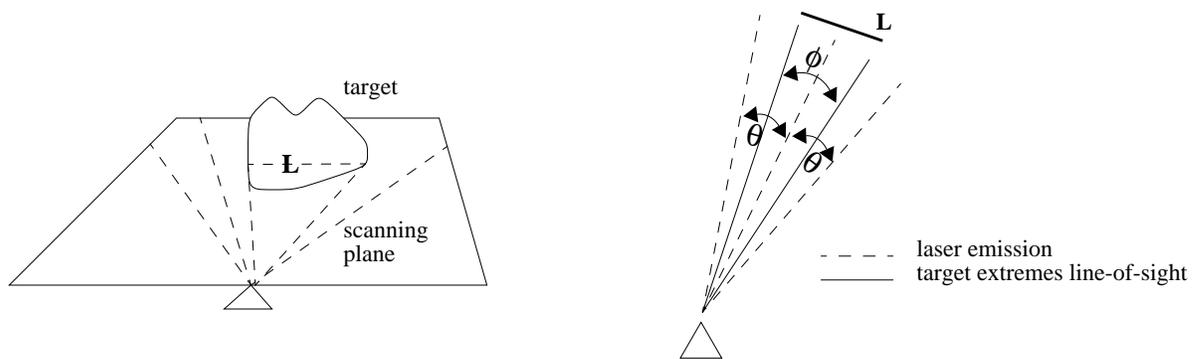


Figure 4-4. Constraints on sensor imaging due to projected target area. In the rangefinder's two-dimensional scanning plane, the target appears as a line segment L (left). In order to guarantee that a target is acquired there must be three samples over the width of the line segment, implying that the target subtends an angle $\phi \geq 2\theta$.

turn, means that different-sized targets will have different-shaped volumes for their detectability.

4.5 Computing Sensor Imaging Constraints

As mentioned previously, sensor imaging constraints limit the possible visibility volume by considering the sensor's ability to image a particular target surface. This constraint is itself represented as a volume V_{imaging} which describes the sensor locations from which a sensor can effectively image the target surface. The factors that contribute to the imaging constraints are the modality of the sensor and the geometric parameters describing its ability to acquire images, such as breakdown angle α (the maximum inclination of the sensor with respect to a surface), depth of field, standoff (which describes the closest the sensor may be to the target), and its far plane and resolution (both of which limit the farthest distance the sensor may be from the target). Each of these parameters affect the

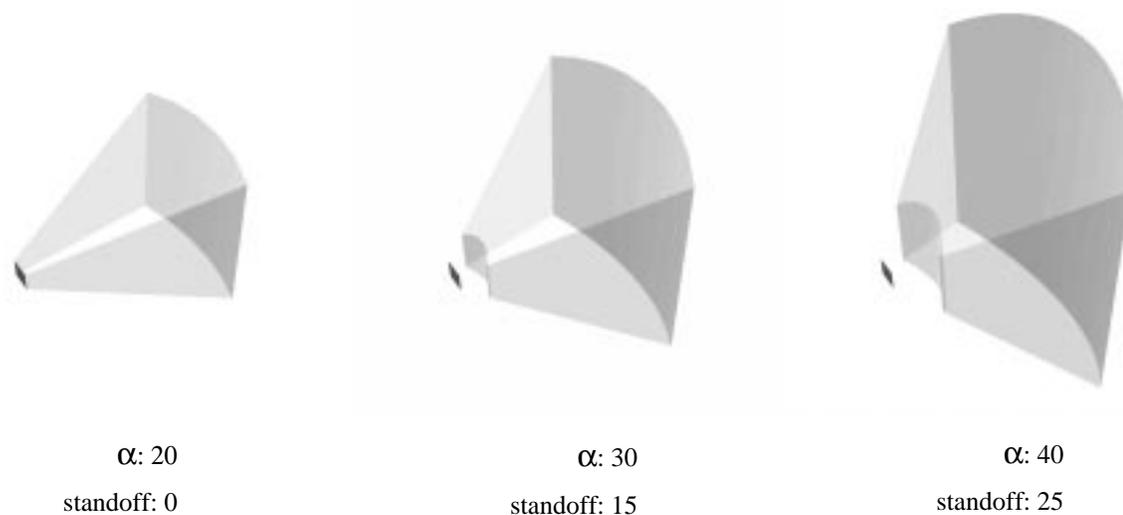


Figure 4-5. Variation in the shape of the volume representing imaging constraints with respect to breakdown angle α and standoff (in mm). Depth of field and resolution are assumed infinite; the target is the small dark polygon to the left of the volume.

shape of a volume representing this constraint: some variations for a rectangular target are shown in figure 4-5. In this figure, V_{imaging} is shown for three hypothetical sensors and a rectangular target surface. In each of the three images, the grey volume describes the set of positions from which the sensor can appropriately acquire the target. Thus, in the left image, from every point in the grey volume the sensor may acquire an image and still be within the breakdown angle of 20° . The second example (middle of figure 4-5) shows V_{imaging} for a greater breakdown angle, which gives the volume a fatter shape. In addition, a standoff distance has been modeled by removing all sensor positions that are closer to the center of the target than the standoff parameter of 15mm. This standoff would be important to consider if the sensor we were modeling could not acquire data at closer ranges than this. The last example, at the right of figure 4-5, shows V_{imaging} for a sensor with a breakdown angle of 40° and a standoff of 25mm.

An assumption made when computing V_{imaging} is that the sensor is able to find the appropriate orientation to acquire the target: we do not explicitly compute an orientation constraint. This means that the final plan assumes that the field-of-view of the sensor is wide enough to completely acquire any targets planned for. In our implementation, it is assumed that the sensor's field-of-view covers the entire workspace, so the field-of-view is guaranteed to be wide enough.

The volume V_{imaging} may be generated from a polygonal target surface by performing an extrusion operator on the surface, in the direction of the surface normal, with a draft angle equal to the breakdown angle α . This computation may be done in time proportional to the number of edges in the target, and thus consumes very little computational resource. If a more sophisticated model of the sensor is used, this computation may become more significant.

4.6 Computing Occlusion Constraints

The occlusion constraints further restrict the positioning of the sensor for a specified target by disallowing all positions in space from which the target is occluded by any part of the environment. Since the current composite model describes all that is known about the layout of the environment at a particular moment in the modeling process, it is the surfaces in the composite model that must be checked to see if they occlude the sensor from the target. For each model surface i that potentially occludes the target, a volume O_i is constructed that represents the space that is disallowed for the sensor. To describe the entire space from which it is not possible to see the target, it suffices to compute:

$$V_{\text{occlusion}} = \bigcup_{\forall i, i \neq \text{target}} O_i \quad (\text{EQ 4-3})$$

– that is, the boolean union of these volumes over all surfaces in the composite model comprises the space from which it is not possible to see the target, and hence captures the occlusion constraints.

It remains to be shown how to decide which model surfaces occlude the target, and how to compute O_i for each of them. For the former problem, it is clear in the general sense that *every* model surface may potentially occlude the target. This means that a union operator must be applied to a number of volumes equal to the number of surfaces in the model (minus one, since the target surface cannot occlude itself) for every target planned for. However, this requirement may be dramatically reduced by considering the following:

- only those surfaces with normals that have some component anti-parallel to the target's normal need be considered – that is, back-face culling may be performed.
- only those surfaces that are within the volume V_{imaging} need be considered; it is not necessary to consider surfaces outside the space in which the sensor may operate.

Together these conditions greatly reduce the number of surfaces that need to be examined for occlusion situations. Further techniques to accelerate this process are given below in section 4.9.

The computation of O_i for a specific target and model surface i is based on the algorithm for feature visibility described in [Tarabanis 1996]. It is based on geometric

decomposition of space into volumes from which a specified model feature either can or cannot be fully imaged by an ideal sensor. If a target surface – that is, an “occluded” surface in the model – is used as a feature, this algorithm may be used to produce valid viewpoints for any target- and occluding-surface pair. The technique is based on binary space decomposition, and operates by traversing the boundary of the target and the occluding surface, as shown in figure 4-6. The traversal generates a partitioning element (a

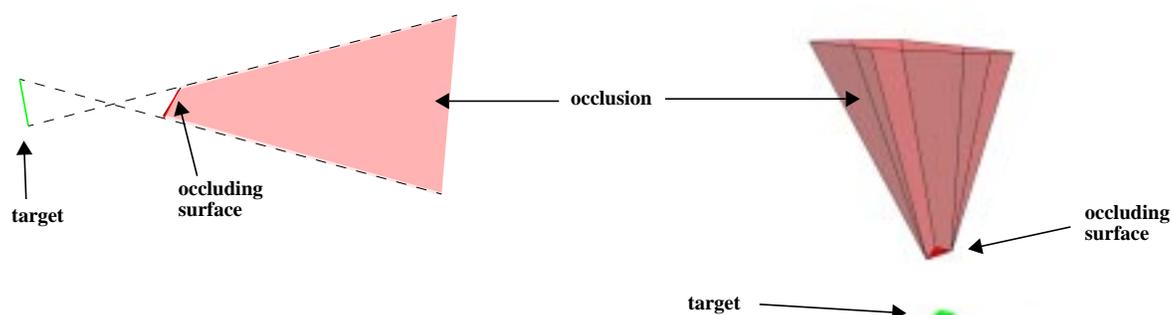


Figure 4-6. Occlusion computation by spatial decomposition. An example in 2 dimensions is shown at the left, where the target and occluding “surfaces” are line segments, and the occluding “volume” is a planar area. At the right is an example in 3 dimensions, using polygons to represent the target and occluding surfaces, and a volume to represent the occlusion.

line in 2-D, a plane in 3-D) for each pair of boundary elements, which is then used to produce the occluding area or volume. In the 2-D example shown, the target (in green) and occluding surface (in red) are line segments, the boundary elements are their endpoints, the partitioning elements are lines constructed from the endpoints, and the occlusion is the red area. For the 3-D example, the target and occluding surfaces are polygons, the boundaries are edges and vertices, the partitioning elements planes, and the occlusion is the volume. Computationally, this technique operates in time linear in the combined number of edges of the target and occluding surfaces. It should be noted that this

algorithm assumes convex boundaries for both the target and occluding surface. If this is not the case, a convex decomposition algorithm must first be applied.

4.7 Computing Sensor Placement Constraints

Sensor placement constraints describe the physical locations in which the sensor may be placed, and are typically derived from a description of the manipulator used to position the sensor or the scene. In some instances, this may be a 6-degree-of-freedom manipulator such as a PUMA 560, in which case the sensor placement constraint may be represented by a sphere of finite radius (assuming there is no overlap between the workspace and the manipulator's unreachable positions). If the manipulator is constrained to lie a fixed distance from the workspace origin, as in [Whaite & Ferrie 1992], or if the sensor is fixed in location and the object is rotated, then this constraint may be represented by a spherical shell of the appropriate radius. In practical systems, it is likely that there will be an inadmissible region near the center of the workspace where the object is typically placed, to prevent the manipulator from attempting to position the sensor there. This would be represented by an additional constraint giving the minimum distance from the origin the manipulator may attain. These volumes representing two of these situations are depicted in the top row of figure 4-7.

In our experimental setup, the manipulator may move the sensor only in the world z coordinate, but there is additional positioning due to the turntable that may rotate the object around the z axis. Thus, the workspace is a cylindrical shell. This configuration is common in laser-rangefinder systems, because it maximizes the stability and repeatability

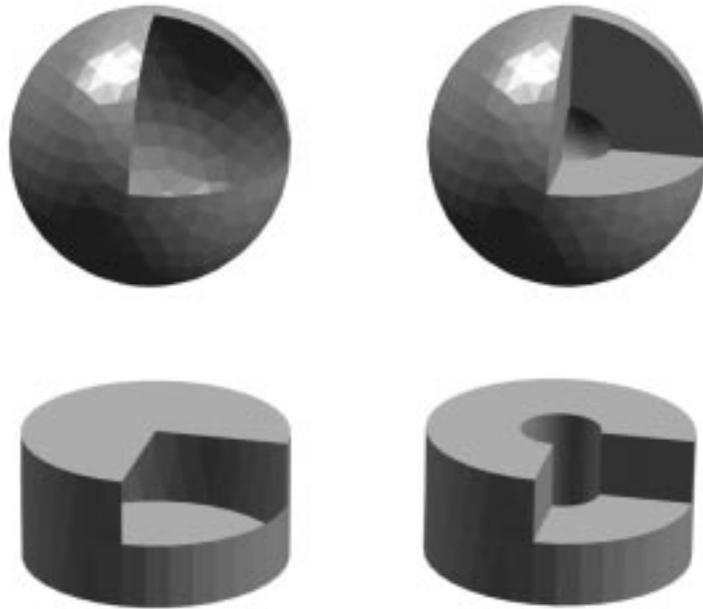


Figure 4-7. Volumetric representation of sensor placement constraints. Examples shown are for a system with a 6-DOF manipulator (top row) and a system with 1-DOF manipulator/turntable combination: shell for positional freedom constrained to a specific distance from the origin (left) and a thick shell for positional freedom constrained to be at least a certain distance from the origin (right).

of the rangefinding unit, while still allowing enough maneuverability to capture most objects. Sensor placement constraints for systems that have a manipulator workspace resembling cylinders are illustrated in the second row in figure 4-7. The implementation described in this thesis is represented by the model on the left.

4.8 Combining Constraints

The three constraints described above must be combined to determine the set of regions or cells from which the sensor may properly acquire the target. Each of these constraints is represented by a set in three-dimensional space, and a solution may be found by applying set operators:

$$(V_{\text{imaging}} - V_{\text{occlusion}}) \cap V_{\text{placement}} \quad (\text{EQ 4-4})$$

Note that the final result is a set of points, lines, surfaces, or volumes, and may be the empty set if there is no solution.

4.9 Computational Considerations

The planning process is one of the more computationally expensive aspects of the model acquisition process. However, the cost may be drastically reduced if the computation of the constraints is done in serial fashion, and if information from one constraint is used to reduce the amount of calculation in the more computationally intensive calculations. To illustrate this, consider the current costs of calculating the constraint volumes for a target surface f :

- sensor imaging constraint V_{imaging} : $O(m)$, where m = the number of edges of f .
- occlusion constraint $V_{\text{occlusion}}$: $O(n^2)$, where n = the number of occluding surfaces. This calculation is dominated by the cost of unioning the O_i volumes.
- sensor placement constraint $V_{\text{placement}}$: $O(1)$, as this is independent of the surfaces in the model, and may be computed off-line.

It is clear that for any real-world situation the most computationally expensive computation is that of the occlusion constraint. However, it can be seen that only those surfaces that intersect the volume described by the sensor imaging constraint need to be considered¹, because no model surfaces outside of that volume may come between the

1. Standoff should not be included here because it may prevent the consideration of occluding surfaces that are closer than the standoff distance.

sensor and the target surface. The sensor imaging constraint can be calculated very rapidly, because the vast majority of surfaces in our model are triangular, the rest being simple polygons with a small number of edges. Thus, if V_{imaging} is calculated first, and then used to determine the candidate model surfaces for the occlusion constraint, a considerable amount of calculation is avoided. In particular, consider that many of the target surfaces have no model surfaces that might possibly block the sensor from them, and so the occlusion calculation is avoided entirely. Candidate model surfaces may be evaluated for possible occlusion simply by testing if they intersect V_{imaging} .

The above ordering of the application of constraints in no way affects the outcome of the planning process. However, in the interest of further reducing the amount of time spent in the planning process, there are two other optimizations that do reduce the accuracy of the computed plan: decimating the surface of the composite model and discretizing the space representing the sensor placement constraint. These methods reduce the accuracy because they use approximations to either the model or the continuous-space plan, and hence are not exact solutions.

The composite model is a dense polygonal representation of all that is known about the currently explored workspace. Because it consists of a great number of small polygonal surfaces, many of which are coplanar, it is a prime candidate for application of a mesh decimation scheme, e.g. [DeRose *et al.* 1993] [Schroeder *et al.* 1992] [Turk 1992]. At the point in the planning process where occlusion is computed, accuracy may be sacrificed for speed by reducing the number of occluding polygons by merging nearby coplanar surface elements. In our implementation, we use a variant of Simplification Envelopes (SE)

described in [Cohen *et al.* 1996]. SE is a method that generalizes offset surfaces to determine an interior and exterior boundary between which the resulting decimated model must lie, and so has the desirable property of having an absolute bound ϵ for the distance between the original surface and the simplified one. The distance from these boundaries to the original surface is given as a single real number input to the algorithm. However, in its original form SE produces models whose representative sets may be subsets of the original model, and which may therefore miss some occlusion situations. To prohibit this effect, in applying the SE method it is necessary to allow only the use of an exterior offset surface, so that the resulting simplified model is always a superset of the model from which it is derived, as shown in figure 4-8. In addition to this, we have modified the algorithm so that

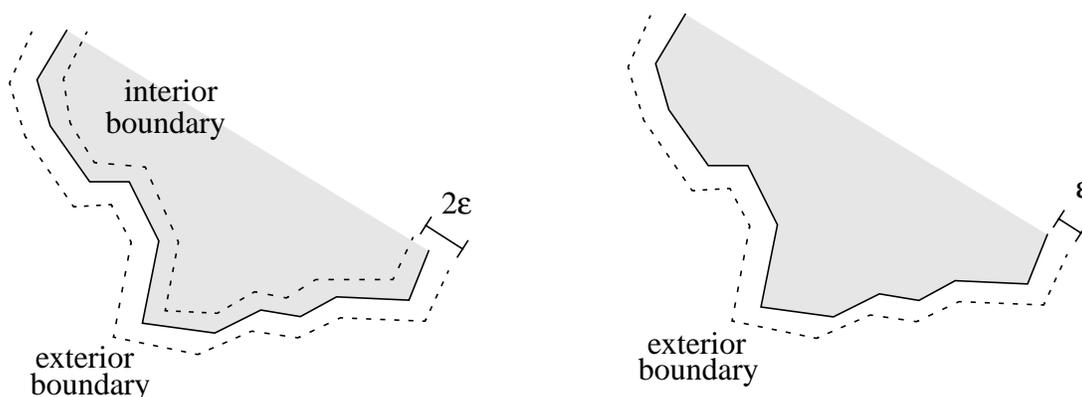


Figure 4-8. Simplification Envelopes. The original algorithm produces a simplified model whose surface lies between an interior and exterior boundary, offset from the model surface by ϵ (left). The variant used to prevent underestimation of model volume produces a model boundary between the original surface and an exterior boundary only (right).

it retains the surface-type tags which describe the surface elements as imaged or occluded, and disallows any merging process between them. This is done to ensure that the resulting surfaces are composed of only one type of modeled surface: it is not clear what the

resulting surface describes if it is due to the merging of an imaged and an occluded surface.

4.10 Terminating Conditions

A primary consideration for a system that plans dynamically is that of determining when planning is no longer needed and the modeling process is complete. There are a variety of methods that may be used, among them the following, in decreasing order of difficulty:

- continue until there are no surfaces tagged “occluded” left.
- continue until there are no surfaces tagged “occluded” that may be planned for.
- continue until there are no surfaces tagged “occluded” with area greater than α .
- continue until the total area of surfaces tagged “occluded” is less than α .
- continue planning dynamically for a certain amount of time, number of views, etc.

Each of these may be applied in different situations, but none of them used alone will give the best results. The first method is not generally applicable because it will fail to terminate in situations where the entire surface of the object is not imageable, as in figure 4-9. The second method provides a solution to this problem: continue to plan for surfaces until the plans generated are no longer attainable. The main drawback to this method is that it will continue to plan in the situation where there are many possible targets, each of which is disconnected and of small area. The third method is more forgiving in that it will only try to plan for targets greater than a certain area. The fourth method just operates until the total unimaged area is less than a threshold, and again it

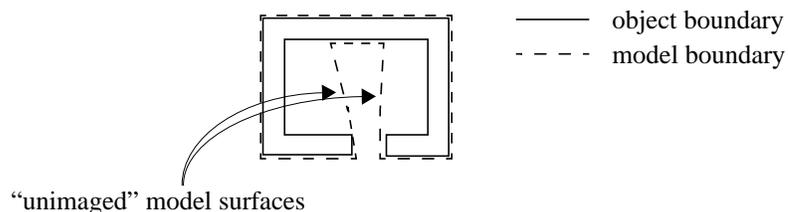


Figure 4-9. A situation where it is impossible to image the entire object surface, or all model surfaces labelled “occluded”.

may not terminate in certain situations. Finally, there are the methods that do not take into account model quality as a terminating condition, and that lie somewhere between the systems with a fixed number of predefined viewpoints and those with full automatic model acquisition.

Our approach combines the second and third conditions to determine if the planning should be discontinued: if there are no unimaged surfaces for which a valid plan may be computed, or there are no unimaged surfaces of area greater than a threshold value, the planner assumes the model has been acquired properly. This guarantees that new scene information will be acquired with each new view, and at the same time there will always be a point at which the system will terminate.

Note that just because a particular unimaged surface has no acceptable plan at a particular stage in the modeling process does not mean it may not be planned for later on. This is because the model changes after every integration operation: an unimaged surface that was previously occluded by other unimaged surfaces may be unoccluded subsequently. However, if an unimaged surface is occluded completely by imaged surfaces (that correctly model parts of the scene) then it is safe to disregard those surfaces for the

remainder of the planning for that scene, as the occlusions are a permanent part of the model and will not change with subsequent views.

As discussed in the previous chapter, the final model retains all surface-type attributes, so that even if the object is not completely imaged, it is still possible to analyze the model appropriately based on the different types of surfaces.

4.11 Example: Single Target, Synthetic Data

To better illustrate the entire planning process, we show in this section plan generation for a synthetic model composed of three surfaces. In this case, one of the surfaces has been chosen as the target, and is depicted as the green polygon in figure 4-10. There are two other model surfaces, shown as red polygons in that figure, which may contribute to occlusion situations for the sensor if they are not considered in plan generation.



Figure 4-10. Synthetic planning example. The model consists of three surfaces: one target (shown in green) and two other model surfaces (shown in red).

The first step is to compute V_{imaging} for the green target surface. For this example, we are using a 40° breakdown angle to generate the volume, which is shown in grey in figure 4-11. In order to better visualize this example, the volume has been cut off at the top of the

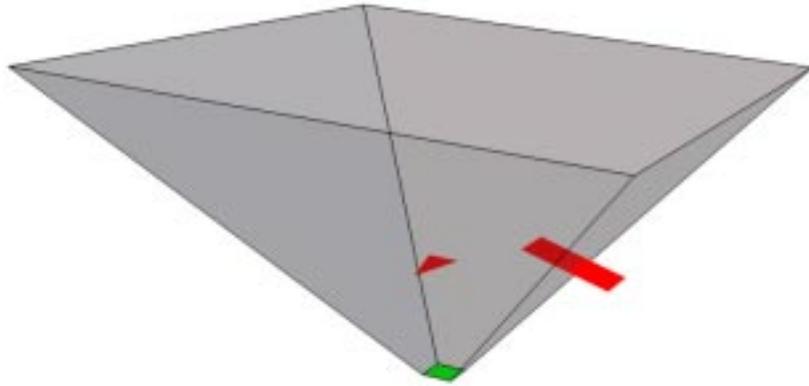


Figure 4-11. Example: sensor imaging constraint. V_{imaging} is shown (in grey) for the green target.

image; for most sensors the far bound is not planar but a curved surface, better approximated by a spherical boundary.

The second step is to compute $V_{\text{occlusion}}$ for the target and the remaining model surfaces. As stated earlier, this is done by computing the occlusion due to each model surface individually (shown as the red volumes in figure 4-12), then unioning them together. In this case, there are only two other model surfaces, and their occlusions are disjoint. Any sensor positions contained inside the red volumes will not be able to properly image the green target due to occlusion by other model surfaces.

This allows the simple computation of the visibility volume V_{target} by subtracting $V_{\text{occlusion}}$ from V_{imaging} . The resulting volume describes, in continuous three-dimensional space, the set of positions from which the sensor has an unoccluded view of the entire target surface (shown in figure 4-13). As mentioned above in section 4.9, Computational Considerations, it is not necessary to compute $V_{\text{occlusion}}$ for every model surface. Only

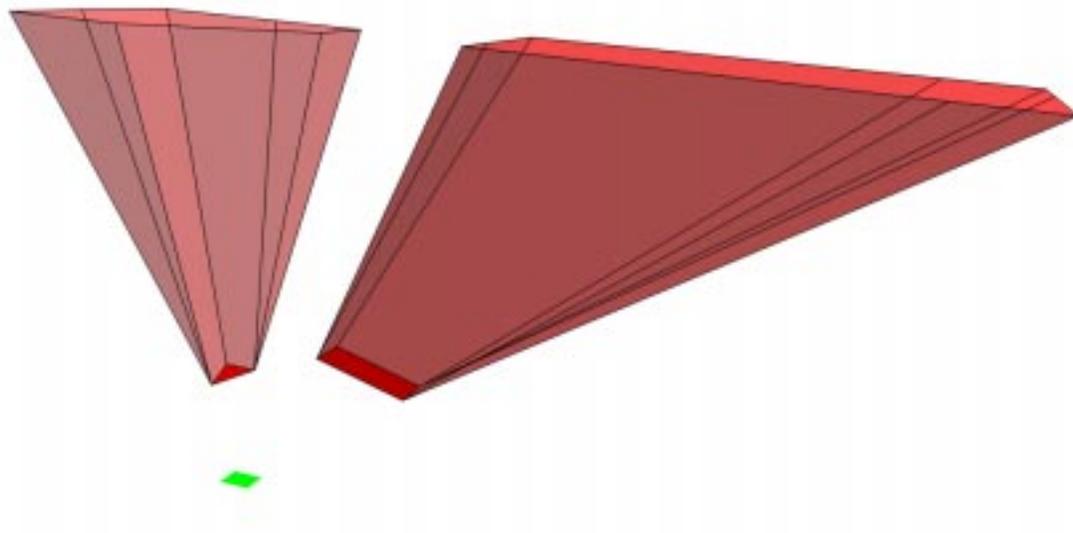


Figure 4-12. Example: occlusion constraint. $V_{\text{occlusion}}$ (shown as the red volumes) computed from the target and occluding surfaces.

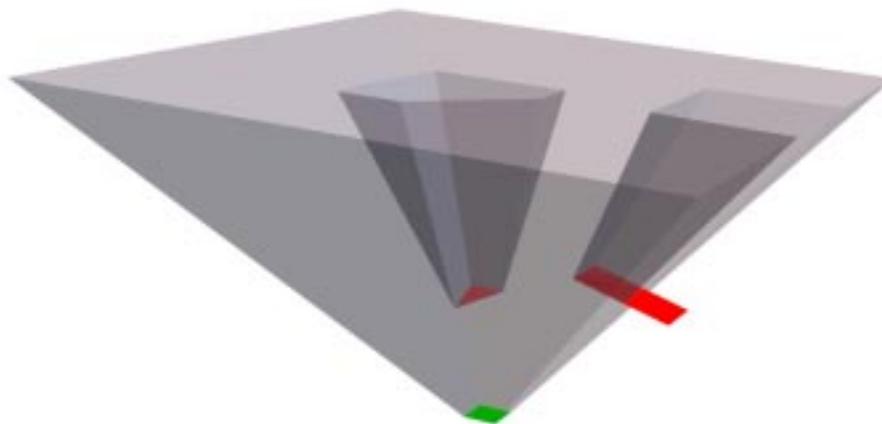


Figure 4-13. Example: Visibility volume. The grey volume is the result of subtracting $V_{\text{occlusion}}$ from V_{imaging} .

those surfaces that intersect V_{imaging} need be considered in the computation of $V_{\text{occlusion}}$, as both red surfaces do in this example.

However, the sensor's positional freedom, which is determined by the type of manipulator used, may prevent parts of this solution from being viable. Thus it is necessary to consider a model of the sensor's positional capability, which is described by $V_{\text{placement}}$. In this example, we will simulate the use of a Cartesian manipulator that can position the sensor within a rectangular space, shown in blue along with the visibility volume V_{target} (in grey) in figure 4-14. As can be seen in this image, there are many positions available to the sensor which will not provide a satisfactory image of the target, due either to occlusion situations or to limitations on the sensor's imaging capabilities. These positions are shown in the figure as those that are interior to the blue volume (i.e., contained by $V_{\text{placement}}$), but exterior to the grey volume (i.e., not in V_{target}). Conversely, there are positions from which the sensor has an unoccluded and otherwise acceptable view of the target but where it cannot be positioned due to the shape of the manipulator's workspace. These positions are those in the figure that are interior to the grey volume (i.e., contained in V_{target}) but exterior to the blue volume (i.e., not in $V_{\text{placement}}$).

The positions in which the sensor can be placed so as to properly image the target, and not be blocked by any other model surface, are described by the intersection of V_{target} and $V_{\text{placement}}$. This intersection is shown in figure 4-15 as the grey volume shown inside of $V_{\text{placement}}$. This volume is a continuous-space solution to the planning problem for the green target, and may now be analyzed in a variety of ways. Because it is represented volumetrically, its volume, center of mass, and higher-order moments may be calculated.

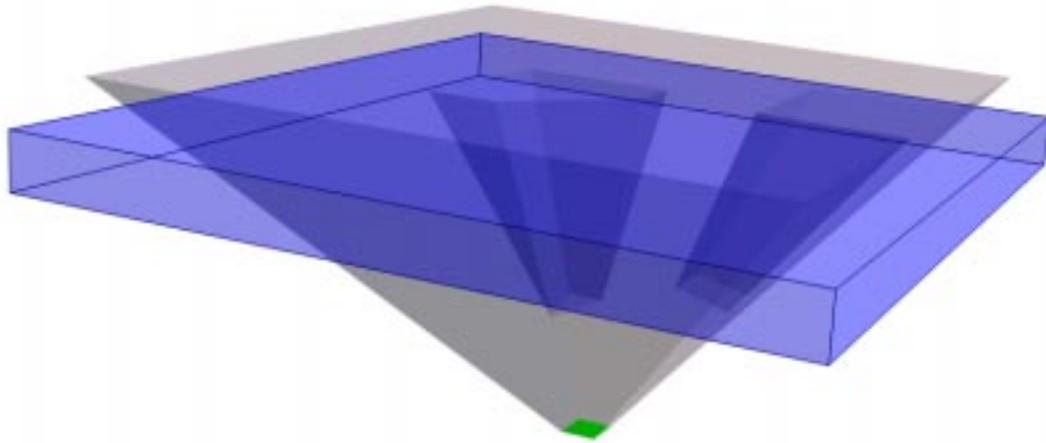


Figure 4-14. Example: including sensor placement constraints. Target surface (in green) with visibility volume shown in V_{target} (grey volume) and sensor placement constraint $V_{\text{placement}}$ (blue volume)

Or, if so desired, it may be used to determine a discretized planning solution if the more general continuous solution is unnecessary. This is shown by example in figure 4-16, where a discrete sensor model is represented by rectangular elements that model individual sensor positions, in this case an 11x11 grid. To compute the discrete plan, all that is necessary is to test each element for inclusion in the continuous solution. In the figure, this is depicted by having acceptable positions shown in green, unacceptable ones in red. As an extension to this, the size and shape of these elements may be used to denote uncertainty in sensor placement, for example by a disk or ball of radius equal to the positional uncertainty.

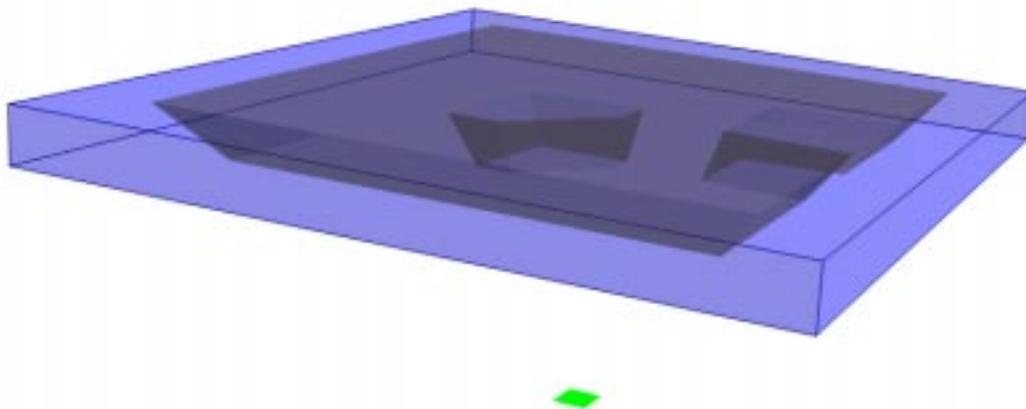


Figure 4-15. Example: continuous-space planning solution. The intersection of V_{target} and $V_{\text{placement}}$, shown in grey inside $V_{\text{placement}}$ (shown in blue).

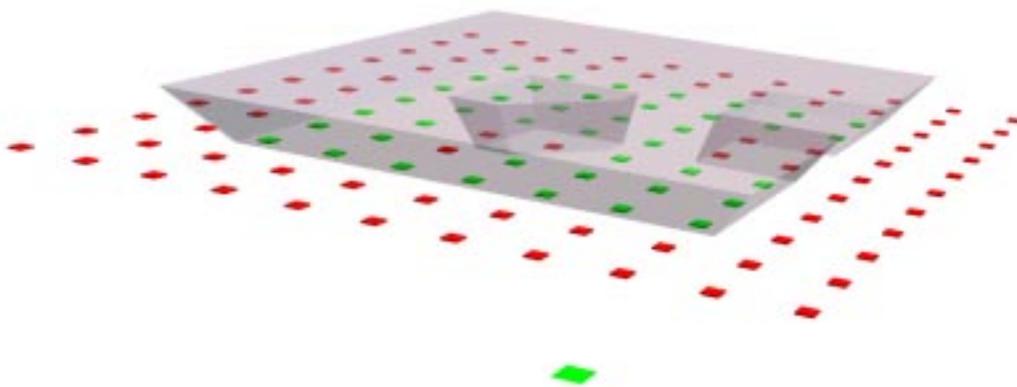


Figure 4-16. Example: discretized solution. If a discrete solution is desired, it may be computed by testing each location for intersection with the continuous-space solution (bottom): here the discrete samples are shown in green if the target is visible, red if not.

4.12 Example: Strut Model

We further demonstrate this planning process by building a model from distinct views of the object shown in figure 4-17, which is a strut-like part. The planning for the sensor orientation is done by the algorithm described 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. The models acquired from these images, and the resulting composite model are shown in figure 4-18 and figure 4-19. The general shape of the part is already quite evident in this composite model after two views. A target is designated on this composite model by selection from one of the surfaces tagged “occluded surface”, and the planning algorithm constructs the plan shown in figure 4-20. This plan is executed by rotating the turntable to place the sensor within the green visibility volume, in this case by 83° . The image taken from that sensor location is used to produce the model shown at the left in figure 4-21. At the right of that figure, the new composite model is shown. Again, a target is designated and a plan produced, which is shown in figure 4-22. The turntable is rotated 134° to move the sensor into this visibility volume, and the model at left in figure 4-23 results. This final model is shown at the right in figure 4-23, and rendered as a mesh in figure 4-24. As 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. Because this is an incremental method, additional scans may be performed to improve the quality of the model.



Figure 4-17. Photograph of strut part.



Figure 4-18. Initial models from strut acquisition. The solid models constructed by sweeping the mesh surface of the first two range images are shown.



Figure 4-19. The composite model of the strut after two views.

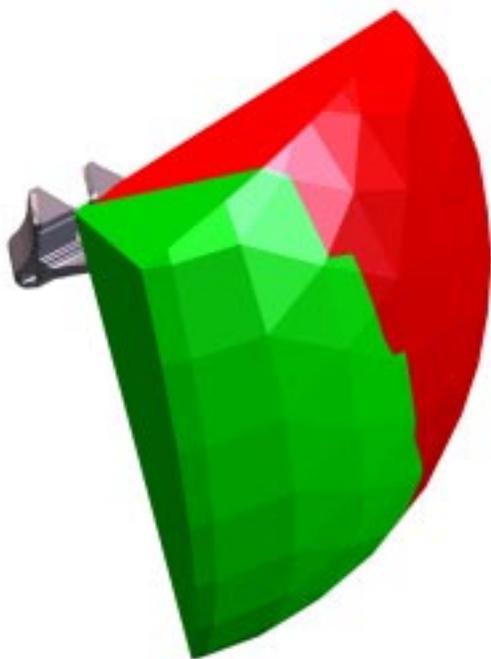


Figure 4-20. Occlusion computation for a target on the composite model. The entire green and red volume represents V_{imaging} for a target from the composite model's "occluded surface". The red space is $V_{\text{occlusion}} = \cup O_i$, the union of sensor positions occluded from the target by model surface i . The green space is the visibility volume $V_{\text{target}} = V_{\text{imaging}} - \cup O_i$, i.e. the valid positions for a sensor viewing the target.

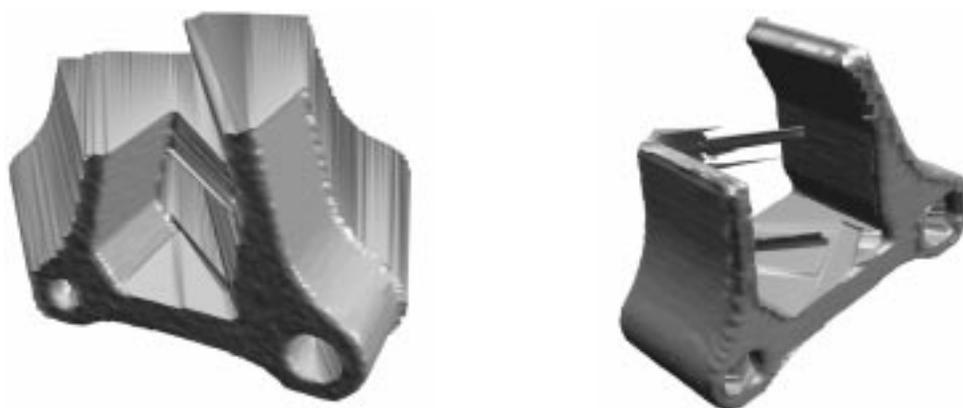


Figure 4-21. The solid acquired from the third range image via the first plan (left), and the composite model after three views. The composite model is now very similar to the object, but there are still some occlusion surfaces between the strut's arms.

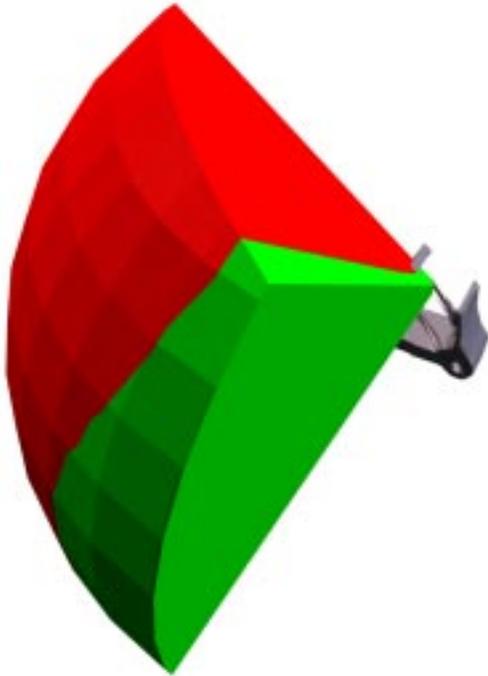


Figure 4-22. Result of sensor planning for a target specified on the “occlusion surface” of the composite model in figure 4-21. Again, red volume specifies points that are occluded from seeing the target, green volume describes the valid sensor positions.

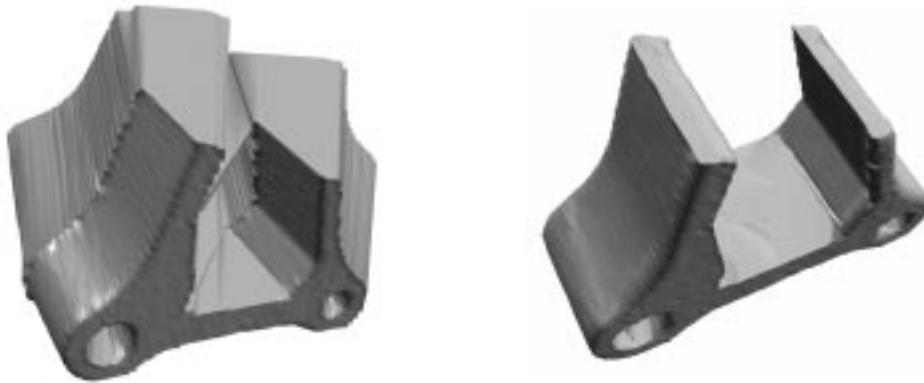


Figure 4-23. Fourth model acquired according to the second plan (left), and the composite model after integration (right).

4.13 Considering Multiple Targets

In the above 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 the targets into account. A more sophisticated method is to plan using multiple targets, and select the sensor position that images the most target area. Although in many sensing

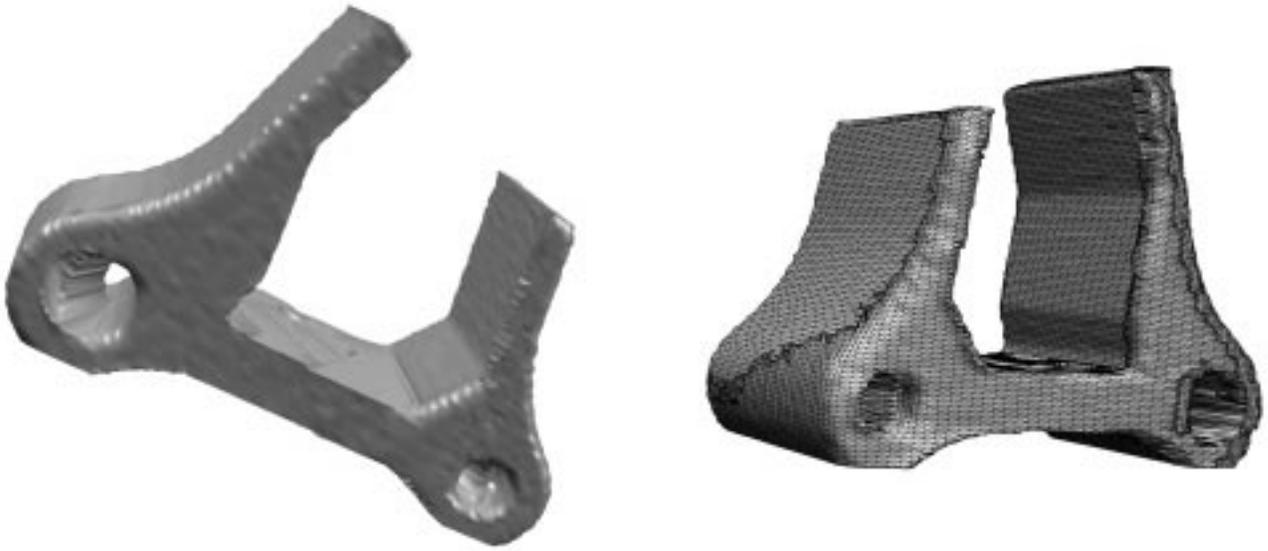


Figure 4-24. Final strut model, shown rendered (left) and as a mesh surface (right). Note the through-hole acquired in the rendered model.

scenarios this may be unnecessary, we show here how multiple targets may be used in planning. In this example, the composite model of the strut after two views is used to determine a plan that considers a number of targets. This is achieved by decimating the surfaces tagged “occluded” and selecting from the result a subset of surfaces with the largest area. In this example, the 30 occluded surfaces with the largest area are selected as targets and the visibility volume V_{target} is generated for each of them.

The plans are computed for two sensor configurations: one with a sensor that can be placed with full rotational freedom on the surface of a sphere, and the other for a sensor that is constrained to lie on the surface of a cylinder from which it can only sense horizontally. This latter configuration is typical of systems such as ours that use a turntable from which a sensor operates at a fixed distance. The plan for full rotational freedom is

shown in figure 4-25. At the left of the figure is the set of visibility volumes for the 30 largest targets by area. At right in the figure is the result of intersecting those volumes with $V_{\text{placement}}$, which in this case is modeled by a sphere. Darker regions in the final intersection represent sets of positions in sensor space where more than one target is visible, thus their individual plans overlap. The same process is applied to the more constrained sensor configuration, which is shown in figure 4-26. Again, overlapping plans appear as darker regions which may then be discretized or searched for the position that images the most target area (see [Preparata & Shamos 1985] for techniques that address this issue). At the bottom of this figure, we show the resulting continuous-space plans “unwrapped” from the cylindrical $V_{\text{placement}}$ onto a plane, making the overlapping regions quite evident.



Figure 4-25. Planning for multiple targets. Shown here is a plan for a sensor with full rotational freedom and with positional freedom constrained to a viewing sphere.

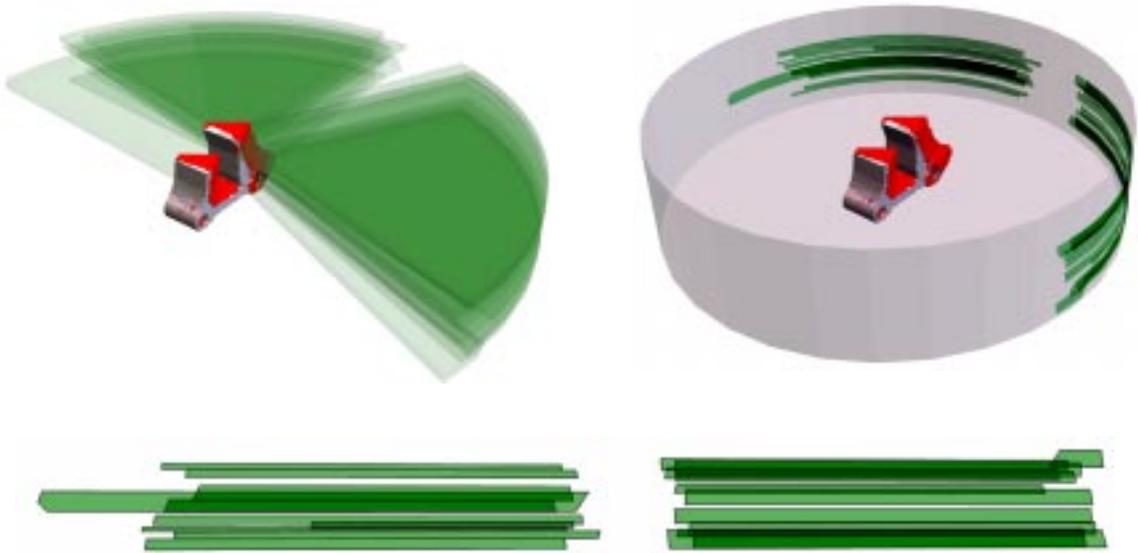


Figure 4-26. Planning for a sensor constrained to move vertically in conjunction with a turntable rotation stage (top). Individual plans “unwrapped” to lie on plane (bottom).

4.14 Example: City Scene

We now show a planning example of a complex scene using multiple targets. This example, which is the city scene shown in figure 4-27, is composed of multiple parts and has extremely high self occlusion. It was chosen because it provides an example of how our system might be used in complex outdoor urban environments. These environments are typified by large structures that encompass a wide range of geometric shapes and typically are grouped so that occlusion is a significant problem. The resulting site models may be used in many different applications ranging from city planning, urban design, fire and police planning, to military applications, virtual reality modeling and others.

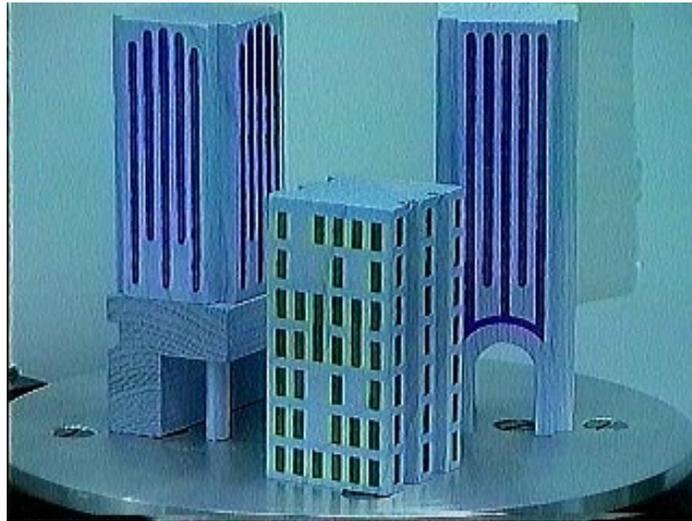


Figure 4-27. The city scene, consisting of three toy buildings. Note the archway under the rightmost building, and one of three pillars visible on the leftmost building.

The modeling process was initiated by the acquisition of four range images, with 90° turntable rotations between them, to produce the preliminary model shown from two perspectives in figure 4-28. In this rendering, “occluded” surfaces are shown in red, while “imaged” surfaces are shown with their edges visible. Approximately 25% of the entire acquirable model surface is at this point composed of “occluded” surface (“acquirable model surface” in this context means those “occluded” surfaces that are not in a horizontal orientation, such as the roofs). After decimating the “occluded” surfaces, the 30 largest by area were chosen and plan was generated for them. V_{target} is shown for each of these 30 surfaces, with a decimated copy of the city scene at the center to allow the reader to observe the relative orientations.

These visibility volumes are then intersected with $V_{\text{placement}}$ to compute the sets of occlusion-free sensor positions for the targets, as shown in figure 4-30. In this planning example, a discrete solution is desired for the proper number of degrees to rotate the

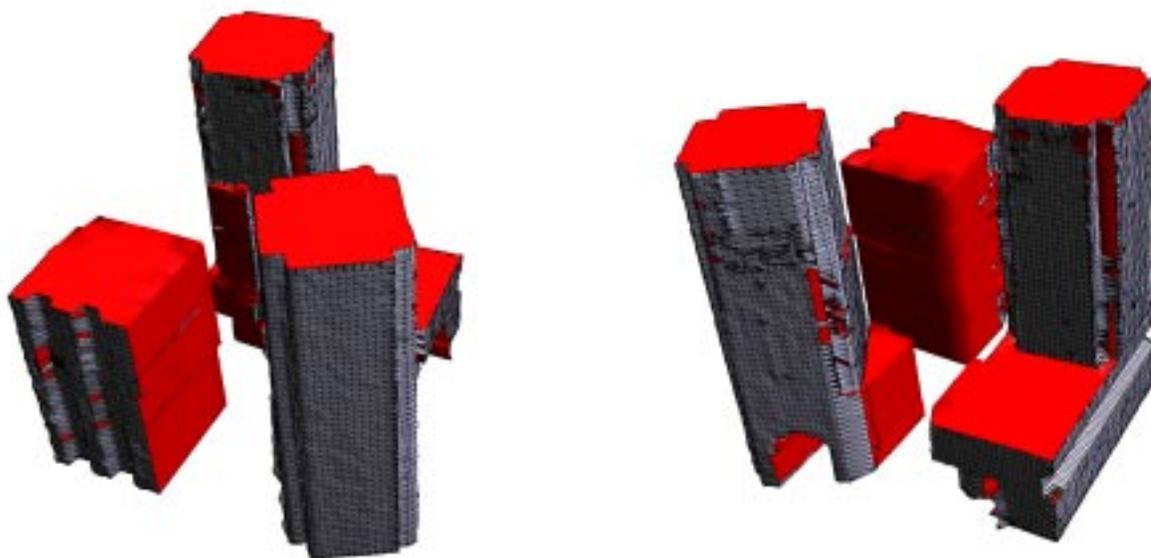


Figure 4-28. Preliminary model of city scene, from two perspectives.

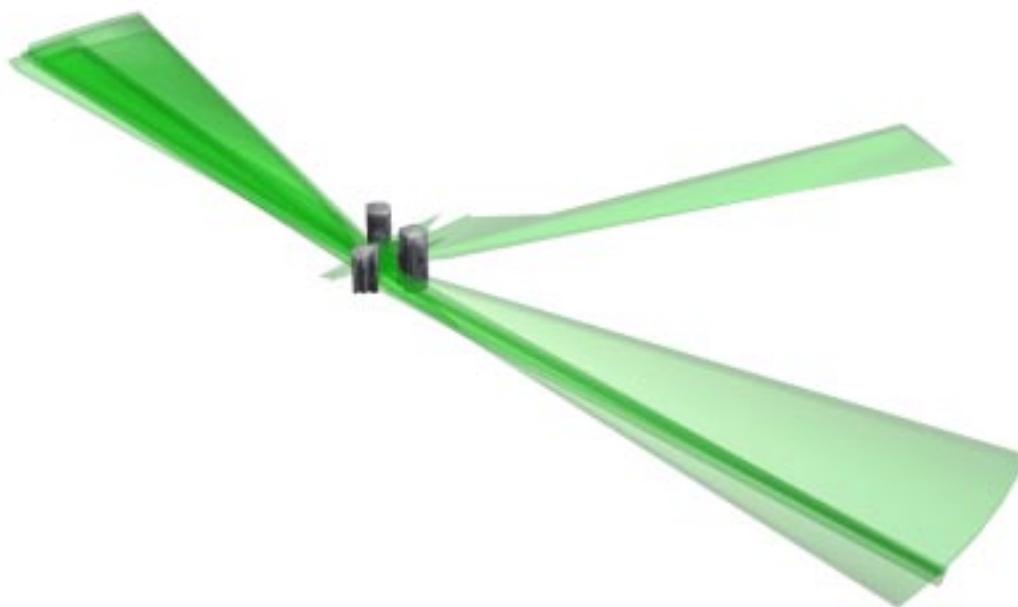


Figure 4-29. Initial planning for city scene. Shown are visibility volumes V_{target} (in green) for the 30 largest “occluded” surfaces by area.

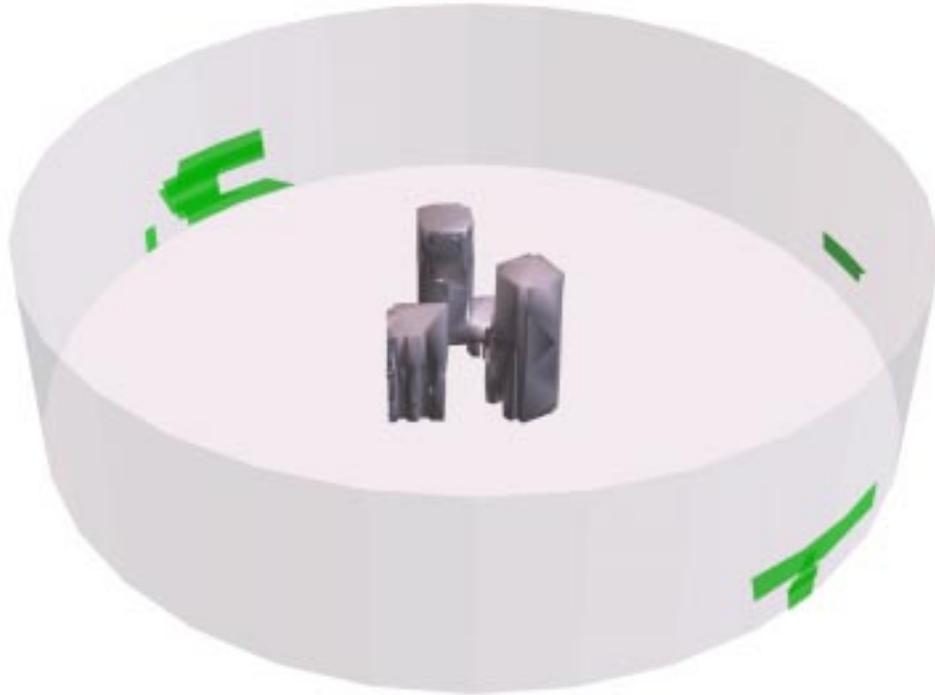


Figure 4-30. Plans for top 30 targets by area, shown in green on the surface of a model of the sensor placement constraint.

turntable. To accomplish this, the sensor space has been discretized every 2° , with the total target area acquired at each position found by testing the continuous-space plans for intersection with a vertical line at the appropriate position on the cylinder representing the sensor placement constraint. The results of this process are shown in figure 4-31 in a “planning histogram”, where the height of each green bar represents the area of target surfaces visible from that sensor location. Thus, higher bars denote desirable sensor locations, lower ones less so.

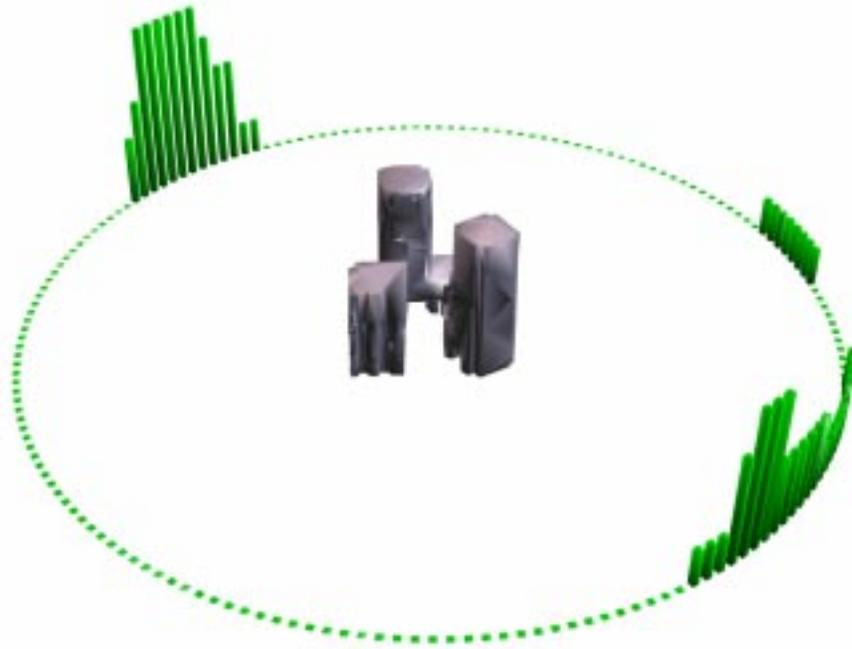


Figure 4-31. Results of discretization of sensor space. The height of each green “bar” represents the area of target surfaces visible from that sensor location.

From this discrete representation, the angle of turntable rotation is found by selecting the peak in the planning histogram. After the next range image is taken, the planning process is restarted with the next model. In figure 4-32, figure 4-33, and figure 4-34 the continuous and discrete plans are shown for the next 6 views.

After a total of 12 images have been automatically acquired, modeled, and integrated, the final model is shown in figure 4-35. Again, red areas denote “occluded” surface, while properly acquired surfaces are shown with their edges visible. Note that because of the constraints on the orientation of the sensor, in this example no surfaces on the buildings’ roofs can be acquired. This model is shown texture-mapped in figure 4-36.

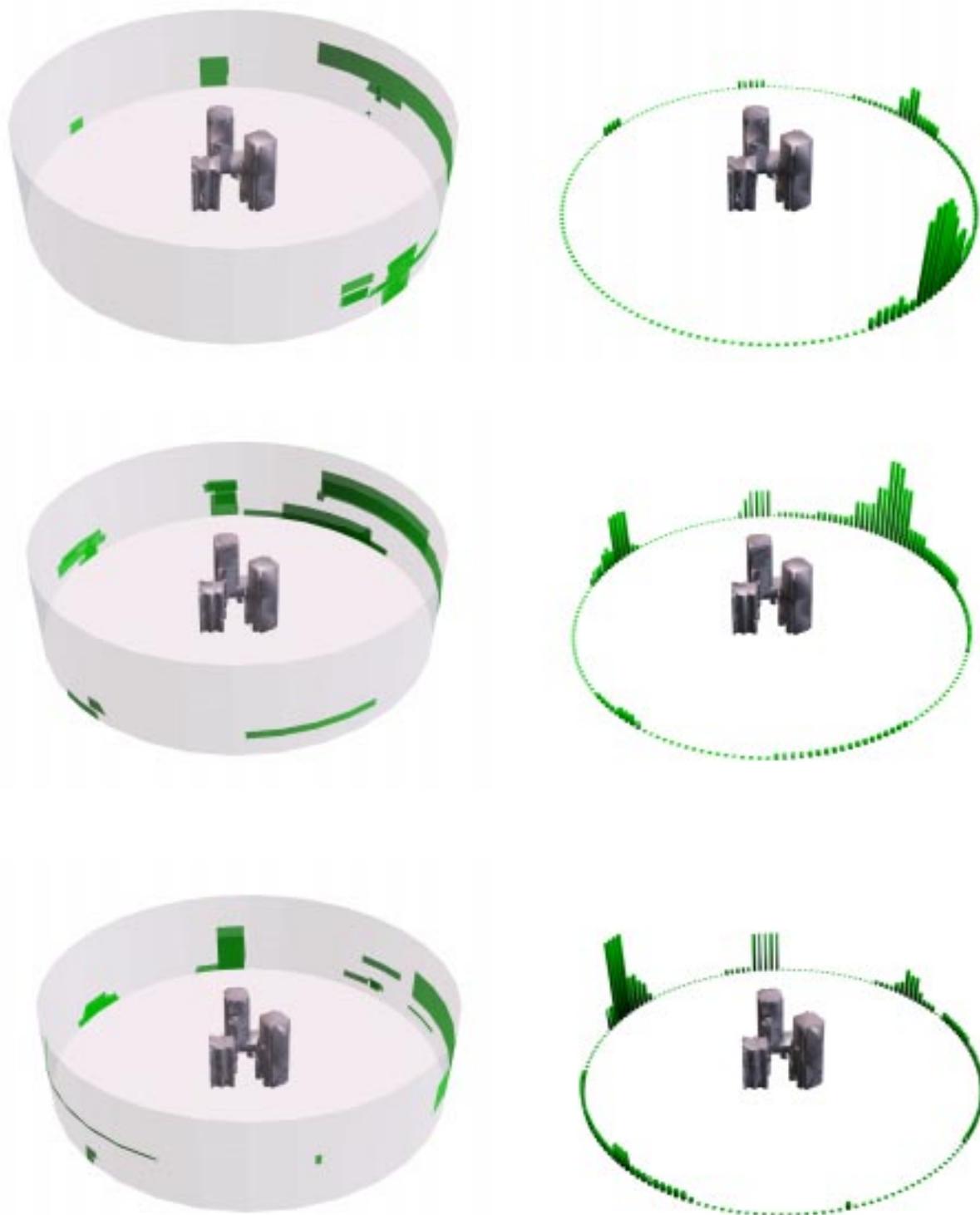


Figure 4-32. Plan generation for views 6, 7, and 8.



Figure 4-33. Plan generation for views 9, 10, and 11.

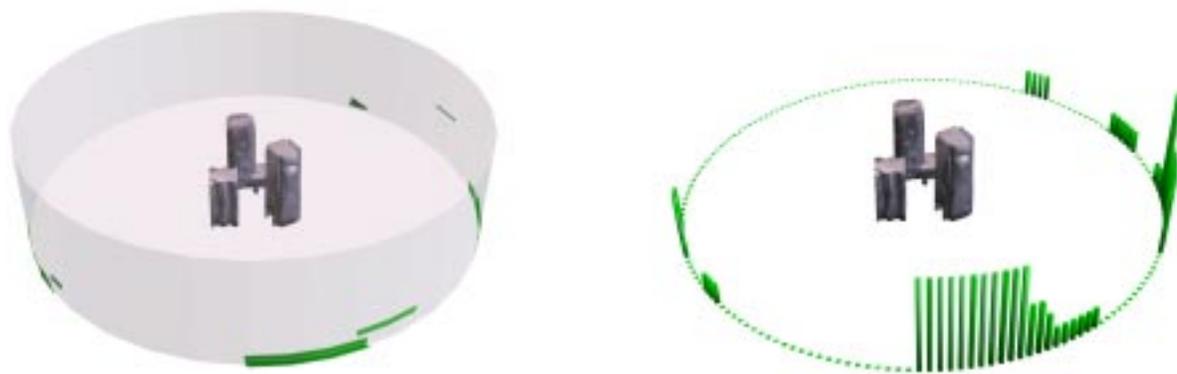


Figure 4-34. Plan generation for view 12.



Figure 4-35. Final model of city scene.

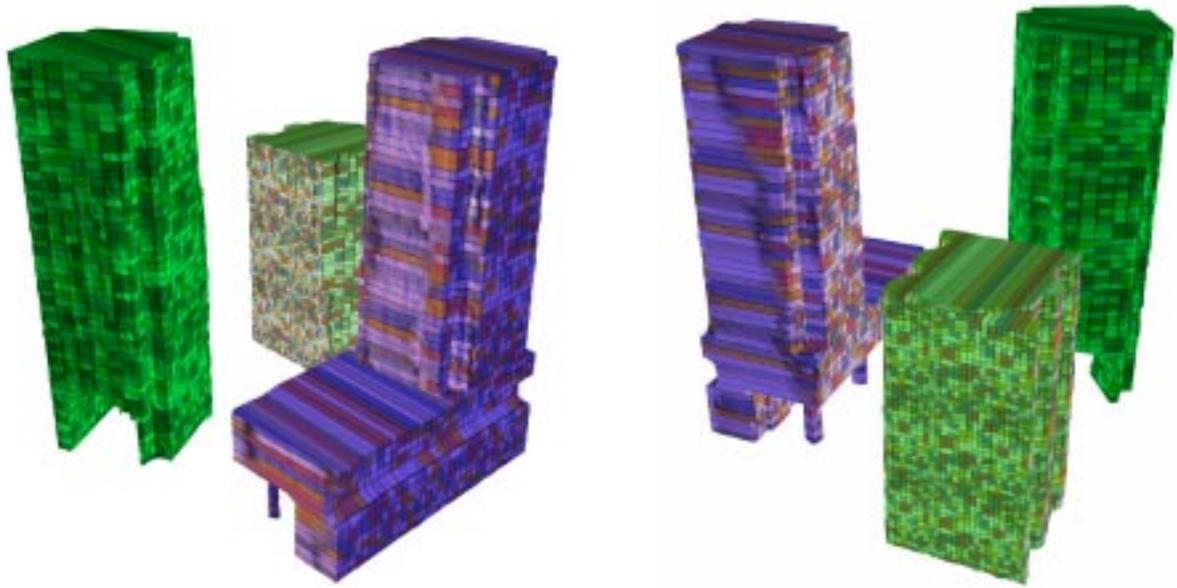


Figure 4-36. Final model of city scene, texture-mapped.

4.15 Analysis: Model City

The results from the city scene acquisition require some analysis. In the table below we list some of the pertinent measurements of the model as acquisition continues:

- Volume – The total volume of the model.
- Surface Area, Total – The total surface area of the model.
- Surface Area, Occluded – The total area of all “occluded” surfaces that have a significant component of their surface normals in the world x-y plane. This prevents the inclusion of “roof” features, which can not be acquired and should not be planned for, in this sum.
- Surface Area, Planned Targets – The total surface area of the targets for which plans have been generated.

- % Target Area Planned – The surface area of planned-for targets, as a percentage of the total “occluded” surface area.

Each of these metrics was calculated algorithmically on the computer model.

As shown in table 4-1, the first 4 views were acquired without any planning. In the data describing the remaining views there are some features that seem intuitive. The total model volume decreases over time, as indeed it must for a system that uses set intersection for integration and has not duplicated any sensor viewpoints. As is shown in the third column, the total surface does not strictly decrease, due to the effect described in section 4.2. Of particular import is the data in the final column. Because the plans are computed using a fixed number of surfaces at each iteration, it is interesting to see what percentage of the total available target area is being planned for. Clearly, if every target surface were considered, this would be 100% each time. Even though only 30 of the largest targets by area are planned for, the percent of the planned area never drops below 10% of the total area, and in most cases is over 20%. This shows that the considerable computational cost saved by selecting a subset of the targets to plan for is a viable strategy. The actual volume of the city scene has been calculated from measurements made by hand as 362 cm^3 .

4.16 Discussion of Trade-Offs between Sensing and Planning

The focus of this section is on the computational costs and trade-offs involved in the sensing and planning phases of the model acquisition process. Although it is not possible to determine either the sensor positions or their number for an unknown scene in advance, we discuss here some of the issues involved in using planned versus unplanned sensing.

TABLE 4-1. Model measurements during acquisition.

View #	Volume (cm ³)	Surface Area: Total (cm ²)	Surface Area: Occluded (cm ²)	Surface Area: Planned Targets (cm ²)	% Target Area Planned
0	4712	1571	1571	–	–
1	1840	1317	942	–	–
2	1052	1151	590	–	–
3	506	733	200	–	–
4	432	658	140	–	–
5	416	656	121	61	50%
6	404	659	104	28	27%
7	391	657	90	12	13%
8	386	647	84	8	10%
9	382	644	75	15	20%
10	380	651	62	7	11%
11	374	622	53	16	30%
12	370	604	36	9	25%

As we have mentioned above, the purpose of the planning phase is twofold. By automatically planning sensor viewpoints, it is possible to ensure that the scene or object is acquired to the desired resolution. In addition, having an algorithmic planning method may lead to a reduction in the number of views, and therefore the time and data, necessary to acquire a model. However, quantifying the reduction exactly can only be achieved if the number of unplanned and planned views is known in advance, which may only be done for scenes so simple that they usually do not require planning at all.

The entire time cost to acquire a model is the sum of the time to acquire and integrate the information from both pre-planned views – such as those from equal 90° turntable rotations – and those which are planned. The cost of pre-planned views may be written as:

$$\text{Cost}_{\text{pre-planned}} = n (C_A + C_M + C_I) \quad (\text{EQ 4-5})$$

where C_A , C_M , and C_I are the costs associated with acquiring, modeling, and integrating each range image and n is the number of pre-planned views. C_A includes the time needed to position the sensor and to acquire the image. Image acquisition time is linear with respect to the size of the image, but positioning time varies widely with the task. For some situations, such as the experimental setup described in Chapter 3, determining the path from one position to the next is done in constant time, and the repositioning motion is fast. However, for situations where positioning the sensor requires the use of advanced path planning algorithms, or repositioning is very expensive, this cost may be considerably larger. C_M accounts for the time spent applying extrusion and set union to each surface element in the mesh constructed from the range image. The extrusion may be done in constant time, but general set intersection operators are approximately quadratic in the number of model edges. However, because the union is of a set of parallel triangular prisms which meet only along their extruded surfaces, each union operation may be carried out in constant time. Thus, C_M is linear in the size of the image. C_I is quadratic, since it relies on a general set intersection operator. However, due to the effects of taking intersections of noisy surface data, the number of intersecting model surfaces from the same object surface decreases over time, thus somewhat countering the effect of the increasing number of model surfaces.

It is possible to estimate a lower bound on n , the number of unplanned views required, if we assume that the space represented by the sensor placement constraint is divided evenly by the number of views – that is, sensor positions are evenly distributed and the first position is chosen randomly. In the case of a turntable-based system, n views would have $360/n$ degrees of rotation between each of them. Under this assumption, if the visibility for a

certain feature f is known to lie within a portion of the sensor placement constraint, then n may be determined. Again, using the turntable example, if a model feature is only visible within a 20° angle, then in order to guarantee that feature is acquired images must be taken at less than 20° increments. Thus, for any feature which can only be imaged within an angle of α , the number of views needed to guarantee acquisition of f is

$$\left\lceil \frac{360}{\alpha} \right\rceil + 1 \quad (\text{EQ 4-6})$$

In the example of the city scene shown above, there is one feature that is only visible for a 9° region of the sensor placement constraint. Thus at least 41 images at equal turntable rotations are needed to guarantee acquisition of that feature if no view planning is performed. There may be other features with even more constrained visibility, which would increase the number of required views, but this is a useful way to estimate a lower bound.

The cost of planned views is more difficult to analyze, since it is necessary to know how many planned views will be required for the scene. The cost for acquiring the planned views is:

$$\text{Cost}_{\text{planned}} = m (C_A + C_M + C_I + C_P) \quad (\text{EQ 4-7})$$

where m is the number of planned views, and C_P is the cost of planning a single view, which was shown in section 4.9 to be quadratic in the number of model surfaces. Thus, both an unplanned view and a planned one are $O(s^2)$, s in this case referring to the number of surfaces in the composite and single-view models. Unfortunately, there is no solution for computing the number of views m that an unknown scene will require, just as there is no solution to determining a set of viewpoints that will completely acquire an unknown

scene. However, it may be possible to compute an estimate based on a qualitative description of the scene. This description may be based on a classification of parts or scene types by an “occlusion index”, from which the number of initial unplanned views may be conjectured. In this thesis the number of initial views was a user-defined variable, found by examining the scene occlusion and estimating the number of initial views needed to acquire a rough model. Because the number of initial views is a single integer parameter, its estimation is an appropriate place to allow user interaction.

In the implementation of our system the planning component has received considerably less attention to optimization, and remains highly instrumented with verification code. Because of this, the time to plan a new sensor viewpoint and acquire an image takes approximately 120 minutes for each of the planned city scenes. This is a factor of 20 greater than the time required to acquire an unplanned view, which is approximately 6 minutes as discussed in the previous chapter. Although the minimum of 41 unplanned images necessary to acquire the city scene would have been faster in this situation, it is easy to imagine a situation where this would not be the case. For example, if by improving the planning code the time to produce a plan was reduced to less than 29 minutes, the planned and unplanned methods would take the same amount of time to produce models of equal quality. However, because using viewpoint planning greatly reduces the number of views (from 41 to 12), it would also reduce the amount of data storage necessary, as well as wear on the positioning system.

Clearly, if the cost of planning is very expensive with respect to the cost of acquiring, modeling, and integrating the data, for many tasks it will be sufficient to use pre-defined

views, as shown in Chapter 3. However, for many scenes it will not be as easy to acquire the sensor data as it is in our experimental setup. One such situation is for systems where the sensor is on a mobile robot, and is typified by the hours required to move the Mars Sojourner a few meters on a remote world. In these situations, C_A has increased dramatically, and far outweighs the cost of planning to reduce the number of views. Another way to reduce the disparity between planned and unplanned views is to reduce C_P either by using more efficient code or by a modification of the planning algorithm. The former is desirable because it retains the generality of the method, and it is expected that a speedup by a factor of 5 to 10 may be easily realized. Modifications to the planning algorithm can include the use of heuristics to select targets with few occlusions, thus reducing the relatively expensive intersections done during the planning operation (see section 4.9, Computational Considerations). A detailed discussion of these heuristics is out of the scope of this work, but those based on the position and orientation of the target surfaces hold some promise: it can be seen that the “occluded” surfaces closest to the borders of the range image cannot be occluded, and therefore may support faster planning.

Although in this thesis we have not identified exact methods of determining n and m for an unknown scene, experience has shown us that planning is most effective when a majority of the object surface has already been acquired. In the examples shown, planning is initiated only after a substantial model has been built, and therefore the planning is isolated to relatively small regions of “occluded” surface. Because the number of initial views depends on the scene, the task, and the system, it may be best to have user interaction specify this quantity.

Chapter 5 Conclusions and Future Work

5.0 Conclusions

This thesis discusses the problem of automatic model acquisition for arbitrary free-form objects and scenes. It has presented an approach for solving this task under a variety of constraints, by decomposing the task into a modeling component and a planning component. It has also presented the results from experiments that show how an implementation of the approach performs on a variety of reconstructions. This chapter summarizes the contributions to computer vision, graphics, and modeling made by this work. Limitations of the system, and the future research that will advance work in this area, are also presented.

5.1 Summary

The automated model acquisition method presented in this thesis consists of two components that operate in an interleaved fashion: a modeling component that acquires,

models, and integrates data acquired from a range imaging camera, and a planning component that analyzes the resulting model and computes the next sensor position.

The modeling component operates by constructing a mesh surface from the range image, and extruding each mesh element into a solid prism. The union of the resulting set of prisms forms a solid model of both the imaged object surfaces and the space occluded from the sensor. A sensor model is then used to annotate the model with tags that describe each surface as a properly-imaged object surface or one that is the result of occlusion in the scene. Models from different viewpoints are integrated using a set intersection operator.

The planning component takes as input the integrated model from the modeling component. The tags that annotate the model surfaces are used to determine volumes of visibility for contiguous areas of unexplored scene. These volumes are combined with models representing the sensor's positional freedom to compute sets of occlusion-free sensor positions that are guaranteed to improve the quality of the model. These sets may be intersected to determine a single best region for the next sensor position, or discretized if a continuous solution is not necessary.

Both components have been tested, individually and as a system, on a variety of different shapes and scenes. Complex parts with holes and both flat and curved surfaces, as well as multi-part scenes with very high self-occlusion, have been properly acquired. The resulting models have been rendered, and in many cases fabricated using an RP machine.

5.2 Contributions

The primary contributions of this thesis are embodied in the techniques described for model acquisition and sensor viewpoint planning:

- Model acquisition of complex scenes – Free-form and multiple objects of arbitrary topological type can be acquired.
- Guaranteed solid models – The resulting model is “watertight” irrespective of the number of images used to acquire it.
- Construction of solid using mesh surface and sweeping – Although there has been work done using extrusion to generate models, this has been with intensity images, and has not been extended to range imagery or mesh surface applications.
- Persistence of surface-type information – “Imaged” and “occluded” surface tags persist through model integration, so that properly-acquired surfaces (and those not so) are identifiable in the final model.
- Incremental model improvement – The model quality may be improved as the application requires by using additional sensing operations.
- Plan generation for best next view – Feature-based viewpoint sensor planning techniques have been applied to situations in which the entire object is fully modeled. Their use in this context is new, and has the following advantages: continuous-space representation, the ability to handle model self-occlusion, applicability to different sensors, and computational cost determined by target surfaces (not by the size of the sensor space).

- Model acquisition system and testbed – It is important, particularly in the geometric modeling arena, that algorithms be able to be usefully applied. The methods presented in this thesis have been implemented in a system that combines the software algorithm with an industrial robot, a laser rangefinder, and a turntable. It should be noted that this system is much more flexible than scanners currently on the market, and therefore may serve as an excellent platform for future research in this field.

5.3 Limitations

As discussed in the preceding chapters, the approach described in this thesis is not without limitations. We summarize these limitations here, and discuss possible solutions that may be carried out to improve the system.

5.3.1 Limitations of the Modeling Algorithm

The modeling component of this thesis has three primary limitations to its effectiveness: one due to a sensing constraint and two due to modeling constraints.

The first and probably the greatest limitation of the domain of objects that may be acquired is that the method requires “surrounding views” of a scene in order to produce a useful model. Thus, it is not effective in modeling the inside of a room from a sensor location at the center. However, this limitation may be overcome if the modeling technique is changed to produce the solid from the mesh surface *to* the sensor origin (instead of sweeping it *away* from the sensor) and if set subtraction is used instead of set intersection.

Although this technique will not create well-bounded models after only a few views, it does permit the modeling of situations such as the room example.

Secondly, this method may run into limitations due to modeling resolution faster than other techniques such as mesh surface merging. This is because at each intersection operation between models, there are many surfaces that differ only because their mesh vertices have been affected differently by noise. Thus, at each intersection operation, the mesh surface there tends to increase in resolution. This is not a problem in many mesh surface-based merging methods because a resulting mesh may have the same resolution as the two input meshes.

Finally, because set intersection is used as the integration method, there is no consideration of the value of overlapping samples, as is done in methods that combine samples to compute a *consensus surface*. In addition, there is a bias effect due to the intersection operation: since each composite model is formed from the intersection of previous models, the bias is towards the model of least volume. A possible solution to this is to provide a topology-preserving post-processing stage in which the vertices from the final model are moved to new positions that are weighted averages of the surfaces that fall within a certain distance in the direction of the surface normal. This technique might be used in conjunction with additional surface tags that give an estimation of the sensor's accuracy when the surface was acquired.

5.3.2 Limitations of the Planning Algorithm

There are three limitations of the planning system. The first is due to the characterization of target surface visibility, the second is due to omission of a field-of-view constraint, and the last is the failure to recognize when certain inappropriate plans are used.

While in the process of acquiring an object, it may become evident that a view that degrades the model has been used. For example, let's assume that the part being imaged is a propeller blade or other thin part. After an initial model has been acquired, it is quite possible that the planner will generate a sensor position that causes the blade to be imaged in profile, i.e. edge-on. Since sampling sensors tend to perform poorly in this orientation, the single-view model created is likely to be only a subset of the true volume. However, it is possible to note when these situations occur. If the surface area of surfaces tagged "imaged" ever decreases, the current single-view model should be suspect. Possible planning behavior at this point would at least include disregarding the current viewpoint and image.

The second limitation is that no orientation constraint is included in the planning process. The reason for this is that the modeling component assumes that the field-of-view completely covers the workspace, so the constraint is not relevant during the planning process. However, in situations where such a field-of-view is not possible, this constraint is important to consider. The reader is directed to [Abrams 1997] and [Sedas-Gersey 1993] for techniques that allow its inclusion.

Finally, because the planning system considers only total visibility for each target, it is possible that some solutions are missed where partial visibility would allow a better result.

“Total visibility” in this context means that every position in the plan can *completely* image the target surface. For example, when planning for two targets, their visibility volumes may be disjoint, meaning that there are no positions that totally image both targets. However, there may be positions that partially image both targets, and whose total imaged area is greater than either target individually. This is an example of a case where planning using total visibility does not provide the best solution. However, in practice it probably does not affect the outcome significantly, since the plans generated with the current methods already have many overlapping visibility volumes for the total visibility case.

5.4 Future Work

The limitations of the system described above point to areas in which there remains much work to be done to increase the effectiveness and flexibility of the current system. Moreover, several other avenues of exploration in this field draw attention. We discuss some of them here.

- **Modeling in diverse environments:** As modeling techniques are applied to larger objects or outdoors, it will become impossible to continue the current practice of viewing the entire surface of the object from its exterior. For example, consider the interior of a large room: A sensor at the center of the room can acquire information about the interior walls, but the methods used to model and integrate this information will be quite different from the case where the outside surface has been completely imaged.

- Rapid modeling systems: There have been recent advances in rangefinding scanners, and some prototypes are now able to produce range images at close to 1Khz. Although it is unlikely that real-time modeling will be realized any time soon, specialized techniques can certainly be developed to improve the speed of current systems, although at a cost to modeling fidelity.
- High-accuracy modeling: As with most systems, there is always the desire to push the envelope with respect to modeling accuracy. Accuracy in systems that use multiple images depends in large part on the method of integration, and so robust merging methods must be developed that can determine optimal surfaces under various constraints.
- Planning and visibility issues: Sensor planning systems ordinarily assume that the sensor requires complete visibility for each target surface. However, finding more appropriate viewpoints may be possible if this constraint is relaxed and partial visibility is considered.
- Model acquisition at the CAD primitive level: Although one may acquire highly tessellated models of real-world objects, these models are far from the level of abstraction of the CAD models with which a human designer would typically work. The problem for the future is to find methods to transform models so that they more closely follow common design principles.

5.5 Final Remarks

The field of 3-D digital imaging and modeling is undergoing a remarkable expansion at this time, in both research and application. The problems that will be encountered before these systems can easily be applied to a wide range of tasks will be both interesting and challenging. It is our hope that this thesis will provide both support and cause for future advancements in the field.

Appendix: Calibration

A.0 Introduction

This appendix discusses the means used to calibrate the rangefinder/robot/turntable system.

A.1 Calibration

In order to merge models acquired from different viewpoints, they must be transformed into a common coordinate frame. In this regard a system may choose between transforming the models into the world coordinate frame or transforming them into some other common frame, usually that of one of the models. All other things being equal, it is better to do the former, since it allows interaction between the model and the world. The latter technique is useful if one is only concerned with generating a model of the object, and may be accomplished by either fitting overlapping parts of the model together or by aligning registration features that are either part of the model or added nearby.

Calibration of our system allows imaged data from the rangefinder to be transformed to world coordinates. There are two parts to this calibration: the first is concerned with calibrating the camera internally, so that it correctly interprets its sensing operations into accurate depth data. The second part of the calibration, and the one which is more completely dealt with here, is concerned with determining the transform from the end effector of the robot to the center of the rangefinder's coordinate system.

A.1.1 Rangefinder / Robot System Calibration

Although single images may be acquired and modeled independently, it is necessary to be able to establish their position and orientation in a common reference frame if multiple images (or models) are to be merged into one. We need to determine the homogeneous transformation matrix $T^{\text{grip-range}}$ that transforms points from the coordinate frame of the rangefinder (denoted P^{range}) to points in the frame of the manipulator's gripper (denoted P^{grip}), as shown in figure A-1. Once we have this transform a sensed point may be brought into the world coordinate frame by multiplying first by $T^{\text{grip-range}}$, and then by $T^{\text{world-grip}}$ (the gripper transform given to us by the robot) to give us P^{world} .

One common way to solve this problem is to include a step during processing in which the distinct images are registered. However, image registration is a computationally expensive process that can be avoided if the inter-image transform is known beforehand. Determining this involves accurately computing the transform from the end effector of the robot to the origin of the coordinate system of the rangefinder.

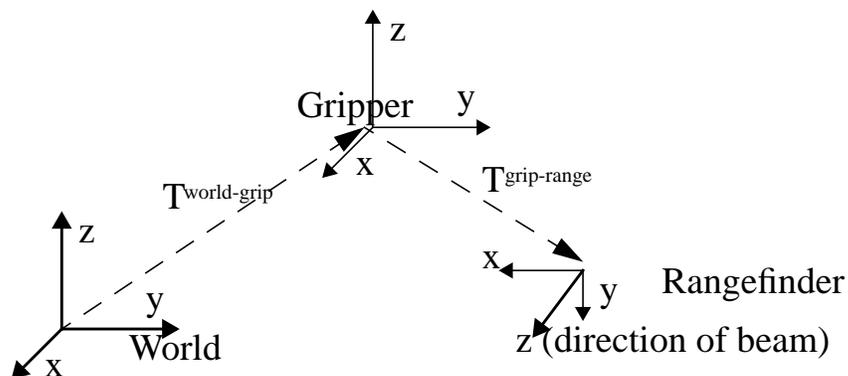


Figure A-1. Transformations between world, gripper, and rangefinder.

Pose #	Position Error (in.)	Orientation Error (deg.)
1	0.456	0.619
2	0.454	0.369
3	0.564	0.665
4	0.481	0.646
5	0.496	1.904
6	0.518	0.641
7	0.363	1.116
8	0.469	0.748
9	0.153	0.757

TABLE A-1. Results of accuracy experiment on Puma-560 manipulator. The manipulator was commanded to move the end effector to one of 9 poses, where the actual pose was then measured using a CMM.

To see why registration is a difficult problem, look at table A-1, which shows the results of an accuracy test reproduced from [Mooring *et al.* 1991] using the type of manipulator initially chosen for the work presented here. Although the repeatability of this manipulator is well suited to our task (mean position error of 0.0041 inch, mean angular error of

0.0905 degrees), this table shows that the accuracy of the manipulator is insufficient for our work. Arbitrarily positioning the sensor to acquire a range image may result in angular errors in excess of one degree, as shown in poses five and seven. Since the laser often does sensing operations from a distance of approximately 0.5m from the target, this will result in a sensed position error of greater than 8mm, which is unsatisfactory.

There are many ways of overcoming this problem: First, do not use the manipulator in arbitrary positions, or do not use anthropomorphic manipulators such as the one above. The joints that have the greatest effect on error are known, and use of these should be minimized. Secondly, calibrate the laser/manipulator system in the pose it will acquire data from. One simple and often-used solution is to use a turntable to rotate the object to be imaged. This allows the rangefinder/manipulator system to be calibrated very precisely within the range of motion required for image acquisition. Third, try to reduce the distance from the rangefinder to the object being imaged: the longer this distance is, the greater the effect of angular errors on the resulting data.

One may wonder at the ubiquitous use of turntables in the commercial products that perform modeling-from-observation tasks. Additional motivation for using a turntable is that it reduces the number of unique transformations in a way that simplifies correction by other registration methods. Without a turntable, and allowing unconstrained placement of the manipulator arm, we have the following situation:

$$P_n^{world} = T_n^{world-gripper} T^{gripper-range} P_n^{range} \quad (\text{EQ A-1})$$

Here n is the number of matching points in the rangefinder and world coordinate frames, and also the number of different gripper transforms (in the general case one point is

extracted per scan, i.e. per gripper position). For n matching world and imaged points, we will have n transforms for the end effector, each of which will introduce additional error. Using a turntable allows us to constrain the arm to a small subset of positions where the error may be accounted for in $T^{\text{grip-range}}$ when $T^{\text{grip-range}}$ is determined. If we use the rangefinder model discussed above and constrain the sensing operation to just one viewpoint, the above equation reduces to:

$$P_n^{\text{world}} = T^{\text{world-range}} P_n^{\text{range}} \quad (\text{EQ A-2})$$

where $T^{\text{world-range}} = T^{\text{world-grip}} T^{\text{grip-range}}$. The benefit of this is that there is only one transformation matrix, instead of the n in the first equation, which permits fine tuning of one transformation to improve registration instead of n independent transformations. Of course, a disadvantage is that now only one view of the object may be imaged. This is where the turntable is useful, since it allows the object to be repositioned with respect to the sensor. Turntables are an inexpensive means to increase the performance of this type of system, since their rotation $R(\theta)$ may be applied very accurately. In order to incorporate a turntable, its relationship $T^{\text{world-turn}}$ to the world coordinate system must be known. This may be measured using the sensor or the end effector of the robot. Now any points measured by the sensor must not only be transformed into the world coordinate frame, but also into that of the turntable so that any turntable rotation may be reversed. This gives us the following equation for transforming data from the sensor into the common coordinate frame of the turntable with rotation $\theta = 0$:

$$P^{\text{turntable}} = R(\theta)^{-1} (T^{\text{world-turn}})^{-1} T^{\text{world-range}} P^{\text{range}} \quad (\text{EQ A-3})$$

A.1.2 Determination of Rangefinder Transform T^r

To determine the relationship $T^{\text{grip-range}}$ between the rangefinder and the manipulator's end effector a least-squares solution is computed to a system of linear equations, analogous to the method for used to determine transformations in tablet systems [Sutherland 1974].

Again, we need to obtain the matrix $T^{\text{grip-range}}$, described by:

$$\begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ T_{41} & T_{42} & T_{43} & T_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} \quad (\text{EQ A-4})$$

where $[x \ y \ z \ 1]$ and $[x' \ y' \ z' \ w]$ are the homogeneous coordinates of a point in the rangefinder and gripper coordinate frame's respectively. Note that, due of the type of sensor, 3-D coordinates are used in both the sensed (rangefinder) and gripper coordinate frames, so the scale factor w is unity. More importantly, note that because the sensor only acquires data in it x - z plane, all the values y are 0. The effect of this is that it is not possible to completely determine the transformation matrix. Fortunately, the matrix can be found up to a determination of the Opposition vector $[T_{12} \ T_{22} \ T_{32}]$, which may then be calculated as shown below.

Multiplying equation A-4 out, and including the knowledge that all y are 0, gives the following 4 equations which hold for all matched sets of points:

$$\begin{aligned} T_{11}x + T_{13}z + T_{14} &= x' \\ T_{21}x + T_{23}z + T_{24} &= y' \\ T_{31}x + T_{33}z + T_{34} &= z' \\ T_{41}x + T_{43}z + T_{44} &= 1 \end{aligned} \quad (\text{EQ A-5})$$

For n points we will then have 4 equations per point in 12 unknowns to solve the general case, therefore requiring 3 pairs of matching points. We can solve this system by computing a solution to $Ax = b$ in the following form:

$$\begin{bmatrix}
 x_1 & z_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 x_2 & z_2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \dots & & & & & & & & & & & \\
 x_n & z_n & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & x_1 & z_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & x_2 & z_2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \dots & & & & & & & & & & & \\
 0 & 0 & 0 & x_n & z_n & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & x_1 & z_1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & x_2 & z_2 & 1 & 0 & 0 & 0 \\
 \dots & & & & & & & & & & & \\
 0 & 0 & 0 & 0 & 0 & 0 & x_n & z_n & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_1 & z_1 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_2 & z_2 & 1 \\
 \dots & & & & & & & & & & & \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_n & z_n & 1
 \end{bmatrix}
 \begin{bmatrix}
 T_{11} \\
 T_{13} \\
 T_{14} \\
 T_{21} \\
 T_{23} \\
 T_{24} \\
 T_{31} \\
 T_{33} \\
 T_{34} \\
 T_{41} \\
 T_{43} \\
 T_{44}
 \end{bmatrix}
 =
 \begin{bmatrix}
 x'_1 \\
 x'_2 \\
 \dots \\
 x'_n \\
 y'_1 \\
 y'_2 \\
 \dots \\
 y'_n \\
 z'_1 \\
 z'_2 \\
 \dots \\
 z'_n \\
 1 \\
 1 \\
 \dots \\
 1
 \end{bmatrix}
 \tag{EQ A-6}$$

With more than 3 pairs of matching points this system will be overdetermined. A least mean-square solution may be found by first multiplying both sides by A^T :

$$A^T A x = A^T b \tag{EQ A-7}$$

As discussed above the sensed points $[x \ y \ z]$ are coplanar in the x - z plane of the sensor's coordinate frame, hence it will not be possible to determine the second column in the $T^{\text{grip-range}}$, i.e. the Opposition vector. However, if it is assumed that $T^{\text{grip-range}}$ describes a right handed coordinate frame, this which may be calculated from cross product of the Normal and Approach vectors (the first and third columns in $T^{\text{grip-range}}$), giving us $T^{\text{grip-range}}$.

Bibliography

-
- [Abrams 1997] S. Abrams. *Sensor Planning in an Active Robot Work-Cell*, Ph.D. Thesis, Computer Science Department, Columbia University, New York, NY, 1997.
- [Abrams *et al.* 1993] S. Abrams, P.K. Allen, and K.A. Tarabanis. Dynamic Sensor Planning, *International Conference on Intelligent Autonomous Systems*, pp. 206–215, Pittsburgh, PA, February 1993.
- [Allen 1987] P.K. Allen. *Robotic Object Recognition using Vision and Touch*, Kluwer Academic Publishers, 1987.
- [Anderson *et al.* 1995] L. Anderson, S. Dorney, T. Peters, and N. Stewart. Polyhedral Perturbations that Preserve Topological Form, *Computer Aided Geometric Design*, 12:785–799, 1995.
- [Asada *et al.* 1992] M. Asada, M. Kimura, Y. Taniguchi, and Y. Shirai. Dynamic Integration of Height Maps into a 3D World Representation from Range Images, *International Journal of Computer Vision*, 9(1):31–53, 1992.
- [Babuska *et al.* 1995] I. Babuska, J. Flaherty, W. Henshaw, J. Hopcroft, J. Olinger, and T. Tezduyar. *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*, Springer-Verlag, 1995.

- [Bajcsy 1988] R. Bajcsy. Active Perception, *Proceedings of the IEEE*, August 1988.
- [Bajcsy & Solina 1987] R. Bajcsy and F. Solina. Three Dimensional Object Representation Revisited, *Proceedings of the International Conference on Computer Vision*, London, June 1987.
- [Ballard 1991] D.H. Ballard. Animate Vision, *Artificial Intelligence*, 48:57–86, 1991.
- [Baumgart 1975] B. Baumgart. A Polyhedron Representation for Computer Vision, *National Computer Conference*, pp. 589-596, AFIPS Conference Proceedings, 1975.
- [Besl 1988a] P. Besl. *Surfaces in Range Image Understanding*, Springer-Verlag, 1988.
- [Besl 1988b] P. Besl. Active, Optical Range Imaging Sensors, *Machine Vision and Applications*, 1:127–152, 1988.
- [Besl & Jain 1986] P.J. Besl and R.C. Jain. Invariant Surface Characteristics for 3D Objects, *Computer Vision, Graphics, and Image Processing*, 33:33–80, 1986.
- [Besl & McKay 1992] P.J. Besl and N.D. McKay. A Method for Registration of 3D Shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–255, 1992.
- [Brown 1991] J. Brown. Vertex Based Data Dependent Triangulations, *Computer Aided Geometric Design*, 8:239–251, 1991.
- [Chen 1988] C.H. Chen. *3D-Poly: A Robot Vision System for Recognizing Objects in Occluded Environments*, Ph.D. Thesis, School of Electrical Engineering, Purdue University, December 1988.
- [Chen & Medioni 1991] Y. Chen and G. Medioni. Object Modeling by Registration of Multiple Range Images, *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pp. 2724–2729, April 1991.
- [Chen & Medioni 1994] Y. Chen and G. Medioni. Fitting a Surface to 3-D Points Using an

- Inflating Balloon Model, *Proceedings of the Second CAD-Based Vision Workshop*, pp. 266-273, 1994.
- [Cohen *et al.* 1996] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Webber, P. Agarwal, F. Brooks, and W. Wright. Simplification Envelopes, *Proceedings of SIGGRAPH*, pp. 119-128, 1996.
- [Connolly 1985] C. Connolly. The Determination of Next Best Views, *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, pp. 432–435, 1985.
- [Connolly & Stenstrom 1989] C. Connolly and J. Stenstrom. 3D Scene Reconstruction from Multiple Intensity Images, *Proceedings of the 1989 IEEE Conference on Robotics and Automation*, pp. 124-130, 1989.
- [Curless & Levoy 1996] B. Curless and M. Levoy. A Volumetric Method for Building Complex Models from Range Images, *Proceedings of SIGGRAPH*, pp. 303-312, 1996.
- [DeRose *et al.* 1993] T. DeRose, M. Lounsbery, and J. Warren. Multiresolution Analysis for Surfaces of Arbitrary Topological Type, Report 93-10-05, Department of Computer Science, University of Washington, Seattle, WA, 1993.
- [Diamond & Kreplin 1994] A. Diamond and T. 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).
- [Dickinson *et al.* 1992] S.J. Dickinson, A.P. Pentland, and A. Rosenfeld. 3D Shape Recovery using Distributed Aspect Matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):174–198, February 1992.
- [Dickinson *et al.* 1997] S.J. Dickinson, D. Metaxas, A. Pentland. The Role of Model-Based Segmentation in the Recovery of Volumetric Parts from Range Data, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):. 259-267, March 1997.

- [Dorai & Jain 1997] C. Dorai and A.K. Jain. Optimal Registration of Object Views Using Range Data, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10): 1131-1138, October 1997.
- [Ferrie & Levine 1987] F.P. Ferrie and M.D. Levine. Integrating Information from Multiple Views, *Proceedings of the IEEE Workshop on Computer Vision*, pp. 117–122, 1987.
- [Ferrie & Levine 1988] F.P. Ferrie and M.D. Levine. Deriving Coarse 3D Models of Objects, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 345–353, 1988.
- [Fua & Sander 1992] P. Fua and P. Sander. Reconstructing Surfaces from Unorganized Points, *Proceedings of the DARPA Image Understanding Workshop*, 1992.
- [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.
- [Hartley & Gupta 1994] R.I. Hartley and R. Gupta. Linear Pushbroom Cameras, *Proceedings of the European Conference on Computer Vision*, pp. 555–566, May 1994.
- [Herman *et al.* 1984] M. Herman, T. Kanade, and S. Kuroe. Incremental Acquisition of a Three-Dimensional Scene Model from Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(3):331–340, 1984.
- [Herman 1985] M. Herman. Matching Three-Dimensional Symbolic Descriptions Obtained from Multiple Views of a Scene, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1985.
- [Hoover *et al.* 1994] A. Hoover, D. Goldgof, and K. Bowyer. Building a B-Rep from a Segmented Range Image, *Proceedings of the 1994 Second CAD-Based Vision Workshop*, pp. 74–81, Champion, PA, February 1994.
- [Hoppe 1994] H. Hoppe. *Surface Reconstruction from Unorganized Points*. Ph.D. thesis, CS Dept, University of Washington, 1994.

- [Ikeuchi & Flynn 1995] K. Ikeuchi and P.J. Flynn. Editorial: Recent progress in CAD-based vision, *Computer Vision and Image Understanding*, 61(3), 1995.
- [Jain & Flynn 1993] A.K. Jain and P.J. Flynn, editors. *Three-Dimensional Object Recognition Systems*, Elsevier, Amsterdam, 1993.
- [Jarvis 1993] R. Jarvis. Range Sensing for Computer Vision. *International Journal of Computer Vision*, pp. 17–56, 1993.
- [Kalvin 1992] A. D. Kalvin. A Survey of Algorithms for Constructing Surfaces from 3D Volume Data. Research Report #77606, IBM Research Division, Yorktown Heights, NY, 1992.
- [Kodiyalam & Saxena 1994] S. Kodiyalam and M. Saxena. *Geometry and Optimization Techniques for Structural Design*, Elsevier, 1994.
- [Kutulakos 1994] K.N. Kutulakos. *Exploring Three-Dimensional Objects by Controlling the Point of Observation*, Ph.D. thesis, Computer Sciences Department, University of Wisconsin, 1994.
- [Kweon & Kanade 1990] I.S. Kweon and T.Kanade. High-Resolution Terrain Map from Multiple Sensor Data, *Proceedings of the IEEE International Workshop on Intelligent Robotics and Systems*, pp. 127–134, Tsuchiura, Japan, 1990.
- [Laurentini 1993] A. Laurentini. Inferring the Shape of the Real Object from the Object Reconstructed by Volume Intersection, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 280-285, 1993.
- [Mantyla 1988] M. Mantyla. *An Introduction to Solid Modeling*, Computer Science Press, 1988.
- [Martin & Aggarwal 1983] W. M. Martin and J. K. Aggarwal. Volumetric Descriptions of Objects from Multiple Views, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158, March 1983.

- [Maver & Bajcsy 1990] J. Maver and R. Bajcsy. How to Decide from the First View Where to Look Next, *Proceedings of the 1990 DARPA Image Understanding Workshop*, pp. 482–496, 1990.
- [McInerny & Terzopolous 1993] T. McInerny and D. Terzopolous. A Finite Element Model for 3D Shape Reconstruction and Nonrigid Motion Tracking, *Proceedings of the International Conference on Computer Vision*, pp. 518-523, Berlin, Germany, 1993.
- [Mooring *et al.* 1991] B.W. Mooring, Z.S. Roth, and M.R. Driels. *Fundamentals of Manipulator Calibration*. John Wiley and Sons, Inc., 1991.
- [Naylor *et al.* 1990] B.F. Naylor, J. Amantides, and W. Thibault. Merging BSP Trees Yields Polyhedral Set Operations. *Computer Graphics*, 24(4):115–124, August 1990.
- [Naylor 1981] B.F. Naylor. *A Priori Based Techniques for Determining Visibility Priority for 3-D Scenes*. Ph.D. thesis, University of Texas at Dallas, May 1981.
- [O'Rourke 1994] J. O'Rourke. *Computational Geometry in C*, Cambridge University Press, NY, 1994.
- [Parvin & Medioni 1992] B. Parvin and G. Medioni. B-Rep from Unregistered Multiple Range Images, *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pp. 1602–1607, May 1992.
- [Peters 1993] J. Peters. Smooth Free-Form Surfaces over Irregular Meshes Generalizing Quadratic Splines, *Computer Aided Geometric Design*, 10:347–361, 1993.
- [Pito 1997] R. Pito. A Registration Aid, *Proceedings of the International Conference on Advances in 3-D Digital Imaging and Modeling*, pp. 85-92, Ontario, Canada.
- [Pito 1996] R. Pito. Mesh Integration Based on Co-Measurements, *Proceedings of the IEEE Conference on Image Processing*, special session on range image analysis, Lausanne, Switzerland, 1996.
- [Pito & Bajcsy 1995] 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, *Proceedings of the SPIE Symposium on Intelligent Systems and Advanced Manufacturing*, Philadelphia, PA, 1995.
- [Potmesil 1982] M. Potmesil. Generating *Three Dimensional Surface Models of Solid Objects from Multiple Projections*, Technical Report 33, Image Processing Laboratory, RPI, October 1982.
- [Potmesil 1986] M. Potmesil. Generating Octree Models of 3D Objects from their Silhouettes in a Sequence of Images, *Computer Vision, Graphics, and Image Processing*, 40: 1-29, 1987
- [Preperata & Shamos 1985] F.P. Preperata and M.I. Shamos. *Computational Geometry*, Springer-Verlag, NY, 1985.
- [Raja & Jain 1994] N. Raja and A. Jain. Obtaining Generic Parts from Range Images using a Multi-View Representation, *CVGIP: Image Understanding*, 60(1):44–64, July 1994.
- [Reed *et al.* 1997a] M. Reed, P. Allen, and I. Stamos. Automated Model Acquisition from Range Images with View Planning, *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, Puerto Rico.
- [Reed *et al.* 1997b] M. Reed, P. Allen, and I. Stamos. 3-D Modeling from Range Imagery: An Incremental Method with a Planning Component, *Proceedings of the International Conference on Advances in 3-D Digital Imaging and Modeling*, Ontario, Canada, 1997.
- [Reed & Allen. 1997] M. Reed and Peter K. Allen. A Robotic System for 3-D Model Acquisition from Multiple Range Images, *Proceedings of 1997 IEEE International Conference on Robotics and Automation*, New Mexico, USA.
- [Reed 1996] M. Reed. Generating Complete CAD Models from Range Imagery, *Proceedings of the 1996 U.S.-Japan Graduate Student Forum in Robotics*, Osaka, Japan, 1996.

- [Reed *et al.* 1995] M. Reed, P.K. Allen, and S. Abrams. CAD Model Acquisition using BSP Trees, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp. 335–340, 1995.
- [Requicha 1980] A.A.G. Requicha. Representations for Rigid Solids: Theory, Methods, and Systems, *ACM Computing Surveys*, 12(4):329–345, 1980.
- [Rutishauser *et al.* 1994] M. Rutishauser, M. Stricker, and M. Trobina. Merging Range Images of Arbitrarily Shaped Objects, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 573–580, 1994.
- [Schroeder *et al.* 1992] W. Schroeder, J. Zarge, and W. Lorensen. Decimation of Triangle Meshes, *Computer Graphics*, 26(2), pp. 65-70, July, 1992.
- [Schumaker 1993] L. Schumaker. Computing Optimal Triangulations using Simulated Annealing, *Computer Aided Geometric Design*, 10:329–345, 1993.
- [Seales & Faugeras 1994] W.B. Seales and O.D. Faugeras. Building Three Dimensional CAD/CAM Models from Image Sequences, *Proceedings of the Second CAD-Based Vision Workshop*, pp. 116–123, Champion, PA, February 1994.
- [Sedas-Gersey 1993] S. Sedas-Gersey. *Algorithms for Automatic Sensor Placement to Acquire Complete and Accurate Information*. Ph.D. thesis, The Robotics Institute, Carnegie Mellon University, May 1993.
- [Shmuel & Werman 1990] A. Shmuel and M. Werman. Active Vision: 3D from an Image Sequence, *Proceedings of the 10th International Conference on Pattern Recognition*, pp. 48–54, June 16-21 1990.
- [Shum *et al.* 1997] H.-Y. Shum, M. Hebert, K. Ikeuchi, and R. Reddy. An Integral Approach to Free-Form Object Modeling, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12), pp. 1366-1370, December 1997.
- [Sobh *et al.* 1995] T.M. Sobh, J. Owen, C. Jaynes, M. Dekhil, and T.C. Henderson. Indus-

- trial Inspection and Reverse Engineering, *Computer Vision and Image Understanding*, 61(3):468–474, may 1995.
- [Soucy & Laurendeau 1992] M. Soucy and D. Laurendeau. Multi-Resolution Surface Modeling from Multiple Range Views, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 348-353, 1992.
- [Spatial Technology 1994] Spatial Technology Inc. *ACIS Geometric Modeler Application Guide*. 1994.
- [Stenstrom & Connolly 1992] J.R. Stenstrom and C.I. Connolly. Constructing object models from multiple images, *International Journal of Computer Vision*, 9(3):185–212, 1992.
- [Stytz *et al.* 1991] M.R. Stytz, G. Frieder, and O. Frieder. Three-Dimensional Medical Imaging: Algorithms and Systems, *ACM Computing Surveys*, 23(4), pp. 421-500, December 1991.
- [Suk & Bhandarkar 1992] M. Suk and S.M. Bhandarkar. *Three-Dimensional Object Recognition from Range Images*, Computer Science Workbench, Springer-Verlag, New York, 1992.
- [Sutherland 1974] I.E. Sutherland. Three-Dimensional Data Input by Tablet., *Proceedings of the IEEE*, 62:453–461, April 1974.
- [Tarabanis *et al.* 1996] K. Tarabanis, R.Y. Tsai, and A. Kaul. Computing Occlusion-Free Viewpoints, *IEEE Transactions Pattern Analysis and Machine Intelligence*, 18(3), March 1996.
- [Tarabanis *et al.* 1995] K. Tarabanis, R. Tsai, and P. Allen. The MVP Sensor Planning System for Robotic Vision Tasks, *IEEE Transactions on Robotics and Automation*, 11(1), pp. 72-85, February 1995.
- [Tarabanis *et al.* 1995b] K. Tarabanis, R. Tsai, and P. Allen. Sensor Planning in Computer

- Vision, *IEEE Transactions on Robotics and Automation*, 11(1), pp. 86-105, February 1995.
- [Tarabanis *et al.* 1994] K. Tarabanis, R. Tsai, and P. 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), pp. 340-358, May 1994.
- [Tarbox & Gottshlich 1995] G.H. Tarbox and S.N. Gottshlich. Ivis: An Integrated Volumetric Inspection System, *Computer Vision and Image Understanding*, 61(3):430–444, May 1995.
- [Thompson *et al.* 1996] W.B. Thompson, H.J. de St. Germain, T.C. Henderson, and J.C. Owen. Constructing High-Precision Geometric Models from Sensed Position Data, *Proceedings of the 1996 ARPA Image Understanding Workshop*, pp. 987–994, February 1996.
- [Turk & Levoy 1994] G. Turk and M. Levoy. Zippered Polygon Meshes from Range Images, *Proceedings of SIGGRAPH*, pp. 311–318, 1994.
- [Turk 1992] G. Turk. Re-Tiling Polygonal Surfaces, *Computer Graphics*, 26(2), pp. 55-64, July, 1992.
- [Underwood & Coates 1975] S.A. Underwood and C.L. Coates. Visual Learning from Multiple Views, *IEEE Transactions on Computers*, pp. 651–661, June 1975.
- [Vemuri & Aggarwal 1986] B. Vemuri and J. Aggarwal. 3D Model Construction from Multiple Range Views using Range and Intensity Data, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 435–438, June 1986.
- [Vemuri & Aggarwal 1987] B. Vemuri and J. Aggarwal. Representation and Recognition of Objects from Dense Range Maps, *IEEE Transactions on Circuits and Systems*, pp. 1351–1363, November 1987.

- [Viewpoint Datalabs 1995] Viewpoint DataLabs, Inc. The Viewpoint Seal of Approval, from *Guidelines for Model Submission*, also available at <http://viewpoint.com>, 1995.
- [Wang & Wang 1994] Y. Wang and J. Wang. On 3D Model Construction by Fusing Heterogeneous Sensor Data, *CVGIP: Image Understanding*, 60(2):210–229, September 1994.
- [Whaite & Ferrie 1992] P. Whaite and F. Ferrie. Uncertain Views, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3–9, 1992.
- [Wheeler 1996] M. Wheeler. *Automatic Modeling and Localization for Object Recognition*, Ph.D. thesis, School of Computer Science, Carnegie Mellon University, 1996.
- [Winston 1970] P.H. Winston. Learning Structural Descriptions from Examples, Technical Report MAC TR-76, Massachusetts Institute of Technology, September 1970.
- [Wohlers 1994] T. Wohlers. State of the Industry, *Rapid Prototyping and Manufacturing '94*, Dearborn, Michigan, April 1994. Society of Manufacturing Engineers and the Rapid Prototyping Association.
- [Wu & Levine 1994] K. Wu and M. Levine. Recovering Parametric Geons from Multiview Range Data, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 156–166, 1994.
- [Yang & Allen 1998] R. Yang and P. Allen. Registering, Integrating, and Building CAD Models from Range Data, *Proceedings of 1998 IEEE International Conference on Robotics and Automation*, Belgium, 1998.