Masters Theses

Graduate School

5-1999

An algorithm and system for finding the next best view in a 3-D object modeling task

Laurana Moy-ping Wong

Recommended Citation

To the Graduate Council:

I am submitting herewith a thesis written by Laurana Moy-ping Wong entitled "An algorithm and system for finding the next best view in a 3-D object modeling task." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

M. A. Abidi, Major Professor

We have read this thesis and recommend its acceptance:

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

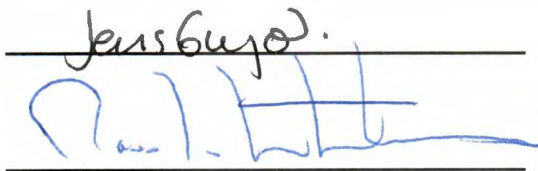(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Laurana Moy-ping Wong entitled "An Algorithm and System for Finding the Next Best View in a 3-D Object Modeling Task." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

M. A. Abidi, Major Professor

We have read this thesis
and recommend its acceptance:

Accepted for the Council:

Associate Vice Chancellor and
Dean of The Graduate School

AN ALGORITHM AND SYSTEM

FOR FINDING THE NEXT BEST VIEW

IN A 3-D OBJECT MODELING TASK

A thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Laurana Moy-ping Wong

May 1999

Dedication

This thesis is dedicated to my parents

Patricia Jo Wong

and

Seok Pin Wong

who have given me the means to get this far
and have never doubted that I could.

Acknowledgements

I want to thank Dr. Mongi Abidi for giving me this research opportunity and for being an excellent mentor. I would also like to thank Christophe Dumont for his assistance and friendship and my graduate committee for their guidance. My thanks go to Dr. Hamed Sari-Sarraf whose incredible teaching skills provided me with a new outlook on Electrical Engineering. My thanks also go to Sarah Cates for bringing me back into focus during a rough first semester of graduate school. Most importantly, I want to thank Josh Cates, whose presence in my life keeps me going through it all.

Abstract

Sensor placement for 3-D modeling is a growing area of computer vision and robotics. The objective of a sensor placement system is to make task-directed decisions for optimal pose selection.

This thesis proposes a Next Best View (NBV) solution to the sensor placement problem. Our algorithm computes the next best view by optimizing an objective function that measures the quantity of unknown information in each of a group of potential viewpoints. The potential views are either placed uniformly around the object or are calculated from the surface normals of the occupancy grid model. For each iteration, the optimal pose from the objective function calculation is selected to initiate the collection of new data. The model is incrementally updated from the information acquired in each new view. This process terminates when the number of recovered voxels ceases to increase, yielding the final model.

We tested two different algorithms on 8 objects of various complexity, including objects with simple concave, simple hole, and complex hole self-occlusions. The first algorithm chooses new views optimally but is slow to compute. The second algorithm is fast but not as effective as the first algorithm. The two NBV algorithms successfully model all 8 of the tested objects. The models compare well visually with the original objects within the constraints of occupancy grid resolution.

Objects of complexity greater than mentioned above were not tested due to the time required for modeling. A mathematical comparison was not made between the objects and their corresponding models since we are concerned only with the acquisition of complete models, not the accuracy of the models.

Contents

List of Tables

List of Figures

1 Introduction

To create a model of a 3-D object, we must acquire $2\frac{1}{2}$-D information from several locations around the object. The question is, how do we choose these locations? One way is to uniformly place enough viewpoints around the object to acquire a complete model. This method, however, is inefficient and may be ineffective when dealing with many different complex objects. It would be better to use an algorithm that makes task-directed decisions about where to place new viewpoints. Such an algorithm greatly decreases the number of views needed to model an object. A decrease in the number of viewpoints shortens the total modeling time and also saves data storage space.

A task-directed algorithm used to make view placement decisions is called a Next Best View algorithm. The *Next Best View* (NBV) is a new sensor position that reveals an optimal amount of unknown information about the object being modeled. We want to determine sensor poses (viewpoints, views, or sensor/view positions) that provide the next best view in a 3-D object modeling task. The goal of a NBV modeling system is to model an object using the smallest number of views. Our NBV system acquires range data from the object of interest and creates a volumetric model. An objective function uses the information in the volumetric model to calculate the best place to position a new viewpoint. Additional range data is then acquired from the pose of the new view. The volumetric model is updated with the new range information, and a new view is calculated. This process repeats until the model is complete.

3-D object modeling and the NBV have many applications. 3-D modeling is used in various areas including manufacturing inspection, prototyping, and virtual reality. We found examples of next best view planning in 3-D Object Modeling, Object Recognition and Localization, Feature Detectability, and Gaze Control (see

Chapter 2).

1.1 Overview

The NBV system consists of three parts: the Range Scanner, the Model Builder, and the NBV Decision Maker. The Range Scanner simulates the acquisition of range data from a computer object (see Section 3.3). For testing purposes, we chose to simulate a range scanner instead of using a real range scanner. However, the output of the Range Scanner is the same as the output of a real range scanner. In fact, the NBV system could be used with range information acquired from a real object as long as the range images were pre-registered. We do not actively register our range data because we use a *viewsphere*. The viewsphere is a spherical shell that encloses the computer object and limits the number of possible camera locations (see Section 3.1.2). The acquired range images are used by the Model Builder to create a volumetric model (see Section 3.4). The volumetric model is an *occupancy grid* that contains all current information about the object being modeled and is composed of *known* and *unknown* data. Since each viewpoint reveals only object surface information or known data, we also want to infer something about what we cannot see. Therefore, we extract unknown data from each surface point that represents the hidden information behind the surface data. The NBV Decision Maker examines the unknown information in the volumetric model and uses an objective function to determine the best place to position a new viewpoint (see Section 4). We have tested two different objective functions, the *Optimal Method* and the *Surface Normal Method* (see Section 4.1). We have also tested the Adaptive Method, a combination of the Optimal and Surface Normal methods. The pose of the new viewpoint is used to acquire more range data. The Model Builder updates the volumetric model with the new range information, and the NBV Decision Maker

2

calculates the next best view. This process repeats until the model is complete.

We use the experimental results of the Optimal Method to set benchmarks that can be used to assess the performance of other NBV algorithms. Benchmarks of this kind are sorely needed. The Adaptive Method is used for real-time implementation. Our system is robust enough to handle objects of any complexity within the constraints of system setup. In particular, the system can model objects with *self-occlusions*. Self-occlusions are areas of an object that are occluded or blocked from particular viewpoints by other areas of the same object. For example, the inside of a coffee mug is self-occluded from viewpoints that do not point directly into the mouth of the mug.

1.2   Thesis Outline

In Chapter 2 we look at the work of others in the area of sensor planning to find the impetus for our work. Chapter 3 discusses the elements of the NBV system and the formulation of our problem. Chapter 4 gives the details of our NBV finding algorithm. Chapter 5 contains experimental results and Chapter 6 discusses the findings of our research and the possibilities for future work.

2 Review of Related Work in Sensor Planning

Sensor planning is the task-directed process of understanding and quantifying the relationship between objects and the sensors that view them [1]. Sensor parameters such as field of view, focus, and pose (sensor position and orientation) are adjusted to satisfy the task at hand. Sensor planning has applications in four main areas: 3-D object or scene modeling, object recognition and localization, feature detectability, and gaze control.

The following sections review work previously done in the area of sensor planning. The focus of our research lies in object modeling, and Section 2.1 covers this area in depth. Section 2.2 examines other topics in sensor planning that relate to the NBV problem.

2.1 3-D Object Modeling

In building a 3-D model of an object, a single $2\frac{1}{2}$-D view of an object is insufficient to create the model. Therefore, 3-D object modeling is performed by integrating the data found in several $2\frac{1}{2}$-D views. Sensor planning is used to minimize the number of views required to construct a complete 3-D model.

There are two principal methods used to solve the NBV problem: search based and occlusion based methods. Search based methods use optimization criteria to search a group of potential viewpoints for the best view. Work on search based methods can be found in [2], [3], [4], [5], [6], and [7]. Our approach is a search based method. Occlusion based methods use the occlusion boundaries in the $2\frac{1}{2}$-D image of the current view to choose the next view. Work on occlusion based methods can be found in [8], [9], [10], [11], [12], [13], and [14]. Other work on the

4

NBV problem can be found in [15], [16], [17], [18], [19], and [20].

2.1.1 Search Based Methods

The work of Connolly [2] is some of the earliest done on the next best view problem. Connolly starts with a partial octree model of a scene, and the octree root is initialized to an *Unseen* state. The Unseen state indicates which volumes have not yet been seen by any view. He presents two algorithms, the planetarium algorithm and the normal algorithm. The planetarium algorithm finds, on a uniformly sampled sphere, the largest Unseen area and uses this point to define the next best view vector. The normal algorithm measures the visibility of Unseen nodes. For each Unseen node face, the number of adjacent *Empty* nodes is counted. A node is Empty if it contains information not on the object's surface. The next best view vector intersects the vertex whose three adjoining cube faces have the highest count values.

Connolly's planetarium algorithm provides good results; the algorithm usually completes modeling within a small number of views (4 or 5 for simple polyhedral objects). It is not apparent, however, if this algorithm can handle objects of any complexity.

Marchand and Chaumette [3] use sensor planning to model a scene of polyhedral objects and cylinders. An initial view of a scene is acquired, and all the primitives in the image are estimated. Local exploration is performed to fully recover primitives that are partially outside the field of view. Global exploration is then used to explore unknown areas and to find new viewpoints. The global strategy is based on the minimization of an objective function. The function reflects the quality of the new viewpoint; the gain of information associated with the new position is weighed against the cost of moving the camera. Additional constraints are incorporated

into the objective function to avoid choosing viewpoints that are unreachable by the robot or near the robot joint limits.

Marchand and Chaumette's algorithm is robust and formulated for practical use, but it does not ensure completion of the model (especially in occluded areas). Because the algorithm explores locally as well as globally, many camera moves are required during the modeling process. The example given in the paper uses 30 camera moves to model a scene of five simple objects (planes and cylinders).

Milroy et al. [4] use view planning to automate the laser scanning of objects during rapid prototyping. *Orthogonal Cross Section* (OCS) is the modeling method used in this work. The OCS model is a surface mesh that traces an object's contours with closed curves that run in the x, y, and z directions only. The first scan is taken from the top view of the object. *Air points* are calculated for the occlusion boundaries found in this view. An air point occurs when a contour curve meets the unseen volume of the OCS model. In other words, an air point is the endpoint of an open contour curve (a complete OCS model will have only closed contour curves and no air points). The outermost air point defines five possible new viewpoints. The air points within a window defined by each viewpoint are assigned *visibility values*. A visibility value reflects the likelihood that the neighboring unseen volume will be imaged in the new view. The viewpoint with the highest total visibility is chosen as the next view.

The method of Milroy et al. can handle complex objects. The model, however, is not always complete because the scanner used is unable to reach certain areas. Because the algorithm uses a surface growing approach, the number of required scans is a little high for the complexity of objects used. For example, a telephone handset required 20 scans.

Sequeira et al. [5] use active sensing for the modeling of 3-D environments. A range image is acquired from an arbitrary initial viewpoint. The areas of occlusion

are detected using jump edges in the range data and the projection of the jump edges. For each occluded area, a set of view directions is found from which the area of occlusion is completely visible. The intersection of the sets of views defines a *global viewing volume* from which the maximum number of detected occlusions can be imaged. The view from the global viewing volume which maximizes an objective function is chosen as the next best view. The objective function rates the quality of a view according to its contribution to the measurement accuracy of the laser range finder used in image acquisition.

The results of this work are not clear from the report. Only results for the corner of a room are given, and it is unclear how complete the model actually is.

Zha et al. [7] use sensor planning in the modeling of 3-D curved objects. Possible new viewpoints are limited to the surface positions on a tessellated sphere enclosing the object to be modeled. The viewpoints near the current viewpoint are also ignored. The resulting set of potential viewpoints is evaluated using a rating function. The viewpoint which maximizes the rating function is chosen as the next best view. The rating function measures three heuristics: the distance of the new view from previous views, the amount of overlapping information between a new view and the current view, and the smoothness of the overlap between a new view and the current view. The last two heuristics help to improve the accuracy of range image registration.

Zha et al. give no examples of complete models. The objects modeled did not contain any holes or difficult self-occlusions, but all results contain areas that have not been sensed. Required run time for modeling was also not given.

2.1.2 Occlusion Based Methods

Maver and Bajcsy [12] use sensor planning during scene modeling. An initial image is acquired from an arbitrary direction. The occluded areas in this image (*range shadows*) are determined and approximated as polygons. Each contour of the range shadows is segmented into a series of straight lines. The breakpoints of the contour segmentation are used to divide the occluding polygons into areas. Then, *viewing angles* for every pixel in the occluding polygons are calculated. A viewing angle is the sector containing all scanning directions from which a pixel is visible. For each area of the occluding polygons, the histogram of viewing angles is calculated. The histogram shows how many pixels can be seen from a certain direction. Histogram decomposition is then performed to find the next scanning direction; the next view direction is selected from the intervals defined by the maxima in the final histogram.

Maver and Bajcsy test their algorithm only on scenes of simple objects. A small number of views is needed for modeling: 3 views for a scene comprised of polyhedra and 5 views for a more complex scene comprised of a polyhedron, a half cylinder, a telephone receiver, and a wedge. However, this method does not always work with occlusions such as holes.

Kutulakos [10] (or see [11] for an abbreviated version) uses purposive vision to recover the global description of arbitrary curved objects. In this work, Kutulakos studies the relationship between the geometry of an object's occluding contour and the way the occluding contour deforms when the viewpoint changes. To solve the modeling problem, the motion of the viewpoint is controlled in a continuous fashion to constrain the deformation of the occluding contour. Modeling is divided into three tasks. Local surface modeling recovers shape for a region around a selected point of the occluding contour. Incremental surface modeling grows the regions found by the local method, and global surface modeling adds rules to the first two

8

strategies.

Kutulakos' approach is not to optimize the modeling process. Instead, he guarantees global modeling. The objects to be modeled must be smooth but may be arbitrarily shaped.

Abidi [21], [8] uses active sensing for the autonomous volumetric modeling of objects. From the first acquired image, the contour of the object is calculated. An entropy-based objective functional that incorporates shape and contrast is used to measure the information content of the object contour. The contour is separated into segments and the objective functional value for each segment is calculated. The segment with the minimum entropy is used to determine the next best view.

The results of Abidi's system are as expected; the number of images needed for modeling increases with object complexity, and the accuracy of the models is higher for objects with simpler shapes. The system can also handle objects with hole-like concavities. However, if an object contains concavities that do not pierce the object all the way through, the resulting models are entirely convex and, thus, inaccurate. This limitation of the system is caused by the use of intensity sensors for information gathering.

Banta et al. [9], [15] uses sensor planning to volumetrically model objects. The surface normal and curvature at each point in the initial image are calculated. The surface normal is used to calculate view positions for the three points with greatest curvature. Ray tracing techniques are then used to the determine the amount of hidden information which may be revealed by each of the three candidate viewpoints. The view which reveals the most hidden information is chosen as the next best view.

Our research is based on Banta's thesis work. We use a similar occupancy grid framework and his concept of a viewsphere (see Chapter 3). The results in Banta's thesis, however, are not optimal, and the methodology is based on heuris-

9

tics. Banta uses different NBV algorithms throughout the modeling process instead of implementing a single objective function.

2.2 Other Related Topics in Sensor Planning

The following sections review work in sensor planning that relates to the NBV problem. Section 2.2.1 contains research in which a sensor is placed to aid in the recognition of an object. Section 2.2.2 reviews literature in which a sensor is placed to satisfy certain constraints. Section 2.2.3 discusses work in the placement of sensors to gather information.

2.2.1 Object Recognition and Localization

In the area of object recognition/localization, model-based sensor planning is used to disambiguate between the interpretations of an object or to determine its pose. Consider the situation in which a machine has a set of objects that it can recognize. A single view of an object may match multiple objects in the set, but an additional view or sequence of views can be used to distinguish the target object from the others. In such a situation, sensor planning is used to choose a sensor pose that will most quickly lead to the recognition of an object. Work on sensor planning for object recognition or localization can be found in [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], and [32].

Grimson [25] performs object recognition on polyhedral objects. In this work, the initial set of possible interpretations is already given. The distinguishability of each pair of interpretations is determined. Two interpretations are well distinguishable if their geometrical features are significantly different. The next sensing position, which best disambiguates the possible interpretations, is chosen to maxi-

10

mize distinguishability.

The work of Hutchinson and Kak [26] applies sensor planned object recognition to polyhedral objects. An arbitrary view of the object is first acquired and hypotheses are made about the identity of the object. The objective is to select a viewpoint which uniquely identifies the object. To rate the ambiguity of a possible new viewpoint, they predict the set of features which would be observed from this viewpoint for each hypothesis. Then, they determine the hypothesis corresponding to each set of features and calculate the ambiguity. The viewpoint that minimizes the ambiguity is chosen as the next best view.

Lee and Hahn [28] perform recognition and localization on natural quadric objects. An initial view is first acquired and the features obtained from this operation are used to construct a *multiple interpretation image* representing all possible interpretations of the object under scrutinization. The next view is limited by system constraints and is chosen according to a measure of discrimination power, i.e. the number of interpretations that can be eliminated by a view.

2.2.2 Feature Detectability

Sensor planning is used in this area to place views according to feature detectability constraints such as resolution, focus, and field-of-view. Also, if an object is in a cluttered workspace, a sensor must be positioned such that the feature of interest is in clear view (unoccluded).

The methods for feature detectability generally take either a generate-and-test or a synthesis approach [1]. In generate-and-test techniques, sensor configurations are generated and then evaluated according to task requirements. Work using the generate-and-test approach can be found in [33], [34], [35], [36], [37], [38]. In the

11

synthesis approach, task constraints are characterized analytically and then used to restrict sensor parameters. Work using synthesis techniques can be found in [39], [40], [41], [42], [43], [44], [45], [46], [47].

Sakane et al. [35] use a depth buffering algorithm to locate regions in a scene that do not occlude the object of interest. They center a triangular meshed geodesic dome around the target object to limit the number of possible views. The radius of the dome is large enough to encompass the workspace. A possible viewpoint is centered on a facet of the dome. If a ray from the viewpoint to the target point is intersected by an object in the workspace, the distance from the target point to the intersecting object is measured. If this distance is smaller than the radius of the dome, the facet corresponding to the viewpoint in question belongs to an area of occlusion.

Cowan [41] introduced the set of feature detectability constraints that is used in most of the literature. His requirements for feature detectability are resolution, focus, field-of-view, and visibility. The resolution requirement constrains the maximum distance between a viewpoint and the feature of interest, while the focus requirement constrains the minimum distance between a viewpoint and the feature. To satisfy the field of view requirement, the feature of interest must lie completely within the camera field of view, and to satisfy the visibility requirement, the feature of interest must be unoccluded.

In [44], Tarabanis et al. use the feature detectability constraints of resolution, focus, and field-of-view defined in [41] and characterize them more analytically. These constraints are integrated by optimizing a function that is the weighted sum of the constraint criteria. The end product of the integration is the *MVP model-based vision sensor planning system*, a system used to automatically locate camera views for a robotic vision system.

Tarabanis et al. [46] compute occlusion-free viewpoints in a polyhedral world. In this work, a viewpoint has a *viewing cone* associated with it. The viewing cone is the set of rays that connect the viewpoint to all points on an object or feature. A view is unoccluded if its viewing cone does not intersect anything in the environment. The intersection of these unoccluded viewing cones are used to calculate the boundaries of a total visibility region.

2.2.3 Gaze Control

Sensor planning is also used to control gaze. Sensors are placed to maximize the information content in a view or to shift the focus of attention to salient features. Because of the general nature of this work, applications for it fall within the other areas of sensor planning. Work in sensor planning for gaze control can be found in [48], [49],[50], [51], [52], [53], [54] [55], [56], and [57].

Hager and Mintz [51] use model-based sensor planning to maximize the information content of a view. Their approach uses decision theoretic techniques to weigh the performance of a view against the cost of implementing that view.

Maver et al. [54] maximize the amount of new information captured in a view. An initial view of a group of objects is captured arbitrarily. The 3-D coordinates of the interesting object silhouette points are calculated. The next view is the point of interest from which the largest number of object vertices can be determined.

Westelius [57] uses sensor planning to focus the attention of a robotic vision system. By methodology similar to a human's shift in attention, the system locates an object by focusing on areas of high complexity and scans the located object by tracking its linear structures. Applications for this work include automatic recognition, inspection, or surveillance of objects in a scene.

2.3 The Impetus for our Work

After reviewing the literature in sensor planning for 3-D object modeling, it is apparent that this topic requires continued research. Most of the reviewed methods cannot handle objects with holes or self-occlusions. The reviewed literature is also lacking in a standard by which to assess performance. There are no optimal methodologies that can provide a comparison; no one is truly able to judge how well their algorithms operate.

Our work provides experimental results that can be used as a benchmark to assess the performance of other NBV algorithms. Our system can also handle objects of any complexity within the constraints of the system setup.

3 The NBV System

Section 3.1 introduces the general 3-D modeling problem. Section 3.2 discusses the software created to test our NBV algorithms. Sections 3.3 and 3.4 give the details of the Range Scanner and the Model Builder respectively. Section 3.5 summarizes the contents of this chapter.

3.1 Problem Formulation

Section 3.1.1 provides a general framework for the NBV problem as applied to 3-D object modeling, and Section 3.1.2 discusses the assumptions made to simplify the general problem.

3.1.1 The General 3-D Modeling Problem

Creating a 3-D model of an unknown object involves acquiring data from the object at different viewpoints, registering the data, and then integrating all acquired information into a single model. Even though our main concern is finding the next best view, we must also construct a system to implement the 3-D modeling problem. The implementation of the general modeling problem can be complex. There are data acquisition and registration issues to consider. There are costs related to moving the sensor or camera to a new position. There may also be constraints on the time allowed to perform each step of modeling.

The first thing to consider is the quality of any acquired view. An image must be readable in order to extract viable information from it. Factors that effect view quality are resolution, focus, and field of view. For adequate resolution, the distance from the camera to the object must not exceed some maximum. For proper focus,

the distance from the camera to the object must not go below some minimum. Field of view constraints require that the desired object appears within the camera's view. This includes restricting the NBV search to viewpoints that are reachable by the camera. For instance, if the object to be modeled is sitting on a surface, the next view chosen cannot not lie beneath the surface.

Another factor to consider is the registration of all acquired views. To provide coherency of information, each view must be integrated into a world model. Therefore, an image obtained from a particular viewpoint must contain information that has already been acquired. This usually demands some measure of overlap between each image acquired.

When implementing a NBV algorithm, the cost of acquiring the next view should also be considered. One factor that contributes to this cost is the distance that the camera must be moved from its present position to reach the next view position. Another factor is the time required to make such a move. An additional time consideration is time to compute the next best view; a view that reveals the largest amount of unknown information may take too long to find.

3.1.2   Assumptions Made to Simplify the General Problem

To simplify the general problem discussed in Section 3.1.1, we want to make a few assumptions about the object to be modeled. We assume that the size of the object is roughly known and that the approximate object center is also known. Assumptions of this kind are not really limiting because the general object size is usually a known parameter. These assumptions allow us to use a *viewsphere* (see Figure 1). The viewsphere is a shell that encloses the object to be modeled and limits the number of possible camera locations. Our viewsphere is spherical but could be any desired shape. Each new view intersects the viewsphere and points
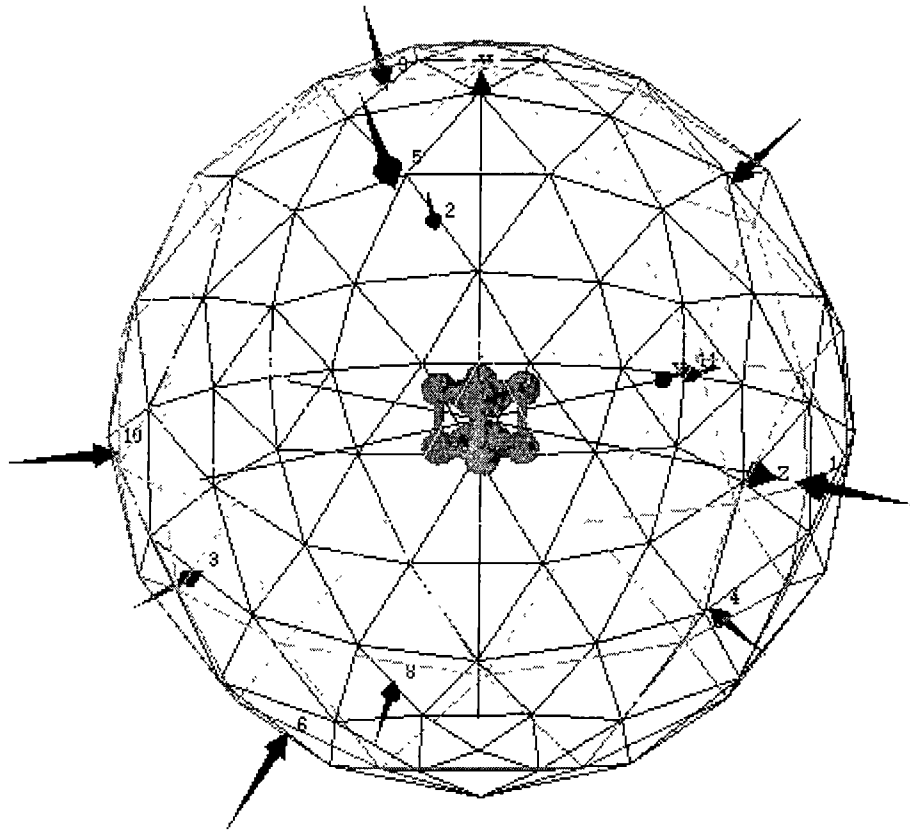
Figure 1: The viewsphere simplifies the general 3-D modeling problem.

17

toward the center of the viewsphere.

The viewsphere simplifies the general modeling problem in several ways. View quality is predefined because the minimum and maximum distances from the camera to the object are already set by the viewsphere. The object is always positioned properly in the camera's field of view because the camera points toward the center of the object. Since the pose of each view is known and the viewsphere remains in a constant position throughout modeling, no overlapping between views is required for registration purposes. For this work, we would like to be able to place the camera anywhere around the object. Therefore, we do not currently consider the costs of moving the camera.

3.2   The NBV System Software

The NBV system software consists of three parts (see Figure 2): the Range Scanner, the Model Builder, and the NBV Decision Maker. The Range Scanner simulates the acquisition of range data from the *archetype*, the object being modeled. The first range image is acquired from an arbitrary position and is used by the Model Builder to create a volumetric model. The NBV Decision Maker then examines the information in the volumetric model and uses an objective function to determine the best place to position a new viewpoint. The pose of the new view is used to acquire more range data. The Model Builder then updates the volumetric model with the new range information, and the NBV Decision Maker calculates the next best view. This process repeats until the model is complete. Sometimes it is not possible to complete the model. Therefore, the NBV system will stop if new information is no longer acquired. After the final model is constructed, it is converted into a surface mesh for better comparison to the original archetype object.
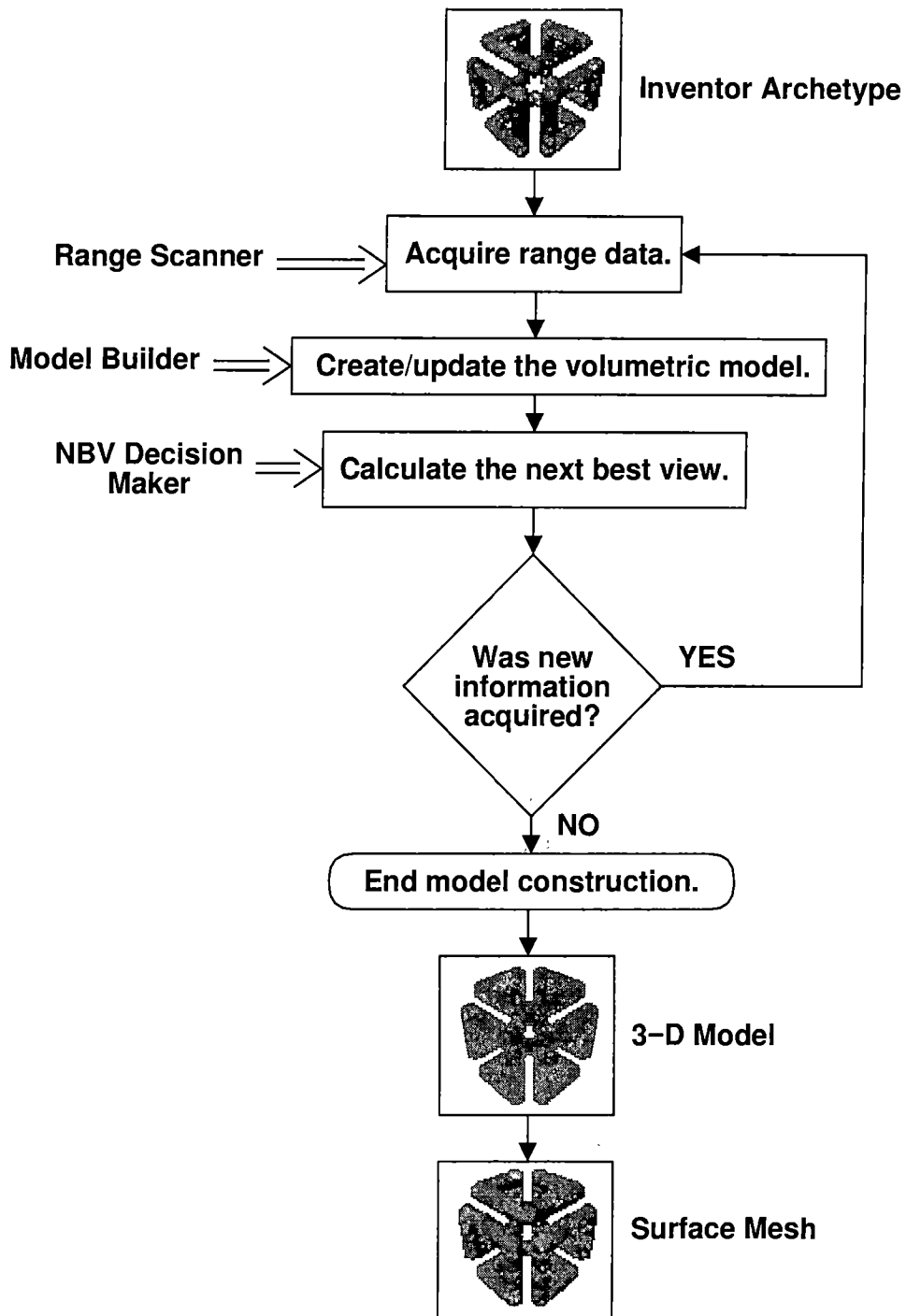
Figure 2: The flow of the Next Best View System.

All parts of the system are written in C++, and we use the command line or a graphical user interface to change all system parameters (see the Appendix). The archetype objects are in Inventor format, the volumetric model is output for viewing in Inventor format, and a surface representation of the model is created from the Inventor model file.

3.2.1   The NBV System Parameters

There are several NBV parameters that determine the setup of the system. A change in these parameters can also effect the results of the NBV program. Variations in workspace size, occupancy grid resolution, range image resolution, field of view, viewsphere radius, and pose of the initial viewpoint contribute to changes in the NBV system results.

The workspace is a cube that is the approximate size of the archetype object. The workspace dictates how far the unknown area extends back from the surface of the first view; the larger the workspace the larger the unknown area. Therefore, the size of the workspace may effect the performance of the NBV program; an increase in the workspace size increases the size of the unknown area.

The occupancy grid is always the same size as the workspace. The size of each voxel in the occupancy grid is determined by the occupancy grid resolution; increasing the occupancy grid resolution decreases the size of each voxel. An increase in the occupancy grid resolution normally causes an small increase in the number of views required to model complex objects. A change in resolution also has an effect on the run time of the NBV system. Increasing the occupancy grid resolution usually requires an increase in the resolution of the range image. Therefore, the time needed to acquire range data and create the volumetric model also increases. The objective function also takes longer to calculate because of the increase in the

number of voxels to consider. The NBV program speed is similarly effected by changes in the workspace size and the view sphere radius.

In order to update each voxel properly, range image resolution must correspond correctly to voxel size. The relationship between the resolution of the range image and the resolution of the occupancy grid has the greatest effect on the results of the NBV system (see Section 3.4).

The field of view determines how much of the workspace can be imaged by the camera. If the field of view is too small, not all of the object is captured in a single view. As the field of view increases, the range image resolution must also increase. If the field of view is too large, the range image will be filled with empty space.

Altering the size of the viewsphere radius changes the number of views required to model an object. Because the range data are acquired with a perspective projection camera, the amount of data gathered in a particular view changes as the distance from the camera to the object changes; a camera positioned very close to the archetype does not acquire as much information as one that is twice as far away.

The pose of the initial viewpoint is chosen arbitrarily but is usually aligned with an axis. Since the information in each possible view is different, a change in the initial viewpoint may alter the final results.

3.3   The Range Scanner

To build a 3-D model of an object we must capture $2\frac{1}{2}$-D data from several locations around the object (see Figure 3). The Range Scanner (see Figure 4) simulates the acquisition of $2\frac{1}{2}$-D range data from the archetypes, 3-D Inventor computer objects.
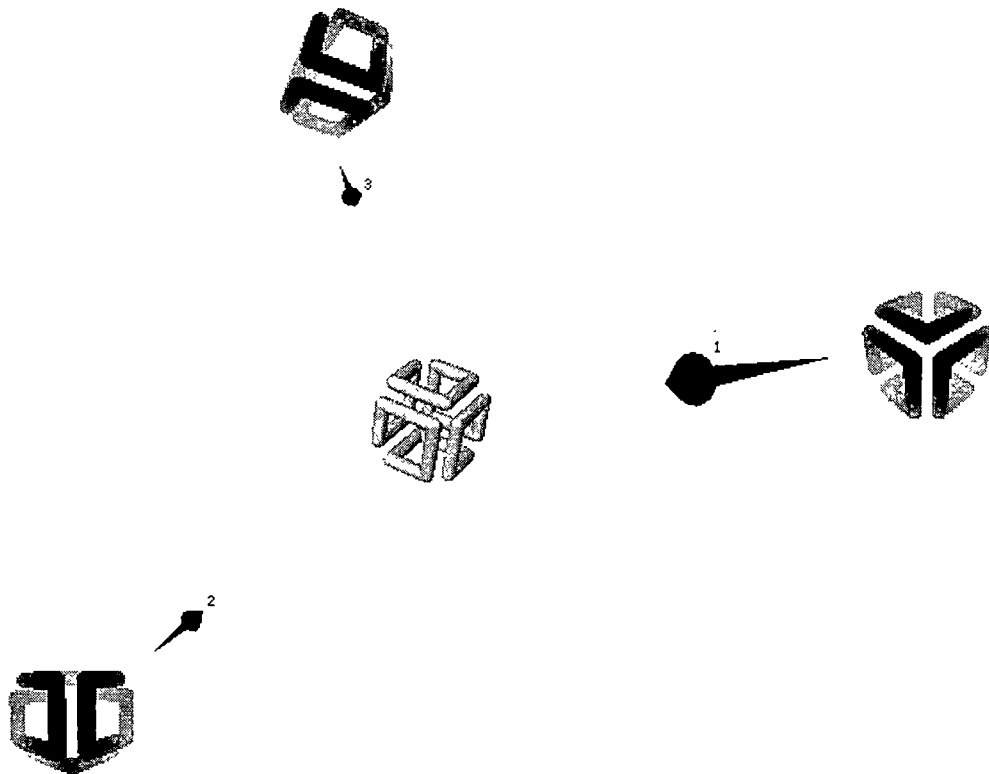
Figure 3: The Range Scanner acquires $2\frac{1}{2}$-D data from several locations around the archetype object.
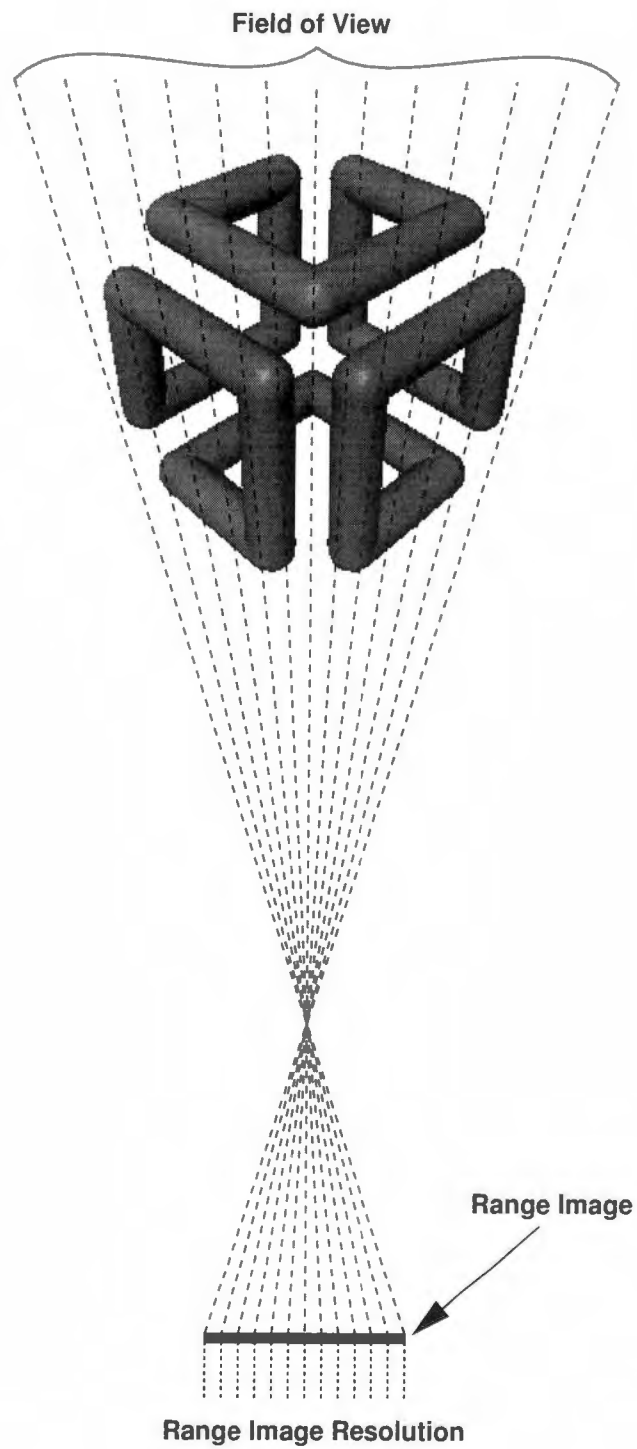
Figure 4: The Range Scanner casts a series of rays to acquire range data points.

We use a perspective projection camera model that always points toward the center of the viewsphere (see Section 3.1.2). Using the Inventor class SoRayPickAction, rays are projected forward from the camera to the archetype object to acquire range data points. The direction of each range data ray is calculated in spherical coordinates according to the field of view angle and the resolution of the range image (see Figure 4). The field of view determines how much of the object can be imaged by the camera. The range image resolution corresponds to the number of pixels in the range image. Even though the Range Scanner acquires range information from synthetic objects, the captured range images are the same as real range images.

3.4 The Model Builder

The first range image captured by the Range Scanner is used by the Model Builder to create a volumetric model. The volumetric model is a cubic 3-D array of voxels called an *occupancy grid*. The occupancy grid is a discretization of the workspace. Figure 5 shows an example of a 2-D occupancy grid. A 3-D occupancy grid is simply a 2-D grid with an added dimension.

Each 3-D pixel or *voxel* in a 3-D occupancy grid represents a part of the object volume. Our occupancy grid contains all current information about the archetype object and is composed of *known* and *unknown* data. Known data (the gray area in Figure 6) is the object surface information. Range information from one view of an object provides only known data. Since each viewpoint reveals only known data, we also want to infer something about what we cannot see. Therefore, we extract unknown data from each surface point that represents the hidden information behind the known data (the black area in Figure 6). After a new viewpoint is acquired, the known and unknown information associated with the view is integrated with the data acquired from all previous views; the model is updated after each new view

24
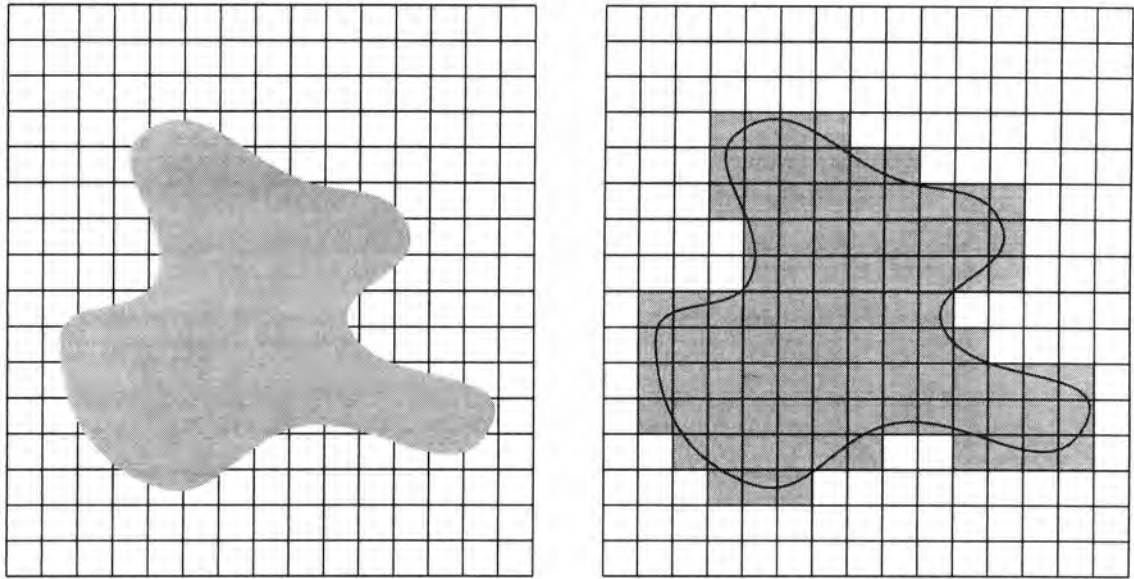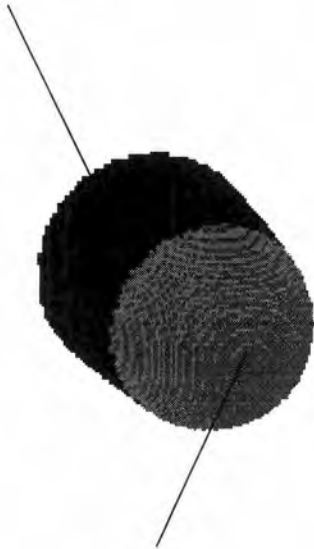
Figure 5: The occupancy grid is a discretization of the workspace. The 2-D occupancy grid (right) for a 2-D object (left).

Figure 6: The occupancy grid model is composed of known and unknown data. The known data is grey and the unknown data is black. The arrow represents the camera position of the acquired view.

to reflect what is currently known about the archetype (see Figure 7). A model is actually represented in software by a known occupancy grid and an unknown occupancy grid. The two grids are combined for visualization of the entire model.

The Model Builder creates the known occupancy grid by taking a point from the range image, calculating its position in the occupancy grid, and updating the corresponding voxel. If the range image resolution is too small compared to the occupancy grid resolution, some voxels will not be updated (see Figure 8). Even a range image resolution that is sufficient for one view may not update all voxels for every view. All results of the NBV system reflect this correspondence problem. To demonstrate the effects of the correspondence problem, however, we have generated one set of results that do not exhibit the correspondence problem. In this case, the Model Builder starts with a voxel, calculates its position in the archetype object space, and then calls the Range Scanner to acquire range data at the calculated point. With this methodology, every voxel is updated properly. These results, therefore, do not exhibit the correspondence problem but are impossible to implement with real range data.

The unknown grid is the combination of two grids created in two different ways. Two grid-filling techniques are needed because each technique was found to have drawbacks. The first implemented technique was to initially empty the grid. Each voxel from the known grid is projected away from the camera (*back-projected*) in the direction of the current view to the extent of the workspace. All voxels in the back-projection path are filled (see Figure 9(a)). This back-projecting technique fills the voxels behind the edges of the known grid (the unknown edge voxels) well but leaves holes inside the grid. Therefore, a second technique was implemented. In this second technique, the grid is initially full. The voxels that are in the background of the range image are emptied (see Figure 9(b)). This carving technique creates a solid grid but does not fill the unknown edge voxels properly. If the unknown edge
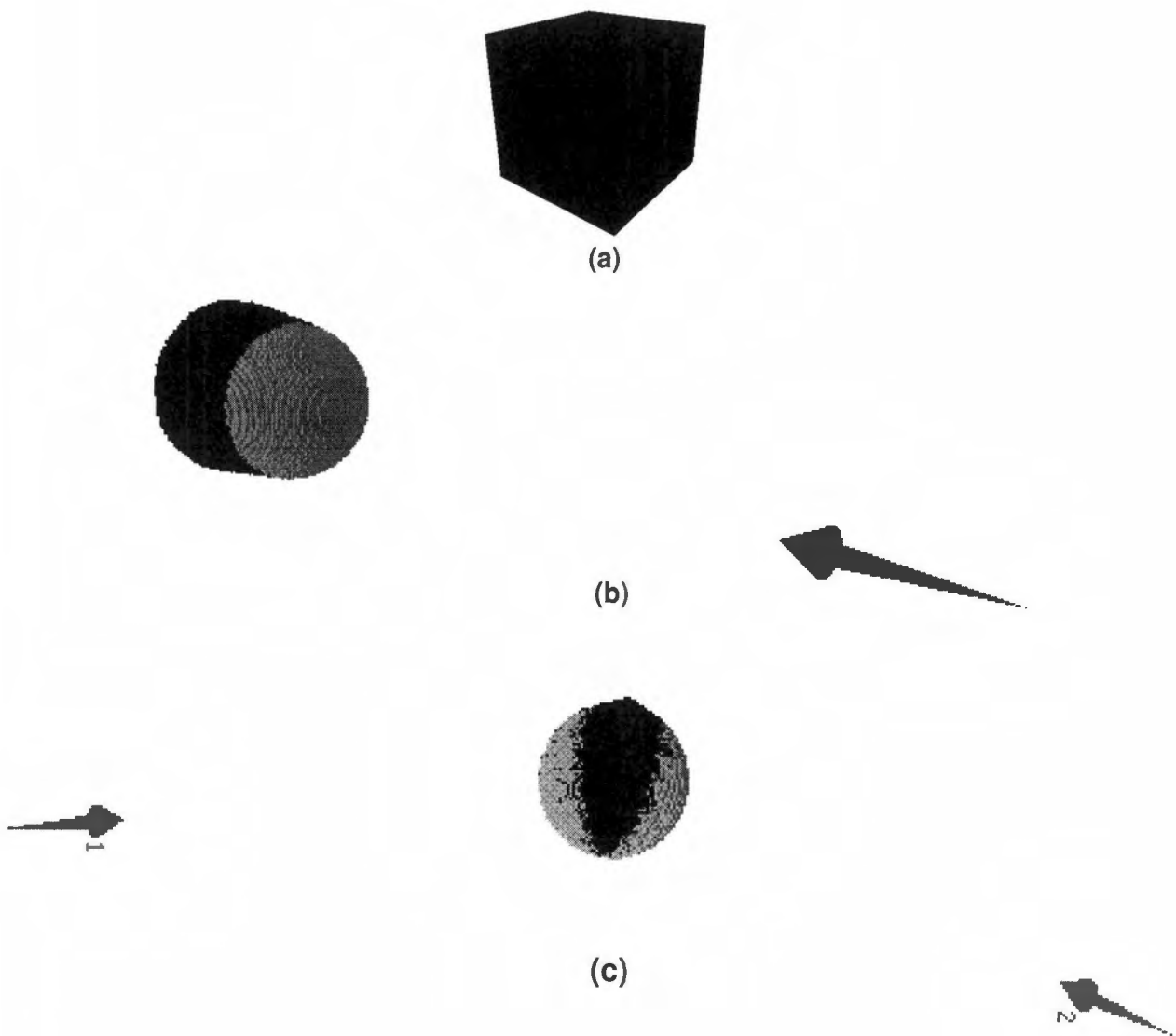
Figure 7: The model is updated after each new view to reflect what is currently known about the archetype. The volumetric model (a) before the acquisition of the first view. The volumetric model (b) after the acquisition of the first view. The volumetric model (c) after the acquisition of the second view. The known data is grey and the unknown data is black. The arrow represents the camera position of the acquired view.
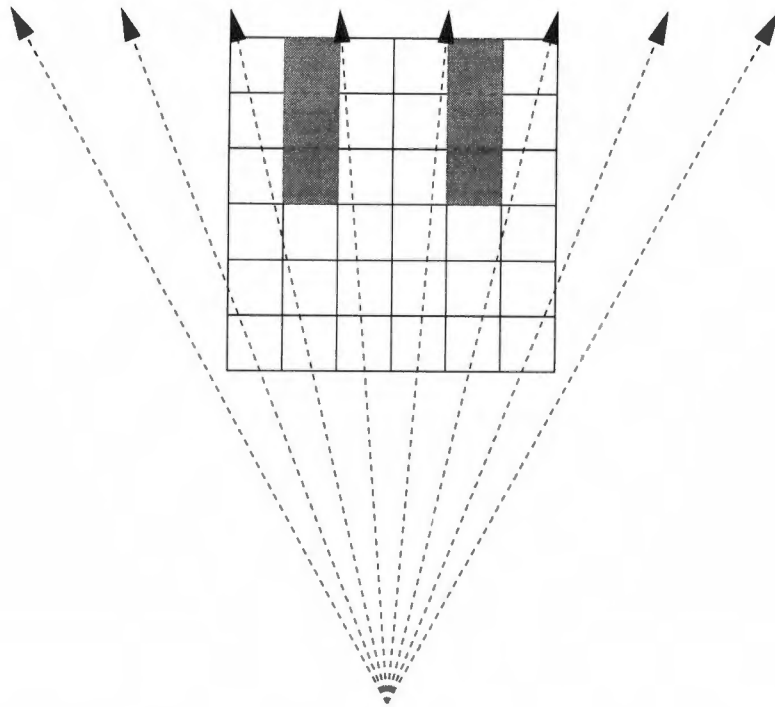
Figure 8: The correspondence problem. The voxels towards the front of the occupancy grid are updated perfectly. The gray voxels, however, are not updated due to the correspondence problem. The dotted lines represent the rays cast to acquire range data.
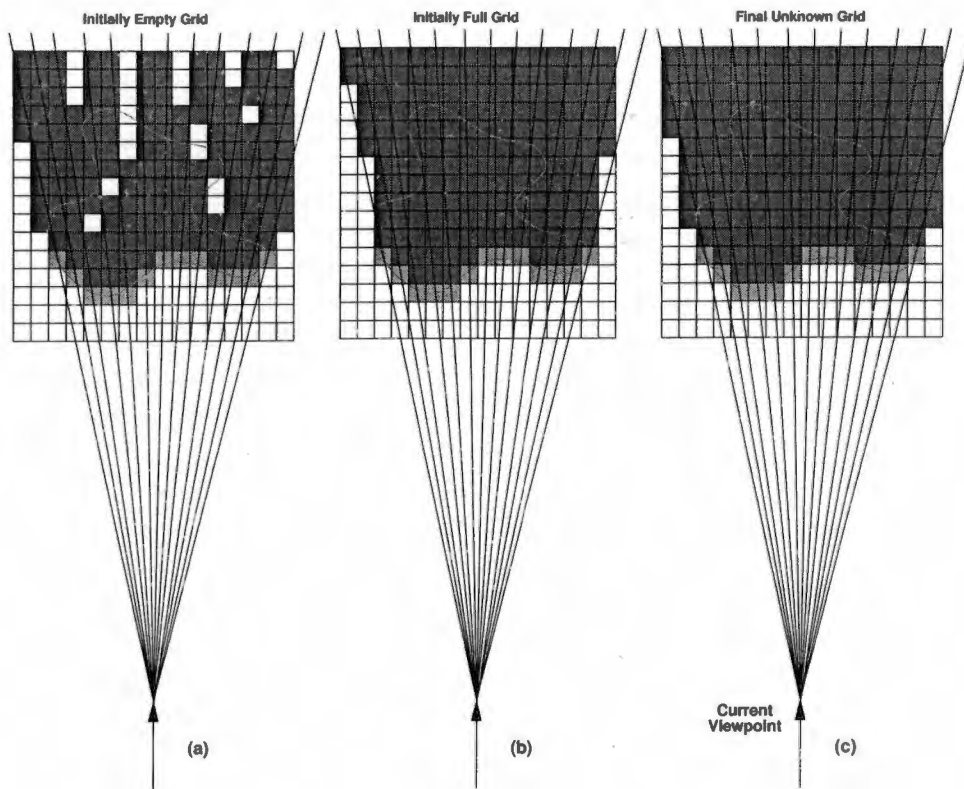
Figure 9: The unknown grid is the combination of two grids created in two different ways. The unknown grid (c) is obtained by ORing the unknown data in grids (a) and (b). The unknown grids are dark gray, and the known grids are light gray.

30

voxels are not filled correctly, there will be empty spaces between the known and unknown grids in the final model. Because of the imperfections of each technique, both techniques are implemented and the resulting two grids are ORed to form the unknown occupancy grid for the current view (see Figure 9(c)). The unknown grid of the current view is then ANDed with the unknown grid constructed for all views thus far. Figure 10 shows how voxels are ANDed and ORed.

To visually compare the volumetric models to the corresponding archetype objects, we use a marching cubes technique (see [58]) to create surface representations of the final occupancy grid models (see Figure 11). The marching cubes algorithm fits a surface to the individual voxel data points.

3.5 Summary of the NBV System

We have discussed the general 3-D modeling problem and the assumptions made to simplify the general problem. The most important assumption is that of the viewsphere. The viewsphere controls view quality and restricts the placement of each viewpoint so that registration does not have to be manually performed.

We also described the NBV system software. The NBV system consists of a Range Scanner, a volumetric Model Builder, and a NBV Decision Maker. The Range Scanner acquires range data from the archetype object. The Model Builder creates and updates the volumetric model, and the NBV Decision Maker determines the best place to acquire more range data. The NBV Decision Maker is described in the next section.

Figure 10: ANDing and ORing voxels. The voxel ANDing operation is demonstrated on the left. The voxel ORing operation is shown on the right.
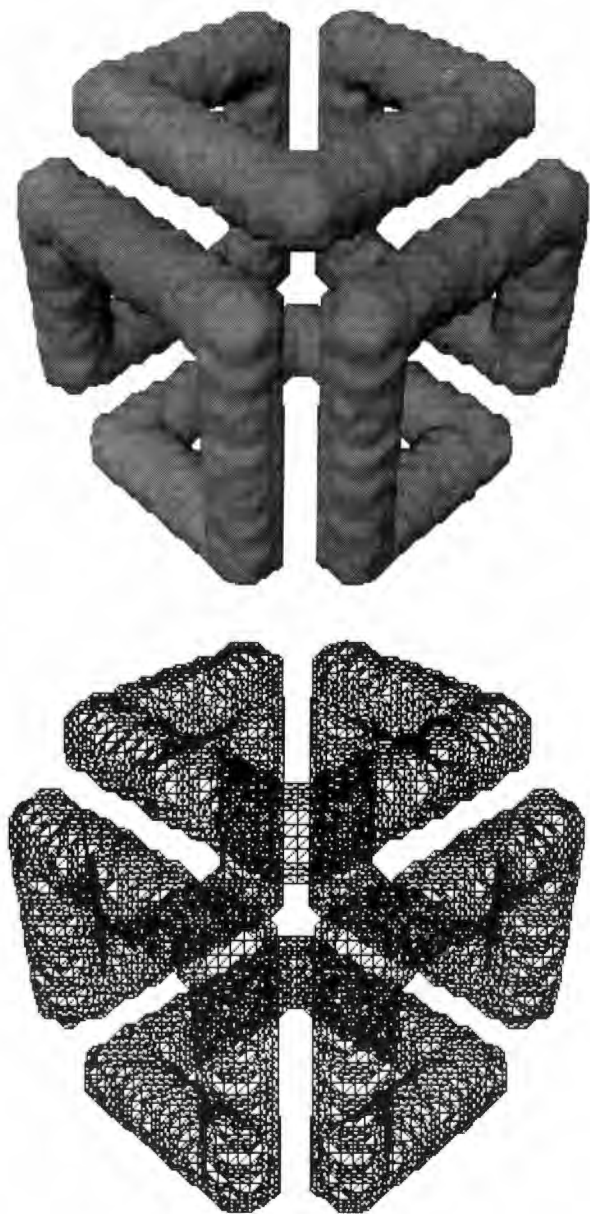
Figure 11: The surface mesh (top) and its wireframe (bottom) created by the marching cubes algorithm.

4 The Next Best View Decision Maker

The following sections describe the NBV Decision Maker. The NBV Decision Maker examines the volumetric model created by the Model Builder (see Section 3.4) and uses an objective function to determine the best place to position a new viewpoint. The objective function used to make NBV decisions is given in Section 4.1. Section 4.2 summarizes the contents of this chapter.

4.1 The Objective Function

The objective function uses the unknown information in the current volumetric model to calculate the next sensor pose. The next best view is defined as the view from which the largest number of unknown voxels can be seen. Before disclosing the equation of the objective function, we must first define some terms.

The pose of the next best viewpoint, $\vec{P}$, is the position and orientation vector of the best view from which to acquire new data. All new viewpoints point toward the center of the viewsphere (see Section 3.1.2). The current occupancy grid, OG, contains the updated volumetric model. The next best view is chosen from the set of potential viewpoints, $\vec{P_i}$. The potential viewpoints are defined in two ways to give two different objective functions.

The *visibility state*, V, of a voxel, n, is the visibility of a voxel from the current viewpoint. To calculate the visibility state of a voxel, it must be determined whether or not the voxel is visible from the current potential viewpoint. If a voxel is occluded on all sides by other known or unknown voxels (see Figure 12), it is not considered further. The voxels that have unoccluded faces are tested for their visibility from every potential viewpoint. A candidate voxel is projected towards the current
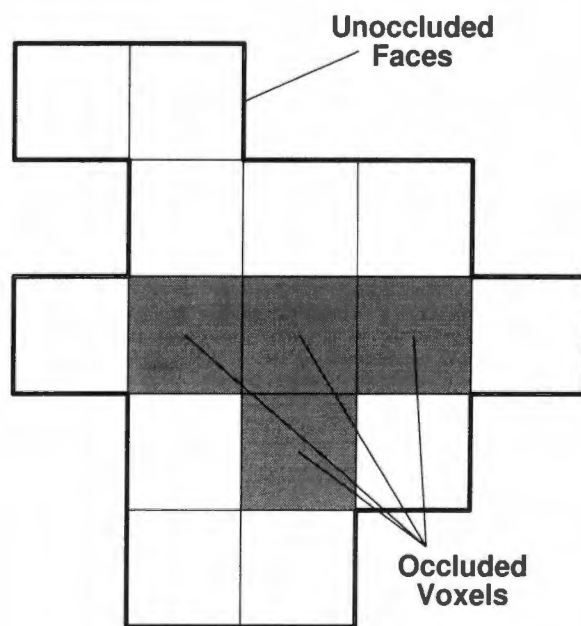
34

Figure 12: An example of occluded voxels. The occluded voxels are gray. The unoccluded voxel faces are outlined in black.

potential viewpoint. If any other voxels occlude the projection path, the voxel is not visible from the potential view, and the value of V is 0. If the projection path is free of obstructions, the voxel is visible from the current viewpoint, and the value of V is 1. The voxel projection direction, $\vec{Proj}_n$, is calculated by subtracting the voxel vector, $\vec{n}$, from the current potential view direction, $\vec{P}_i$:

$$Pr\vec{oj}_n = \vec{P}_i - \vec{n} \tag{1}$$

See Figure 13 for a graphical representation of Equation 1.

The objective function is an optimization of the visibility state for all unknown voxels in the current model. The optimization is performed for all potential viewpoints, and the viewpoint that maximizes the objective function is used to acquire new range data. In other words, the next best view is defined as the view from which the largest number of unknown voxels can be seen:

$$\vec{P} = max_i \sum^{OG(n)|\vec{P}_i} V(n, unknown) \tag{2}$$

We have tested three different objective functions: the *Optimal Method*, the *Surface Normal Method*, and the *Adaptive Method*. The results of the Optimal Method are used to create a benchmarking standard. The Surface Normal Method is fast but often cannot complete the model. The Adaptive Method, a combination of the Optimal and Surface Normal Methods, is not optimal but is fast and performs well.

4.1.1 The Optimal Method

One way to calculate the potential viewpoints is simply to place a number of viewpoints uniformly around the viewsphere. This method chooses each new view opti-

Candidate Voxel

n

$\vec{n}$

$\vec{Proj}_n$

Occupancy Grid
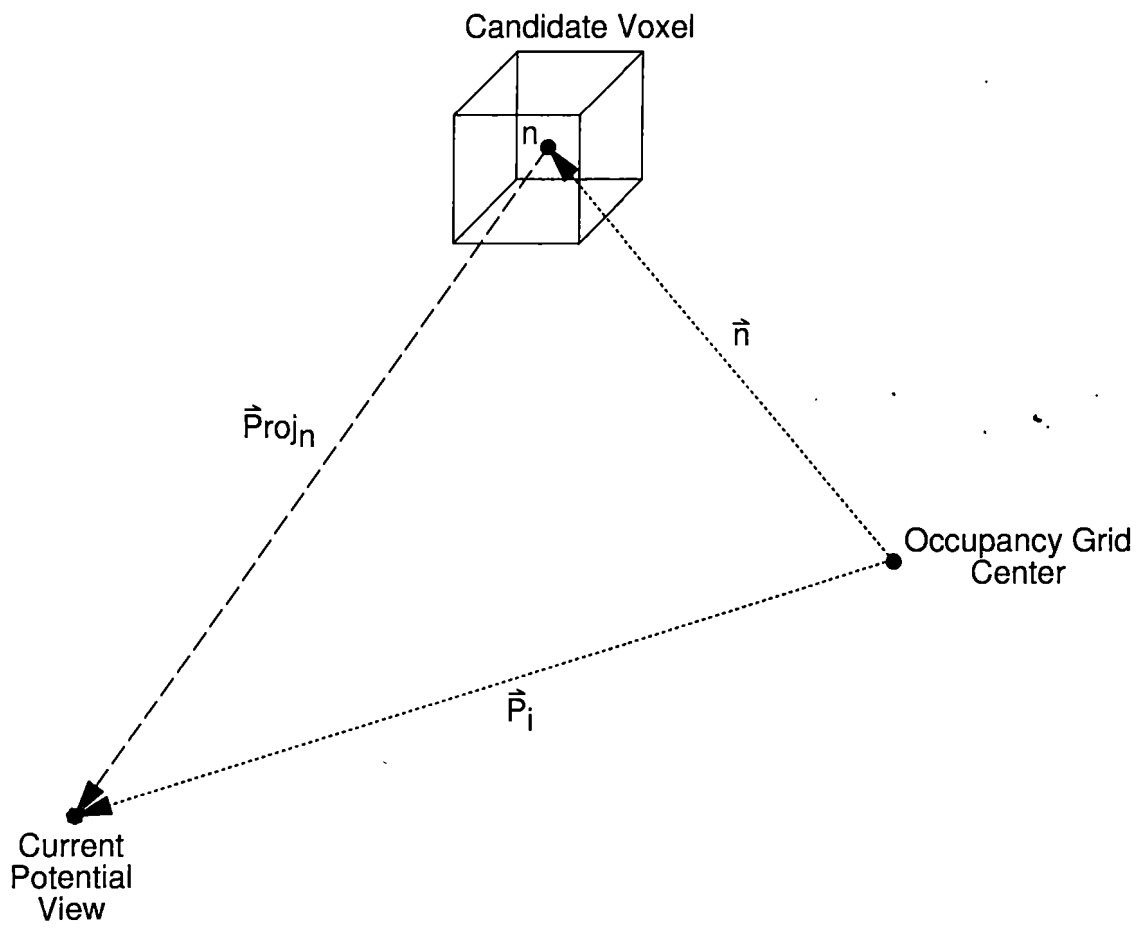Center

$\vec{P}_i$

Current
Potential
View

Figure 13: Calculation of the projection vector to gauge the visibility of a voxel.

mally if the viewpoints are dense enough; there must be enough potential views to sufficiently cover the surface of the viewsphere. The only drawback to this method is the lengthy calculation time required to process so many potential viewpoints (see Section 5.1). Since we use this method to create a benchmarking standard to assess the performance of other NBV finding algorithms, the run time for this method is not a serious concern. The same potential views are considered for each next best view calculation:

$$\vec{P_i} = \{\vec{P_0}(\theta_0, \phi_0), \vec{P_1}(\theta_1, \phi_1), ..., \vec{P_i}(\theta_i, \phi_i)\}, \tag{3}$$

where

$$\theta_i = \frac{360 * i}{div * 2}, \tag{4}$$

$$\phi_i = \frac{180 * i}{div}, \text{ and} \tag{5}$$

$$i = (2 * div) * (div + 1) - 2 * (2 * div - 1). \tag{6}$$

The parameter *div* defines the number of potential viewpoints (see Figure 14).

4.1.2 The Surface Normal Method

Due to the lengthy time required to consider hundreds of potential viewpoints using the Optimal Method, another method of choosing potential views was devised. This method involves summing the surface normals of the visible unknown voxel faces that point in each axis direction. The six calculated surface normal values are used to define eight viewpoints that give an estimation of poses from which to acquire
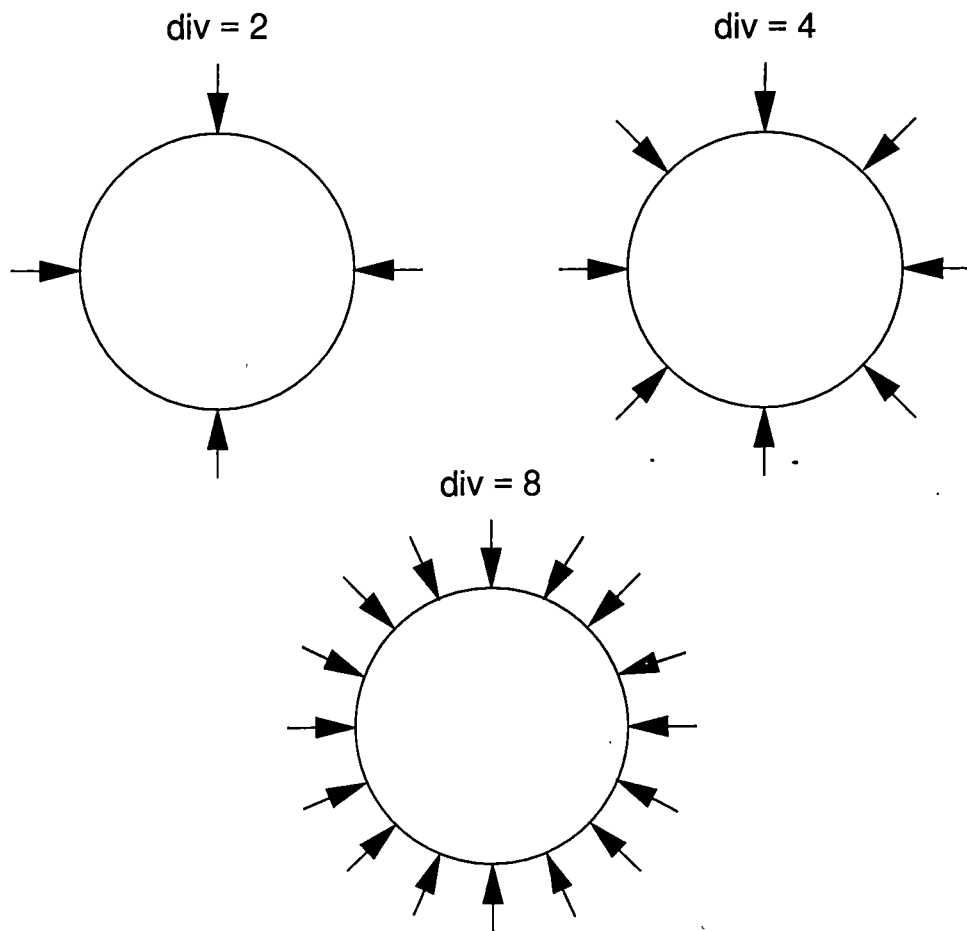
38

Figure 14: The parameter *div* defines the number of potential viewpoints. A cross-section of the viewsphere is shown for different values of *div*. The arrows represent potential viewpoints.

unknown information. New potential views are calculated for each next best view decision:

$$\vec{P}_i = (\pm\vec{x} \cdot n_x) + (\pm\vec{y} \cdot n_y) + (\pm\vec{z} \cdot n_z), \qquad (7)$$

where n_x, n_y, and n_z are the number of visible unknown voxel faces pointing in each axis direction. This method was developed because it is optimal for a single voxel. Figure 15 shows the surface normals for a voxel. Using equation 7 for a single voxel occupancy grid, the eight potential potential viewpoints are: (1,1,1), (1,1,-1), (1,-1,1), (1,-1,-1), (-1,1,1), (-1,1,-1), (-1,-1,1), (-1,-1,-1). In other words, the Surface Normal Method chooses the eight corners of a single voxel occupancy grid as the potential views, and a viewpoint centered on the corner of a voxel captures the most data. Figure 16 shows a more complex example of the Surface Normal Method potential view calculation.

There is one drawback to this method. For objects that contain self-occluded areas, the Surface Normal NBV calculation will often return viewpoints from which the unknown patches being targeted are occluded by the object itself. Objects such as the frame and mug, which contain many self-occluded areas, cannot be completely modeled (see Figure 17). Therefore, we devised the Adaptive Method.

4.1.3 The Adaptive Method

The Adaptive Method is a combination of the Optimal and Surface Normal Methods. This method defaults to the Surface Normal Method unless it returns a viewpoint that is occluded from the target area. When this occurs, the Optimal Method is used to choose another view. This method is fast and can model objects of any complexity within the constraints of system setup.

40
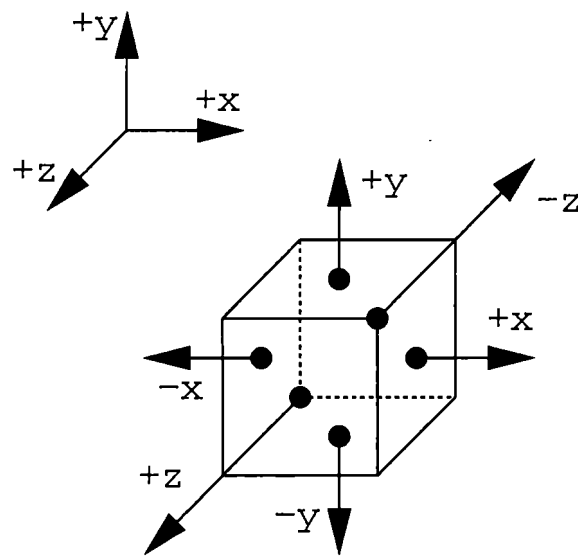
Figure 15: The surface normals for a single voxel. A surface normal is represented by an arrow and its corresponding axis direction.
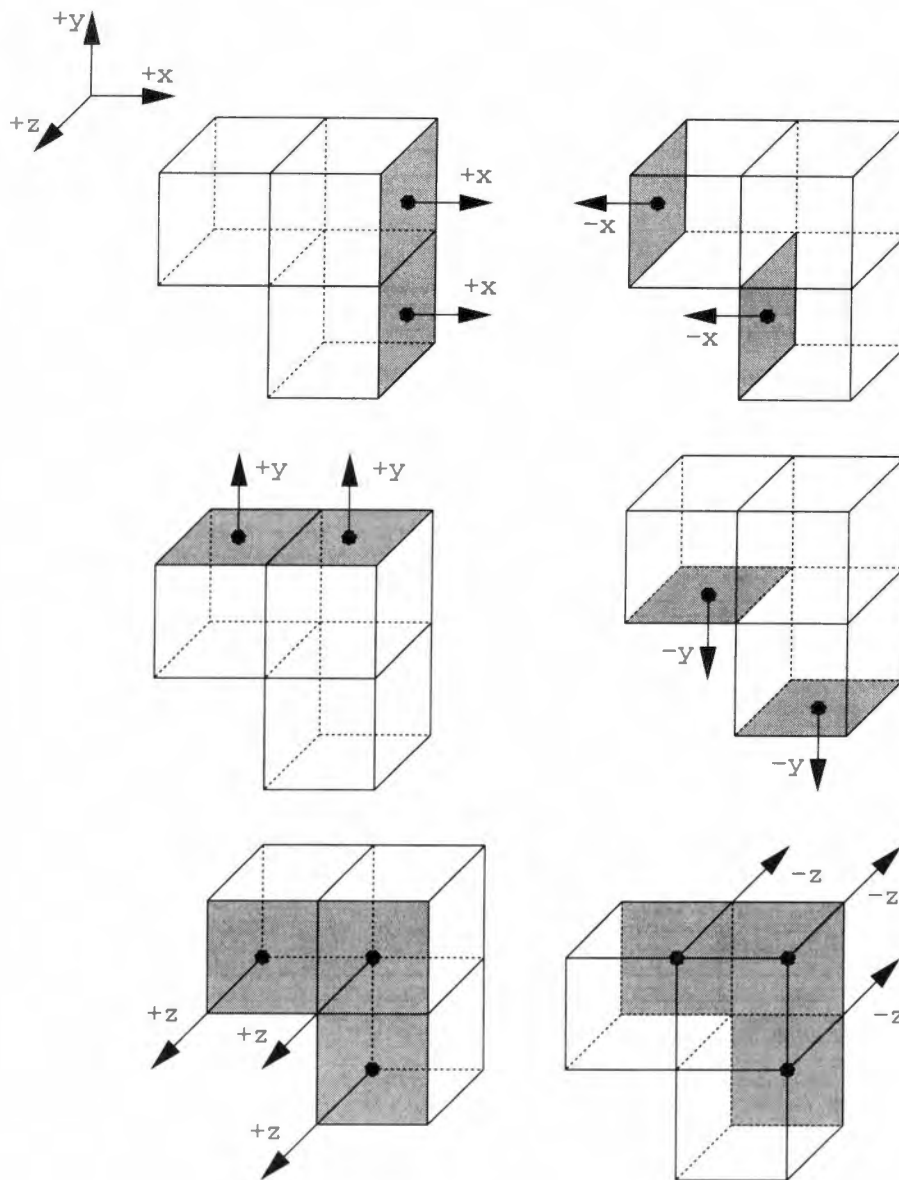
Figure 16: An example of the Surface Normal Method potential view calculation. The surface normals for each axis direction are shown separately. A surface normal is represented by an arrow and its corresponding axis direction. Using Equation 7 for the three-voxel occupancy grid above, the surface normals pointing in each axis direction give the viewpoints (2,2,3), (2,-2,3), (2,-2,3), (2,-2,-3), (-2,2,3), (-2,-2,3), (-2,-2,3), and (-2,-2,-3).
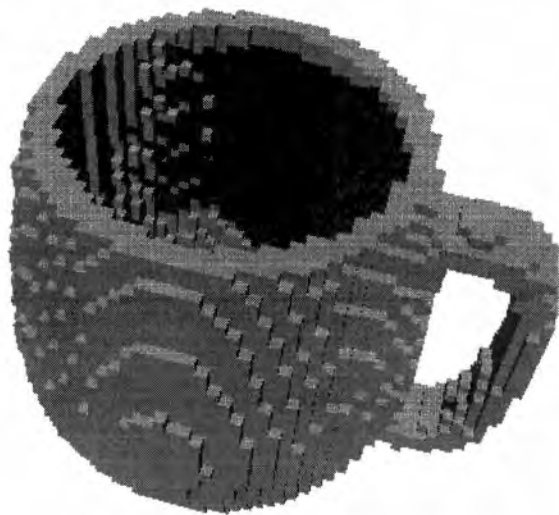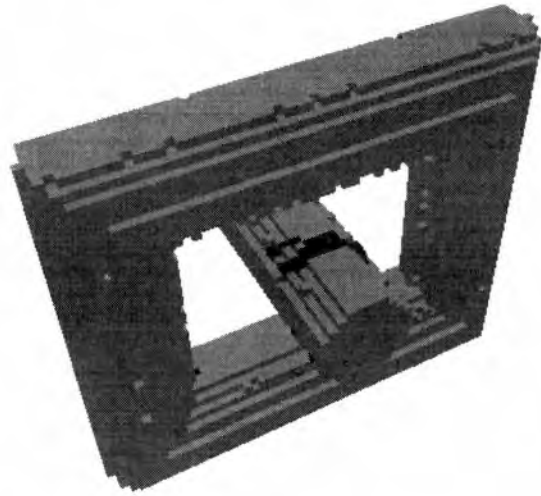
Figure 17: The Surface Normal Method cannot completely model all objects. The frame and mug models cannot be completed by the Surface Normal Method because viewpoints are selected from which the unknown patches being targeted are occluded by the object itself.

4.2 Summary of the Next Best View Decision Maker

The unknown information in the current volumetric model is used to calculate the next best view. The next best view is defined as the view from which the largest number of unknown voxels can be seen. Three methods were developed to calculate new potential views: the Optimal Method, the Surface Normal Method, and the Adaptive Method. The Optimal Method selects potential views by uniformly placing viewpoints around the viewsphere and is used as a benchmark to assess the performance of other NBV finding algorithms. The Surface Normal Method uses the surface normals of the current volumetric model to choose new potential views. The Surface Normal Method is fast but does not always complete the model. The Adaptive Method is a combination of the Optimal and Surface Normal Methods.

5 Experimental Results

The Inventor objects used for testing are shown in Figure 18. The cube and sphere are important objects to test because of their simplicity. The martini and the mug contain simple concave *self-occlusions*. Self-occlusions are areas of an object that are occluded or blocked from particular viewpoints by other areas of the same object. For example, the inside of the mug is self-occluded from viewpoints that do not point directly into the mouth of the mug. The frame and torus both contain simple hole self-occlusions, and the lattice and the logo contain complex hole self-occlusions.

The occupancy grid resolution used for testing all objects is 64 x 64 x 64 which requires a range image resolution of 256 x 256. The viewsphere radius is three times the size of the occupancy grid. The NBV program was run on a 195 MHz Silicon Graphics Octane workstation with a MIPS R10000 processor and 128 MB of RAM.

5.1 Optimal Method Results

The Optimal Method places a number of new potential viewpoints uniformly around the viewsphere. This method chooses each new view optimally if there are enough potential views to sufficiently cover the surface of the viewsphere. The Optimal Method can be lengthy to calculate, but this is of little concern since we use the method to create a benchmarking standard.

Figure 19 demonstrates how the Optimal Method progresses. The volumetric models after the acquisition of each view are shown. Each model is oriented in the direction of the view to be acquired next. Notice that the unknown areas (shown in black) decrease in size throughout the modeling process. Since no unknown voxels
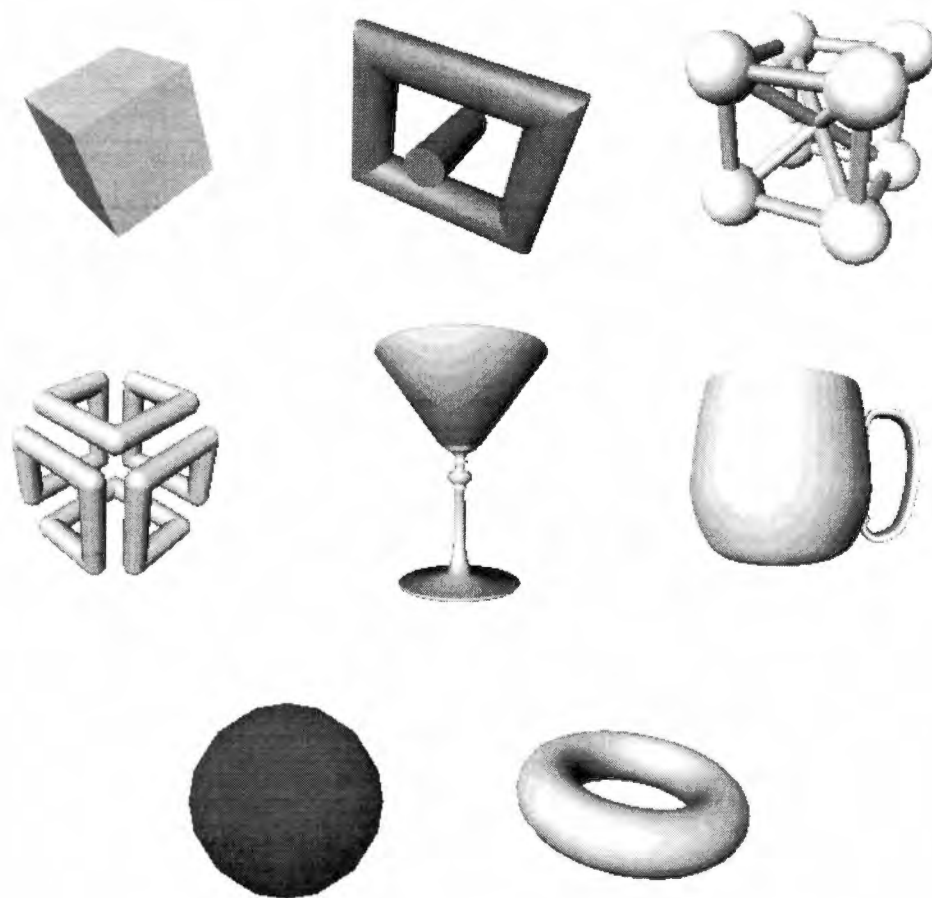
Figure 18: The Inventor test objects from left to right and top to bottom: cube, frame, lattice, logo, martini, mug, sphere, and torus.
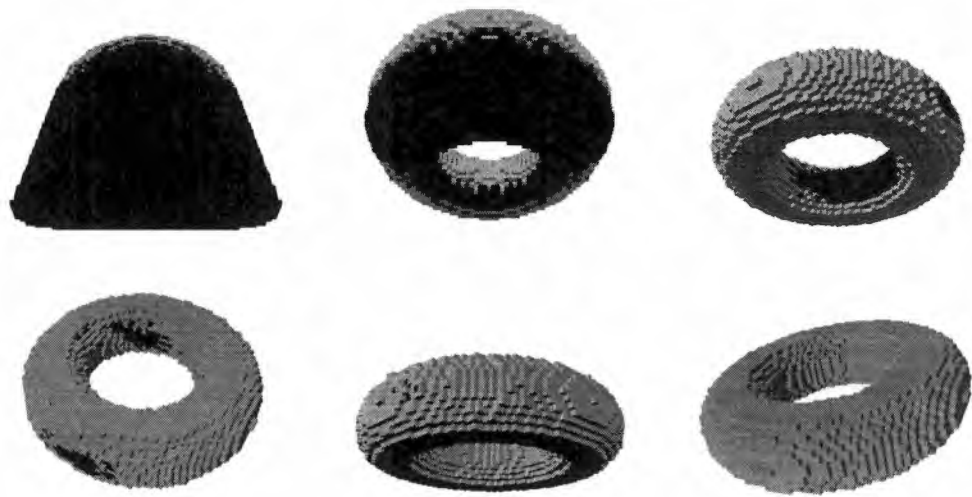
Figure 19: The Optimal Method volumetric models after the acquisition of each view (from left to right and top to bottom). Each model is oriented in the direction of the view to be acquired next.

remain after the sixth view, the modeling process ends. The absence of unknown voxels also means that the entire object surface has been scanned.

Before reviewing some of the more complicated objects, it is important to look at the Optimal results for simple objects. Modeling an object such as the cube gives an indication of how the NBV system performs. The cube was modeled in three views, and this indicates that the NBV system is optimal for the cube; a cube can only be modeled in less than three views only if the orientation of the cube is known a priori. Figure 20 indicates the sensor poses used to model the cube. Figure 21 shows the volumetric models constructed by the NBV system with the Optimal Method.

To visually compare the volumetric models to the corresponding archetypes, we use a marching cubes technique (see Section 3.4) to create surface meshes of each occupancy grid model. Figure 22 shows a comparison between the surface meshes and the archetypes. A visual comparison is adequate since we are concerned only with the acquisition of a complete model, not the accuracy of the model. The models match well with the archetypes but are slightly larger. This occurs because each range data point acquired from the archetype corresponds to a voxel in the model. Therefore, the models are 'fatter' than the archetypes. The surfaces of the models are not smooth because of the aliasing that occurs in binary representations and because of the limitations of the marching cubes algorithm used. The only problem with the models are in areas of great detail. For example, the contours of the martini stem are not truly represented in the model. This discrepancy is due to the low occupancy grid resolution that we chose to use for these results. A higher resolution could be used to recover more detail in the models but this would drastically increase the system run time.

Table 1 shows the Optimal Method experimental data. The information in this

Figure 20: The three views chosen to model the cube using the Optimal Method. The arrows represent the viewpoints from which range data was acquired.

Figure 21: The Optimal Method volumetric models (from left to right and top to bottom) of the cube, frame, lattice, logo, martini, mug, sphere, and torus.
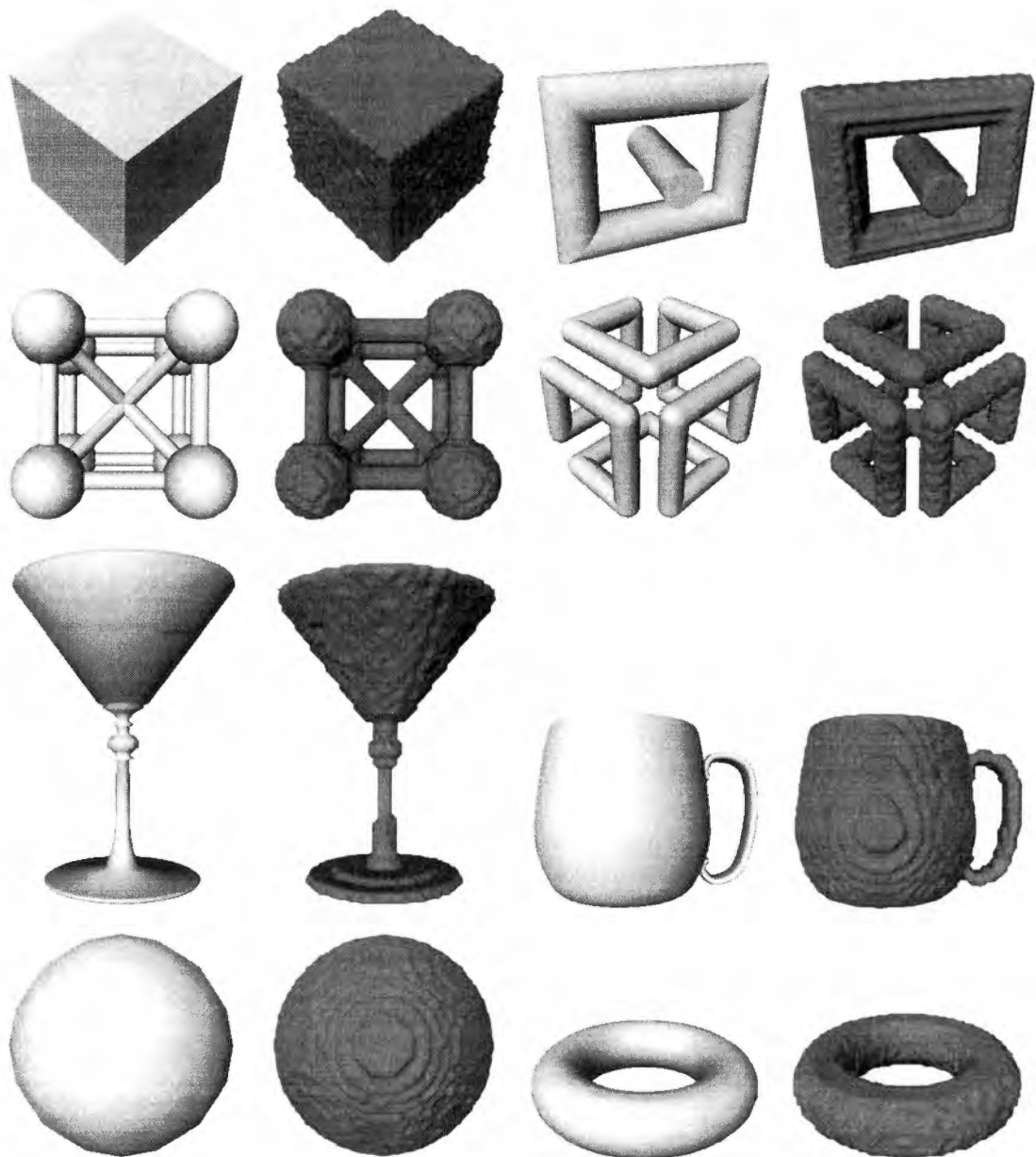
Figure 22: A visual comparison of the Optimal surface models and their corresponding archetype objects.

Table 1: Experimental Results for the Optimal Method.

	-CP	CP	
Object	# Views	# Views	Max Time (s)
cube	3	3	107
frame	5	7	175
lattice	11	11	192
logo	13	13	149
martini	6	6	120
mug	6	10	118
sphere	5	5	118
torus	5	6	97

table is used for the benchmarking standard. 114 potential viewpoints were used for these results. Results under the heading *CP* exhibit the correspondence problem. Results under the heading -*CP* do not exhibit the correspondence problem. These results are shown for comparison purposes only (see Section 3.4) and will not be discussed further. *# Views* refers to the number of viewpoints required to scan the complete object surface. *Max Time* is the time it takes for the system to decide on the NBV after the initial viewpoint is acquired. The Optimal objective function calculations increase in speed as modeling progresses because the number of unknown voxels decreases after each view (see Figure 23 for an example).

5.2   Adaptive Method Results

Due to the the lengthy time required to consider hundreds of potential viewpoints using the Optimal Method, another method of choosing potential views was devised. The Adaptive Method finds potential viewpoints by summing the surface normals of the visible unknown voxel faces that point in each axis direction. This technique is fast and optimal for voxels. When the next best view is occluded from the target area, the Optimal Method is called to select a new view.

Figure 24 demonstrates how the Adaptive Method progresses. The Adaptive Method is obviously not as effective in choosing views as the Optimal Method, but it is not meant to be. Figure 25 demonstrates how the Adaptive Method chooses different sensor poses than the Optimal Method to model the cube. Figure 26 shows the volumetric models constructed by the NBV system with the Adaptive Method. Figure 27 compares the surface meshes with the original objects. Table 2 shows the Adaptive Method experimental results. The Adaptive objective function calculations increase in speed as modeling progresses but there will be spikes where the Optimal Method is employed to complete the model (see Figure 28 for an
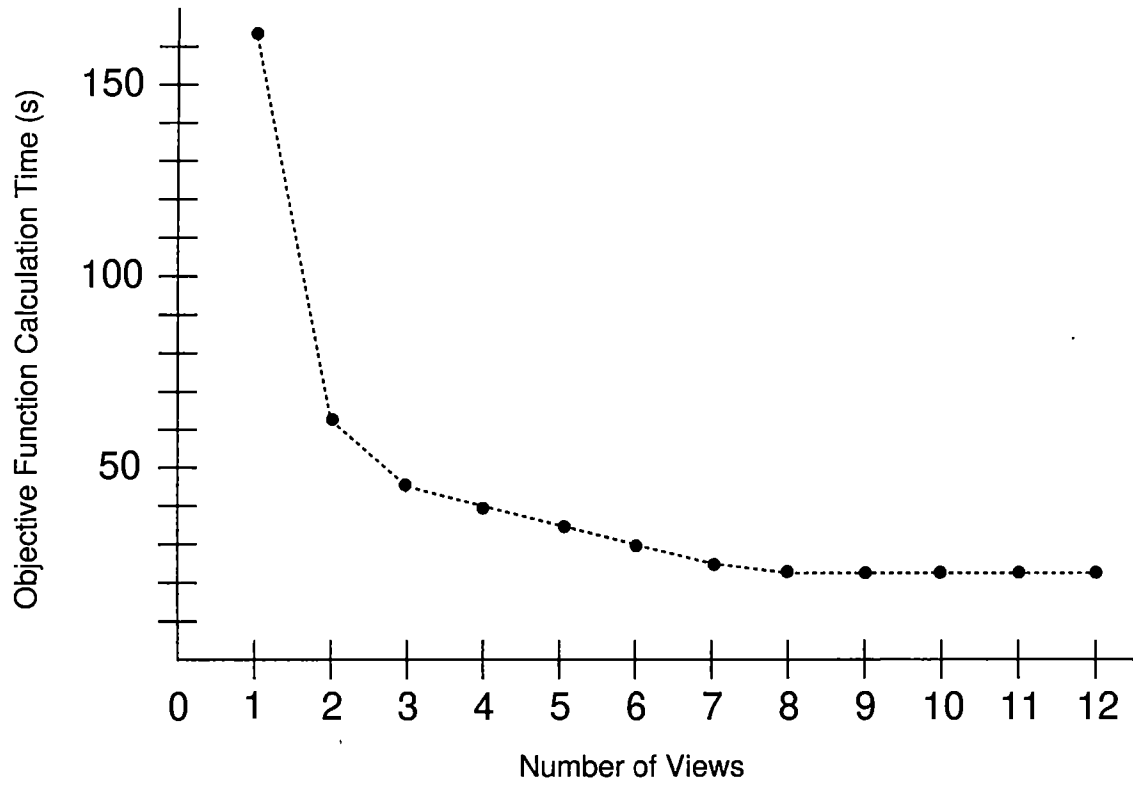
Figure 23: The speed of the Optimal objective function for each view of the logo.
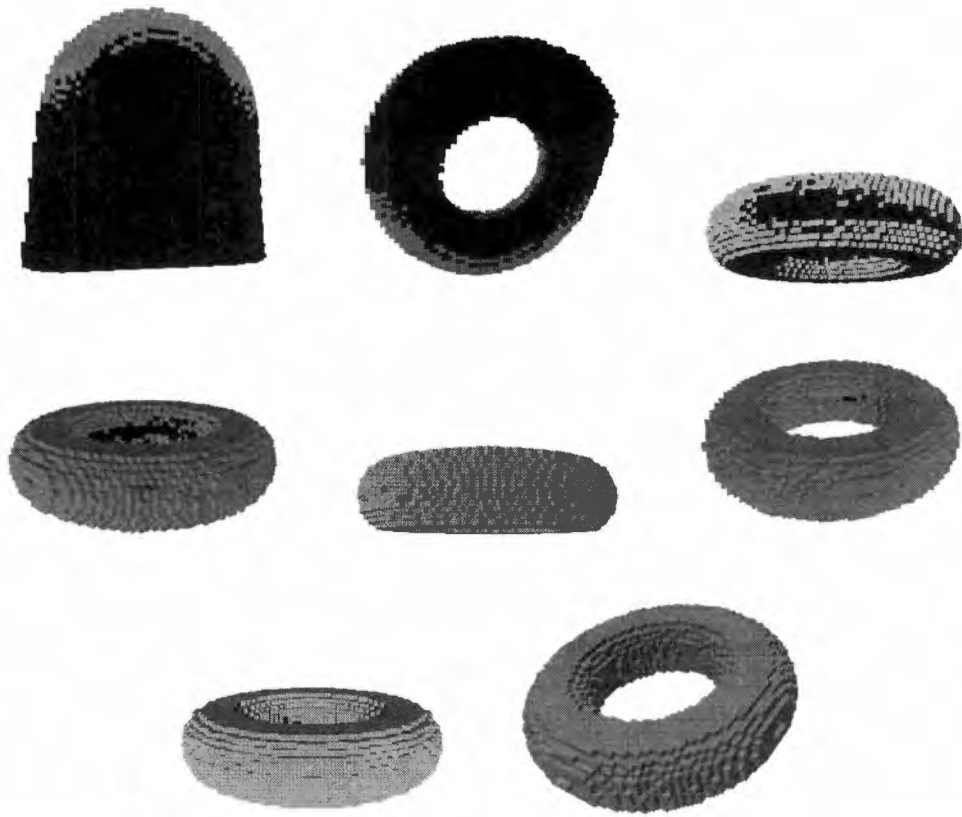
Figure 24: The Adaptive Method volumetric models after the acquisition of each view (from left to right and top to bottom). Each model is oriented in the direction of the view to be acquired next.
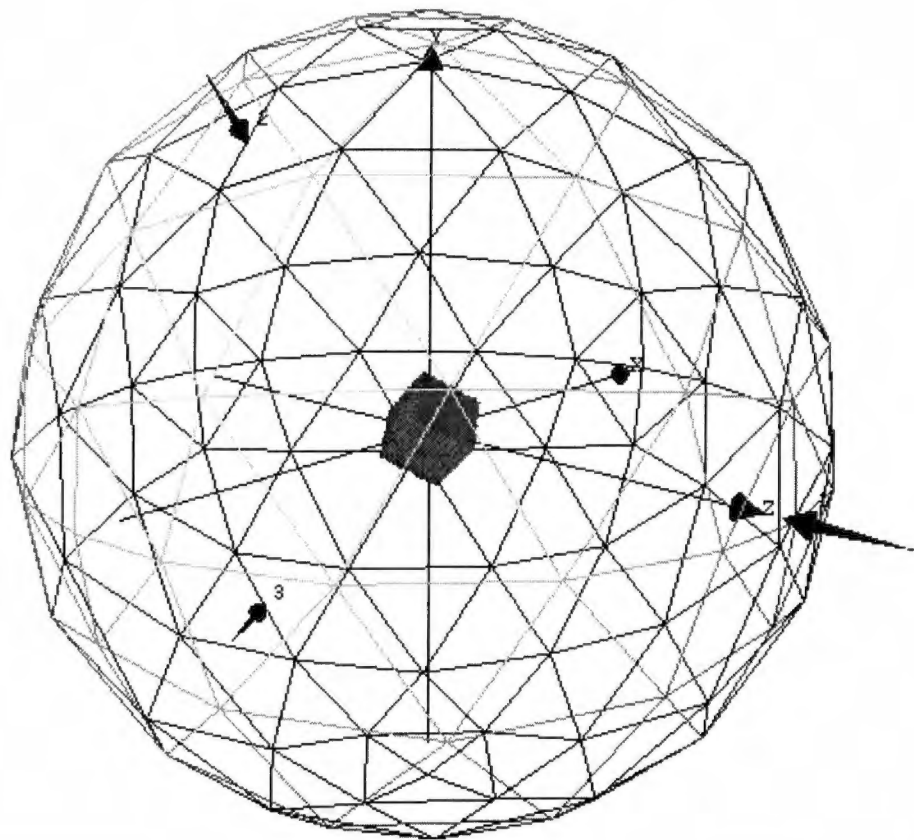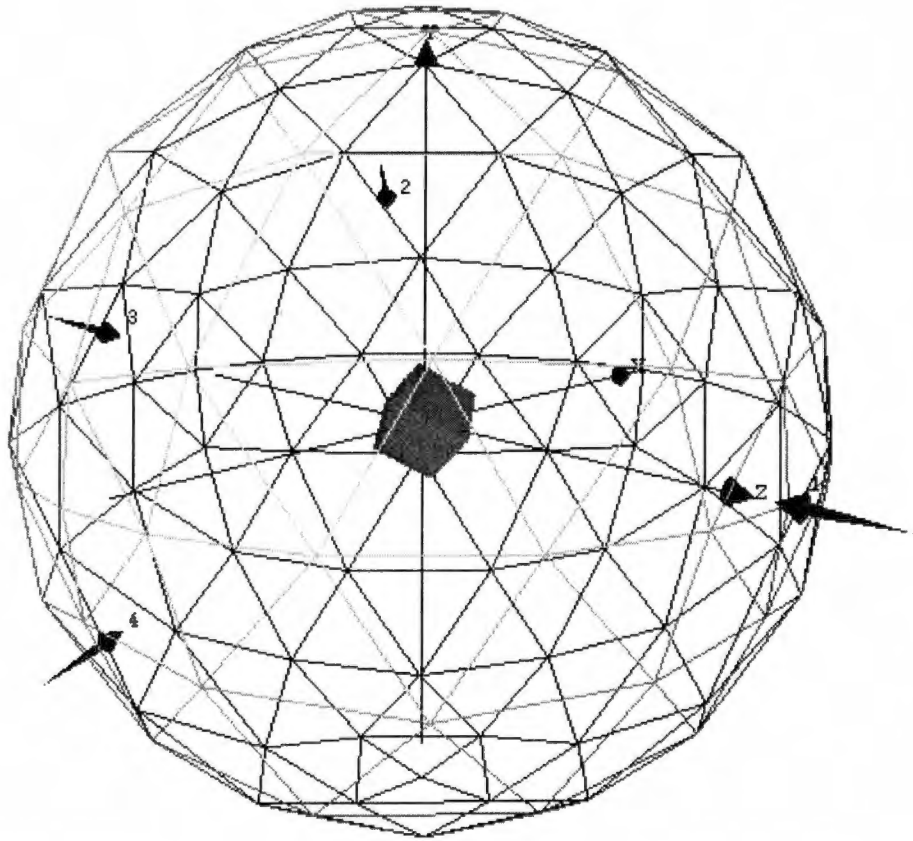
Figure 25: The four views chosen to model the cube using the Surface Normal Method. The arrows represent the viewpoints from which range data was acquired.

Figure 26: The Adaptive volumetric models (from left to right and top to bottom) of the cube. frame, lattice. logo. martini, mug, sphere, and torus.
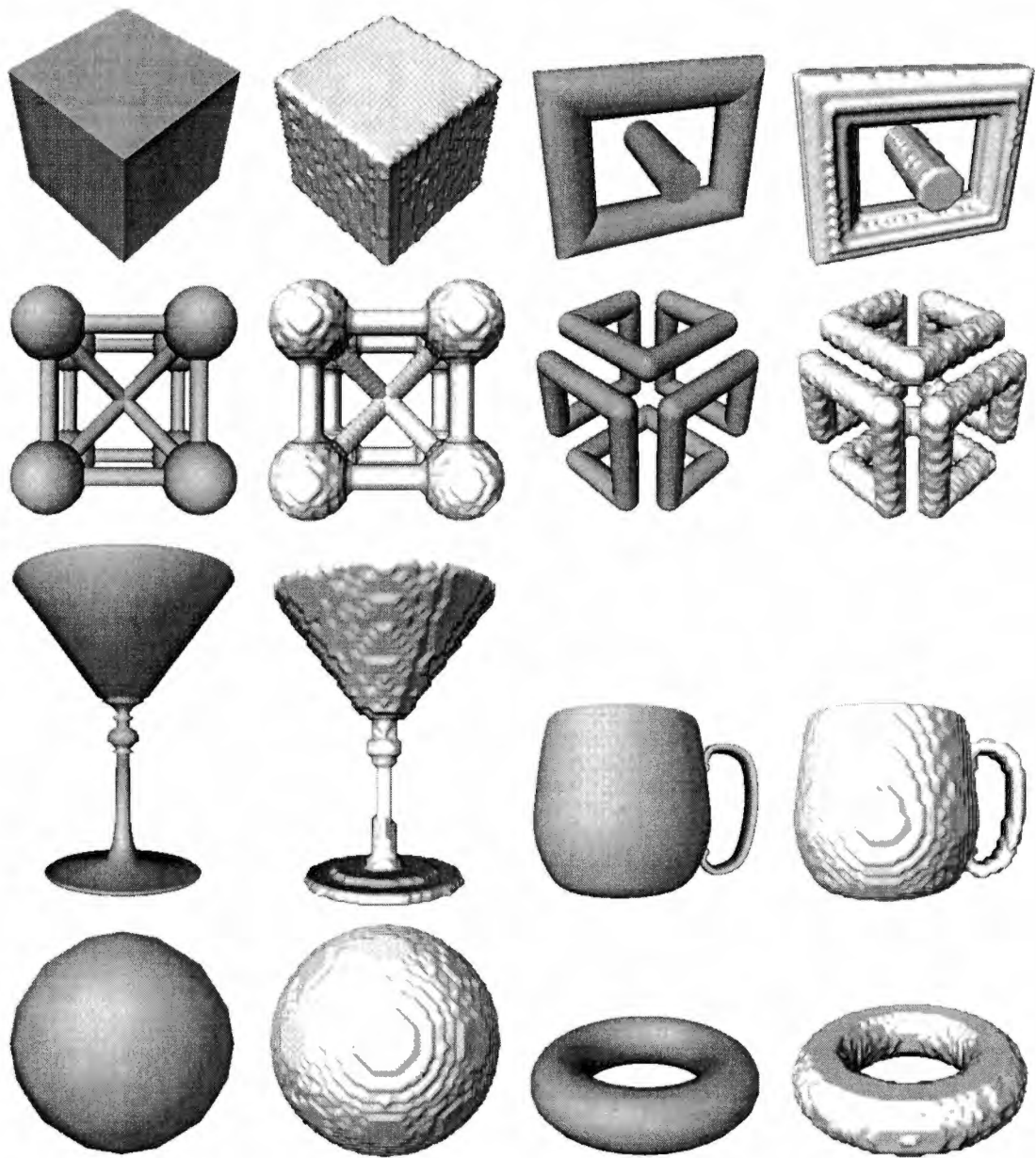
Figure 27: A visual comparison of the Adaptive surface meshes and their corresponding archetype objects.

Table 2: Experimental Results for the Adaptive Method.

	-CP	CP	
Object	# Views	# Views	Max Time (s)
cube	4	4	9
frame	11	12*^{CP}	17
lattice	11	19	14
logo	11	15	9
martini	6	7	10
mug	8	12	11
sphere	5	5	11
torus	6	8	8

*^{CP} The shape of the model is complete but a few visible unknown voxel faces remain due to the correspondence problem (see Section 3.4).
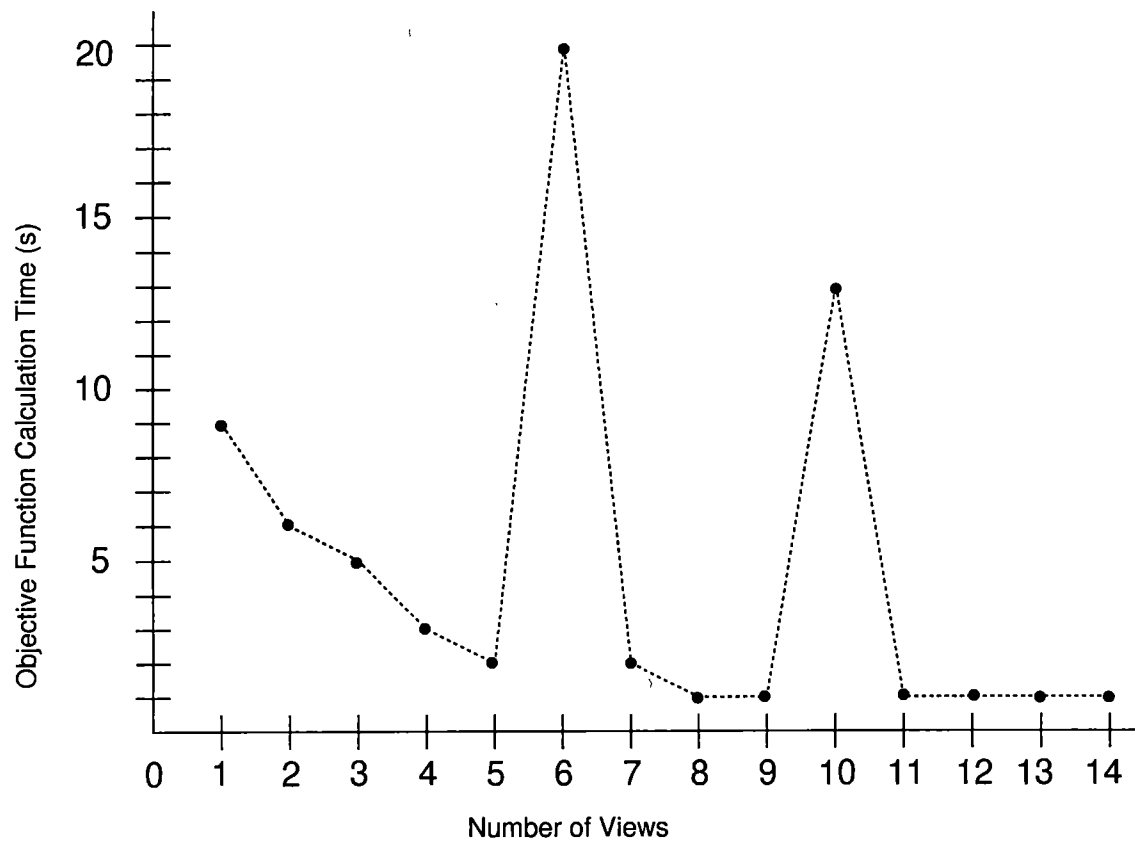
Figure 28: The speed of the Adaptive objective function for each view of the logo.

example). Since the Optimal Method is never used in the early stages of modeling, the objective function calculation is not slowed down drastically.

Table 3 shows a comparison of results for the Optimal and Adaptive Methods. The Adaptive Method is much faster than the Optimal Method but normally requires more views to model an object. To rate the performance of the Adaptive Method, we can look at the state of the Adaptive models when the Optimal Method models are complete. Table 4 shows the percentage of the Surface Normal model volume that is unknown when the Optimal model is complete. A percentage of zero means that the model is complete. High percentages (e.g. the percentages for the mug or the lattice) mean that the number of visible unknown voxels is high compared to the total object volume. Table 5 shows the number of views beyond the Optimal Method needed to complete the Adaptive models. Figure 29 compares the surface meshes for the Optimal and Adaptive Methods.

5.3 Conclusions of Experimental Results

The Optimal Method results given in this chapter and especially in Table 1 will be used as a benchmarking standard. The Optimal Method chooses new views optimally and can handle objects of any complexity within the constraints of system setup. With this method, The NBV system is optimal for a cube but not necessarily optimal in general. The final surface meshes produced by the Optimal Method compare well visually to the archetypes. The Adaptive Method is fast and can handle objects of any complexity within the constraints of system setup. The final surface meshes also compare well visually with the archetype objects. Visual comparisons are adequate since we are concerned only with the acquisition of a complete model, not the accuracy of the model.

Table 3: Experimental Results for the Optimal and Adaptive Methods.

Object	Optimal		Adaptive	
	# Views	Max Time (s)	# Views	Max Time (s)
cube	3	107	4	9
frame	7	175	12*^{CP}	17
lattice	11	192	19	14
logo	13	149	15	9
martini	6	120	7	10
mug	10	118	12	11
sphere	5	118	5	11
torus	6	97	8	8

*^{CP} The shape of the model is complete but a few visible unknown voxel faces remain due to the correspondence problem (see Section 3.4).

Table 4: The percentage of the Adaptive model volume that is unknown when the Optimal model is complete.

Object	% Unknown
cube	0.0069
frame	0.37
lattice	0.35
logo	0.026
martini	0.095
mug	0.021
sphere	0
torus	0.066

Table 5: The number of views beyond the Optimal Method needed to complete the Adaptive model.

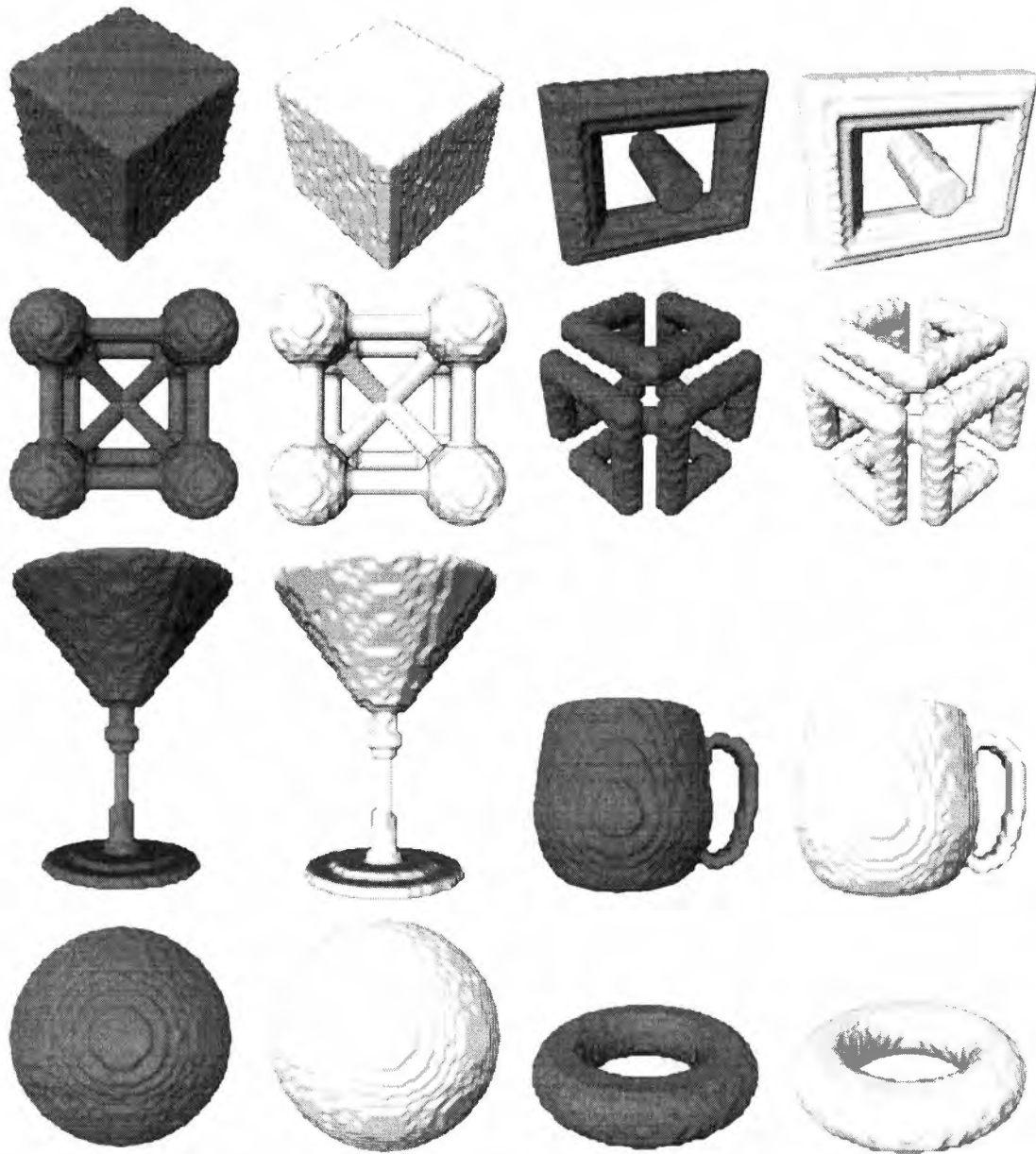| Object	More Views
cube	1
frame	5
lattice	8
logo	2
martini	1
mug	2
sphere	0
torus	2

Figure 29: A visual comparison of surface meshes for the Optimal and Adaptive Methods.

6 Conclusions and Future Work

The goal of a Next Best View modeling system is to completely model an object using the smallest number of views. A NBV algorithm determines new sensor positions that reveal optimal amounts of unknown information about the object of interest. Our NBV algorithms calculate potential viewpoints and choose the one from which the largest number of visible unknown voxels can be seen. The first method places a number of potential views uniformly around the viewsphere. The second method is a summation of surface normals of the visible unknown voxel faces that point in each axis direction.

The NBV system consists of three parts: the Range Scanner, the Model Builder, and the NBV Decision Maker. The Range Scanner simulates the acquisition of range data from a computer object. The acquired range images are used by the Model Builder to create and update the volumetric model. The NBV Decision Maker examines the unknown information in the volumetric model and uses an objective function to determine the best place to position a new viewpoint.

The Optimal Method chooses new views optimally. With this method, the NBV system is optimal for a cube but not necessarily optimal in general. This method, however, requires too much time to calculate the objective function in the early stages of modeling. The Adaptive Method is not optimal but drastically reduces the NBV system run time. The models tested for both methods are complete except when effected by the correspondence problem.

Because of the difficulty in rating the performance of a NBV algorithm, the Optimal results presented in this paper will be used as a benchmark to judge the performance of future objective functions. The next step in the continuation of this work is to add parameters to the objective function so that the system is applicable

66

in the real world; we will consider the overlap of sequential views for registration purposes and the restriction of view placement. We will also use non-synthetic range data of objects for further testing.

We would eventually like to apply this work to the exploration of hazardous environments. In this setting, an autonomous mobile robot must explore a space that has been contaminated with hazardous waste. The robot maneuvers inside the contaminated area by acquiring data from its surroundings. Since there may be a limited amount of time to explore, the NBV would be ideal in choosing the positions from which to acquire data. The NBV system, however, would need to be adjusted to handle scenes instead of objects. Since a viewsphere would no longer be useful in such a situation, all acquired range images would have to be registered. The set of potential viewpoints must also be limited to account for the walls and the floor of the contaminated space.

References

References

[1] Konstantinos A. Tarabanis, Peter K. Allen, and Roger Y. Tsai. A survey of sensor planning in computer vision. *IEEE Transactions on Robotics and Automation*, 11(1):86–104, 1995.

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

[3] Éric Marchand and François Chaumette. Controlled camera motions for scene reconstruction and exploration. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 169–176, 1996.

[4] M. J. Milroy, C. Bradley, and G. W. Vickers. Automated laser scanning based on orthogonal cross sections. *Machine Vision and Applications*, 9(3):106–118, 1996.

[5] Vitor Sequeira, Joao G. M. Gonvalves, and M. Isabel Ribeiro. Active view selection for efficient 3d scene reconstruction. *Proceedings of the International Conference on Pattern Recognition*, 1:815–819, 1996.

[6] G. H. Tarbox and S. N. Gottschlich. Planning for complete sensor coverage in inspection. *Computer Vision and Image Understanding*, 61(1):84–111, 1995.

[7] Hongbin Zha, Ken'ichi Morooka, Tsutomu Hasegawa, and Tadashi Nagata. Active modeling of 3-d objects: planning on the next best pose (nbp) for acquiring range images. *Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pages 68–75, May 1997.

[8] Besma A. Abidi. Automatic sensor placement. *Proceedings of the SPIE*, 2588:387–398, 1995.

[9] J. E. Banta, Y. Zhien, X. Z. Wang, G. Zhang, M. T. Smith, and M. A. Abidi. A best-next-view algorithm for three-dimensional scene reconstruction using range images. *Proceedings of the SPIE*, 2588:418–429, 1995.

[10] Kiriakos Neoklis Kutulakos. *Exploring three-dimensional objects by controlling the point of observation.* PhD thesis, University of Wisconsin-Madison, 1994. Available from ftp://ftp.cs.wisc.edu/computer-vision.

[11] Kiriakos N. Kutulakos and Charles R. Dyer. Global surface reconstruction by purposive control of observer motion. *Artificial Intelligence*, 78(1-2):147–177, 1995.

[12] Jasna Maver and Ruzena Bajcsy. Occlusions as a guide for planning the next view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):417–433, 1993.

[13] Michael K. Reed, Peter K. Allen, and Ioannis Stamos. 3d modeling from range imagery: an incremental method with a planning component. *International Conference on Recent Advances in 3-D Digital Imageing and Modeling*, pages 76–83, May 1997.

[14] Yu Zhien, Wang Ke, and Yang Rong-Guang. Next best view of range sensor. *Proceedings of the IEEE IECON*, 1:185–188, 1996.

[15] Joseph Banta. Best-next-view: minimizing the number of range images for 3d model reconstruction. Master's thesis, University of Tennessee, Knoxville, December 1996.

[16] Richard Pito and Ruzena 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*, 2596:78–89, 1995.

[17] Nageswara S. V. Rao, S. S. Iyengar, B. John Oommen, and R. L. Kashyap. Terrain acquisition by point robot amidst polyhedral objects. *Proceedings of the Conference on Artificial Intelligence Applications*, pages 170–175, 1987.

[18] K. Shanmukh and Arun K. Pujari. Volume intersection with optimal set of directions. *Pattern Recognition Letters*, 12(3):165–170, 1991.

[19] Kazunori Yoshida, Hiromi Tanaka, Jun Ohya, and Fumio Kishino. Active 3d modeling by recursive viewpoint selection based on symmetry. *Proceedings of the SPIE*, 2588:326–336, 1995.

[20] Xiaobu Yuan. A mechanism of automatic 3d object modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3):307–311, 1995.

[21] Besma Roui-Abidi. *Automatic sensor placement for volumetric object characterization*. PhD thesis, The University of Tennessee, Knoxville, August 1995.

[22] Alec Cameron and Hugh Durrant-Whyte. A bayesian approach to optimal sensor placement. *International Journal of Robotics Research*, 9(5):70–80, 1990.

[23] R. E. Ellis. Planning tactile recognition paths in two and three dimensions. *International Journal of Robotics Research*, 11(2):87–111, 1992.

[24] Keith D. Gremban and Katsushi Ikeuchi. Planning multiple observations for object recognition. *International Journal of Computer Vision*, 12(2/3):137–172, 1994.

[25] W. Eric L. Grimson. Disambiguating sensory interpretations using minimal sets of sensor data. *Proceedings of the IEEE International Conference on Robotics and Automation*, 1:286–292, 1986.

[26] Seth A. Hutchinson and Avinash C. Kak. Planning sensing strategies in a robot work cell with multi-sensor capabilities. *IEEE Transactions on Robotics and Automation*, 5(6):765–783, 1989.

[27] Hwang-Soo Kim, Ramesh C. Jain, and Richard A. Volz. Object recognition using multiple views. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 28–33, 1985.

[28] Sukhan Lee and Hernsoo Hahn. An optimal sensing strategy for recognition and localization of 3-d natural quadric objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1018–1037, 1991.

[29] Xueyin Lin, Jianchao Zeng, and Quixiang Yao. Optimal sensor planning with minimal cost for 3d object recognition using sparse structured light images. *Proceedings of the IEEE International Conference on Robotics and Automation*, 4:3484–3489, 1996.

[30] Huiqun Liu and Xueyin Lin. Model-based next view planning by using rules: automatic feature prediction and detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 773–776, 1994.

[31] Michael Magee and Mitchell Nathan. Spatial reasoning, sensor repositioning, and disambiguation in 3d model based recognition. *Proceedings of the Spatial Reasoning and Multi-sensor Fusion Workshop*, pages 262–271, 1987.

[32] David Wilkes and John Tsotos. Behaviors for active object recognition. *Proceedings of the SPIE*, 2055:225–239, 1993.

[33] Lester A. Gerhardt and Kwangik Hyun. Three-dimensional view planning for noncontact dimensional inspection. *Proceedings of the SPIE*, 2665:177–184, 1996.

[34] K. W. Khawaja, A. A. Maciejewski, D. Tretter, and C. A. Bouman. Camera and light placement for automated assembly inspection. *Proceedings of the IEEE International Conference on Robotics and Automation*, 4:3246–3252, 1996.

[35] Shigeyuki Sakane, Masaru Ishii, and Masayoshi Kakikura. Occlusion avoidance of visual sensors based on a hand-eye action simulator system: Heaven. *Advanced Robotics*, 2(2):149–165, 1987.

[36] Bill Triggs and Christian Laugier. Automatic camera placement for robot vision tasks. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2:1732–1737, 1995.

[37] Yulin Yao and Peter Allen. Computing robust viewpoints with multi-constraints using tree annealing. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2:993–998, 1995.

[38] Jiang Yu Zheng, Fumio Kishino, Qian Chen, and Saburo Tsuji. Active camera controlling for manipulation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 413–418, 1991.

[39] Steven Abrams and Peter K. Allen. Sensor planning in an active robotic work cell. *Proceedings of the SPIE*, 1611:267–277, 1991.

[40] Steven Abrams, Peter K. Allen, and Konstantinos A. Tarabanis. Computing camera viewpoints in a robot work-cell. *Proceedings of the IEEE International Conference on Robotics and Automation*, 3:1972–1979, 1996.

[41] Cregg K. Cowan. Model-based synthesis of sensor location. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2:900–905, 1988.

[42] A. D. Marshall and D. R. Roberts. Automatically planning the inspection of three-dimensional objects using stereo computer vision. *Proceedings of the SPIE*, 2599:94–105, 1996.

[43] Scott O. Mason and Armin Grün. Automatic sensor placement for accurate dimensional inspection. *Computer Vision and Image Understanding*, 61(3):454–467, 1995.

[44] Konstantinos A. Tarabanis, Roger Y. Tsai, and Peter K. Allen. Analytical characterization of the feature detectability constraints of resolution, focus, and field-of-view for vision sensor planning. *CVGIP: Image Understanding*, 59(3):340–358, 1994.

[45] Konstantinos A. Tarabanis, Roger Y. Tsai, and Peter K. Allen. The mvp sensor planning system for robotic vision tasks. *IEEE Transactions on Robotics and Automation*, 11(1):72–85, 1995.

[46] Konstantinos A. Tarabanis, Roger Y. Tsai, and Anil Kaul. Computing occlusion-free viewpoints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3):279–292, 1996.

[47] E. Zussman, H. Schuler, and G. Seliger. Analysis of the geometrical features detectability constraints for laser-scanner sensor planning. *International Journal of Advanced Manufacturing Technology*, 9(1):56–64, 1994.

[48] Andrea Califano, Rich Kjeldsen, and Ruud M. Bolle. Data and model driven foveation. *Proceedings of the International Conference on Pattern Recognition*, 1:1–7, 1990.

[49] James J. Clark and Nicola J. Ferrier. Modal control of an attentive vision system. *Proceedings of the International Conference on Computer Vision*, pages 514–523, 1988.

[50] Hugh F. Durrant-Whyte. Uncertain geometry in robotics. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 851–856, 1987.

[51] Greg Hager and Max Mintz. Searching for information. *Proceedings of the Spatial Reasoning and Multi-sensor Fusion Workshop*, pages 313–322, 1987.

[52] Yue Liu and Denis Laurendeau. Processing of a multi-scale triangulated surface model of a 3-d scene for a robotics application. *Proceedings of the SPIE*, 2031:392–403, 1993.

[53] Claus B. Madsen and Henrik I. Christensen. A viewpoint planning strategy for determining true angles on polyhedral objects by camera alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):158–163, 1997.

[54] Jasna Maver, Aleš Leonardis, and Franc Solina. Planning the next view using the max-min principle. In *Computer Analysis of Images and Patterns*, pages 543–547. Springer-Verlag, 1993.

[55] Jun Miura and Katsushi Ikeuchi. Task-oriented generation of visual sensing strategies. *Proceedings of the International Conference on Computer Vision*, pages 1106–1113, 1995.

[56] Amir Shmuel and Michael Werman. Active vision: 3d from an image sequence. *Proceedings of the International Conference on Pattern Recognition*, 1:48–54, 1990.

[57] Carl-Johan Westelius. *Focus of attention and gaze control for robot vision.* PhD thesis, Linkoping University, Sweden, 1995.

[58] W. E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3d surface construction algorithm. *Computer Graphics*, 21:163–169, 1987.

Appendix

Appendix

Here is a bit about how to run the Next Best View system software. The code requires the SGI Open Inventor Toolkits. Run time information is sent to standard out.

There are several scripts that will build the executables and run the programs with all command line options set properly:

o_opt   Optimal method

o_ada   Adaptive method

o_cp_opt   Optimal method without the correspondence problem

o_cp_ada   Adaptive method without the correspondence problem

o_all   all methods

To build the executables, type make and the desired executable name:

nbv_opt   Optimal method

nbv_ada   Surface Normal method

nbv_cp_opt   Optimal method without the correspondence problem

nbv_cp_ada   Adaptive method without the correspondence problem

To run the executables, set these command line options:

-model   [model name]

-method   [opt, ada, cp_opt, or cp_ada]

-numviews   [number of views to acquire]

-og   [resolution of one side of the cubic occupancy grid (OG)]

-wksp   [size of one side of the cubic workspace in Inventor space]

the Inventor object must fit inside the workspace

-rad   [multiple of OG size to determine view sphere radius]

should always be greater than 1

-res   [resolution of the range images]

Vita

Laurana Moy-ping Wong was born in Memphis, Tennessee in 1972. She attended The University of Washington, Seattle for two years and then transferred to The University of Tennessee, Knoxville. During her career at UT, she co-oped three semesters for Machine View, a subsidiary of CSI in Knoxville. She graduated from UT in 1996 with a Bachelor of Science in Electrical Engineering. After the acceptance of this thesis, Laurana will be leaving UT with her Master's degree in Electrical Engineering focusing on Machine Vision.