To the Graduate Council:

I am submitting herewith a thesis by W. Harvey Gray entitled "Integration of multiple range and thermal pairs using a volumetric method to create textured 3D models." I have examined the final paper 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

Mongi A. Abidi, Major Professor

We have read this thesis and recommended its acceptance:

Daniel B. Koch

Acceptance for the Council:

Vice Provost and Dean of Graduate Studies

INTEGRATION OF MULTIPLE RANGE AND THERMAL IMAGE PAIRS USING A VOLUMETRIC METHOD TO CREATE TEXTURED 3D MODELS

A thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

William Harvey Gray Jr.
December 2002

THESIS
2002
.G73

Acknowledgements

This work was supported by the U. S. Army's TACOM program and by Radian, Inc. under contract DAAE07-96-X157.

I would like to thank my parents, Harvey and Nancy Gray, for providing me with limitless amounts of support and patience throughout my graduate career. I would also like to thank Dr. Mongi A. Abidi for selecting me as a Graduate Research Assistant, and for providing such an interesting topic to work on. Thanks also go to Dr. Daniel Koch and Dr. Jens Gregor for serving on my thesis committee. I would also like to thank both Christophe Dumont and Dr. Jeff Price for their valuable ideas on how to attack my thesis topic. Finally, I'd like to send a special thanks to my friends, both in the lab and out, for keeping me sane throughout the years.

Abstract

In this thesis we present a volumetric approach to 3D modeling that considers both object texture and geometry in the reconstruction process. This method assumes that there are multiple range and thermal image pairs of an object, all registered to a global coordinate system. The individual range images are used to create a surface mesh to which the thermal images are applied as a texture map. These textured range meshes are then used to update a volume grid. The volume grid can then be triangulated to yield a fully textured 3D model.

The motivation for the research is the simulation of a thermal tire inspection system. Integrating 3D geometry information with 2D thermal images permits the thermal information to be displayed as a texture map on the tire structure, enhancing analysis capabilities. Additionally, constructing the tire geometry during the inspection process allows the tire to be examined for structural defects that might be missed if the thermal data was textured onto a predefined model.

Experimental results demonstrate the efficacy of the proposed approach and quantitative experiments indicate that the volumetric integration technique compares favorably to a state-of-the-art mesh-based integration approach in terms of geometrical accuracy. Future research goals are also noted.

Contents

1 Introduction **1**

1.1 Overview 2

1.2 Thesis Outline 3

2 Review of Related Work in 3D Object Model Reconstruction **4**

2.1 3D Object Model Reconstruction 4

2.1.1 Volume Based Approaches 7

2.1.2 Surface Based Approaches 12

2.2 Other Related Topics in Defect Detection 13

2.3 The Impetus for our Work 14

3 System Overview **15**

3.1 Simulation Environment and Problem Formulation 15

3.1.1 Assumptions Made to Simplify the Problem 19

3.2 System Software 19

3.2.1 System Software Parameters 20

3.3 Range and Thermal Scan Simulators 22

3.4 Summary of the System 24

4 The 3D Object Model Reconstruction Algorithms **25**

4.1 The Volume Grid Data Capture Algorithm 25

4.1.1 General Structure 26

4.1.2 Data Acquisition 27

4.2 The Volume Grid 3D Model Builder 34

4.2.1 The Marching Cubes Algorithm 35

4.2.2 The Marching Cubes Algorithm with Adaptive Edge Lengths and Texture Mapping. 37

4.3 Summary of the 3D Object Model Reconstruction Algorithms 41

5 Experimental Results **42**

5.1 Incremental Steps in the Reconstruction Process 42

5.2 Final Results of Reconstructed Models 44

5.3 Comparison with a Mesh Based Approach 47

5.4 Conclusion of Experimental Results 60

6 Conclusions and Future Work **61**

References **63**

Appendix **67**

Vita **69**

List of Figures

1 The Dynamometer Model. 16

2 The detailed tire model rendered in wireframe (a), and shaded with thermal information (b). 17

3 Data captured from a single view of the tire model to be used for 3D reconstruction. 18

4 An example set of viewpoints for the range and thermal scan simulators. . . . 23

5 An illustration of how the voxel grid is scanned. 28

6 Example stages of a line scan. 30

7 Overlay of the two range meshes scanned from camera positions 1 and 2. . . 31

8 Illustration of the voxel values after the initial scans in the x and y directions. 32

9 Illustration of the voxel values after the scans in the x and y directions of the second range mesh. 33

10 The 15 Marching Cubes cases. 36

11 The voxel occupancy value is the summation of the values from the individual rays in the x, y, and z directions. 39

12 An example cube formed by eight voxels. 40

13 Surfaces constructed by triangulating the range images from two viewpoints, shown with the registered thermal data as the texture map. 43

14 Three views of the surface produced from the volumetric reconstruction algorithm using only the single view from Figure 12(a). 45

15 Three views of the surface produced from the volumetric reconstruction algorithm after integrating the two views from Figure 12. 46

16 Model reconstructed by integrating $N = 8$ views using an $M = 16$ (16 x 16 x 16) volume grid. 48

17 Model reconstructed by integrating $N = 8$ views using an $M = 32$ (32 x 32 x 32) volume grid. 49

18 Model reconstructed by integrating $N = 8$ views using an $M = 64$ (64 x 64 x 64) volume grid. 50

19 Model reconstructed by integrating $N = 16$ views using an $M = 16$ (16 x 16 x 16) volume grid. 51

20 Model reconstructed by integrating $N = 16$ views using an $M = 32$ (32 x 32 x 32) volume grid. 52

21 Model reconstructed by integrating $N = 16$ views using an $M = 64$ (64 x 64 x 64) volume grid. 53

22 Mean error percentages for the volumetric and mesh-based reconstruction algorithms using eight views. 55

23 Mean error percentages for the volumetric and mesh-based reconstruction algorithms using sixteen views. 56

24 Mean error percentages for the volumetric reconstruction algorithm using differing numbers of views. 57

25 Number of triangles in the final reconstructed 3D models. 59

1 Introduction

The reconstruction of high detail 3D scenes and object models is a process that consists of several independent, but highly interrelated steps. The quality of each individual step will affect the overall quality of the final 3D object model. The first step consists of acquiring data about the original object in question. Such data could be range maps, intensity images, thermal images, or any number of other forms of data. The next stage involved is the registration process where the relationships between different sets of data are defined. The final stage is where the registered sets of data are integrated to form a final 3D object model. The final stage is known as the reconstruction stage, and will be the focus of this paper.

In the reconstruction algorithm we present, multiple range and thermal pairs from different viewpoints are integrated to form a complete 3D tire model, with the thermal data applied as a texture map. The motivation behind the reconstruction of the tire model is to simulate a thermal inspection system. As the equipment involved in such a system can be expensive, fragile, and/or cumbersome to move about, simulation offers a distinct advantage over a typical heuristic approach. Through simulation, sensor placement can be optimized and equipment performance can be evaluated quickly and without risk to costly components, thereby providing fast, low-cost design capabilities.

1.1 Overview

The reconstruction system consists of two main parts: the volume grid data capture module, and the volume grid 3D model builder. One other part of note is the range and thermal simulator. In order to accurately measure the results from our algorithms we need the capability of a simulated range scanner to drive the reconstruction process. As a result we can compare any results we obtain versus the original baseline 3D model. The use of a simulated range scanner also eliminates the need of a registration stage, again useful because imperfections in that stage could bias the overall results. Thermal images are also simulated from the same viewpoints and are used as texture maps on the resulting range images. The reconstruction system accepts the resulting data as inputs, stored as Open Inventor files, thus any real world set of data could possibly work with this algorithm. One should note that the data should be pre-registered before being used with the reconstruction algorithm.

Once the set of range and image pairs are generated we are ready to begin the reconstruction process. First, a volume grid data structure is created and initialized. Next the reconstruction algorithm is passed a range and thermal pair, which it uses to update the data structure. This process is repeated for each pair of range and texture data. The next step is to use the stored data to create our final reconstructed 3D model. The data in the volume grid is then triangulated, and the thermal textures are

applied to the generated triangles. The final step is to put the triangles into a readable format, again an Open Inventor file.

1.2 Thesis Outline

In Chapter 2, we review the work others have done in the area of 3D object model reconstruction, and determine the goals of our algorithm. In Chapter 3, we provide a system overview detailing the simulation environment that lead to our problem formulation. In Chapter 4, we describe in detail the operation of our 3D object model reconstruction algorithm. In Chapter 5, we provide the results from various reconstructions of the tire model, along with a comparison with a state-of-the-art mesh-based method. We finish in Chapter 6 with some conclusions drawn from our research and note future work that is possible.

2 Review of Related Work in 3D Object Model Reconstruction

In Section 2.1, we describe the problems involved in displaying a 3D model of a scene and how these factors are taken into account in the 3D reconstruction problem. The various methods used in 3D object model reconstruction are then listed. In Section 2.2, we delve into topics that are related to our work in the area of defect detection. We finish in Section 2.3 with a summary of this chapter.

2.1 3D Object Model Reconstruction

As the name implies, the goal of a 3D object model reconstruction algorithm is to generate a digital model of an object using information from multiple viewpoints. To truly understand the objectives of these algorithms we must first look at general 3D models. The most common representation of a 3D model is the vertex-face representation where each vertex corresponds to a 3D point, and the faces are formed by connecting several, typically three, of the vertices together. By using this representation we can give the 3D model a distinct geometric shape, and can provide a basic means for coloration of the model. Coloration depends on the interaction between the three main attributes of the scene: the color of the model, the surface normals of the model, and the lighting of the scene. The color of the model can be attached to the model in either a per-vertex or per-face format. Per-vertex is more interesting to us because the colors can be interpolated across any single face. The

surface normals of the model can be used to provide some basic shading for the model. These simple shading schemes are known as flat, Gouraud, and Phong shading [28]. Flat shading is where any one face is given a single surface normal for shading purposes. Gouraud shading goes one step further by specifying a surface normal per vertex. This normal is then used in the lighting equations to modify the color for output. These modified colors are then interpolated across the face to produce a smooth shading effect. Phong shading is the most realistic of these three shading schemes. Phong shading interpolates the per-vertex surface normals across the face. It provides a level of detail greater than Gouraud shading by more accurately imaging specular effects on the surface of the object. The lighting of the scene is less important to us and is not covered in this section.

These simple shading routines work quite well when a face corresponds to the approximate size of an output pixel when rendered. When the size of a face lies outside these general bounds there are troubles in providing a visually appealing image to view. When a face is smaller than a pixel the problem is one of minification, when a face is larger than a pixel the problem is one of magnification. Minification typically involves dealing with level of detail of the model. Mesh reduction is most closely related to this aspect, and is outside the scope of this thesis. The method used to combat the magnification problem is much simpler, texture mapping. Texture mapping was first proposed by Catmull [1] as a way of adding photo-realism to coarse 3D models. An image is mapped to the surface of a 3D object to provide details rather

than increasing the number of vertices. As such the realism of the display of the model is then dependant on the resolution of the texture map rather than the model itself.

There are two primary ways of mapping a texture onto a 3D model. The first is a function-based approach that can be as simple, such as a planar mapping, or as complex, such as a spherical or user generated mapping, as the user decides is necessary. The texture coordinates are determined by analysis of the function, and can be a viable method for certain models. The problem lies in that many of these mapping functions cannot take into account occluded areas and will often provide a less than optimal result. Per-vertex mapping, the second approach, takes care of the occlusion problem by providing the user a means of specifying one texture coordinate for each vertex. These texture coordinates are then used in conjunction with a shading function. The benefits of both methods are obvious, rather than simply relying on a vertex-to-vertex interpolation of color the texture coordinates are used to interpolate inside an image providing greater detail.

When designing a 3D object model reconstruction algorithm it is necessary to take into account the various methods pertaining to displaying a general 3D model. It is highly desirable to reconstruct a model that provides textures for the surfaces, and provides accurate geometry and surface normals. Our reconstruction process was geared towards this task.

In general there are two primary ways of tackling the reconstruction problem: volume based and surface-based approaches. These approaches are detailed in the following sub-sections.

2.1.1 Volume Based Approaches

A volume based approach generally revolves around estimating the probability of occupancy of a voxel in a volume grid. As such, the final model is typically well ordered and can be more robust than a method that looks only at surface interactions. To view a volumetric result a voxel based rendering scheme, or a method that triangulates an iso-surface [2] [3] can be used.

Initial work in volumetric methods, such as Connolly [4], tended to look for ways to reduce the amount of data that was incurred when using a volumetric representation. The octree data structure takes care of this need by using a hierarchy to allocate memory only when a voxel intersection takes place. This ordering works well in certain cases such as simple objects or sparse scenes. This particular method maximizes occupied space, and is very susceptible to outliers.

Pulli et al. proposed a similar method [5] in that an octree data structure was used. There were two primary differences: first, Pulli chose to project the octree cubes directly to the depth maps, where as Connolly chose to convert the maps to quadtrees and project these structures into an octree; and second, Pulli chose to favor proof of

empty space. The projection of the octree cubes to the depth maps eliminated two problems: too much carving, and carving at an unnecessarily fine resolution. By choosing to maximize the amount of empty space they managed to get rid of the worst of the outliers. Unfortunately any outlier that lies behind the surface of the object will cause the algorithm to incorrectly carve away a part of the object that it shouldn't. While neither of these approaches are very similar to the reconstruction described here, it is important to note that there is a need for a better approximation of occupancy rather than just maximizing empty space or occupied space.

Curless and Levoy [6] propose a scheme that alleviates many of the problems associated with the carving bias from the previous two methods. This method defines a volumetric function based on the average signed distance to the set of range images. This function is discretely analyzed at points on a uniform grid and uses those sample values to approximate the zero set of the volumetric function. This scanning provides a very pleasing surface, but has two moderate drawbacks. First, due to the particular way the volume grid is scanned, features smaller than the grid spacing can be missed. This is not a very large drawback as the resolution of the feature size to be imaged should be a function of the volume grid resolution. The second drawback is that this procedure takes up a considerable amount more time and memory than octree methods. It is up to the user of a specific application to determine whether the trade-off of longer processing is worth the better results.

Pollefeys and Koch, *et al.* [7] [8] [9] provide a series of series of papers that deal primarily with structure from motion. They are of interest to our research due to their particular visualization implementation. As the reconstruction process takes place using a sequence of intensity images, texture mapping is intrinsically tied to the reconstruction process. The actual method used to reconstruct the data is to use the corresponding points to generate a dense 3D map that textures can be applied to. The particular implementation used is to generate a hyper-texture, the average of the set of textures that can be back projected from any one particular face upscaled to the highest sampled texture resolution. This texturing method provides both advantages and disadvantages in its implementation. The primary benefit is that specular effects and the like can be eliminated simply by averaging them out with other textures of the given set. The primary drawback is that confidence information is not used in the texture recombination process, thus giving additional weight to potentially poor textures.

Mangan [10] provides a similar framework as the above method, but instead of using dense 3D maps he uses level sets of volumes and deformable models to estimate the most likely location for the surface of the object. It should also be noted that Mangan uses range images to drive the reconstruction process rather than structure from motion. After the estimated surface is decided the texturing process is much the same as in the Pollefeys method. Any polygon that can back project to a particular texture

will be combined at the highest possible resolution with all of the benefits and drawbacks as before.

Pulli, Shapiro, *et al.* return with [11] and [12] to extend their previous work with octrees [5]. The new system still uses an octree basis in order to utilize the space savings that are the hallmark of such methods. In order to provide a more pleasing output model than a traditional octree will provide they adopt a new framework for the display of such models. First off the model is smoothed to eliminate the blockiness traditionally associated with volumetric methods that lack additional shape information. Next the updated model is textured using view-dependant texturing. This form of texturing takes into account the position of the viewer relative to the model as the primary decision of what texture to use. While this method can provide pleasing visual results for a viewer it has yet to be proven accurate for applications that need both accurate geometry and textures.

Eisert, Steinbach, Girod, *et al.* take a unique approach to the coloration of voxel grids by coloring the voxels themselves [13]. They provide a multi-hypothesis framework in that each individual image influences the final voxel color. If a particular voxel has corresponding colors across all viewpoints that can project to it the voxel remains that color. If on the other hand the voxel has conflicting color information it is set to be transparent. Another interesting method is used in [14] in which a deformable model is fit over a volumetric model. This deformable model is then textured using the various images from the different views. A novel approach is used in that this texture

can slide across the surface of the deformable model in order to more accurately represent further integrated views. Finally, the principles of anisotropic projection are used in [15] as an extension of [13]. Rather than assuming a projection of a voxel to an image plane used as a texture is square, they perform perspective calculations to get the actual mapping of the voxel. This method leads to better results than by the previous method. While both [13] and [15] show promise they cannot escape the magnification problem that lead to the invention of texture mapping so long ago.

The reconstruction method employed here is most similar to the work done by Curless and Levoy [6] in that the voxel implementation is supplemented by a distance function to more accurately model the surface of the reconstructed 3D object model. Our method differs from the previous method in several key areas however. The first, and most easily recognizable difference is that we chose to include texture mapping in the reconstruction process. Second, the actual determination of the location of the surface is dissimilar. The voxel occupancy status consists of a weighted sum determined by the measured surface interactions that allows us to efficiently eliminate outliers. This method is in contrast to the methods that maximize either empty or occupied space, both of which are very susceptible to outliers. After the general location of the voxel surface has been established, we then check all of the available results in the local neighborhood of 8 voxels. We then choose the intersections with the highest confidence value in any given range rather than averaging. We chose this method for several reasons: the first of which is that the voxel occupancy should reject outliers

that could have an effect on surface position, and second this method lends itself to texture mapping without the need for back-projection. This reconstruction method will be elaborated on in Chapter 4.

2.1.2 Surface Based Approaches

Surface based approaches tend to take a more localized view to the reconstruction process. In these methods individual surfaces are meshed and then combined together using a variety of combination functions. When dealing with an area of an object with no overlap, these methods can give better results than a volumetric method. However, when dealing with areas that contain significant amounts of overlap the robustness of the volumetric methods tend to win out.

Turk and Levoy [16] treat each mesh as a separate entity and then use a function to "zipper" together the regions that overlap. Soucy, *et al.* [17] [18] describe a system that first implements a surface object that consists of the integration of several range views. A hierarchical system is then use to eliminate vertices and develop an object model of reduced complexity in the areas of overlap. Pito [19] uses confidence information to eliminate triangles in areas of overlap, and then stitches the resulting areas together. Sun, *et al.* [20] uses a modification of [19] extended to take advantage of texture mapping and some other minor modifications. Hoppe *et al.* [21] looks at the set of range images as a set of un-organized points, and describe a signed distance

function to the set. The zero-set is then used to reconstruct a surface mesh, which is then put through a mesh reduction routine. This simplified mesh is then used to fit a piecewise smooth surface representation back to the original data.

2.2 Other Related Topics in Defect Detection

An area of importance to the system we are developing is the area of defect detection. A quality control system such as the one detailed in Chapter 3 aims at capturing information about an object in order to determine whether or not it is defective. Defects are to be detected using specific sensors that acquire multi-modal information about the particular object. Sensors can also be used in conjunction with specific lighting systems to improve results in some cases, such as with the work done by Perard and Beyerer [22]. The multi-modality of the system aids in the detection process: a thermal camera may show a manufacturing defect under the surface of an object that may not be properly detected by range/intensity data and vise-versa.

Camera placement is a key issue for efficient defect detection. Aluze, *et al.* [23] use a black and white camera in association with a structured lighting system. This system is then set up relative to an object so that defects appear clearly in the resulting images. While some systems place cameras based on a heuristic approach there are also many applications that use a formal methodology, or even attempt to simulate vision systems in order to optimize the systems parameters. One such methodology is

used by Coulot, *et al.* [24] to choose an optimal lighting system for the purpose of detecting defects located on the surface of metallic objects.

2.3 The Impetus for Our Work

After reviewing topics in the area of 3D object model reconstruction it is apparent that there is much room for improvement. Volumetric systems based on structure from motion are easily textured, but the geometry of such objects typically leaves a lot to be desired. Conversely volumetric systems based on data taken from range scanners yield good geometry, but are difficult to extend to include data from other modalities on the surface of the object. Surface meshing algorithms can yield good results in certain cases, but in general are not robust enough to accurately handle areas of multiple overlap.

We propose to create a volume based reconstruction process that gives the benefits of a range based method, while extending the process to tackle the problem of including textures on the surface. We also will show a comparison with a surface based method that illustrates how overlap can be used to the benefit of our system.

3 System Overview

In Section 3.1, we describe the simulation environment and how it relates to the general 3D reconstruction problem, and list the assumptions we used to simplify it. In Section 3.2, we discuss the elements that compose the system software. In Section 3.3, we provide details about the range and thermal scan simulators. We finish in Section 3.4 with a summary of this chapter.

3.1 Simulation Environment and Problem Formulation

The goal of any imaging system is to provide users with quality data about an object. In this case the object is a tire with a specific thermal signature. The setup of such a system relies on the interrelation between several key sensor parameters and the positioning of those same sensors relative to the tire in the system. In this case the sensors are a laser range finder and a thermal camera, both simulated.

The tire model is to be placed in a scene simulating a dynamometer (see Figure 1) test machine, also known as an endurance drum test. The dynamometer test machine consists of three major components: a steel drum to rotate the tire against, the tire mount, and a hydraulic ram to keep the tire in contact with the drum. The machine allows a tire to be rotated at a given speed and load in order to generate heat and determine wear.

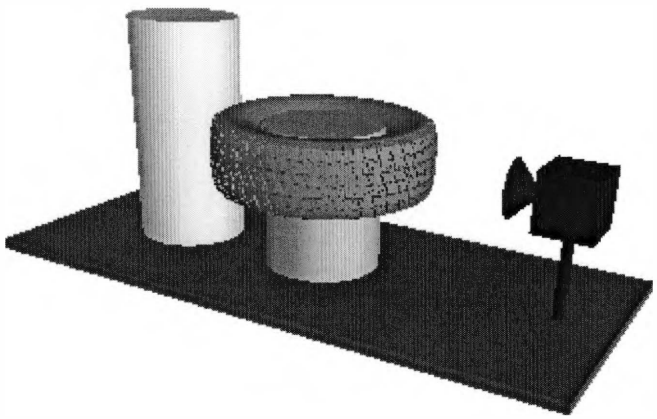
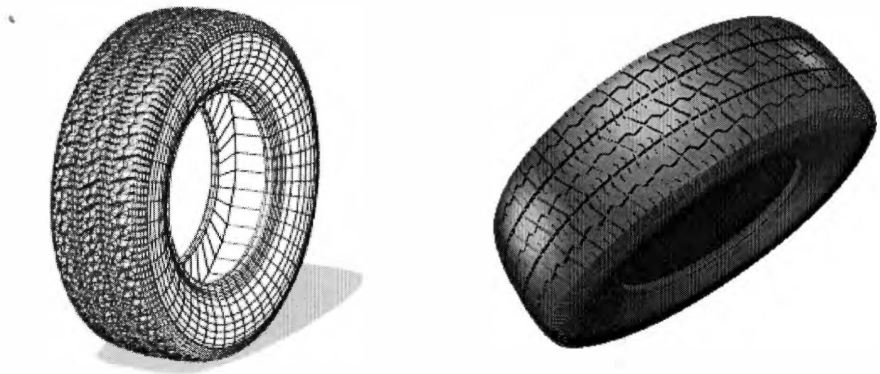


Figure 1: The Dynamometer Model.

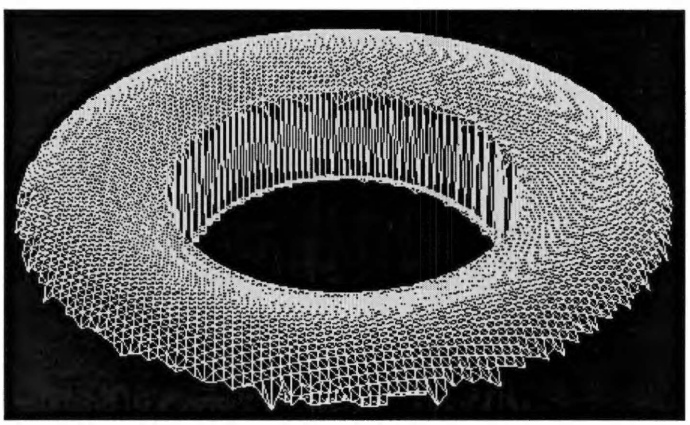
The tire model itself is the key component of the dynamometer system. This model contains approximately 25,000 polygons (see Figure 2(a)). It is vertex shaded (see Figure 2(b)) to give the appearance of having thermal data on the tire surface. The vertex shading is necessary to generate a model that will mimic real world thermal data.

The tire model must then be scanned by both a range and thermal camera (see Figure 3). Both of these cameras are modeled using the pinhole model [25]. The quality of these acquired views will be influenced by several factors. Resolution and field of view are the two primary attributes that we are concerned with. Resolution of the range scan should be such as to give a finely detailed surface mesh. Resolution of the thermal camera (typically higher than the range scan) will provide the texture maps for the reconstruction stage. Finally the field of view should be such that the tire appears in each camera view.



(a) (b)

Figure 2: The detailed tire model rendered in wireframe (a), and shaded with thermal information (b).



(a)



(b)

Figure 3: Data captured from a single view of the tire model to be used for 3D reconstruction. (a) Surface mesh created by triangulating a simulated range image, shown slightly rotated to illustrate depth. (b) A sequence of simulated thermal images.

3.1.1 Assumptions Made to Simplify the Problem

In order to simplify the problem detailed in Section 3.1, we must make several assumptions. First we assume that we know size and locations of all the objects that are a part of the dynamometer test setup and location of the associated sensors. This assumption is an intrinsic part of the process as we are actively trying to model the system before setup. These initial assumptions allow us to not only select the proper resolution and field of view for the range and thermal cameras, but to determine the location and necessary size of the volume grid as well.

The next assumption is that all of the range and thermal pairs are registered to each other. We also assume that these individually registered pairs are registered to a global coordinate system before passing through to the reconstruction stage. The registration stage is necessary to provide coherency of information, and is beyond the scope of this thesis.

3.2 System Software

The system software consists of three primary parts: the range and thermal scan simulators, the volume grid data capture algorithm, and the volume grid 3D model builder. We use the range and thermal scan simulators to acquire data to be used in the reconstruction process. These scans are taken in locations determined by the user. When the scans are finished the results are triangulated to create range meshes with

thermal textures, and the results are passed to the next stage. The volume grid data capture algorithm examines these textured meshes one at a time, and updates the volume grid to reflect the information. After each update of the volume grid 3D model builder is run on the information contained in the volume grid. A surface mesh containing all of the information and relevant texture maps is generated. The process is complete when all of the views have been integrated in the volume grid, and the resulting information has been used to create a reconstructed object model. The volume grid data capture algorithm and 3D model builder are explained more fully in Section 4.

All parts of the system software were run on SGI workstations using C++ and the Open Inventor API. A command line interface is used for changing all parameters and file paths. Alternately it was found that an executable batch file could speed up the process of reconstructing a model with multiple views considerably. The tire model, the intermediate textured meshes, and the final reconstructed models are all in Inventor (.iv) format. The thermal texture maps are generated in the SGI RGB format.

3.2.1 System Software Parameters

There are several parameters that control the output of the reconstruction system. Changes in these parameters will affect the output of the system. Range and thermal scan position, resolution, and field of view will affect the quality of textured meshes that are passed to the reconstruction algorithm. The number of textured meshes

integrated will help determine the quality of the reconstructed 3D object model. Volume grid size and resolution will also have a large impact on the quality of the reconstructed model.

The range and thermal scan parameters can change the results for the remaining stages, so it is imperative that they are optimized to produce textured meshes adequate for the reconstruction stages. The interrelation of the range and thermal cameras position and field of view determines how much of the object is imaged. Ideally we want these parameters to provide views wherein the object is completely imaged from that direction while empty space is minimized. Any increase in the distance from the object, or increase in the field of view tends to increase empty space or allows more of the object to be imaged. To decrease the amount of empty space in a range scan the opposite must be done.

The range and thermal scan resolutions are another key parameter for getting good results. These parameters do not have to be equal. In fact we found better results when the thermal resolution was larger than the range resolution. The range resolution determines the size of the output textured mesh, and has a direct effect on the runtime of both the range scanner and the reconstruction stages. The smaller the resolution the faster all later stages run. The resolution of the thermal image has a very small effect on the initial capture and reconstruction stages. However, large texture sizes can cause the refresh rate to plummet when displaying the final reconstructed model.

The output of the final reconstructed 3D model is controlled by the volume grid size and resolution. The size of the volume grid should roughly correspond to the size of the tire. In order not to clip off parts of the tire it is good to provide a small buffer area. The volume grid resolution indirectly controls the number of triangles in the final output models. As the volume grid resolution increases so does the amount of triangles in the reconstructed model. The relationship is indirect in that there is no algorithm to determine exactly how many triangles a given volume grid resolution will provide.

3.3 Range and Thermal Scan Simulators

In order to fully reconstruct the tire model we must take multiple range and texture scans. These scans must be arranged (see Figure 4) in order to provide full coverage for the tire model. At each selected point we must take a range scan and a thermal scan. The range scan is considered $2\frac{1}{2}$D information in that it contains 3D data, but only from one particular view. After the range data is captured we proceed to triangulate the information to create a surface mesh. Each face of the triangle is assigned a confidence value based upon the dot product of the triangle normal and the given view direction. In this particular case all of the values should lie between zero and negative one. Zero being the lowest possible confidence (where the triangle normal is tangent to the incoming view vector), and negative one being the highest possible confidence (where the triangle normal points directly at the view vector).

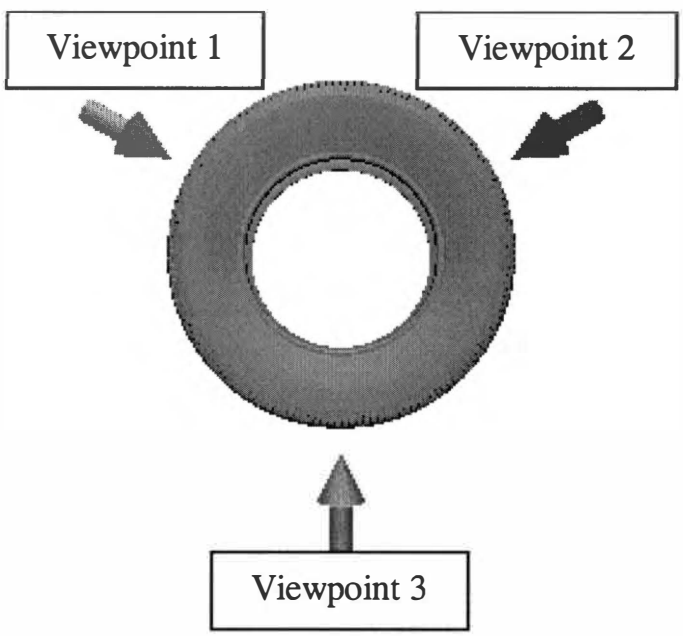


Figure 4: An example set of viewpoints for the range and thermal scan simulators. The viewpoints are chosen to provide full coverage of the tire.

These values are then used to index into a color table. These colors are how we pass the confidence values along to the next stage. The thermal image is then taken, and the texture coordinates corresponding to this image are assigned to the surface mesh. All of the surface mesh information is written to an Inventor file, and the thermal texture is written to a RGB file.

3.4 Summary of the System

We have illustrated the simulation environment and how it relates to the general 3D reconstruction problem while discussing the assumptions needed for our system to work. The system software used to attack the problem has been described. This software consists of the range and thermal scan simulator, the volume grid data capture algorithm, and the volume grid 3D model builder. The range and thermal scanner creates textured surface meshes that are operated on by the remaining parts of the system. The volume grid data capture algorithm and 3D model builder are described in the next section.

4 The 3D Object Model Reconstruction Algorithms

In Section 4.1, we provide details on how the Volume Grid Data Capture Algorithm (VGDCA) takes the input range and thermal scans and transfers them into our volume grid. In Section 4.2, we describe how the Volume Grid 3D Model Builder (VG3MB) takes the data from the VGDCA and renders a textured 3D model. We finish in Section 4.3 by providing a summary of the operations involved in the 3D object model reconstruction algorithms.

4.1 The Volume Grid Data Capture Algorithm

The goal of the VGDCA is to capture the data from each of the input range and texture pairs and integrate them into a representation that can be used to visualize a 3D object model. The number of views and their general size are known *a priori*, but the VGDCA makes no assumptions based upon camera parameters or locations. It should be noted, however, that well registered data is a must for obtaining good results. We also note that our reconstruction algorithm employs an occupancy grid approach similar to that found in the next-best-view (NBV) work of Wong *et al.* [26]. Adopting a NBV approach into our system would permit the reconstruction of unknown objects, where the number of views (and the associated range and texture images) is not specified ahead of time.

4.1.1 General Structure

The first stage of the process is the initialization of the volume grid and its associated data structures. As the user specifies the volume grid size, we have to allocate the necessary memory at run-time. The two primary parts of the data structure are the voxel grid and the three edge grids corresponding to the directions x, y, and z. The voxel grid is an M x M x M array of signed integers that contains information on whether the voxel is occupied or not, where M is the specified dimension of the volume grid. The edge data structures consist of information necessary for the rendering process. It is a NULL array of size M x M x (M-1) x 3 x N, where N is the number of range and texture pairs. Memory is allocated as needed in this array as the data structure is updated. The elements of the edge data structure are the confidence of the point on the edge, the location of the point, and the texture coordinates of the point. The confidence, an unsigned integer, is encoded on the surface of the range mesh as detailed in the last chapter. The location, a floating point number, is the distance along the ray between two adjacent voxels. The texture coordinates, a pair of floating point numbers, are recovered from the surface of the range mesh, and are used to accurately apply the texture to the final 3D model. The voxel grid is initially set to all ones, and the edge grids are all NULL.

4.1.2 Data Acquisition

The following sequence of steps are performed for each view v = 1,...,*N*. First, the range mesh R^v is loaded. Next, we project orthographic rays along the voxel scan lines, in this case the x, y, and z axes, to update the voxel grid and store the surface intersection information in the edge data structure. An illustration of how the voxel grid is scanned is shown in Figure 5. The number of surface intersections drives the updating process. If a ray hits nothing inside of the volume grid data structure we consider the scan line empty. If the ray hits the object inside of the data structure we proceed to the next stage. First we break the individual surface intersections into their respective cells inside the volume grid. If there are multiple intersections inside any one cell we must make certain decisions. If the number of intersections is even that means that there is a "double crossing" in that particular cell location. In this case we throw out these intersections. If the number is odd, we keep the first intersection, and throw out the remaining intersections. In other words we consider the first intersection valid, and the remaining intersections "double crossings." The reasoning for throwing out the "double crossings" is that they represent features that are too small for the volume grid to accurately image. In order to accurately image these features the voxel resolution must be increased.

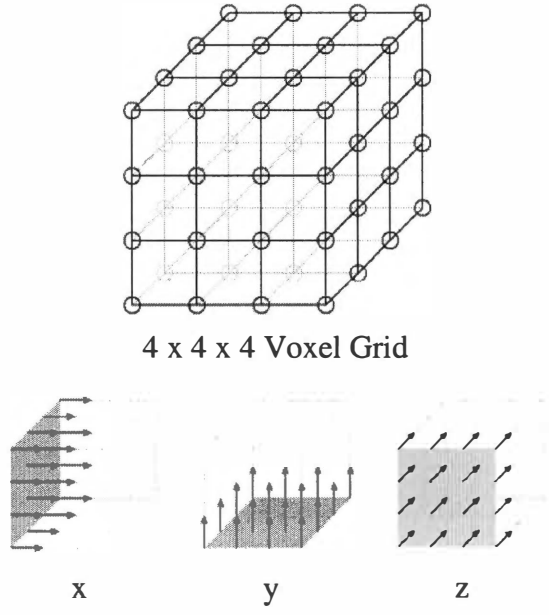


Figure 5: An illustration of how the voxel grid is scanned. The rays in x, y, and z directions capture any edge intersections on the scan lines.

Once we have either a single or no surface intersection for each cell we can begin to update the information contained in the voxel grid. If there were no hits along the entire orthographic ray then we update all of the voxels along the ray by –1, signifying that they are considered empty by that particular view. If there is at least one hit along the ray we step down the ray checking the surface intersections for each cell, until we reach a cell that has a hit. First we check the dot product between the surface normal at the point of intersection and the orthographic ray. This check will allow us to determine the general orientation of the surface, and will allow us to properly update the voxel grid. If the dot product is negative we can deduce that the ray is entering the surface mesh, and the voxels outside the intersection should be counted as empty. We accomplish this task by propagating a –1 value back up the ray, decrementing each

cell until we reach the last cell that contains a surface intersection (or the end of the voxel grid). If the dot product is positive we can deduce that the ray is exiting the surface mesh, and the voxel grid values inside the intersection should be considered unknown. As such there is no need for propagating a value back up the ray.

The confidence at the surface intersection is then checked versus a value chosen by the user, in this case the value corresponds to an angle of incidence of approximately 60 degrees, although other values can be entered. If the confidence is sufficient to meet the user's criteria we then increment the desired voxel by +3, and decrement the voxel on the other side by -1. It is necessary to give this unbalanced weighting to help reduce errors in areas that have conflicting information, such as a tread pattern, which contains noise like attributes when sampled at low resolutions. The edge data structure is also updated with the confidence information, the location, and the texture coordinates at the point on the surface. An illustration of this process on a single ray is shown in Figure 6. This process is repeated until all of the rays have been scanned.

The process for Figure 6 is as follows: in the first step all hits along the scan line are accumulated, and the double crossing detected (highlighted by the dashed box) is removed. The first of the hits is taken care of in step 2 by adjusting the voxel values up to the intersection, and by storing the edge information. The same process is repeated for the second hit in step 3. Finally the remainder of the ray is considered in step 4 by considering the orientation of the last hit dealt with.

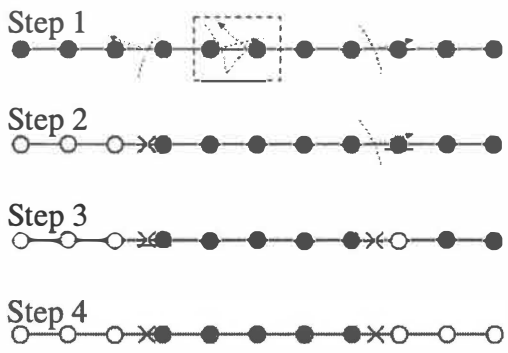


Figure 6: Example stages of a line scan.

The next step is to repeat the process of casting rays for each of the range and texture pairs supplied by the user. After all of the pairs have been integrated into the volume grid we are ready to run the Volume Grid 3D Model Builder. An example sequence showing the integration of two range scans and the updating of a volume grid is shown in Figures 7 – 9.

Figure 7 shows a 2D depiction of a sample object being scanned by two range cameras. The dashed region denotes the area where the integration will take place. The range cameras each produce a surface mesh for the object. Of particular note are the dotted lines on each of the surface meshes. They depict areas on the mesh with low confidence, in this case where the range scans miss the object.

Figure 8 illustrates how the process shown in Figure 6 is applied to the range mesh from camera 1. This process is applied sequentially to all of the rays in the X direction, and then repeated on all of the rays in the Y direction that line up with the

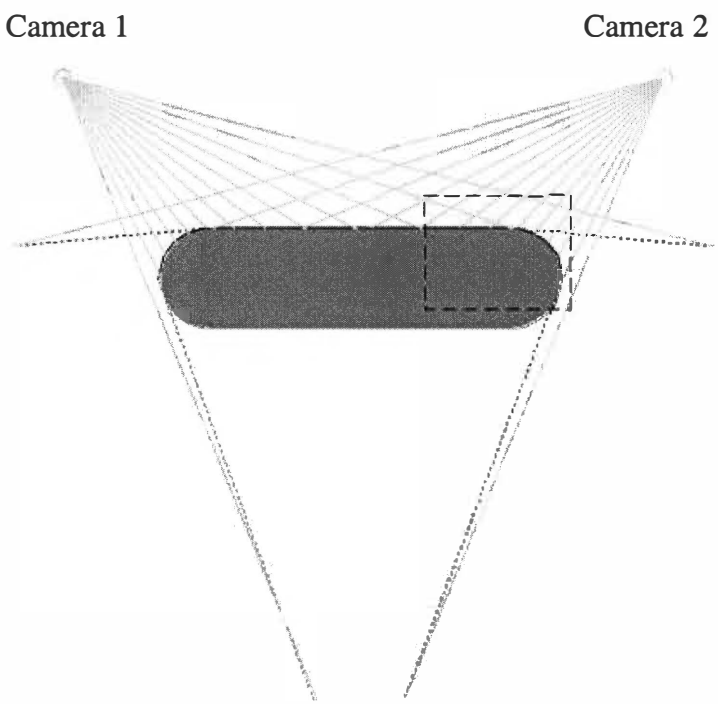


Figure 7: Overlay of the two range meshes scanned from camera positions 1 and 2. The dashed box represents where the volumetric integration will take place.

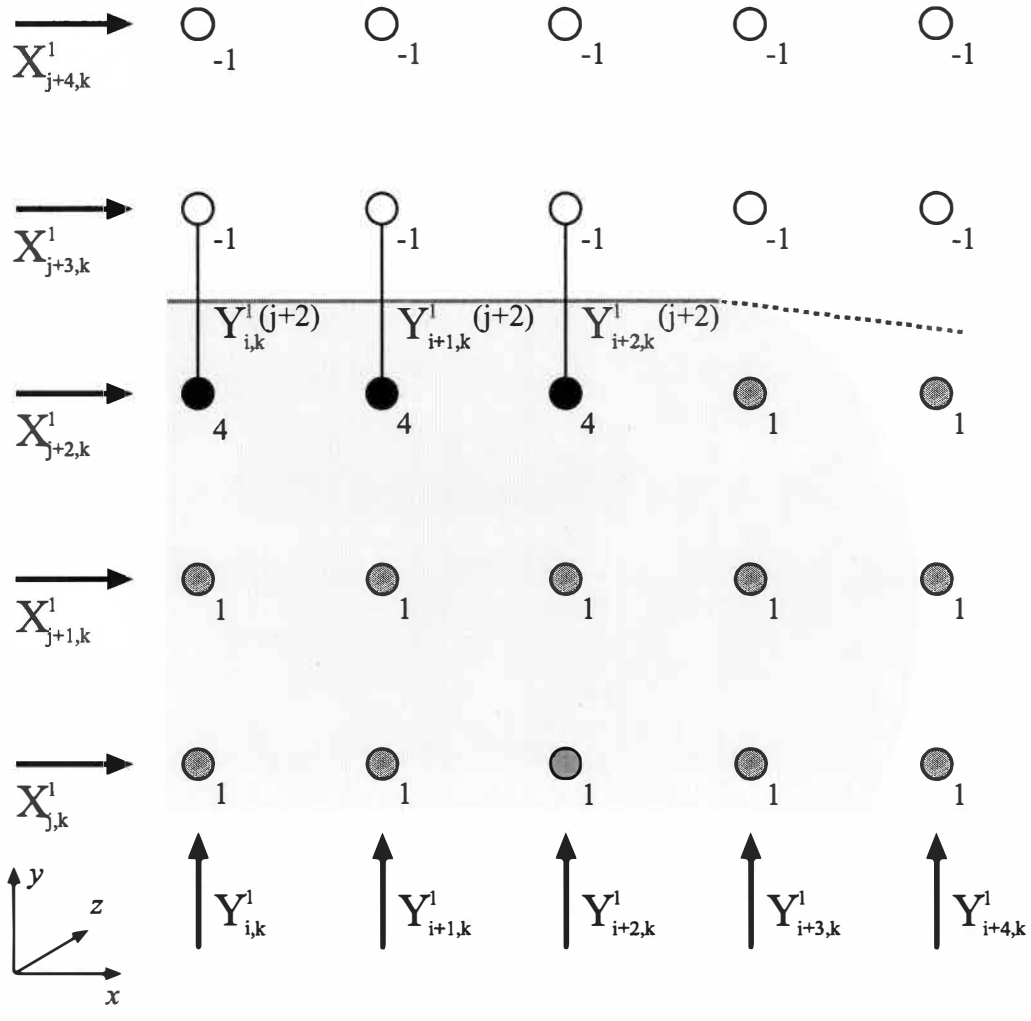


Figure 8: Illustration of the voxel values after the initial scans in the x and y directions. The voxel colors indicate how they were updated during the pass: white signifies decremented, black signifies incremented, and gray signifies untouched. A solid line between two voxels signifies a stored edge.

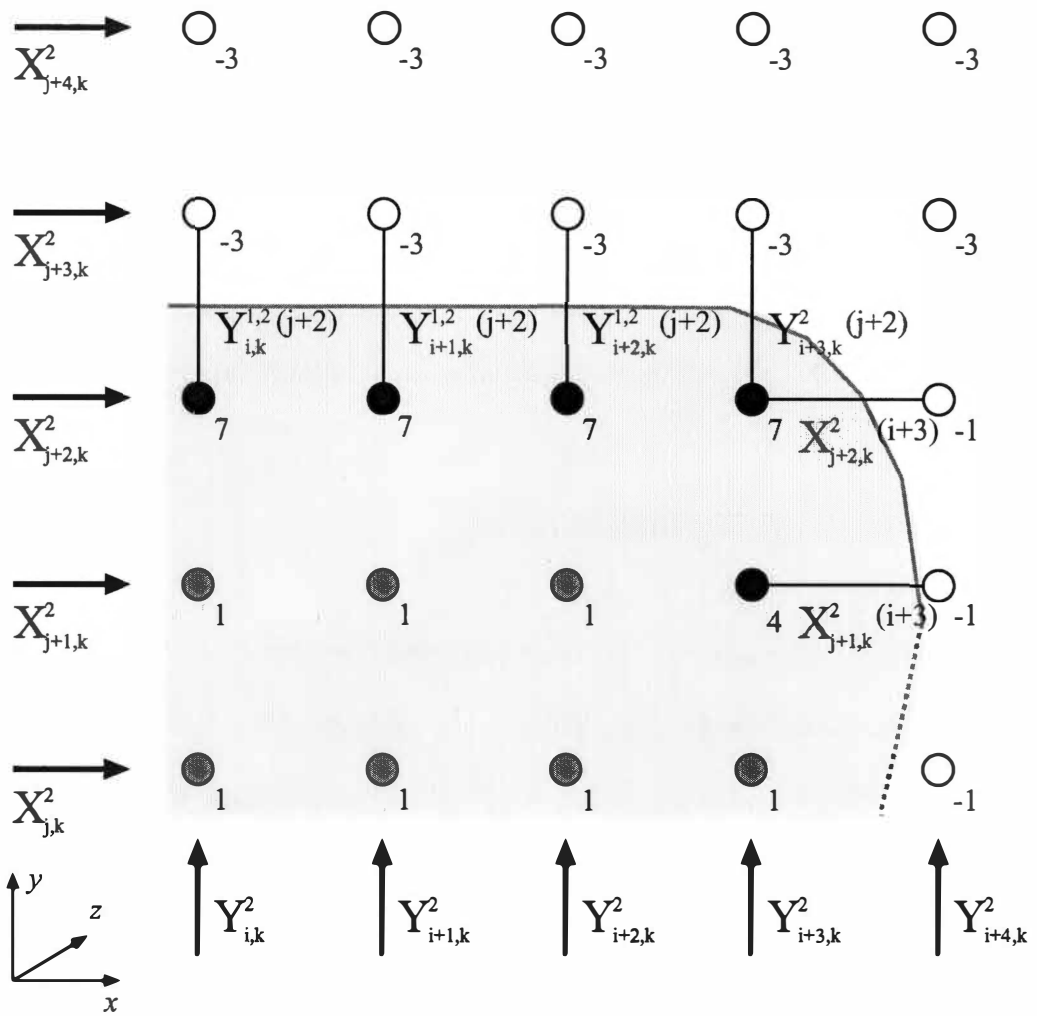


Figure 9: Illustration of the voxel values after the scans in the x and y directions of the second range mesh. The voxel colors indicate how they were updated during the pass: white signifies decremented, black signifies incremented, and gray signifies untouched. A solid line between two voxels signifies a stored edge. Note in this case that some Y edges contain information from both the first and second range meshes.

voxel grid. Note in particular that the voxel values are being incremented once by each direction. Also note that edge information is stored in several occasions.

Figure 9 repeats the process shown in Figure 6 on the range mesh from camera 2, but uses the same voxel grid as in Figure 8. It thus shows the integration of two individual range meshes into a volume grid. Edge information is stored for this range mesh and well. Note that there is even some redundant edge information for particular edges. This redundancy will be explored in greater detail in the reconstruction process.

4.2 The Volume Grid 3D Model Builder

Once we have a volume grid that is filled with data we would like to produce a triangulated, textured surface for visualization. In this case, the surface is the iso-surface between positive and non-positive values in our volume grid. To do this task we use the Volume Grid 3D Model Builder (VG3MB) algorithm. The VG3MB algorithm is a Marching Cubes Algorithm (MCA) (Lorensen and Cline [2]) that has been extended by using adaptive edge lengths and texture mapping. The finished tire model is output to an Open Inventor (.iv) file.

4.2.1 The Marching Cubes Algorithm

The MCA is a method for visualizing a voxel grid by creating a surface mesh of an iso-surface contained in the grid. It works by looking at eight voxels that form the corners of a cube. Each of these voxels are assigned a binary value of either on or off determined by the threshold value of the iso-surface. After the binary values are assigned the eight voxel cube can be categorized into one of 256 possible states. The complexity of this procedure can be reduced down to 15 possible states by allowing the cube to rotate around any primary (x, y, or z) axis, mirror across any primary axis, or invert the state of all voxels. Once the state of the voxel cube is known a look-up can be performed to get the triangulation of the iso-surface. This portion of the iso-surface may contain from zero to four triangles, see Figure 10. The triangles that are created by the MCA have vertices that lie on the edges of the cube formed by the eight voxels, and the vertices always lie at the center of these edges. The MCA repeats the triangle selection process for each of the eight voxel cubes in the voxel grid, in essence it "marches" through all of the cubes which gives the algorithm its unique name.

Note that the standard MCA has several deficiencies. First is the lack of interpolation along the edge. By always placing the vertex of the created triangles at the center of the edge between two voxels the MCA typically yields jagged models that are

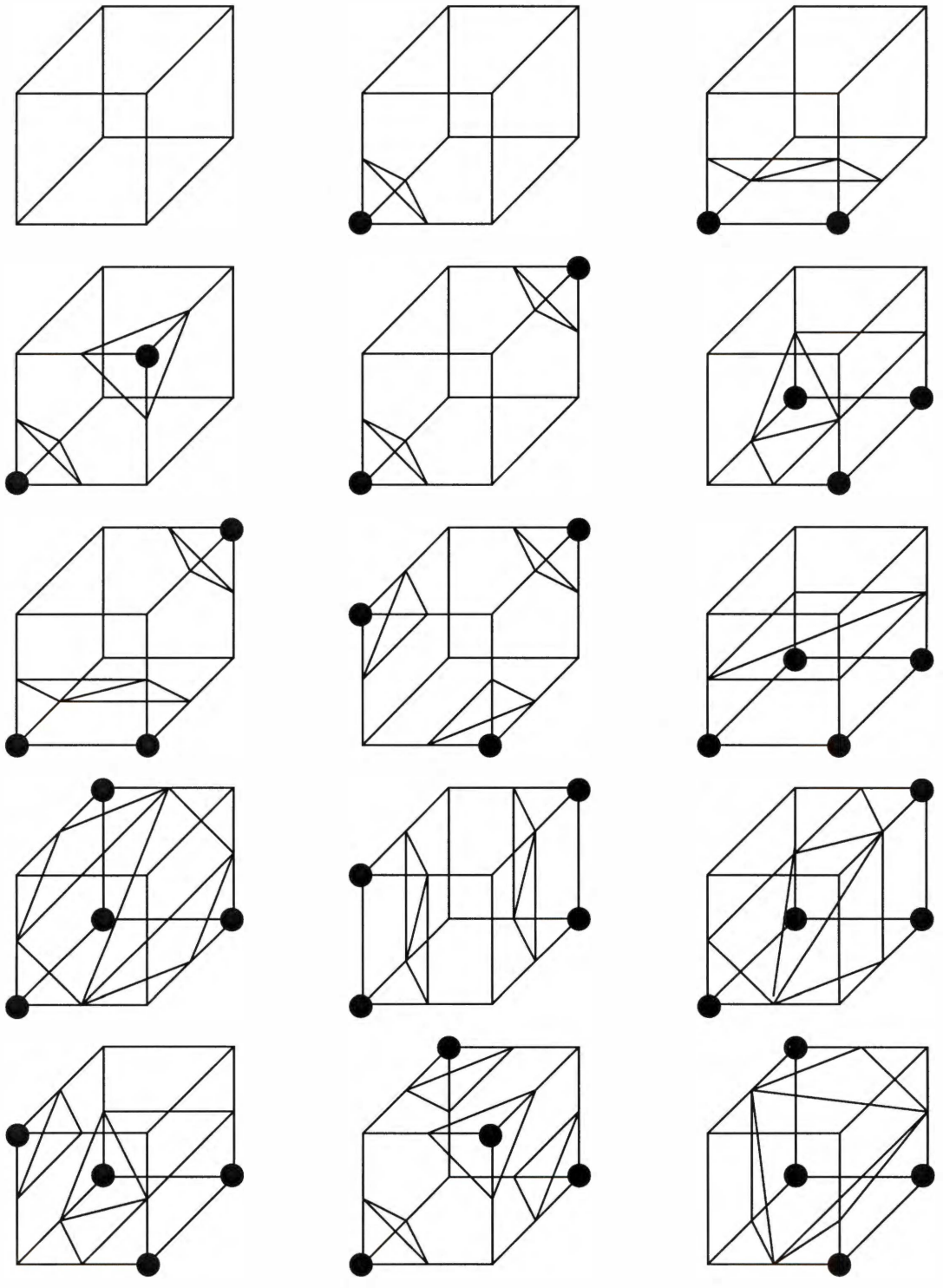


Figure 10: The 15 Marching Cubes cases.

unsuitable for texture mapping. Second is the lack of confidence information past the assignment of binary values to the voxels. After that point all voxels are considered either on or off, and all of the confidence information gained in the scanning process is lost.

4.2.2 The Marching Cubes Algorithm with Adaptive Edge Lengths and Texture Mapping

Our extensions to the Marching Cubes Algorithm consist of three interrelated parts. First is the inclusion of adaptive edge lengths. Second is the ability to retain confidence information for the individual sets of voxels. The final extension is the ability to apply multiple texture maps to the finished model. All of these extensions rely on the edge information stored in the VGDCA.

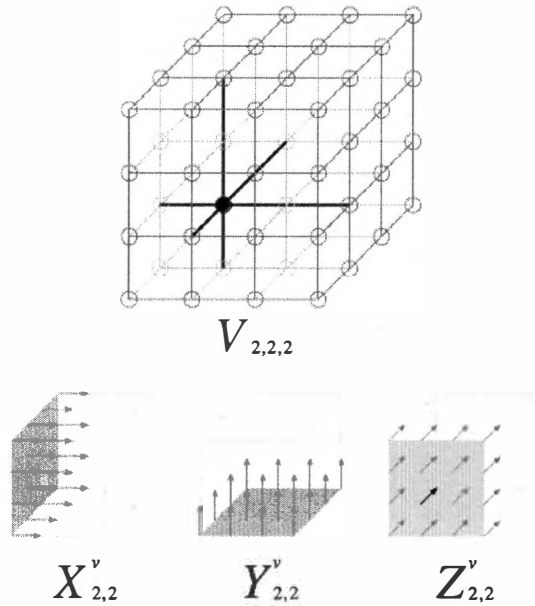
Our algorithm modifies the constraint of always placing a vertex at the median between two voxels by allowing the vertices to move along the edges between the voxels. This movement allows the surface to adapt to the shape of the input models, and provides a much smoother iso-surface as an output.

The inclusion of confidence information allows our algorithm to choose the best location to place a vertex between two voxels. In addition it allows us to intelligently texture the resulting model using stored images.

Texturing the model provides a level of realism in the resulting model. The ability to map multiple images on to a single reconstructed object means that occlusions from a particular camera can be overcome. By interacting with the confidence information we can chose the best texture available for any given triangle.

In our algorithm the threshold value for occupancy is a positive number resulting from the updated voxel values from the Volume Grid Data Capture Algorithm. An illustration of how the individual rays interact to yield the voxel occupancy can be seen in Figure 11. Also note that the edge information is saved by each of these rays as well, and is used to drive the extensions to the MCA.

Our implementation of the MCA incorporates confidence information from the multiple viewpoints to select the best intersection value and texture coordinates using the described adaptive method. To illustrate this selection process we refer to Figure 12, where one eight-voxel "cube" is displayed, and Table 1, where the confidence information is summarized. In this example, only two voxels ($V_{i,j,k}$ and $V_{i+1,j,k}$) are filled (i.e., have positive occupancy values), their neighbors are unoccupied (i.e., have zero or negative occupancy values). These values, when indexed into the MCA, indicate that two triangles (ABC and CBA) should be created with vertices along the cube edges indicated. To find the exact positions of the vertices and the appropriate



$$V_{i,j,k} = \sum_{v=1}^{N} X^{v}_{j,k}(i) + Y^{v}_{i,k}(j) + Z^{v}_{i,j}(k) \quad \text{where X, Y, Z = +3, 0, -1 for each view v.}$$

Figure 11: The voxel occupancy value is the summation of the values from the individual rays in the x, y, and z directions. This process is repeated for each input range mesh.

texture coordinates for the triangles, the appropriate edge data structures must be examined. As an illustrative example, we suppose that the number of views that are being integrated for the triangles is $N = 4$.

Each vertex position is taken from the view with the highest confidence value for that edge. Therefore, in the example of Table 1, the position of vertex A, along edge $Y^{v}_{i,k}(j)$, would be given by the intersection value for view 1, as the confidence value

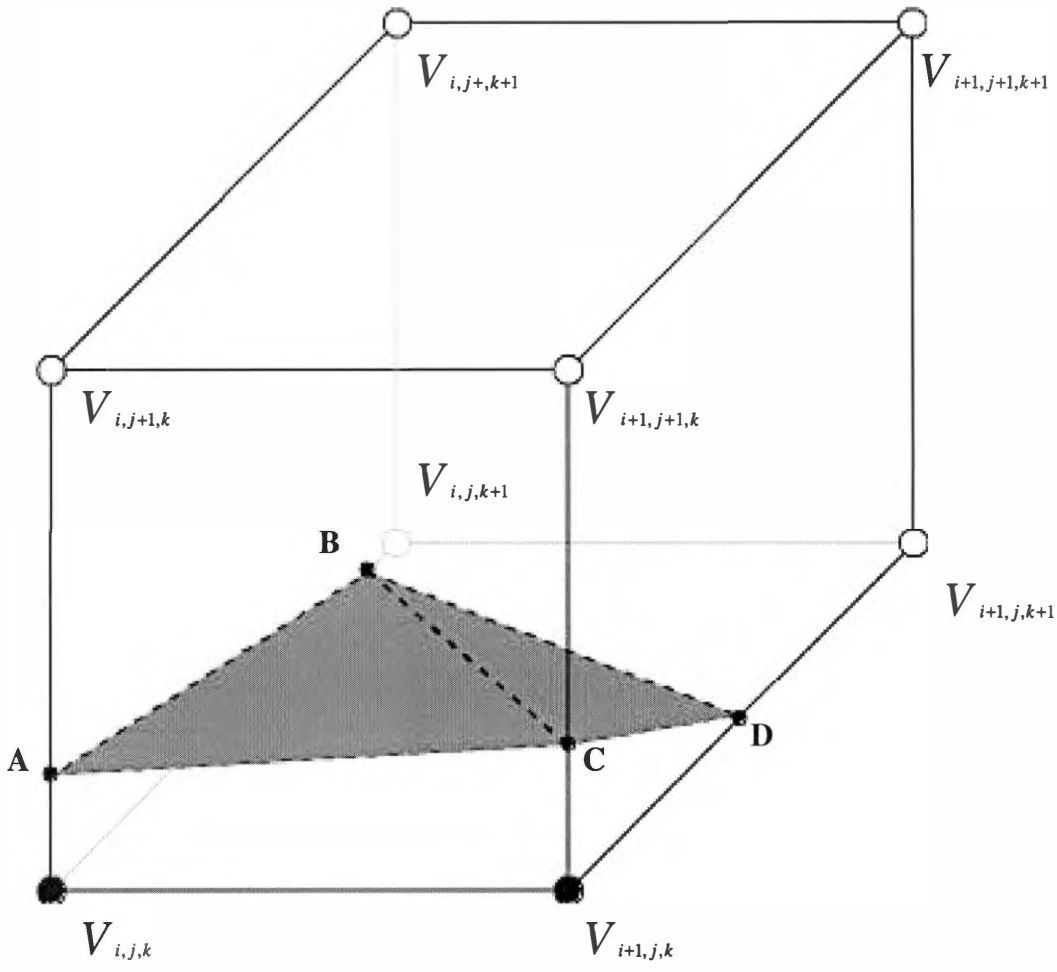


Figure 12: An example cube formed by eight voxels. The vertices A, B, C, and D are used to form the triangles signified by the dashed lines.

Table 1: Confidence information from edge data structures corresponding to the two triangles being constructed.

View v	Vertex A $Y^v_{i,k}(j)$	Vertex B $Z^v_{i,j}(k)$	Vertex C $Y^v_{i+1,k}(j)$	Vertex D $Z^v_{i+1,j}(k)$	Triangle ABC	Triangle BCD
1	90	90	20	-	200	-
2	30	50	70	60	150	180
3	-	-	-	80	-	-
4	-	-	-	-	-	-

of 90 is highest. By this same rationale the position of vertex B is determined by view 1, vertex C by view 2, and vertex D by view 3. Finally the texture coordinates for the triangles are taken from the single view with the highest total confidence that contains information for all three vertices. In the example of Table 1, the texture coordinates for triangle ABC would be taken from view 1, and the texture coordinates for triangle CBD would be taken from view 2. We note that it is possible that no single view contains information for all three vertices. In this case, the triangle is labeled unknown and is not textured. Examples of these unknown triangles will appear in Chapter 5 and we will make further mention of them in the Chapter 6.

4.3 Summary of the 3D Object Model Reconstruction Algorithms

We have discussed the algorithms necessary to generate high quality textured 3D models using a volumetric method when given multiple range and texture pairs. The Volume Grid Data Capture Algorithm is used to reconstruct the raw data into a volumetric format. The Volume Grid 3D Model Builder then takes this volumetric information and creates a 3D object model that uses the initial thermal scans as texture maps.

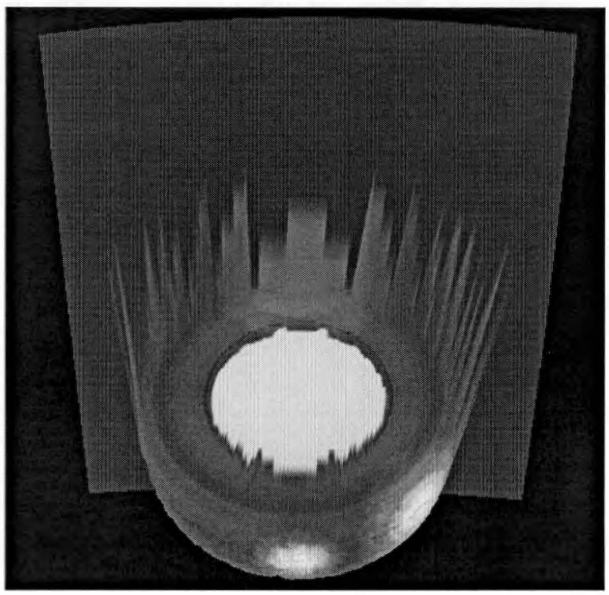
5 Experimental Results

In Section 5.1, we show the progress of our algorithms by displaying the incremental results obtained by incrementally updating our volume grid. In Section 5.2, we display the complete results of the volumetric reconstruction algorithms using differing numbers of views and volume grid resolutions. In Section 5.3, we compare our volumetric reconstruction algorithm with an advanced mesh-based algorithm. We conclude in Section 5.4 by summarizing the results.

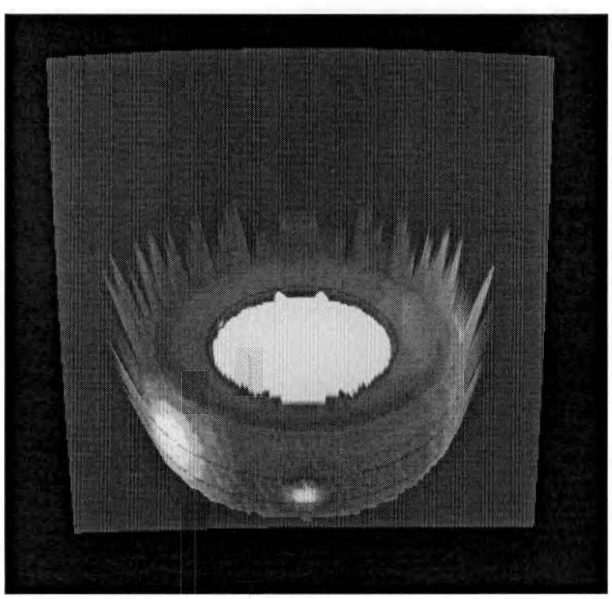
In the figures for this chapter, light gray texturing on a reconstructed 3D model indicates the absence of adequate texture data. This lack of data can be caused by the triangle being labeled as unknown, as noted in Chapter 4, or by the lack of simulated thermal data (in empty regions, for instance). The range scans in this chapter are 64 x 64, the thermal images are 256 x 256.

5.1 Incremental Steps in the Reconstruction Process

In order to accurately display the incremental steps of the reconstruction process we must provide some example stages of the reconstructed 3D model. In order to illustrate this process we show the reconstruction after the integration of two different viewpoints (see Figure 13). These viewpoints are simulated using our thermal tire model, and contain both the simulated range and thermal information. There is a



(a)



(b)

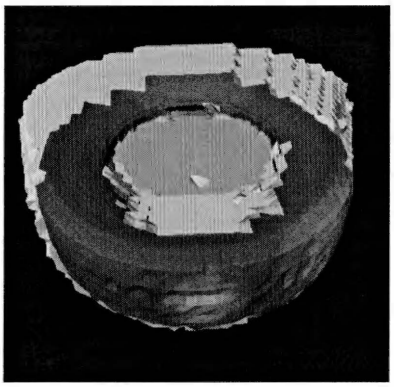
Figure 13: Surfaces constructed by triangulating the range images from two viewpoints, shown with the registered thermal data as the texture map. (a) View 1. (b) View 2.

significant amount of overlap between the two views, as our algorithms use the redundant information to increase the accuracy of the reconstructed 3D model. In Figure 14, we show the reconstructed 3D model after integrating only the view in Figure 13(a). In Figure 15, we show the reconstructed 3D model after integrating both of the views shown in Figure 13.

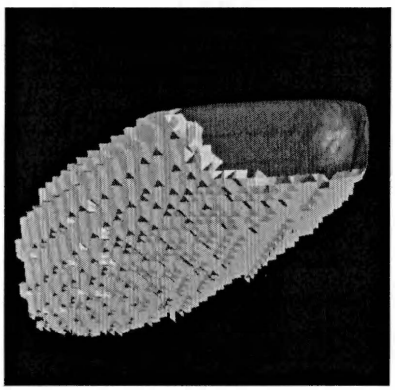
Both of the reconstructed 3D models are shown from the front, side, and top to give the viewer a better feel for the 3D nature of the models. The steps should also give a better idea of how the reconstruction process etches away the unknown areas of the volume grid. In addition, the sequence shows the replacement of low confidence areas with information of higher confidence in areas along the sidewall of the tire.

5.2 Final Results of Reconstructed Models

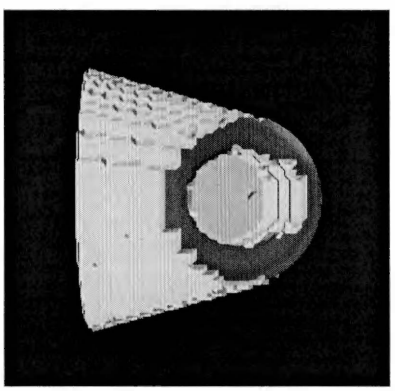
The ultimate goal of the reconstruction process is to recover a 3D model that is fully textured with thermal information of the highest possible confidence. The 3D model that is generated is highly dependant on the number of views used in the reconstruction process and the resolution of the volume grid. The number of views used for a full reconstruction need to provide at least a minimal amount of overlap, but better results are typically acquired with the use of more views.



(a)

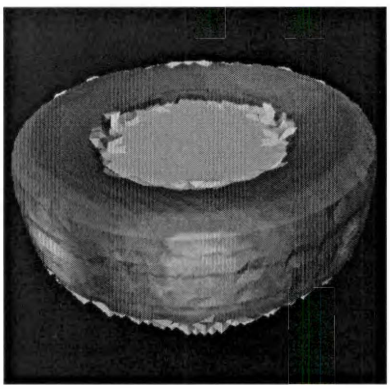


(b)

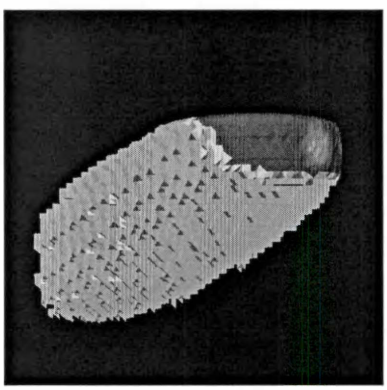


(c)

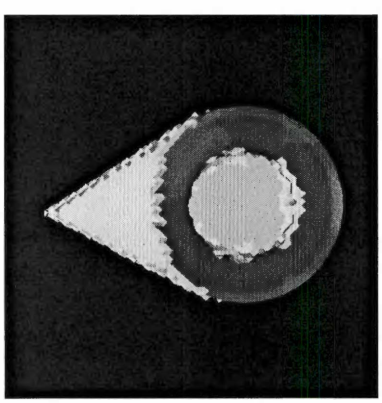
Figure 14: Three views of the surface produced from the volumetric reconstruction algorithm using only the single view from Figure 13(a). (a) Front view. (b) Side view. (c) Top view.



(a)



(b)



(c)

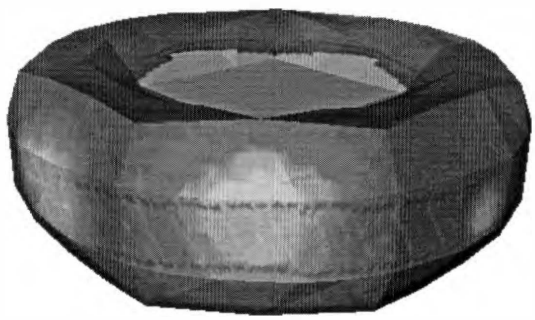
Figure 15: Three views of the surface produced from the volumetric reconstruction algorithm after integrating the two views from Figure 13. (a) Front view. (b) Side view. (c) Top view.

Volume grid resolution can be increased to provide more detailed geometry resolution in the reconstructed 3D model. A drawback to increasing the volume grid resolution is that more views are typically required to provide a completely textured model.

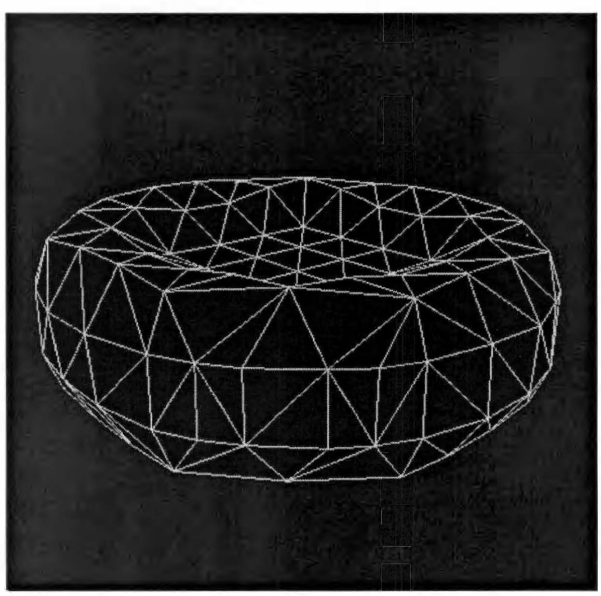
The results for a reconstruction using eight views are shown in Figures 16 – 18. Figure 16 displays the results using a volume grid resolution of M = 16 (16 x 16 x 16), Figure 17 uses M = 32, and Figure 18 uses M = 64. Increasing the number of views to sixteen yields Figures 19 – 21. The same sequence of volume grid resolutions are shown; Figure 19 has M = 16, Figure 20 has M = 32, and Figure 21 has M = 64. Of particular note is the difference between Figures 18 and 21. The increased number of views allows many of the unknown triangles to be properly textured.

5.3 Comparison with a Mesh Based Approach

To demonstrate the merits of our volumetric approach, we compare our results to models reconstructed using the mesh based integration approach proposed by Sun *et al.* [20]. To accurately determine how well both sets of reconstructed models represent the original model, we use the Metro mesh comparison software by Cignoni *et al.* [27]. Metro compares the differences between a pair of surfaces using a surface sampling approach and is well recognized in the literature as a reasonable tool for comparing mesh accuracy. Of particular benefit is that Metro makes no assumption on the particular approach used to generate the reconstructed 3D model.

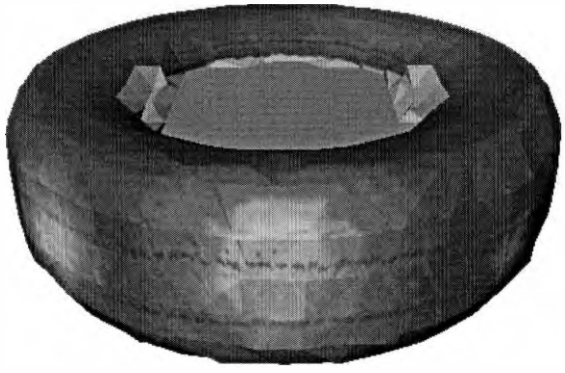


(a)

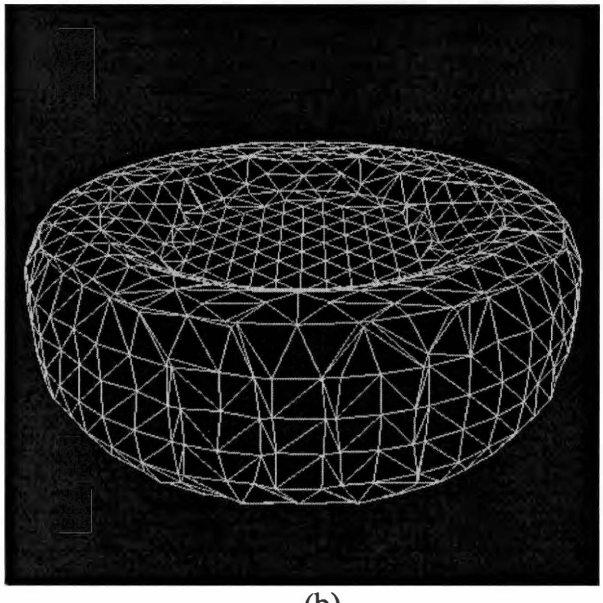


(b)

Figure 16: Model reconstructed by integrating $N = 8$ views using an $M = 16$ (16 x 16 x 16) volume grid. (a) Surface textured with thermal data. (b) Wireframe model.

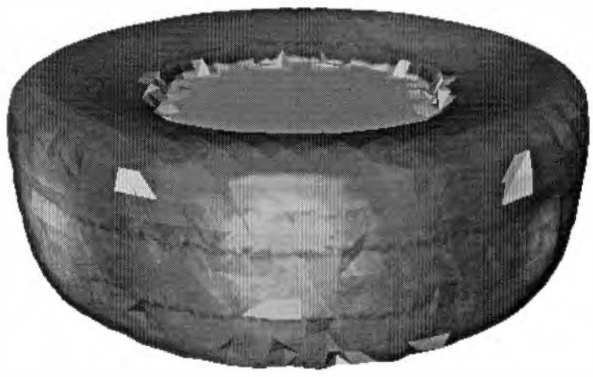


(a)

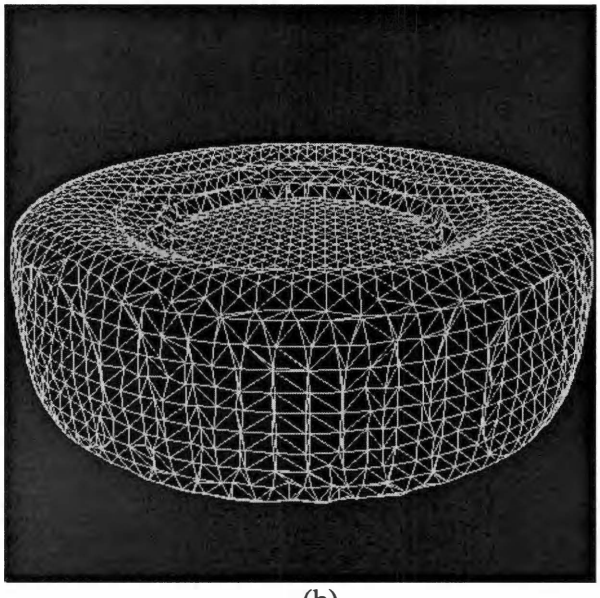


(b)

Figure 17: Model reconstructed by integrating $N = 8$ views using an $M = 32$ (32 x 32 x 32) volume grid. (a) Surface textured with thermal data. (b) Wireframe model.

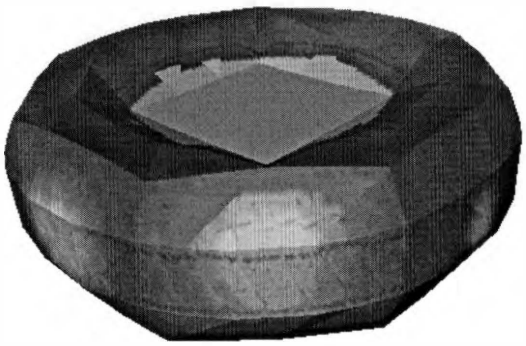


(a)

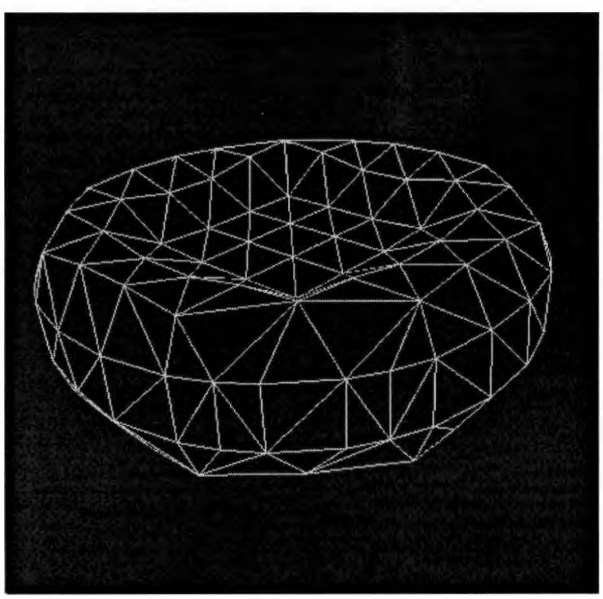


(b)

Figure 18: Model reconstructed by integrating $N = 8$ views using an $M = 64$ (64 x 64 x 64) volume grid. (a) Surface textured with thermal data. (b) Wireframe model.

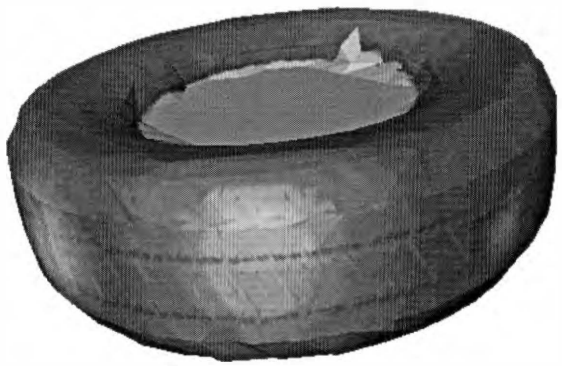


(a)

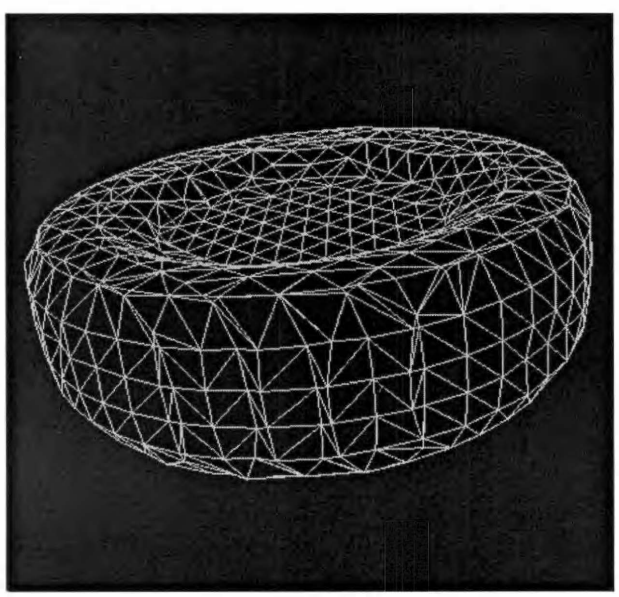


(b)

Figure 19: Model reconstructed by integrating $N = 16$ views using an $M = 16$ (16 x 16 x 16) volume grid. (a) Surface textured with thermal data. (b) Wireframe model.



(a)

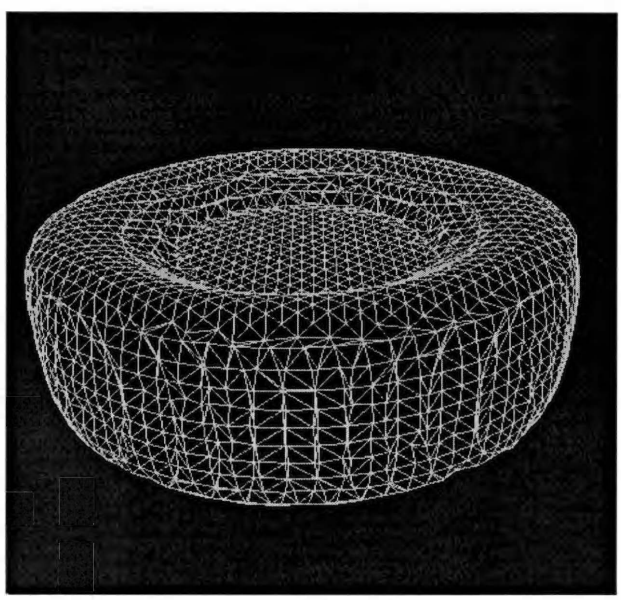


(b)

Figure 20: Model reconstructed by integrating $N = 16$ views using an $M = 32$ (32 x 32 x 32) volume grid. (a) Surface textured with thermal data. (b) Wireframe model.



(a)



(b)

Figure 21: Model reconstructed by integrating $N = 16$ views using an $M = 64$ (64 x 64 x 64) volume grid. (a) Surface textured with thermal data. (b) Wireframe model.

In Figure 22, the percentage of the mean error is shown for the reconstructed models using eight views. As can be seen, the mean error is reduced as the resolution of the volume grid increases, which is to be expected. The mesh-based result has the best percentage for this set of reconstructed data, but the volumetric method with M = 96 comes close. Similar results can be seen in Figure 23, but in this case the volumetric method with M = 96 has the lowest percentage of mean error. It should be noted that at the higher resolutions the added redundancy of extra viewpoints improves the percentage of mean error for the volumetric method, as can be seen in Figure 24, whereas the mesh-based results are slightly worse.

More complete Metro analysis results are given in Table 2. Of particular interest are the comparisons between the volumetric method with M = 96 and the mesh based method. As can be seen, both methods generate reconstructed models with approximately the same amount of vertices and triangles (see Figure 25). The results favor the volumetric approach if the user hopes to achieve the smallest mean squared error possible. However, if the user desires the smallest maximal error, the mesh-based approach seems to be the better solution in this case.

Finally, we should note that these comparisons are based purely upon geometry and do not consider the accuracy of the texture map. A tool to quantify such multimodal accuracy is the subject of future work.

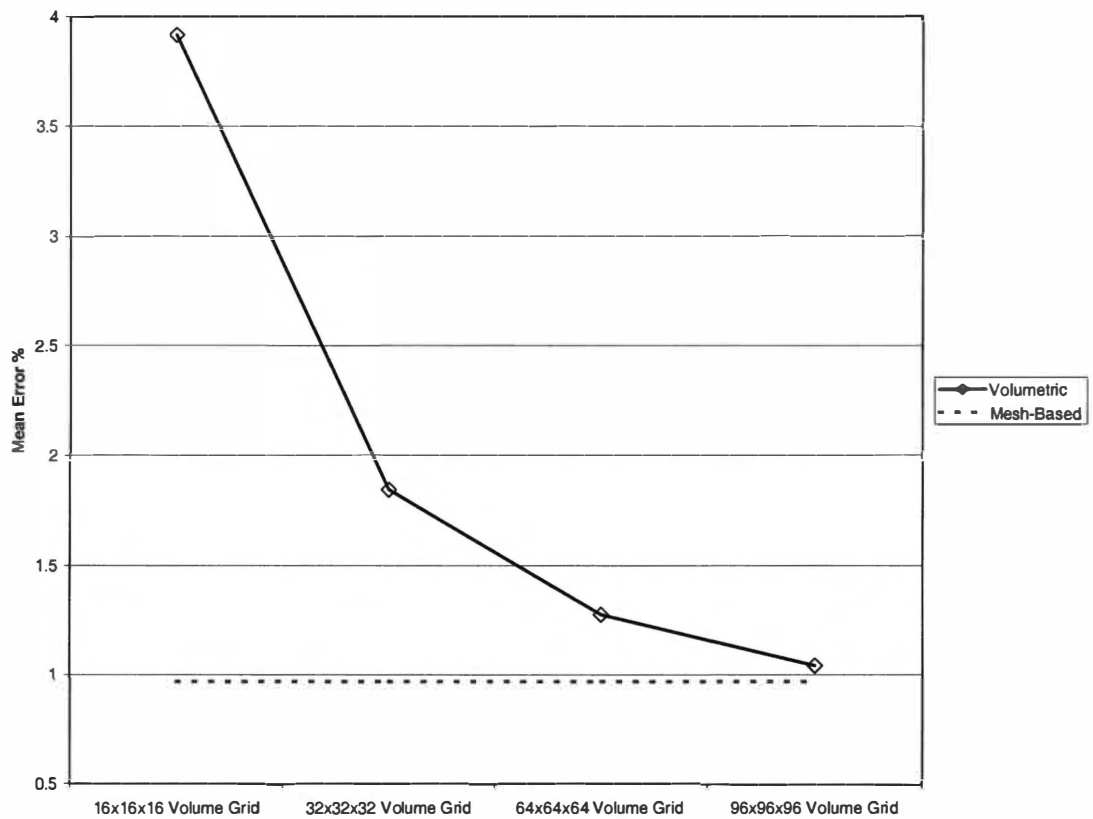


Figure 22: Mean error percentages for the volumetric and mesh-based reconstruction algorithms using eight views.

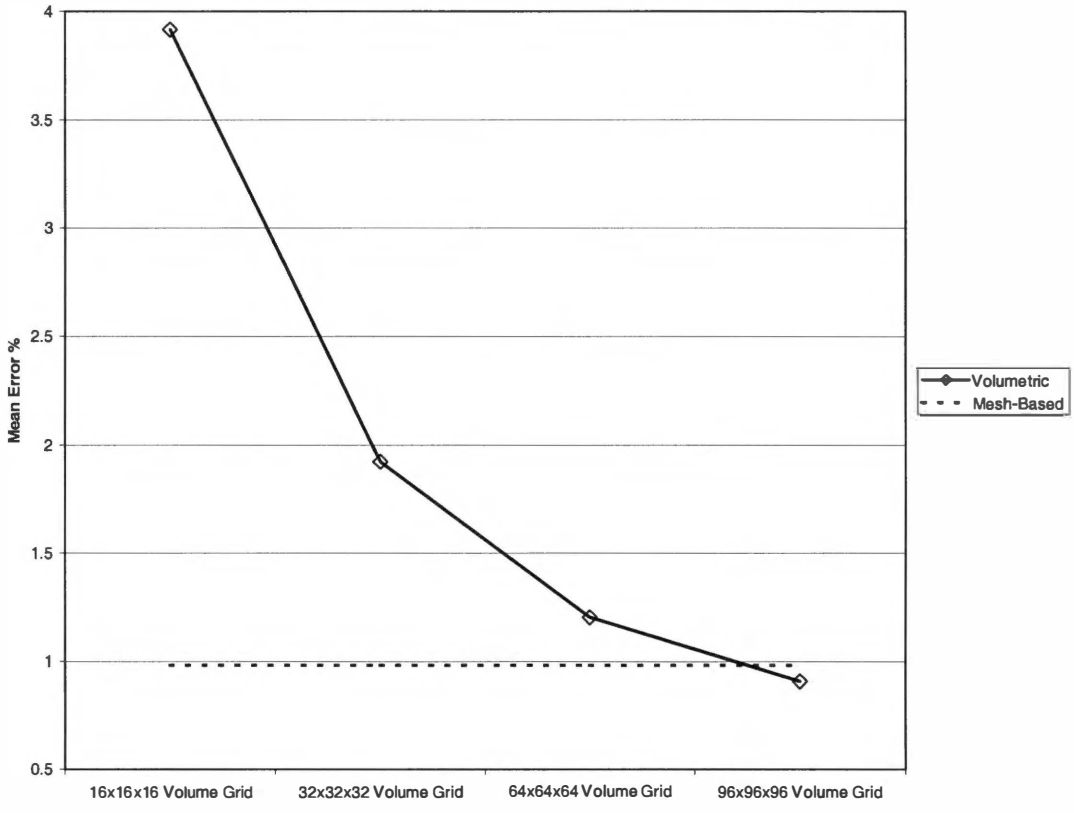


Figure 23: Mean error percentages for the volumetric and mesh-based reconstruction algorithms using sixteen views.

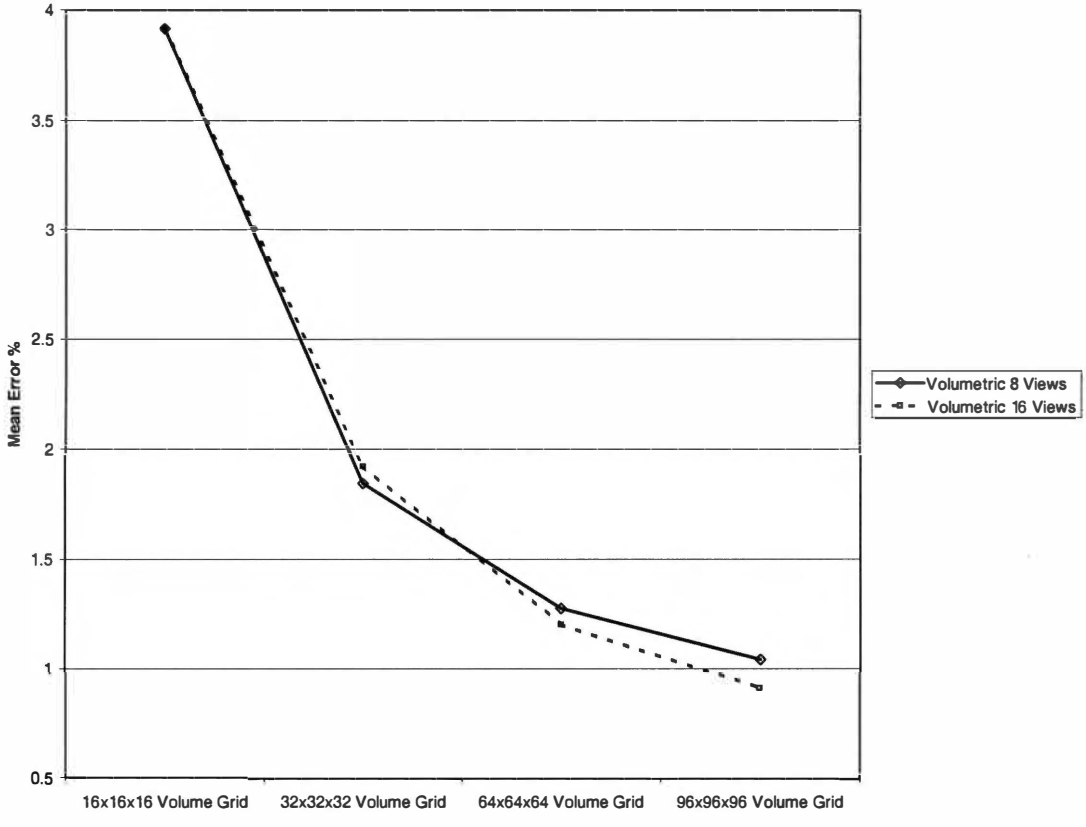


Figure 24: Mean error percentages for the volumetric reconstruction algorithm using differing numbers of views.

Table 2: Complete listing of Metro results.

	Vertices	Triangles	Maximal Error (%)	Mean Error (%)	Mean Square Error (%)
Original	18534	37032	-	-	-
16x16x16 (8 views)	168	332	10.9925	3.9142	4.5711
16x16x16 (16 views)	168	332	10.9345	3.9167	4.5734
32x32x32 (8 views)	982	1960	5.5842	1.8459	2.3101
32x32x32 (16 views)	980	1954	5.6184	1.9234	2.3672
64x64x64 (8 views)	3831	7600	4.755	1.2761	1.6164
64x64x64 (16 views)	4050	7973	4.7358	1.202	1.5102
96x96x96 (8 views)	9122	18225	4.4672	1.0429	1.4158
96x96x96 (16 views)	9734	19387	4.4639	0.9099	1.2681
Mesh-Based (8 views)	8964	17688	3.933	0.969	1.4882
Mesh-Based (16 views)	10772	21352	3.939	0.9836	1.5093

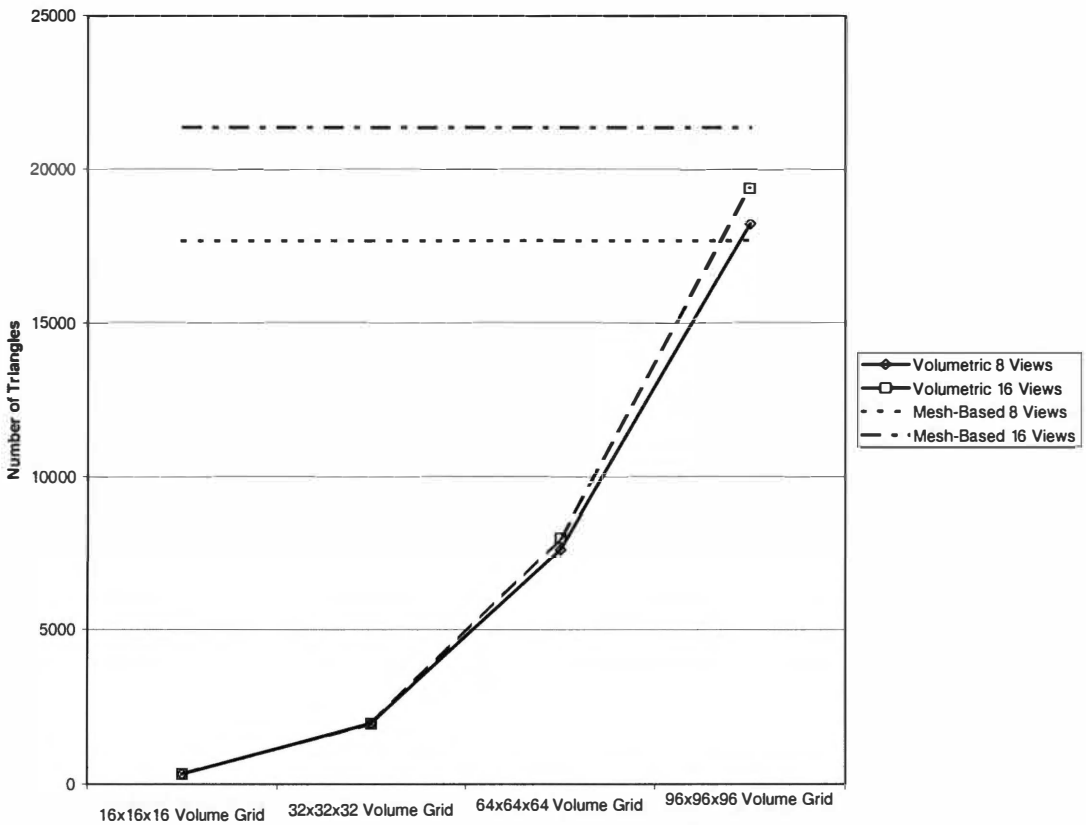


Figure 25: Number of triangles in the final reconstructed 3D models.

5.4 Conclusion of Experimental Results

We have displayed the incremental steps inherent in the volumetric reconstruction process. We have also displayed the final 3D models reconstructed from varying numbers of views and volume grid resolutions. The problems and solutions pertaining to texturing with high resolution volume grids have been illustrated. A detailed comparison with a mesh integration approach has been discussed.

6 Conclusions and Future Work

We have presented a volumetric method for building 3D, textured models from multiple range and texture image pairs. Unlike previous volumetric methods, we have explicitly considered texture (thermal data, in this case) in the reconstruction process. We have shown that this volumetric method compares favorably to a state-of-the-art, mesh-based integration approach in both resulting model size and error.

The impetus of this work was the simulation of a thermal tire inspection station. Incorporating 3D reconstruction into the thermal inspection process permits mapping the thermal data to the true tire structure, enhancing analysis capabilities, while simultaneously allowing inspection of the tire geometry to identify structural defects.

The primary drawback of the proposed algorithm, as noted previously, is that increasing the resolution of the volume grid requires increasing the number of views so that the unknown-texture triangles can be eliminated. One possible solution to this problem is to back-project the unknown-texture triangles to the available view planes. Assuming no occlusions, if the confidence of an unknown-texture triangle is high with respect to some view (i.e., the triangles normal is nearly parallel to the view direction), we should be able to map the texture from

that view to the triangle. If there are occlusions in the object this approach will not work, and a much more involved back-projection method must be considered. A second possible solution would be to analyze the distribution of points in the 2D texture space. In other words, the texture coordinates could be found by interpolating between the neighbor texture coordinate points in the 2D space. The unknown-texture triangle problem is the subject of future work.

Another goal of future work concerns triangles able to be textured from multiple views. These triangles could benefit from the data fusion approach wherein the textures from the multiple views are combined before being applied to the triangles. The confidence information could be used as a weighting mechanism for this texture combination. This methodology could be used to improve the results of this research.

[1] E. Catmull, "A Subdivision Algorithm for Computer Display of Curved Surfaces." Ph. D. thesis, Dept. of C. S., U. of Utah, (1974).

[2] W. Lorensen, H. Cline, "Marching Cubes: a High Resolution 3D Surface Reconstruction Algorithm," in *Proc. SIGGRAPH*, 163-169 (1987).

[3] A. Hilton, A. Stoddart, J. Illingworth, T. Windeatt, "Marching Triangles: Range Image Fusion for Complex Object Modeling," in *Proc. IEEE Int'l Conf. Image Processing*, 2, 381-384 (1996).

[4] C. Connolly, "Cumulative Generation of Octree Models From Range Data," in *Proc. Int'l. Conf. on Robotics and Automation*, 25-32 (1984).

[5] K. Pulli, T. Duchamp, H. Hoppe, J. McDonald, L. Shapiro, W. Stuetzle, "Robust Meshes from Multiple Range Maps," in *Proc. Int'l Conf. on Recent Advances in 3-D Digital Imaging and Modeling*, 205-211 (1997).

[6] B. Curless, M. Levoy, "A Volumetric Method for Building Complex Models from Range Images," in *Proc. SIGGRAPH*, 303-312, (1996).

[7] R. Koch, M. Pollefeys, L. Van Gool, "Realistic 3-D Scene Modeling from Uncalibrated Image Sequences," in *Proc. IEEE Int'l Conf. Image Processing*, 2, 500-504 (1999).

[8] M. Pollefeys, R. Koch, M. Vergauwen, B. Deknuydt, L. Van Gool, "Three-Dimensional Scene Reconstruction From Images," in *Proc. of the SPIE*, 3958, 215-226, (2000).

[9] R. Koch, M. Pollefeys, L. Van Gool, "Realistic Surface Reconstruction of 3D Scenes from Uncalibrated Image Sequences," in *Journal of Visualization and Computer Animation*, 11(3), 115-127 (2000).

[10] A. Mangan, "Photo-Realistic Surface Reconstruction," Master's Thesis, Univ. of Tennessee, Knoxville, (1999).

[11] K. Pulli, M. Cohen, T. Duchamp, H. Hoppe, L. Shapiro, W. Stuetzle, "View-Based Rendering: Visualizing Real Objects from Scanned Range and Color Data," in *Rendering Techniques '97, Proc. of the Eurographics Workshop*, 328, 23-34, (1997).

[12] K. Pulli, L. Shapiro, "Surface Reconstruction and Display from Range and Color Data," in *Graphical Models*, 62(3), 165-201 (2000).

[13] P. Eisert, E. Steinbach, B. Girod, "Multi-Hypothesis, Volumetric Reconstruction of 3-D Objects from Multiple Calibrated Camera Views," in *IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, 6, 3509-3512, (1999).

[14] P. Eisert, E. Steinbach, B. Girod, "Automatic Reconstruction of Stationary 3-D Objects from Multiple Uncalibrated Camera Views," in *IEEE Trans. on Circuits and Systems for Video Technology*, 10(2), 261-277, (2000).

[15] E. Steinbach, B. Girod, P. Eisert, A. Betz, "3-D Object Reconstruction Using Spatially Extended Voxels and Multi-Hypothesis Voxel Coloring," in *Proc. 15th Int'l Conf. on Pattern Recognition, IEEE Comput. Soc.*, 1, 774-777, (2000).

[16] G. Turk, M. Levoy, "Zippered Polygon Meshes from Range Images," in *Proc. SIGGRAPH*, 311-318, (1994).

[17] M. Soucy, G. Goudin, R. Barineau, F. Blais, M. Rioux, "Sensors and Algorithms for the Construction of Digital 3-D Colour Models of Real Objects," in *Proc. IEEE Int'l Conf. Image Processing*, 3, 409-412, (1996).

[18] M. Soucy, D. Laurendeau, "A General Surface Approach to the Integration of a Set of Range Views," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 17(4), 344-358, (1995).

[19] R. Pito, "Mesh Integration Based on Co-Measurements," in *Proc. IEEE Int'l Conf. Image Processing*, 2, 397-400 (1996).

[20] Y. Sun, C. Dumont, M. Abidi, "Mesh-Based Integration of Range and Color Images," in *Sensor Fusion: Architectures, Algorithms, and Applications IV, Proc. SPIE*, 4051, 110-117, (2000).

[21] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, W. Stuetzel, "Piecewise Smooth Surface Reconstruction," in *Proc. SIGGRAPH*, 295-302, (1994).

[22] D. Perard, J. Beyerer, "Three-Dimensional Measurement of Free-Form Surfaces with a Structured Lighting Reflection Technique," in *Machine Vision Application in Industrial Inspection, Proc. SPIE*, 3204, 74-80, (1997).

[23] D. Aluze, C. Coulot, F. Meriaudeau, P. Gorria, C. Dumont, "Machine Vision for the Control of Reflecting Non-Plane Surface," *Journal of the Machine Vision Association, SME*, 14(3), 1-4, (1998).

[24] C. Coulot, S. Kohler-Hermmerlin, C. Dumont, D. Aluze, B. Lamalle, "Simulations of Lighting for an Optimal Defect Detection by Artificial Vision," *IEEE/ASME Int'l Conf. on Advanced Intelligent Mechatronics*, 35, (1997)

[25] R. Gonzales, P. Wintz, Digital Image Processing, Addison-Wesley (1992).

[26] L. Wong, C. Dumont, M. Abidi, "A Next Best View Algorithm for Object Reconstruction," in *Sensor Fusion and Decentralized Control in Robotic Systems, Proc. SPIE*, 3523, 191-200, (1998).

[27] P. Cignoni, C. Rocchini, R. Scopigno, "Metro: Measuring Error on Simplified Surfaces," in *Computer Graphics Forum*, 17(2), 167-174, (1998).

[28] D. Hearn, M. Baker, Computer Graphics, Prentice Hall (1994).

Appendix

Software Usage

Here are instructions on how to use the software that was developed in conjunction with this research. Both of these executables require the SGI Open Inventor Toolkits. They were designed to run in batch mode, as processing took a while on larger range scans and input models.

The first executable, range, is the range and texture simulator. It may be run by the following command:

range [rres] [ires] [focal] [x] [y] [z] [name]

where:

rres = range resolution, one integer shared as both the X and Y resolution
ires = image resolution, one integer shared as both the X and Y resolution
focal = focal length, one floating point
x, y, z = camera location, three floating point numbers
name = name of the file to output to (.iv and .rgb extensions will be added)
The object to be scanned is assumed to be centered at the origin.

The second executable, volgrid, constructs the 3D model from the individual range and texture scans provided by range. It may be run by the following command:

volgrid [bound] [size] [input]

where:

bound = size of the bounding box that should contain the model, one integer. The resulting bounding box will be +bound, -bound in the X, Y, and Z directions
size = size of the volume grid, one integer. The number of voxels in the grid is size * size * size
input = list of all [name] files that the user wishes to integrate.

Vita

William Harvey Gray Jr. was born in Asheville, North Carolina in 1974. He grew up in Oak Ridge, Tennessee and attended high school there, graduating in 1992. He then attended Vanderbilt University graduating in 1996 with a Bachelor of Engineering, Electrical Engineering. Upon the acceptance of this thesis Harvey will have completed his Master's degree in Electrical Engineering from the University of Tennessee, Knoxville. His primary focus during his stay at UT has been in the areas of machine vision and scene reconstruction.

8774 5976 7
03/05/03 MRB