

Integration of multiple range and intensity image pairs using a volumetric method to create textured three-dimensional models

W. Harvey Gray
Jeffery R. Price

University of Tennessee
Department of Electrical and Computer Engineering
IRIS Laboratory
Knoxville, Tennessee 37996-2100

Christophe Dumont
University of Burgundy
Le2I Laboratory
IUT Le Creusot
Le Creusot, 71200 France

Mongi A. Abidi*
University of Tennessee
Department of Electrical and Computer Engineering
IRIS Laboratory
Knoxville, Tennessee 37996-2100
E-mail: abidi@utk.edu

Abstract. We present a volumetric approach to three-dimensional (3D) object modeling that differs from previous techniques in that both object texture and geometry are considered in the reconstruction process. The motivation for the research is the simulation of a thermal tire inspection station. Integrating 3D geometry information with two-dimensional 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 were 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. © 2001 SPIE and IS&T. [DOI: 10.1117/1.1329340]

1 Introduction and Background

The reconstruction of high-detail three-dimensional (3D) scenes and object models is a major goal of current research.¹⁻⁴ Such scenes and models can be useful in machine vision applications and various visualization tasks. The motivation for the work presented here is the simula-

tion of a thermal tire inspection station. As the equipment involved in machine vision applications can be fragile, expensive, and/or cumbersome, simulation is often a very important component when constructing a machine vision system. 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.

In the reconstruction algorithm we present, multiple range and thermal image pairs from different viewpoints are integrated to form a complete 3D tire model, with the thermal data applied as a texture map. The geometry of the surface is determined by a discrete-state voxel grid, where a voxel is either occupied (i.e., on the surface), empty, or unknown (i.e., inside the surface). Much previous research,¹⁻¹¹ including some volumetric approaches,^{2,5-8} has focused on integrating multiple range images to construct 3D models. Of these many approaches, only a few^{1,2,4} have considered surface texture explicitly, and of those, only Koch *et al.*² employ a volumetric approach. Our approach differs from that of Koch *et al.* in that we use confidence information to guide the texture selection in addition to the geometry estimation.

The remainder of this paper is organized as follows. In Sec. 2, we discuss the simulation environment and present the evolution of the thermal tire imaging simulator from

*Author to whom correspondence should be addressed.

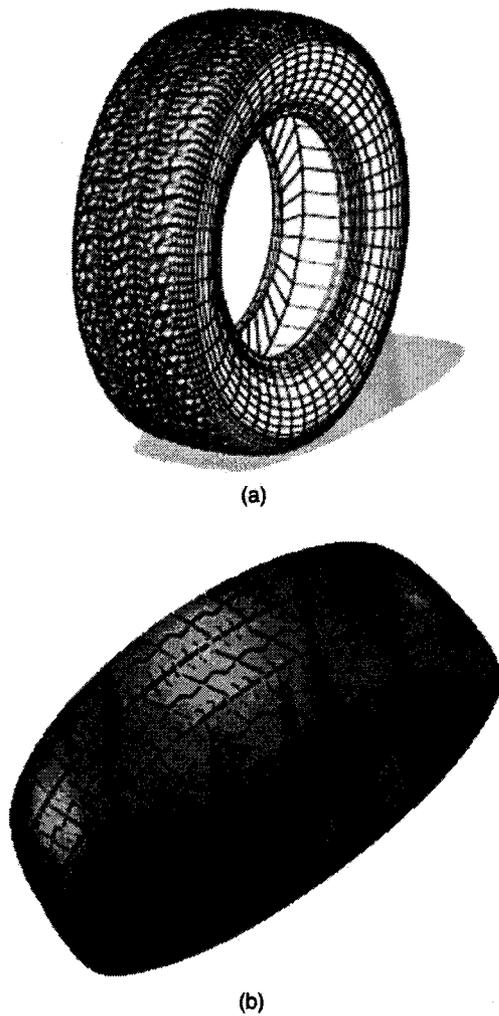


Fig. 1 Tire model used in simulations. (a) Detailed wire-frame model. (b) Model with artificial thermal data applied as a texture map.

two-dimensional (2D) to 3D. In Sec. 3, we describe the operation of our 3D reconstruction algorithm in detail, using some simple, illustrative examples. Then, in Sec. 4, we provide some example results of the algorithm applied to reconstructing tire models in the simulation system. In Sec. 5, we compare our results to those from a state-of-the-art mesh-based approach.¹ We close in Sec. 6 with some concluding remarks and note the direction of some future research.

2 Simulation Environment

The first key component in the simulation environment is the tire model. A detailed wire-frame tire model is shown in Fig. 1(a). Artificial thermal data are applied to this tire model as a texture map, as shown in Fig. 1(b). Initially, we place the tire in an environment simulating an endurance drum (dynamometer) test machine, as illustrated in Fig. 2. 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. A thermal camera model, based on the ubiquitous pinhole model,¹² is placed into this environment to image

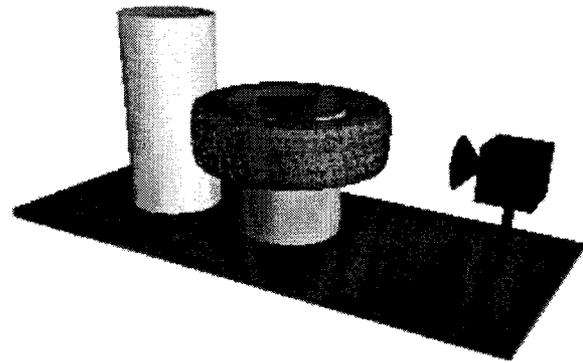


Fig. 2 Coarse mock-up of 2D thermal inspection station illustrating an endurance drum (dynamometer), the tire model, and the thermal camera.

the tire's thermal profile. To demonstrate, we show three different thermal images of the tire model in Figs. 3 and 4.

Although the 2D thermal imaging system described above can provide good results, several assumptions must be made if we intend to visualize the thermal images on a 3D model. First, we must assume that a sufficiently detailed model is available. Additionally, we must also assume that the model is indeed accurate and that the actual tire being tested does not deviate significantly from the model due to manufacturing irregularities or defects. For these reasons, we propose to reconstruct the tire geometry during the inspection process. We accomplish this by capturing and integrating range and thermal data from several viewpoints. In Fig. 5, we show an example of range and thermal data captured from the same viewpoint. Note that the range data in Fig. 5(a) have been triangulated for easy visualization and rotated slightly to highlight the fact that it is a depth map. To construct a 3D tire model with the thermal images applied as texture, we must integrate multiple pairs of images, such as the pair in Fig. 5, from several different viewpoints. This integration is the subject of the following section.

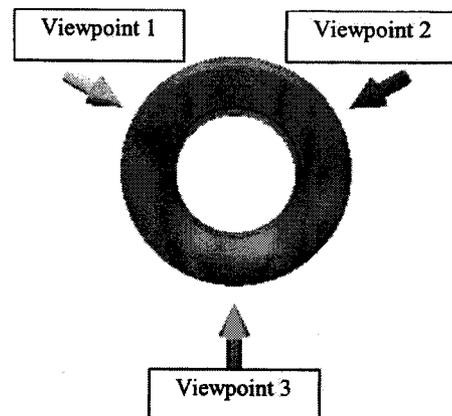
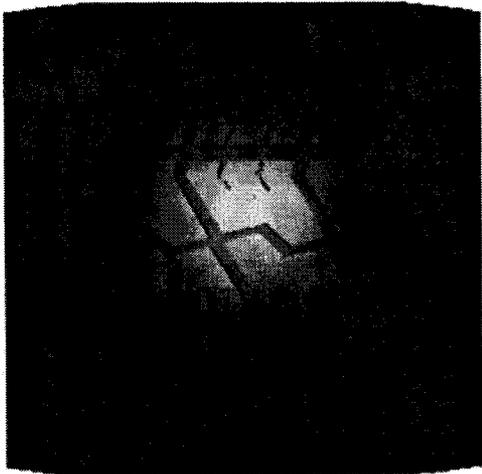


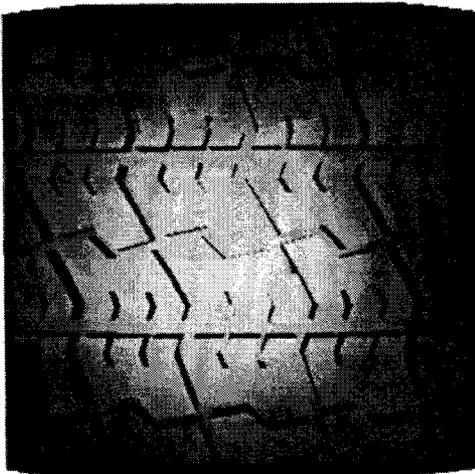
Fig. 3 Illustration of multiple views to be imaged with the thermal camera model in the simulation environment of Fig. 2.



(a)

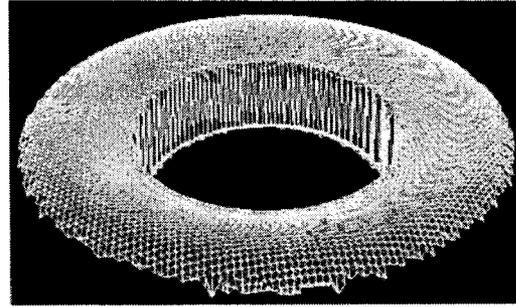


(b)

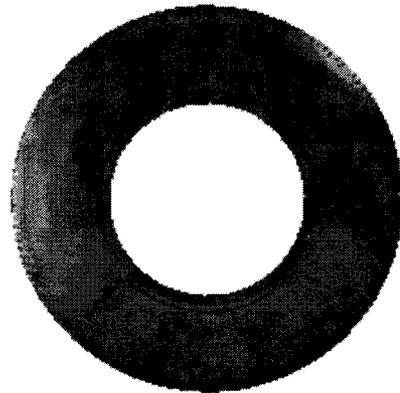


(c)

Fig. 4 Simulated thermal images captured from the views shown in Fig. 3 using a pinhole thermal camera model and the tire model from Fig. 1. (a) view 1, (b) view 2, and (c) view 3.



(a)



(b)

Fig. 5 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 the depth. (b) Simulated thermal data.

3 3D Reconstruction

In this section, we describe our volumetric reconstruction algorithm. After initializing the voxel grid to an $M \times M \times M$ array of zeros, the reconstruction algorithm, as outlined in the pseudocode of Fig. 6, comprises two primary steps: (1) data capture and storage, and (2) triangulation and texture mapping. The details of these two steps are described in the following subsections.

3.1 Data Capture and Storage

A set of N range and thermal image pairs is to be captured, where both the number of views N and the view locations are specified *a priori*. We assume that the parameters necessary to register the range and thermal images are known. As we are simulating a well-defined machine vision system, which will be imaging known objects (tires), the *a priori* specification of N , the viewpoints, and the registration parameters are not a significant restriction. We note, however, that our reconstruction algorithm employs an occupancy grid approach similar to that found in the next-best-view (NBV) work of Wong *et al.*¹³ Adopting our approach to employ the NBV algorithm would permit the

Data Capture and Storage.

- For each view $v = 1, \dots, N$:
 - Capture range image R^v .
 - Capture thermal image T^v .
 - Triangulate R^v .
- For each axis $d = x, y, z$:
 - Cast orthographic rays along voxel scan lines in d direction.
 - Update voxel grid according the ray/surface intersections.
 - For every surface intersection, store the scale, confidence, and texture coordinates in the edge data structure E_d^v .

Triangulation and Texture Mapping.

- For each triangle to be constructed with adaptive marching cubes:
 - Get corresponding information from the edge data structures for every view.
 - Set scales (i.e., positions of vertices) according to the view with the highest confidence for each vertex.
 - Map texture from the thermal image T^v , where v indicates the edge data structure with the highest total confidence.
 - If no single view contains information for all three vertices, then label the triangle texture unknown.

Fig. 6 Pseudocode for volumetric 3D reconstruction algorithm.

reconstruction of unknown objects, where N and the view locations are not specified ahead of time.

As noted in the pseudocode of Fig. 6, the following sequence of steps is performed for each view $v=1, \dots, N$. First, a range image R^v and thermal image T^v are captured. Next, R^v is triangulated to approximate the surface from viewpoint v . One illustration of such a surface is shown inside the bounding cube of the entire voxel grid in Fig. 7. Next, we project orthographic rays along voxel scan lines in all three directions to update the voxel occupancy status and store the voxel grid/surface intersection information. The process is illustrated for the x and y directions in Figs.

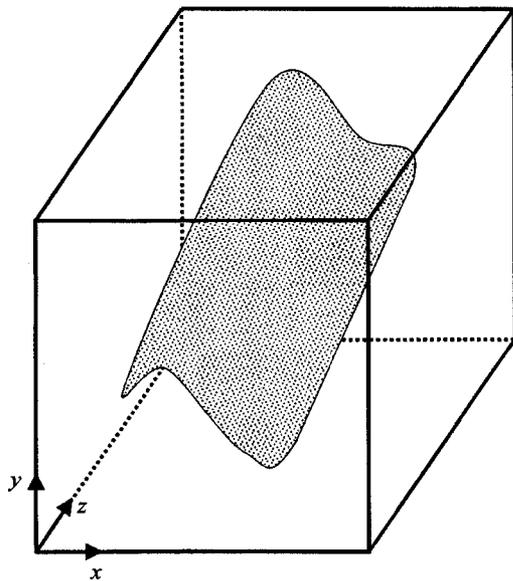


Fig. 7 Example surface obtained from one view, shown inside the bounding cube of the volume grid. The surface is approximated by triangulating the range image from the given view. The occupancy status of each voxel and the edge data structures are updated by orthographically projecting rays along each voxel scan line in the x , y , and z directions.

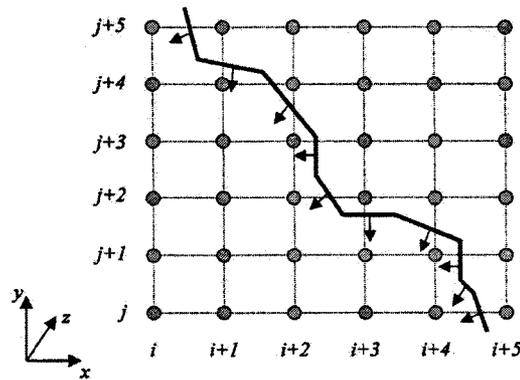


Fig. 8 An xy cross section ($z=k$) of an example triangulated surface and the corresponding region of the voxel grid. The normals of the surface triangles are indicated with the arrows.

8–12. First, in Fig. 8, an xy cross-section of an example surface is shown, with the surface normals indicated, along with a region of the voxel grid. In Fig. 9, orthographic projection along the x axis is shown. The occupancy value of empty voxels is decremented by -1 (indicated by white in Fig. 9), voxels on the opposite side of a surface intersection are incremented by $+4$ (indicated by black in Fig. 9), and the other voxels are left unchanged. Figure 10 demonstrates the same process in the y direction for the example surface. It has been found experimentally that giving more weight to the occupied voxels (i.e., those that are incremented by $+4$) helps to reduce errors in areas that have conflicting information, such as the tread pattern, which contains noise-like attributes when sampled at low resolutions.

To ensure accurate surface reconstruction and texture mapping, information characterizing the voxel grid intersections with the surface must also be stored. For the example surface of the previous figures, the x and y edges for which this information is stored are illustrated by the bold, dashed lines in Figs. 11 and 12, respectively. For each view v , and for each surface intersecting edge in direction d ,

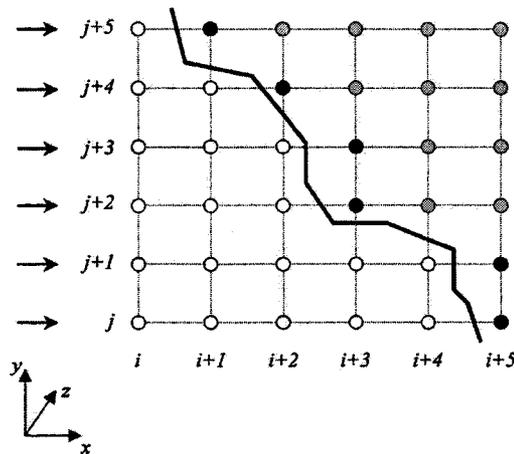


Fig. 9 Voxel occupancy update by orthographic projection along x voxel scan lines for the example surface from Fig. 8. The white voxels are decremented (-1), the black voxels are incremented ($+4$), and the gray voxels are left unchanged.

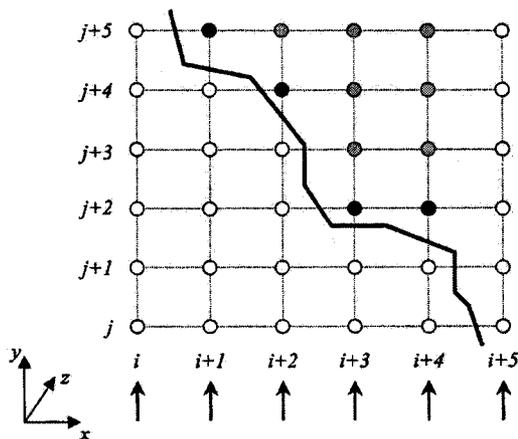


Fig. 10 Voxel occupancy update by orthographic projection along y voxel scan lines for the example surface from Fig. 8. The white voxels are decremented (-1), the black voxels are incremented ($+4$), and the gray voxels are left unchanged.

three quantities are stored in the appropriate edge data structures, denoted E_d^v . These three quantities are the intersection, confidence, and texture coordinates. The intersection is a number between 0.0 and 1.0 indicating where along the given edge the intersection occurs. When later triangulating the surface for visualization, the intersection values will determine where to place triangle vertices. The texture coordinates note where in the thermal image T^v the point of intersection lies. We adopt the prevalent practice in the literature^{6,10,11} and compute the confidence as the dot product of the intersecting triangle with the given view direction. A large positive value indicates that the triangle in question is well imaged from the given viewpoint. A negative value would indicate that the triangle is facing opposite

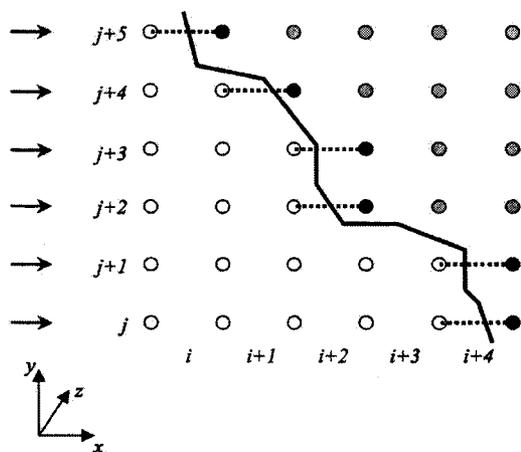


Fig. 11 Bold dotted lines indicate edges in the x direction for which the intersection, texture coordinates, and confidence are stored for the example surface. The intersection is a number between 0.0 and 1.0 indicating where along the edge the surface intersection occurs, the texture coordinates give the location of the intersecting point in the texture map (a simulated thermal image in this application) for the given view, and the confidence is given by the dot product of the intersecting triangle's surface normal with the view direction. Note that the x indices here indicate the edges between voxels rather than the voxels themselves.

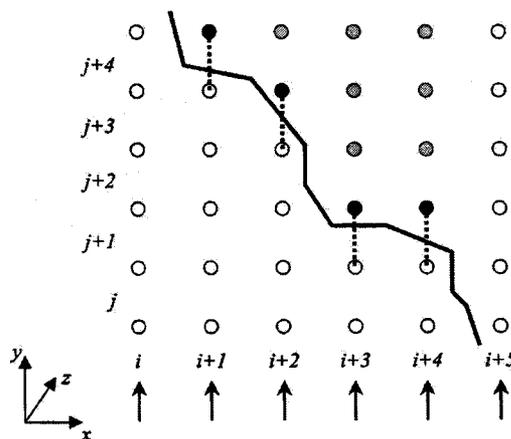


Fig. 12 Bold dotted lines indicate edges in the y direction for which the intersection, texture coordinates, and confidence are stored for the example surface. Note that the y indices here indicate the edges between voxels.

the given viewpoint. As described in Sec. 3.2 below, the confidence values will determine from which views to select the intersection values and texture coordinates.

It should also be noted that the volume grid data can be analyzed at this point, prior to rendering for visualization purposes. As the volume data lies on a regular grid, standard signal processing techniques can be applied. One other point to note is that the integration technique is independent of the rendering scheme. The results can be used with a voxel rendering scheme or by any number of algorithms used to triangulate volumetric grids.

3.2 Triangulation and Texture Mapping

Once all N views have been completed and the corresponding information stored, we would like to produce a triangulated, textured surface for visualization. We employ an adaptive implementation of the marching cubes algorithm¹⁴ that incorporates confidence information from the multiple viewpoints to select the best intersection value and texture coordinates. To illustrate this selection process we refer to Fig. 13, where one eight-voxel "cube" is displayed. In this example, only voxel (i,j,k) is filled (i.e., has positive occupancy value), its neighbors are unoccupied (i.e., have zero or negative occupancy values). This indicates that a triangle ABC should be created with vertices along the cube edges indicated. To find the exact position of the vertices and the appropriate texture coordinates for triangle ABC, the three edge data structures for each view must be examined. As an illustrative example, we suppose that the number of views is $N=8$ and that five views (1, 4, 5, 6, and 8) contain information for at least one of the edges where vertices A, B, and C lie. The corresponding confidence information is summarized in Table 1. 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 $E_x^v(i,j,k)$, would be given by the intersection value for view 6, as the confidence value of 75 is highest. Similarly, the position of vertex B, along $E_y^v(i,j,k)$, would be determined from view 6, while the position of vertex C, along $E_z^v(i,j,k)$, would be determined by view 8. Finally,

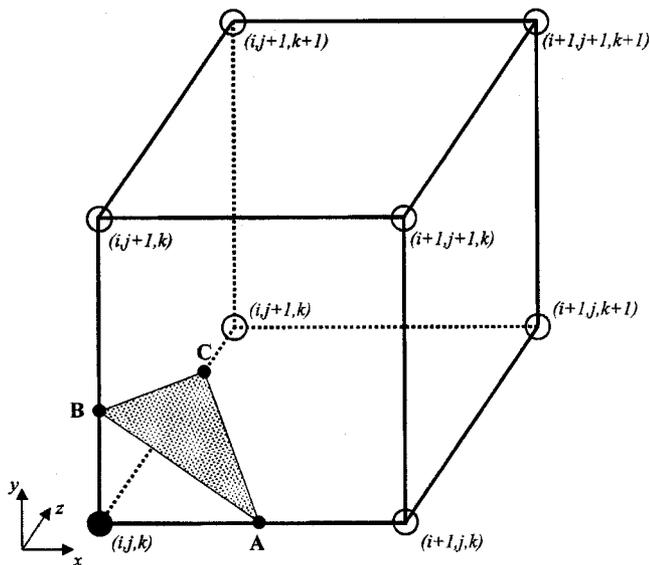
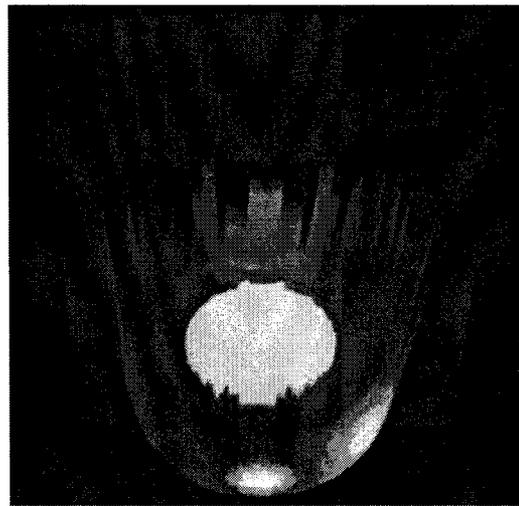


Fig. 13 Example of triangulation by adaptive marching cubes. Only voxel (i, j, k) is occupied (i.e., has positive occupancy value), the others comprising the cube are empty. To find the exact location of the three triangle vertices A, B, and C, and the corresponding texture coordinates, the edge data structures E_x^v , E_y^v , and E_z^v must be examined for each view v , as described further in Table 1. Note that, for each view v , information on vertex A is stored in $E_x^v(i, j, k)$, B in $E_y^v(i, j, k)$, and C in $E_z^v(i, j, k)$.

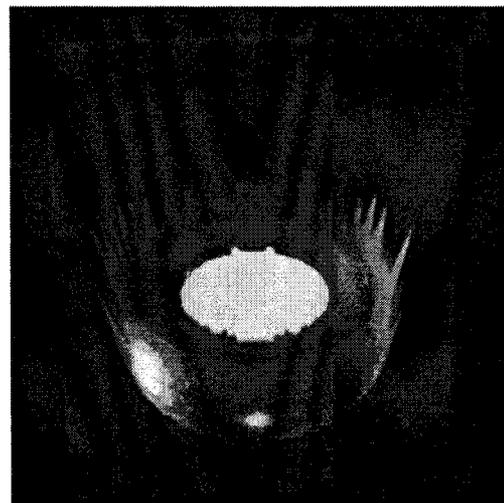
the texture coordinates for ABC are taken from the single view with the highest total confidence that contains information for all three vertices. In the example of Table 1, the texture coordinates would be taken from view 6. We note that it is possible that no single view contains information for all three vertices. In this case, the triangle texture is labeled unknown. Examples of these unknown-texture triangles will appear in Sec. 4 and we will make further mention of them in the future research portion.

Table 1 Confidence information from edge data structures corresponding to triangle ABC in Fig. 13. The confidence is computed as the dot product of the triangulated range image normal at the point of intersection with the given view direction. Here, the confidence is scaled to lie between 0 and 100 for illustration. In this example, only views 1, 4, 5, 6, and 8 have information about ABC (i.e., all the entries for all other views would be null, as indicated by dashes). The vertex positions are selected from the views with the highest corresponding confidence. In this example, vertices A and B would be determined by view 6, and vertex C by view 8. The texture coordinates are taken from the view with the highest total confidence (sum of the individual confidence measure) that has information about all three vertices, which is view 6 in this example. In the case that no single view has information about all three vertices, the triangle texture is labeled unknown.

View v	$E_x^v(i, j, k)$ confidence	$E_y^v(i, j, k)$ confidence	$E_z^v(i, j, k)$ confidence	Total confidence
1	10	...	70	...
4	...	20	30	...
5	15	60	90	165
6	75	80	60	215
8	95	...



(a)



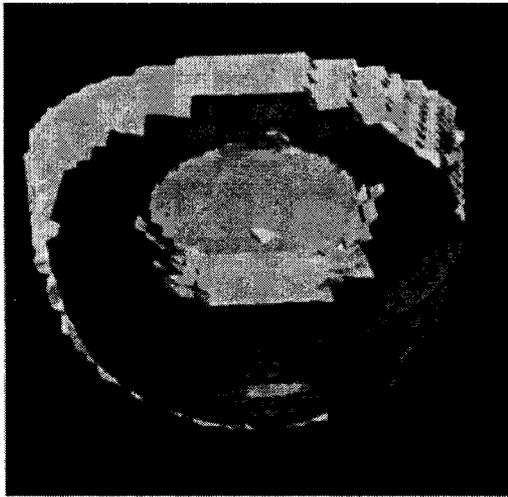
(b)

Fig. 14 Surfaces constructed by triangulating the range images from two viewpoints, shown with the registered thermal data as the texture map. (a) View 1 and (b) view 2.

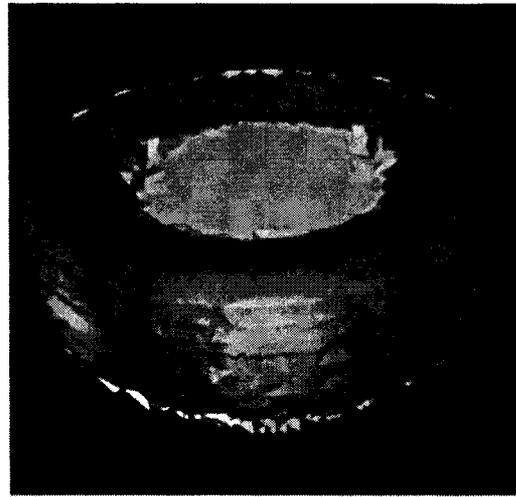
4 Examples

Here, we provide some examples of 3D tire models reconstructed using our algorithm and simulated range and thermal data. In the figures for this section, light-gray texturing indicates the absence of true texture data, caused either by the absence of simulated thermal data (in empty regions, for instance) or because the triangle texture is labeled unknown as noted at the end of Sec. 3.2.

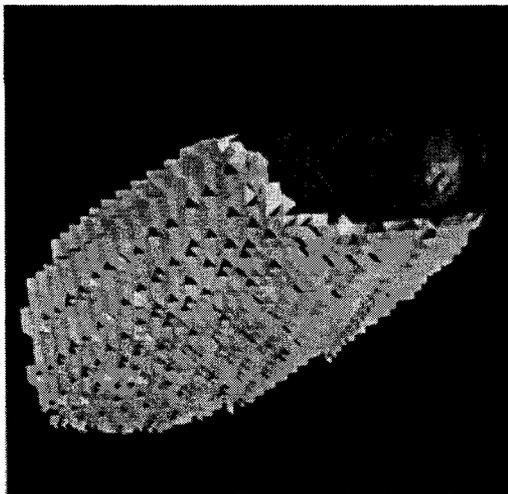
In Fig. 14, the triangulated range images from two different viewpoints are shown with the corresponding thermal data applied as a texture map. The significant overlap evident in the two views of Fig. 14 is necessary to ensure accurate modeling. In Fig. 15, we show three different views of the model produced by our algorithm using only the data from Fig. 14(a). In Fig. 16 we show the same three views as in Fig. 15 after integrating the data from Figs. 14(a) and 14(b). In Figs. 17–22 we show the reconstructed tire model using different resolution volume grids with varying numbers of views. Examining Figs. 17–22, it is



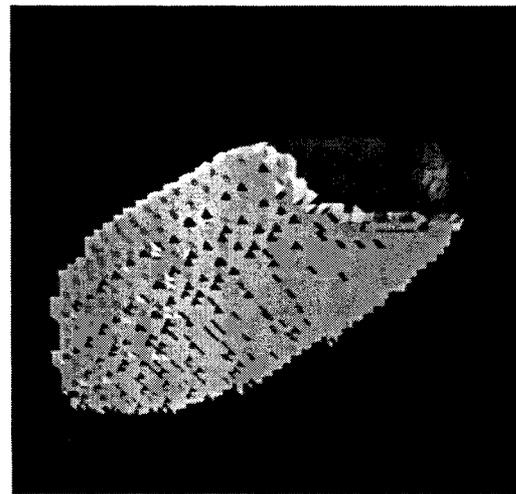
(a)



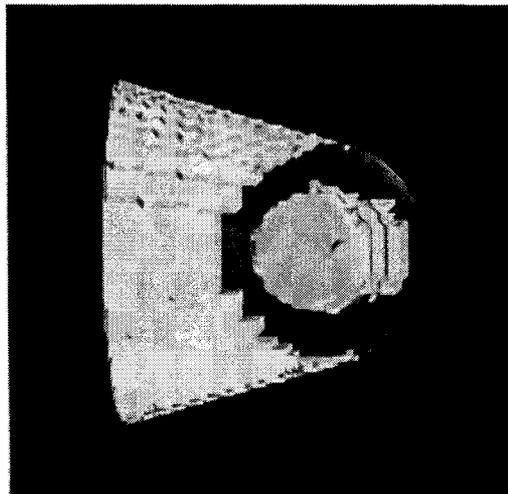
(a)



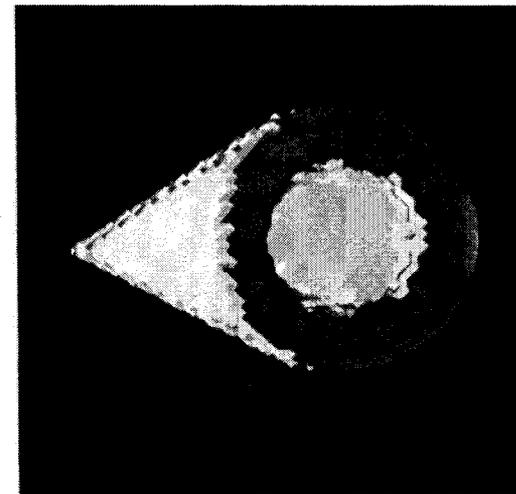
(b)



(b)



(c)



(c)

Fig. 15 Three views of the surface produced from the volumetric reconstruction algorithm using only the single view from Fig. 14(a). (a) Front view, (b) side view, and (c) top view.

Fig. 16 Three views of the surface produced from the volumetric reconstruction algorithm after integrating the two views from Fig. 14. (a) Front view, (b) side view, and (c) top view.

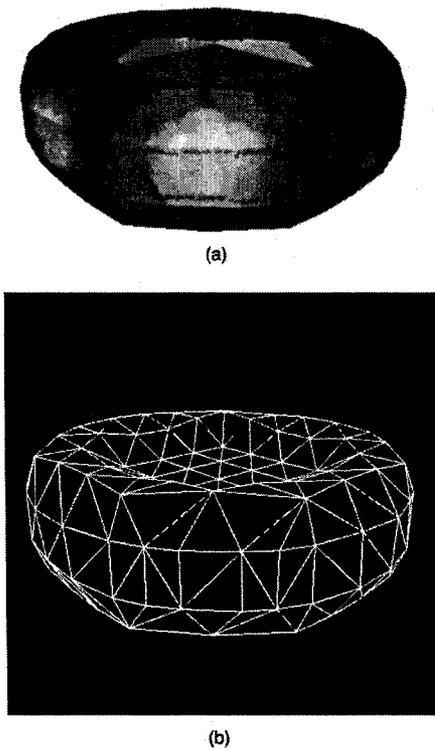


Fig. 17 Model constructed by integrating $N=8$ views using an $M=16$ ($16 \times 16 \times 16$) volume grid. (a) Surface textured with thermal data and (b) wire-frame model.

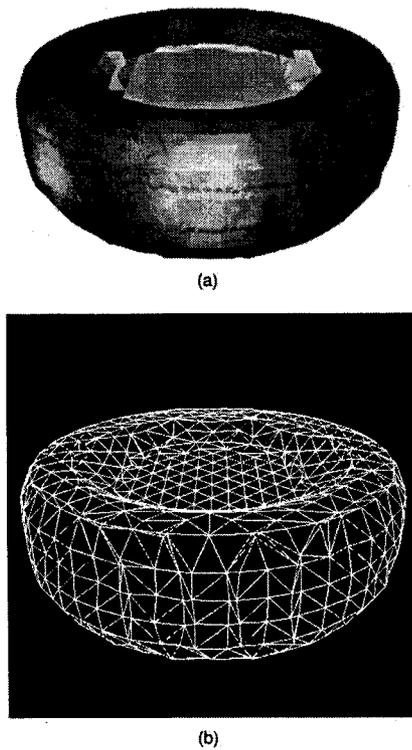


Fig. 18 Model constructed by integrating $N=8$ views using an $M=32$ volume grid. (a) Surface textured with thermal data and (b) wire-frame model.

evident, as we would certainly expect, that increasing the resolution of the volume grid M permits increased geometry resolution. Note the increasing evidence of the tread pattern in going from $M=16$ to $M=64$. A drawback to increasing M , however, is that more views are required to obtain an accurate and complete model, as evidenced by the unknown-textured triangles on the tire tread in Figs. 19 and 22.

5 Comparison With A Mesh-Based Approach

To demonstrate the merits of our volumetric approach, we compare some results to models reconstructed using the mesh integration approach proposed by Sun *et al.*¹ This mesh integration technique¹ is a state-of-the-art algorithm that is based upon earlier work by Soucy and Laurendeau.¹⁰ To accurately determine how well both sets of the reconstructed models represent the original model, we use the METRO¹⁵ mesh comparison software by Cignoni *et al.* 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 model.

In Fig. 23, 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, of course, to be expected. The mesh-based result has the best percentage for this set of reconstructed data, but the $96 \times 96 \times 96$ volume grid comes quite close. Similar results can be seen in Fig. 24, but this

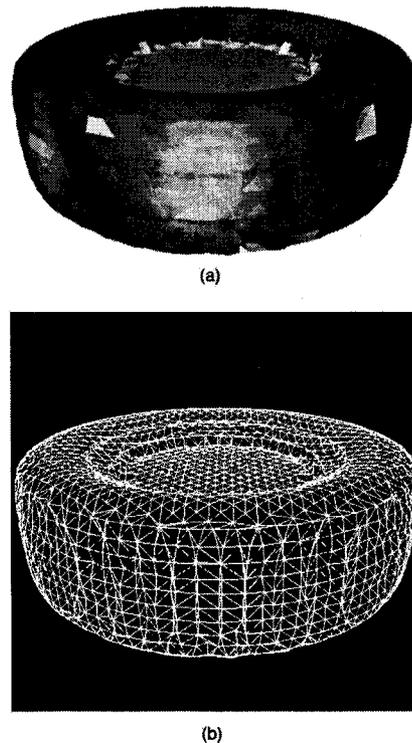


Fig. 19 Model constructed by integrating $N=8$ views using an $M=64$ volume grid. (a) Surface textured with thermal data and (b) wire-frame model.

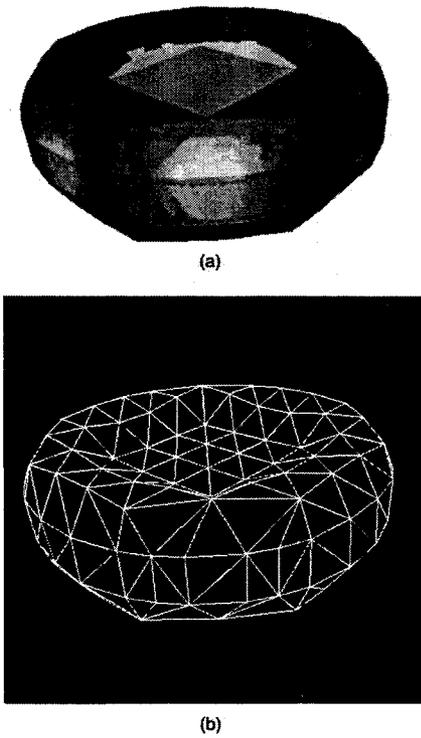


Fig. 20 Model constructed by integrating $N=16$ views using an $M=16$ volume grid. (a) Surface textured with thermal data and (b) wire-frame model.

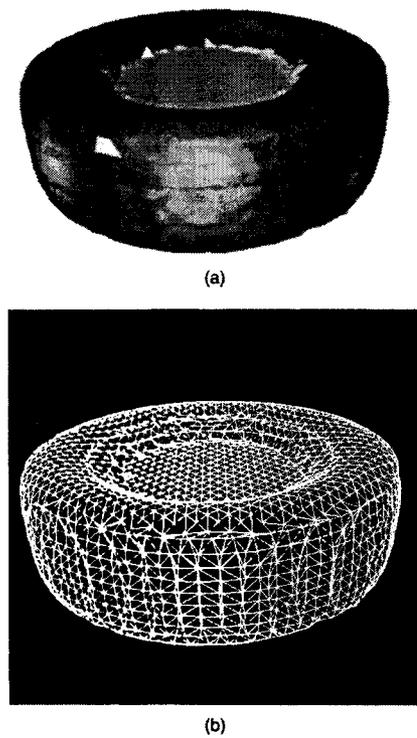


Fig. 22 Model constructed by integrating $N=16$ views using an $M=64$ volume grid. (a) Surface textured with thermal data and (b) wire-frame model.

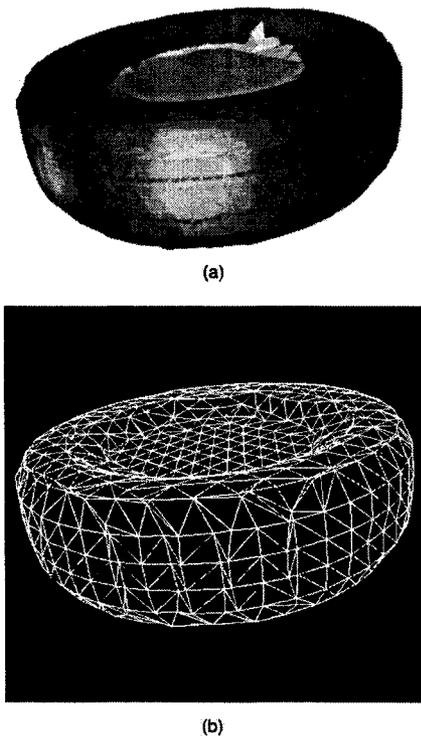


Fig. 21 Model constructed by integrating $N=16$ views using an $M=32$ volume grid. (a) Surface textured with thermal data and (b) wire-frame model.

time the $96 \times 96 \times 96$ resolution volume grid has the lowest percentage of mean error. It should be noted that at the higher resolutions the added redundancy of extra view-points improves the percentage of mean error for the volumetric method, as can be seen in Fig. 25, whereas the mesh-based results are slightly worse.

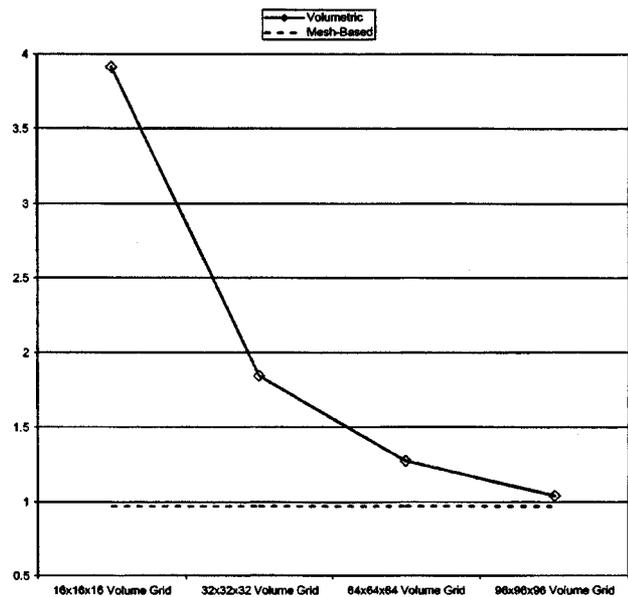


Fig. 23 Mean-error percentage for volumetric and mesh-based reconstruction using eight views.

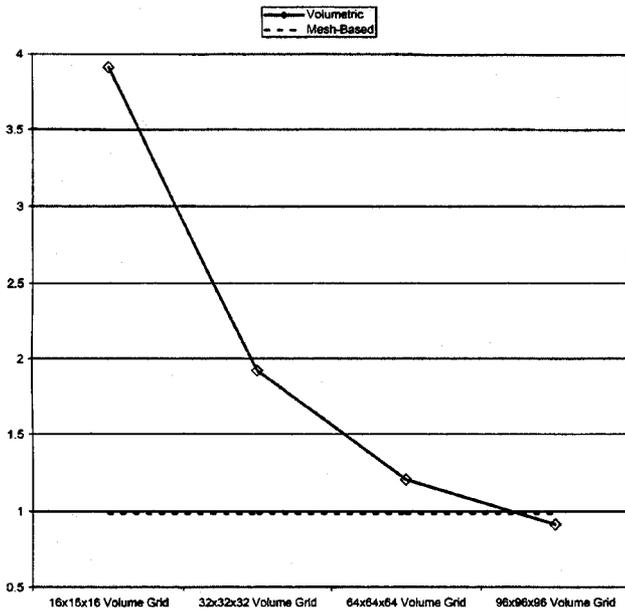


Fig. 24 Mean-error percentage for volumetric and mesh-based reconstruction using 16 views.

More complete METRO analysis results are given in Table 2. Of particular interest are the comparisons between the $96 \times 96 \times 96$ volumetric method and the mesh-based method. As can be seen in Table 2, both methods generate reconstructed models with approximately the same amount of vertices and triangles (see Fig. 26). The results favor the volumetric approach if the user hopes to achieve the smallest mean-squared error possible. If, however, a user desires the smallest maximal error, the mesh-based approach seems to be the better solution in this case.

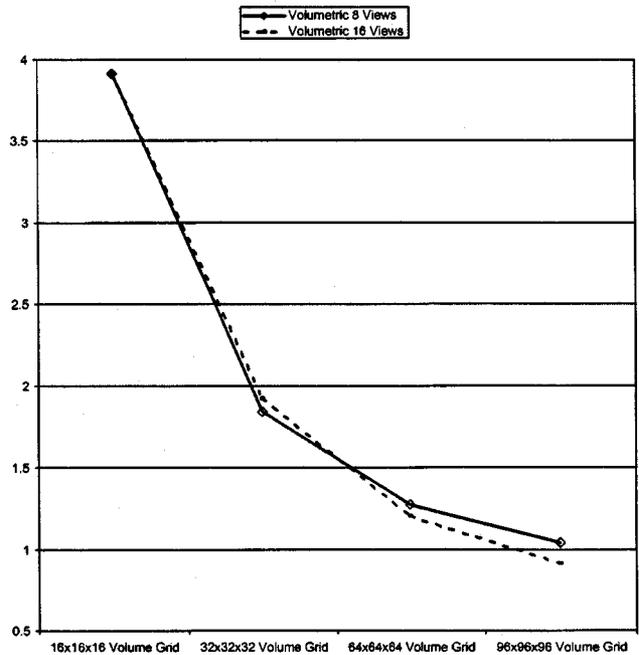


Fig. 25 Mean-error percentage for volumetric reconstruction with differing numbers of views.

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.

6 Conclusions and Future Research

We have presented a volumetric method for building 3D, textured models from multiple range and texture images.

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

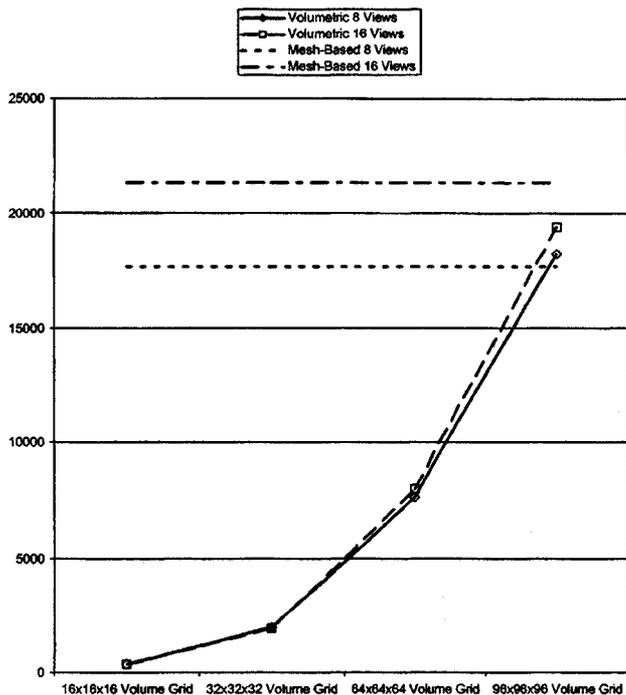


Fig. 26 Number of triangles in final reconstructed models.

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.

One 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. Another possible solution to this problem, however, is to back-project the unknown-texture triangles to the available view planes. If the confidence of an unknown-texture triangle is high with respect to some view (i.e., the triangle's normal is nearly parallel to the view direction), we should be able to map the texture from that view to the triangle. If such an unknown-texture triangle is visible from more than one view, a data fusion approach may be applied to improve the results. This idea of back-projection to solve the unknown-texture triangle problem is the subject of future research.

Acknowledgments

This work was supported by the U.S. Army's TACOM program and by Radian, Inc., under Contract No. DAAE07-96-C-X157.

References

1. Y. Sun, C. Dumont, and M. Abidi, "Mesh-based integration of range and color images," in *Sensor Fusion: Architectures, Algorithms, and*

Applications IV, Proc. SPIE 4051, 110-117 (2000).

2. R. Koch, M. Pollefeys, and L. Van Gool, "Realistic 3D scene modeling from uncalibrated image sequences," *Proc. IEEE Intl. Conf. Image Process.* 2, 500-504 (1999).
3. N. Amenta, M. Bern, and M. Kamvyselis, "A new Voronoi-based surface reconstruction algorithm," *Proc. SIGGRAPH 98*, 415-421 (1998).
4. M. Soucy, G. Goudin, R. Baribeau, F. Blais, and M. Rioux, "Sensors and algorithms for the construction of digital 3D color models of real objects," *Proc. IEEE Intl. Conf. Image Process.* 3, 409-412 (1996).
5. K. Pulli, T. Duchamp, H. Hoppe, J. McDonald, L. Shapiro, and W. Stuetzle, "Robust meshes from multiple range maps," in *Proc. Intl. Conf. on Recent Advances in 3D Digital Imaging and Modeling*, pp. 205-211 (1997).
6. B. Curless and M. Levoy, "A volumetric method for building complex models from range images," *Proc. SIGGRAPH 96*, 303-312 (1996).
7. A. Hilton, A. Stoddart, J. Illingworth, and T. Windeatt, "Reliable surface reconstruction from multiple range images," in *Proc. European Conf. Computer Vision*, pp. 117-126 (1996).
8. A. Hilton, A. Stoddart, J. Illingworth, and T. Windeatt, "Marching triangles: Range image fusion for complex object modeling," in *Proc. IEEE Intl. Conf. Image Process.* 2, 381-384 (1996).
9. R. Pito, "Mesh integration based on co-measurements," in *Proc. IEEE Intl. Conf. Image Process.* 2, 397-400 (1996).
10. M. Soucy and D. Laurendeau, "A general surface approach to the integration of a set of range views," *IEEE Trans. Pattern Anal. Mach. Intell.* 17(4), 344-358 (1995).
11. G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *Proc. SIGGRAPH 95*, 311-318 (1995).
12. R. Gonzales and P. Wintz, *Digital Image Processing*, Addison-Wesley, Reading, MA (1992).
13. L. Wong, C. Dumont, and 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).
14. W. E. Lorensen and H. E. Cline, "Marching cubes: A high-resolution 3D surface construction algorithm," in *Proc. SIGGRAPH 87*, 163-169 (1987).
15. P. Cignoni, C. Rocchini, and R. Scopigno, "METRO: Measuring error on simplified surfaces," *Comput. Graph. Forum* 17(2), 167-174 (1998).



Jeffery R. Price received the BSEE degree from the U.S. Naval Academy in 1993, and the MS and PhD degrees from the Georgia Institute of Technology in 1997 and 1999, respectively. While at Georgia Tech, he was the recipient of the Kodak Fellowship. In 1998, he worked with the Imaging Science Division at Kodak Research Laboratories. From 1999-2000, Dr. Price was a Research Assistant Professor with the Imaging, Robotics, and Intelligent Systems (IRIS) Laboratory at the University of Tennessee. Currently, he is a staff member in the Image Science and Machine Vision Group at Oak Ridge National Laboratory. His current research interests include image enhancement and restoration, image reconstruction, 3D object and scene modeling surface registration, and 3D segmentation. Dr. Price is a member of SPIE and IEEE.

Mongi A. Abidi received his Master's of Engineering in Electrical Engineering in 1985 and his PhD in Electrical Engineering in 1987, both from the University of Tennessee, Knoxville. In 1986, he joined the Department of Electrical and Computer Engineering at The University of Tennessee, Knoxville, where he is a W. Fulton Professor and Associate Department Head. His interests include image processing, multisensor processing, three-dimensional imaging, and robotics. He has published over 110 papers in these areas. He co-edited a book entitled: *Data Fusion in Robotics and Machine Intelligence*, Academic Press, 1992. He is a member of Tau Beta Pi, Phi Kappa Phi, and Eta Kappa Nu. He received the First State Award in primary graduation, the First State Award in secondary graduation, and the First Presidential Principal Engineer Award prior to joining the University of Tennessee. He is the recipient of the 1994-95 Chancellor's Award for Excellence in Research and Creative Achievement. Dr. Abidi is or has been a member in the professional societies including IEEE, Computer Society, Pattern Recognition Society, and SPIE.