NTT-MIT Variable Viewpoint Reality Tech Report

Please do not distribute outside of NTT

Exact Voxel Occupancy with Graph Cuts

Dan Snow
snow@ai.mit.edu

Paul Viola
viola@ai.mit.edu

Massachusetts Institute of Technology
Cambridge, MA 02139

Ramin Zabih
rdz@cs.cornell.edu
Cornell University
Ithaca, NY 14853

Abstract

Voxel occupancy is one approach for reconstructing the 3-dimensional shape of an object from multiple views. In voxel occupancy, the task is to produce a binary labeling of a set of voxels, that determines which voxels are filled and which are empty. In this paper, we give an energy minimization formulation of the voxel occupancy problem. The global minimum of this energy can be rapidly computed with a single graph cut, using a result due to Greig, Porteous and Seheult [7]. The energy function we minimize contains a data term and a smoothness term. The data term is a sum over the individual voxels, where the penalty for a voxel is based on the observed intensities of the pixels that intersect it. The smoothness term is the number of empty voxels adjacent to filled ones. Our formulation can be viewed as a generalization of silhouette intersection, with two advantages: we do not compute silhouettes, which are a major source of errors; and we can naturally incorporate spatial smoothness. We give experimental results showing reconstructions from both real and synthetic imagery. Reconstruction using this smoothed energy function is not much more time consuming than simple silhouette intersection; it takes about 10 seconds to reconstruct a one million voxel volume.

1 Introduction

Reconstructing an object's 3-dimensional shape from a set of cameras is a classic vision problem. In the last few years, there has been a great deal of interest in it, partly due to a number of new applications (such as [13]) that require good reconstructions. Voxel occupancy is a well-known approach to this problem, dating back at least to the early 1980's [11]. In voxel occupancy, the scene is represented as a set of 3-dimensional voxels, and the task is to label the individual voxels as filled or empty. In this paper, we present an energy minimization formulation of the voxel occupancy problem, where the exact global minimum can be rapidly computed with by finding the minimum cut on an associated graph.

We begin with a review of related work. In section 3 we formulate the voxel occupancy problem as the task of finding the binary labeling of voxels that minimizes an energy which depends both upon the observations and the smoothness of the shape. Section 4 shows that the labeling whose energy is the global minimum can be rapidly computed using a single graph cut. We present experimental results in section 5, using both real and synthetic imagery.

2 Related work

Most work on the voxel occupancy problem uses silhouettes (in fact, the problem is sometimes called shape-from-silhouettes). These methods work by computing the object's silhouette from each camera, typically via image differencing. Each silhouette is back-projected into the voxels, to yield a set of voxels that may be occupied. The silhouettes from the different cameras are then intersected (the algorithm is sometimes referred to as silhouette intersection). Of course, additional processing may be done on the resulting set of voxels, such as representing them with an octree [17]. Another related problem is to reconstruct the shape from a single camera, for example by placing the object on a turntable [4, 19, 16].

In silhouette intersection, the silhouettes essentially act as hard constraints (i.e., the object's volume must lie within its silhouette). The silhouette computation can obviously contain errors (for instance, if some pixels on the object happen to have a similar intensity to the background). A single pixel error in an individual silhouette will typically lead to a hole through the reconstructed object. In practice some kind of local morphological operator [15] is usually applied as a cleanup phase, either to the 2D silhouetted images or to the 3D volume. Unfortunately, these operators tend to introduce noticeable artifacts.

Our method, in contrast, does not explicitly compute silhouettes from each image, and allows global spatial smoothness to be incorporated. We essentially replace the silhouettes' hard constraint with a soft constraint, which is that the energy (which contains both a data term and a smoothness term) must be minimized. This allows voxels where the observed data is ambiguous to take on a value that is consistent with their neighbors' values. In silhouette intersection, ambiguous data creates difficulties, because the silhouette computation must make a binary decision; if this binary decision is incorrect, it is extremely difficult for the morphological cleanup operator to obtain good answers. In fact, our method can be viewed as a generalization of silhouette intersection; we will show at the end of section 3 that with the appropriate data term and smoothness term, minimizing the energy is equivalent to intersecting the silhouettes.

The voxel occupancy problem that we address is significantly simpler than the related problem of *voxel coloring* [5, 9, 14]. In voxel coloring, the task is to label every voxel with its color plus its transparency. Voxel coloring requires handling difficult issues, such as visibility relationships and non-lambertian surfaces. One advantage of voxel coloring is that the shape of the object can be estimated more accurately, since a mis-match in camera intensities can be used to prune away empty voxels. In addition the resulting voxel colors can be directly used to generate new views. Voxel occupancy, on the other hand, is an easier problem to solve. It is also much less sensitive to the geometric and photometric calibration of the camera system.

It is important to note that the views of an object from multiple cameras do not uniquely determine the shape of the object. This is true even with an infinite number of cameras. The visual hull of an object [10] is defined to be the maximal object that gives the original object's silhouette from any viewpoint. It is impossible to distinguish two different objects that have the same visual hull by relying purely on silhouettes.

One can however introduce prior information about the shape of objects in order to disambiguate the true object shape among the infinite set of objects which may have generated the observed silhouettes. Sullivan and Ponce introduce a smoothness term (using the formalism of splines) over the space of objects and show how to find the smoothest object which is consistent with the observation [16]. Their approach represents the object's shape analytically which avoids many of the difficulties of a voxel based solution. While our approach is similar in motivation, the formalization and algorithm we propose are quite different.

3 Problem formulation

Our input consists of k images from calibrated cameras. In addition, for each camera there is a background image I', which was taken with no object present. We will denote the set of all pixels in all images as $\mathcal{P}$. There will be a 3-dimensional set of voxels $\mathcal{V}$, with a neighborhood system $\mathcal{N} \subset \mathcal{V} \times \mathcal{V}$ that connects a voxel with the adjacent voxels.

Our output will be a binary labeling of $\mathcal{V}$, where a voxel is labeled with 1 if it is occupied and 0 if it is empty. We will write a labeling as f, and for any voxel $v \in \mathcal{V}$ we will write the label that f assigns to v as f_v.

A pixel p corresponds to a solid angle, which intersects a set of voxels denoted $V(p)$. We will assign a voxel v to be an element of $V(p)$ if p's solid angle contains more than $\frac{1}{2}$ of v's volume. The intensity difference $\Delta(p) = I(p) - I'(p)$ between the observed intensity and the background gives us information about which voxels are occupied and which are empty.

3.1 Energy minimization

To set this up as an energy minimization function, we will obtain from the observed intensity data a penalty for assigning a particular label f_v to a particular voxel v. We will write this penalty as $D_v(f_v)$. In addition, there will be a penalty of λ for assigning different labels to a pair of adjacent voxels. We seek the binary labeling that is both consistent with the observed data and spatially smooth. Specifically, we wish to obtain the labeling f^* that minimizes

$$E(f) = \sum_{v \in \mathcal{V}} D_v(f_v) + \lambda \sum_{v,v' \in \mathcal{N}(v)} (1 - \delta(f_v - f_{v'})). \quad (1)$$

Here, δ is the unit impulse function which is 1 at the origin and 0 elsewhere. The constant λ (often called the regularization parameter) controls the degree of spatial smoothness.

There are many ways to define $D_v(f_v)$, and our method does not depend upon its exact form. In general we expect that if $f_v = 1$ there will be large intensity differences in $O(v)$, while if $f_v = 0$ there will be small intensity differences in $O(v)$. More details regarding this function are given in the experimental section.

3.2 Relationship with silhouette intersection

It is easy to see that our problem formulation generalizes the silhouette intersection algorithm. In silhouette intersection (at least in the standard algorithm) there is no notion of spatial smoothness, so we let $\lambda = 0$. The term $D_v(f_v)$ will be binary valued. If there is some pixel p such that $v \in V(p)$, and the pixel p lies outside the silhouette in p's image, then $D_v(f_v) = f_v$; otherwise, $D_v(f_v) = 1 - f_v$.

With this choice of λ and D_v, there is a unique global minimum of E at a labeling f^* where $E(f^*) = 0$. The labeling f^* is precisely the labeling computed by silhouette intersection.

4 Fast exact energy minimization

We now face the task of minimizing the energy E given in equation 1. The form of this equation is typical of the regularization-based energy functions that arise in early vision (see [12] for some additional examples). In general, minimizing these energy functions is intractable [18], and so one has to rely on local search heuristics or simulated annealing. However, there are some interesting classes of energy functions which can be exactly minimized using graph cuts [2, 7, 8], and it turns out that E is in one of these classes.

4.1 Graph cuts

Let $\mathcal{G}$ be an undirected weighted graph with two distinguished terminal vertices $\{s, t\}$ called the source and sink. A *cut* $\mathcal{C} = S, T$ is a partition of the vertices into two sets such that $s \in S$ and $t \in T$. The cost of the cut, denoted $|\mathcal{C}|$, equals the sum of the weights of the edges between a vertex in S and a vertex in T.

The minimum cut problem is to find the cut with smallest cost. This problem can be solved very efficiently by computing the maximum flow between the terminals, according to a theorem due to Ford and Fulkerson [6]. There are a large number of fast algorithms for this problem (see [1], for example). The worst case complexity is low-order polynomial; however, in practice the running time is nearly linear.

4.2 Minimizing the energy

Greig, Porteous and Seheult showed in [7] that the global minimum can be rapidly computed for certain energy minimization problems via graph cuts. The class of energy functions they address, expressed in our notation, is as follows. Let $\mathcal{V}$ be a set of variables, with some neighborhood system $\mathcal{N} \subset \mathcal{V} \times \mathcal{V}$, and let $\mathcal{L} = \{0, 1\}$. The task is to find the binary labeling $f : V \mapsto \mathcal{L}$ that minimizes the energy. Let $D : V \times \mathcal{L} \mapsto \Re^+$ be arbitrary, and let $C : V \times \mathcal{L} \times V \times \mathcal{L} \mapsto \Re^+$ obey $C(v, 0, v', 1) = C(v, 1, v', 0)$ and $C(v, 0, v', 0) = C(v, 1, v', 1) = 0$ for all $v, v' \in \mathcal{V}$. Then the energy to be minimized is

$$E(f) = \sum_{v \in \mathcal{V}} D(v, f(v)) + \sum_{v, v' \in \mathcal{N}} C(v, f(v), v', f(v')).$$

Obviously, the energy minimization problem given in equation 1 is a special case of the one addressed by [7].

The graph $\mathcal{G}$ is constructed as follows. For every voxel $v \in \mathcal{V}$ there will be a node in $\mathcal{G}$. The only other nodes are the terminals. There will be links with weight λ between any voxel v and its neighbors in $\mathcal{N}$. Finally, there will be links between each voxel v and the terminals. The weight of the link between v and s will be $D_v(0)$, while the weight of the link between v and t will be $D_v(1)$.

There is a natural correspondence between cuts on $\mathcal{G}$ and labelings. If $\mathcal{C} = S, T$ is a cut, then the corresponding labeling $f^{\mathcal{C}}$ is defined by

$$f^{\mathcal{C}}(v) = \begin{cases} 1 & \text{if } a \in S, \\ 0 & \text{if } a \in T \end{cases}$$

The following theorem is the central result of [7].

Theorem 1 *If $\mathcal{C} = S, T$ is the minimum cut on $\mathcal{G}$, then the corresponding labeling $f^{\mathcal{C}}$ is the global minimum of the energy E.*

The proof follows trivially from the construction of $\mathcal{G}$ and the mapping between cuts and labelings.

5 Experimental results

A number of experiments, involving both synthetic and real data, were performed to demonstrate this algorithm. In order to simplify the subsequent description, both the synthetic and real experiments were performed with the same number of cameras, in the same positions, with the same sized volume, and using the same camera and volume resolutions.

5.1 Acquisition environment

We have constructed a 16 camera 3D scanning suite which can scan volumes approximately 2 meters square, large enough to reconstruct the human body (see Figure 1). The cameras are distributed in a "girdle-like" fashion outside of the volume looking inward: four of the cameras are 2 meters above the floor and look slightly downward, four are 70 centimeters above the floor and look slightly upward, 8 are roughly one meter high and look directly inward. There are no cameras directly above or below the volume. The

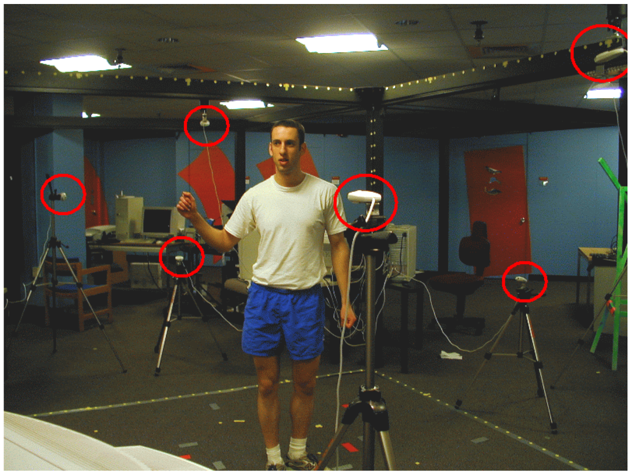

Figure 1: The 3D volume acquisition area. The acquisition volume is roughly 2 meters square, and is surrounded by 16 cameras. The camera which are visible in this image are highlighted with red circles.

cameras are connected to frame-grabbers in 8 computers which can synchronously acquire up to 30 frames per second. The cameras capture images at a resolution of 320 by 240 pixels (see Figure 2 for example images).

Since the cameras can be flexibly repositioned, they must be calibrated before data acquisition. This is performed using conventional calibration fiducials as well as a fine-scale refinement procedure. The resulting camera calibration are accurate to less than a centimeter.

5.2 Implementation details

The reconstructed volume is 1.5 meters wide, 1.5 meters deep, and 2 meters high. It is represented at a resolution of 2.5 centimeters. A graph is constructed containing one node for each voxel and the two terminal nodes, which we will call *object* (the source s) and *background* (the sink t). The voxel nodes are each connected to 6 neighbors using a weight of λ. Each voxel node is also connected both to the *object* and *background* node. The weight from *object* to a voxel v is $D_v(0)$, and the weight from v to *background* is $D_v(1)$.

The minimum cut algorithm cuts the minimal set of edges so that *object* and *background* are left unconnected. Since each voxel is connected to both of these nodes, either the *object* or the *background* edge must be cut. As a result it is only the difference between these weights that affects the final answer. In the experiment below the *background* edges (i.e., $D_v(1)$) are always set to 300. The *object* edge's weight $(D_v(0))$ is made larger if each of the cameras sees a large difference from the background. Intuitively, $D_v(0)$ is the cost of labeling a voxel as empty, which is large in this case.

Since there is a great deal of flexibility in selecting these weights, some of the experiments described below are designed to explore this choice. The voxel connection weight, λ, is of equal importance to the quality of the reconstructed volume. Intuitively it plays two roles. It determines whether a single voxel v may be occupied if its neighbors aren't; this is only possible if 6λ is less than the difference between $D_v(0)$ and $D_v(1)$. It also determines the smoothness of the resulting reconstruction. Narrow filaments are likely to be removed as λ is increased. In all of the experiments λ is 30.

The minimum cut is found using the max-flow code[1] due to Cherkassky and Goldberg [3]. The computation time of the 3D volume in all of the experiments is the same, roughly 9 seconds on a 500Mhz Intel PIII. The time is divided into three computations: the graph is prepared from the images (7 secs), the max-flow of the graph is computed (1.1 secs), the minimum cut labeling of the voxels is computed from the max-flow (1 sec). Note, the preparation of the graph involves a loop over every voxel where it is projected into each of the 16 images. This is the same sort of operation required to compute a simple silhouette intersection. The additional computation required to find the minimum energy voxel occupancy is minimal.

5.3 Synthetic experiment

Surprisingly the "quality" of silhouette intersection is often worse for synthetic objects than for real objects like the human body or a flower vase. Upon visual inspection, the silhouette intersection reconstruction of a cylinder using 16 images is poor (see Figure 3). The key difficulty is that while a cylinder is defined by its large flat ends and its straight, parallel sides, the reconstruction has none of these properties. Since this data is synthetic and perfect silhouettes are available, this poor reconstruction is due entirely to the fundamental limitations of silhouette intersection.

In order to demonstrate the value of a spatial smoothing , a simplified experiment was performed on this data. Using perfect silhouettes, reconstruction was performed using $\lambda = 30, \text{background} = 300$ and object $= \frac{400N}{16}$ where N is the number of cameras which believe this voxel is inside of the silhouette. Since the total number of cameras is 16, this weight

[1] See http://www.star-lab.com/goldberg/soft.html for the source code.

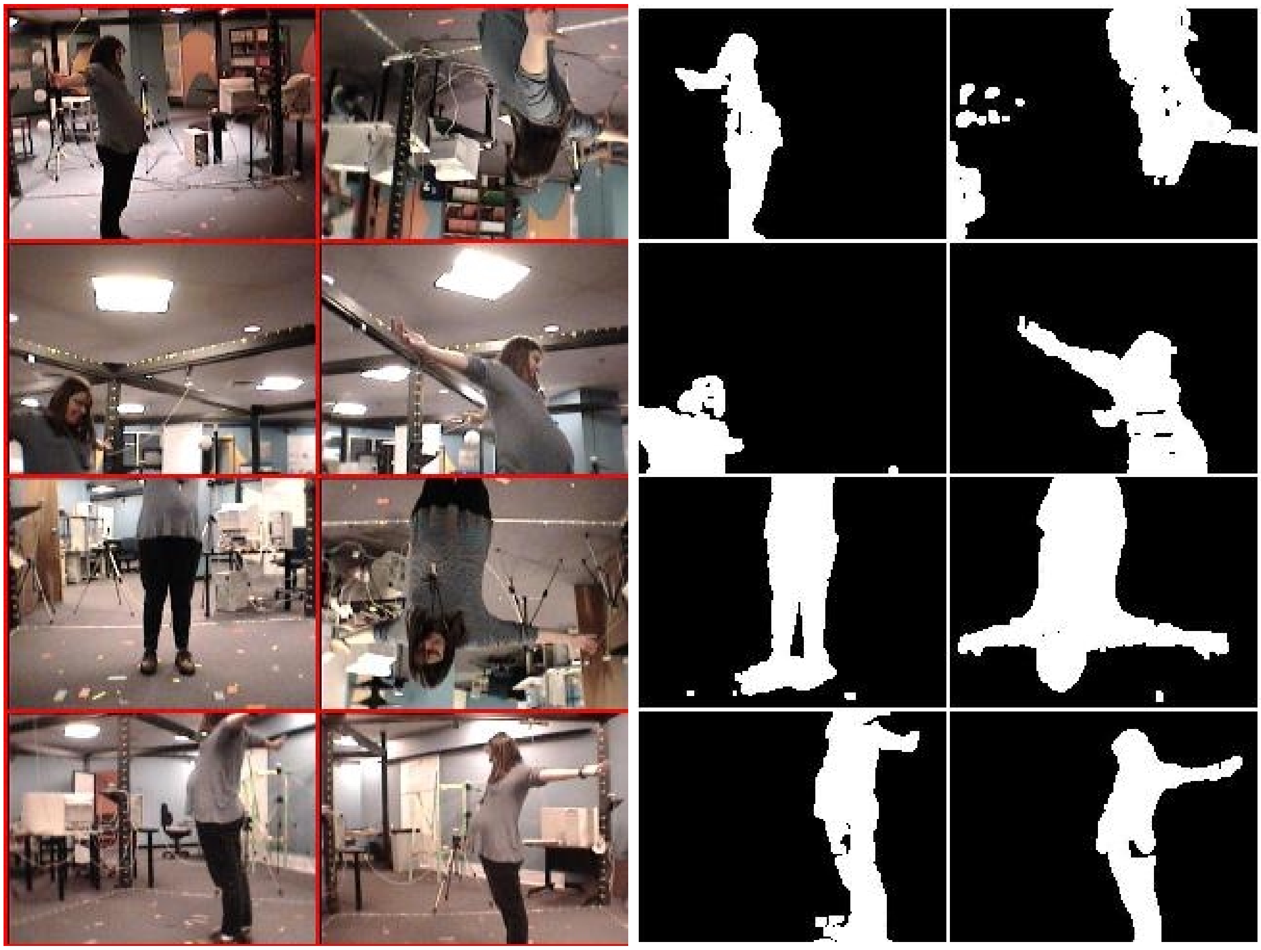

Figure 2: Left: Eight of the 16 images captured using within the acquisition volume. Right: Silhouettes computed from these images. Notice that there has been no attempt to artificially simplify the image processing necessary to compute silhouettes. The lighting and background is complex and uncontrolled. The subject is also wearing natural clothing which often matches the color of the background.

has a maximum value of 400. Our reconstruction is shown in Figure 3. Notice that the surfaces of this object are much flatter.

5.4 Real Experiments

Several real datasets were acquired using the multi-camera system (see Figure 2 for one example). Reconstruction proceeds much as above using $\lambda = 30, D_v(1) = 300$, except the edge weights between *object* and a voxel v is a function of $O(v)$, the observed differences in intensities at pixels that intersect v

$$D_v(0) = \frac{\sum_{\Delta \in O(v)} \min(\Delta^2, 400)}{16}.$$

This cost function uses a truncated quadratic function to determine the significance of the pixel differences. If the difference is small in many images $D_v(0)$ will be small. If $\Delta(p)$ is very large in one or a few images, the $D_v(0)$ will still be relatively small. Only in the case where $\Delta(p)$ is large in most of the images will the weight be large.

Figure 4 shows several 3D reconstructions from the images in Figure 2. The top reconstruction is performed using our method. Notice that the shape is quite smooth. One apparent artifact in the reconstruction is a swelling of the abdomen. This swelling is in fact the pre-natal child of one of the authors!

The middle reconstruction is performed using conventional silhouette intersection. Silhouettes were found by thresholding the difference images, followed by an erode and dilate operation. The low quality of

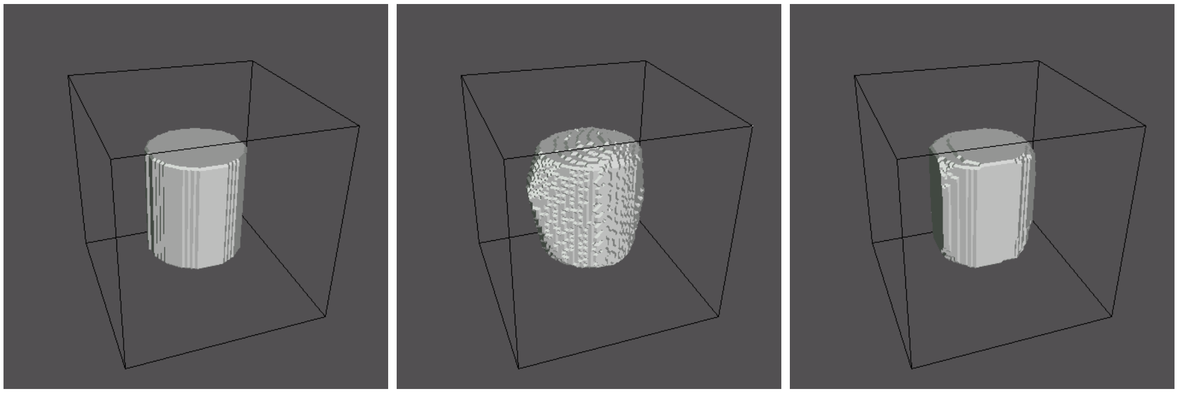

Figure 3: Ground truth voxel reconstruction of a synthetic cylinder (left), silhoutte intersection reconstruction (middle), and our reconstruction (right).

this reconstruction is due to the inaccuracies of the silhouettes, which is in turn related to the quality of the original images. Looking back at these images, notice that they are very "realistic". There is no artificial lighting, the backgrounds are quite complex, and the subject is wearing colors which appear in the background. All these issues conspire to make estimation of the silhouette difficult with simple algorithms.

The reconstruction at the bottom of the figure uses a heuristic mechanism to improve silhouette intersection. Classic silhouette intersection requires that each occupied voxel project to within the silhouette of *every* image. The heuristic reconstruction labels a voxel occupied if it projects into the silhouette of 3 out of 4 cameras. This heuristic does a good job of filling the holes, but it often yields reconstructions which are much larger than they should be.

A second reconstruction is shown in Figure 5. In this case the reconstructed volume was limited to the area of the torso.

6 Conclusions

This paper presents a new formulation of the voxel occupancy task. The classic formulation, silhouette intersection, often yields unsatisfactory results because of silhouette ambiguity and a lack of spatial smoothness. The new formulation never computes a silhouette, so it can handle noise in the original images as a well as situations where the object and background are of similar colors. The new formulation also naturally incorporates spatial smoothness which can improve the final results. An algorithm is presented which is based on graph cuts that can efficiently determine the 3D shape with lowest cost – the smoothest shape which is consistent with the observations. Finally a number of experiments demonstrate that the approach can rapidly and effectively reconstruct 3D volumes of real objects.

Acknowledgements

We thank Shai Avidan and Steve Seitz for helpful comments and references.

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, 1993.

[2] Yuri Boykov, Olga Veksler, and Ramin Zabih. Markov random fields with efficient approximations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–655, 1998.

[3] B. V. Cherkassky and A. V. Goldberg. On implementing push-relabel method for the maximum flow problem. In *4th Integer Programming and Combinatorial Optimization Conference*, volume 920 of *LNCS*, pages 157–171, 1995.

[4] R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *International Journal of Computer Vision*, 9(2):83–112, November 1992.

[5] J. DeBonet and P. Viola. Roxels: Responsibility weighted 3D volume reconstruction. In *International Conference on Computer Vision*, pages 418–425, 1999.

[6] L. Ford and D. Fulkerson. *Flows in Networks.* Princeton University Press, 1962.

[7] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279, 1989.

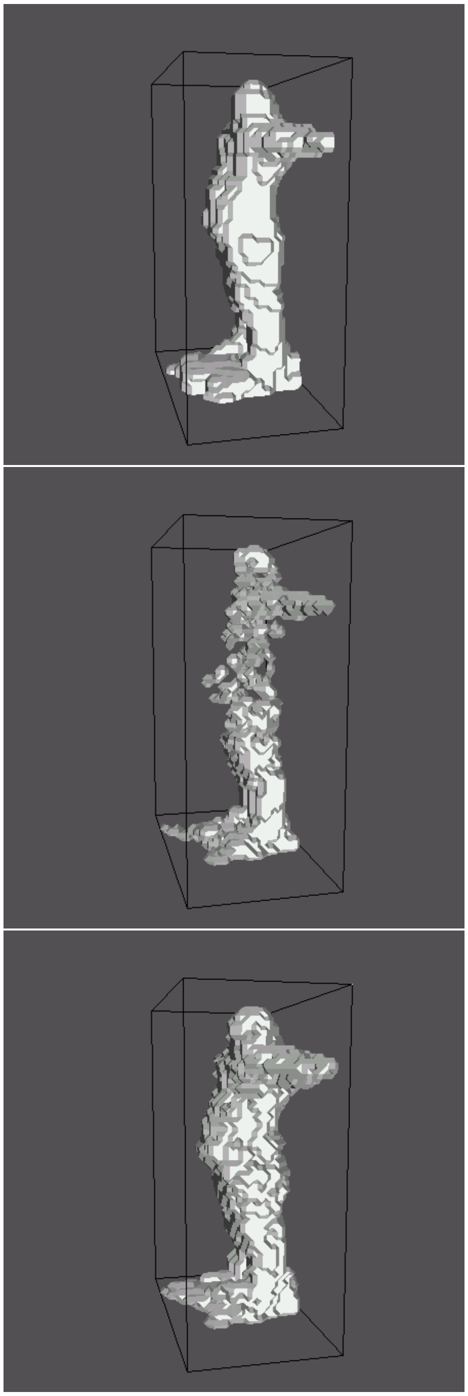

Figure 4: Three reconstructions from the images show in Figure 2. Top: reconstruction using our method. Middle: reconstruction using silhouette intersection (silhouettes were computed using image differencing, and morphological operations to remove noise). Bottom: robust silhouette intersection, where a voxel is consider occupied if 3 out of 4 cameras agree that it is within the silhouette.

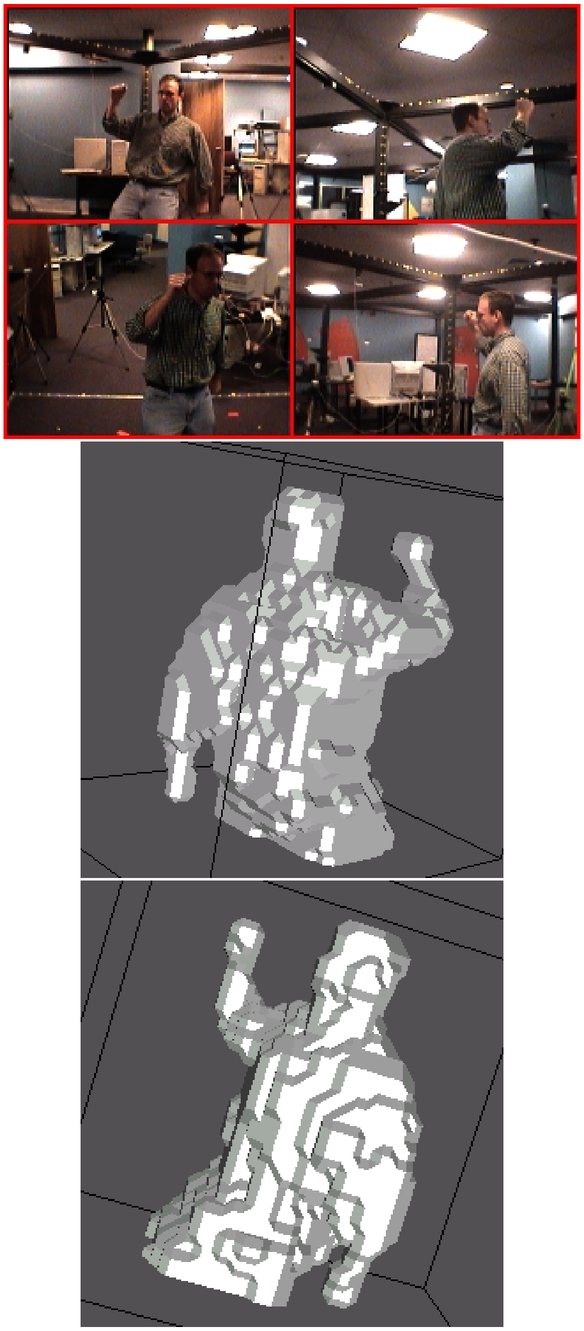

Figure 5: Top: four of the 16 views captured. Middle: one view of our reconstructed volume. Bottom: another view of the reconstructed volume.

[8] H. Ishikawa and D. Geiger. Segmentation by grouping junctions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 125–131, 1998.

[9] K. Kutulakos and S. Seitz. A theory of shape by space carving. In *International Conference on Computer Vision*, pages 307–314, 1999.

[10] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, February 1994.

[11] W.N. Martin and J.K. Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158, March 1983.

[12] Tomaso Poggio, Vincent Torre, and Christof Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.

[13] Peter Rander, P.J. Narayanan, and Takeo Kanade. Virtualized reality: Constructing time-varying virtual worlds from real events. In *IEEE Visualization '97*, volume 552, pages 277–283, October 1997.

[14] S.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1067–1073, 1997.

[15] Jean Serra. *Image Analysis and Mathematical Morphology.* Academic Press, 1982.

[16] S. Sullivan and J. Ponce. Automatic model construction and pose estimation from photographs using triangular splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1091–1096, October 1998.

[17] R. Szeliski. Rapid octree construction from image sequences. *Computer Vision, Graphics and Image Processing*, 58(1):23–32, July 1993.

[18] Olga Veksler. *Efficient Graph-based Energy Minimization Methods in Computer Vision.* PhD thesis, Cornell University, July 1999.

[19] J.Y. Zheng. Acquiring 3-D models from a sequence of contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):163–178, February 1994.