A Probabilistic Framework for Surface Reconstruction from Multiple Images

Motilal Agrawal and Larry S. Davis
Department of Computer Science,
University of Maryland,
College Park, MD 20742, USA
email:{mla,lsd}@umiacs.umd.edu

Abstract

This paper presents a novel probabilistic framework for 3D surface reconstruction from multiple stereo images. The method works on a discrete voxelized representation of the scene. An iterative scheme is used to estimate the probability that a scene point lies on the true 3D surface. The novelty of our approach lies in the ability to model and recover surfaces which may be occluded in some views. This is done by explicitly estimating the probabilities that a 3D scene point is visible in a particular view from the set of given images. This relies on the fact that for a point on a lambertian surface, if the pixel intensities of its projection along two views differ, then the point is necessarily occluded in one of the views. We present results of surface reconstruction from both real and synthetic image sets.

1. Introduction

Reconstruction of surfaces from stereo images has been a central research problem in computer vision for a long time. Early work in this area focussed on developing stereo algorithms for binocular camera configurations. There is a volume of literature on binocular stereo with a number of algorithms that work well on many types of images. More recently, however, due to significant advances in computational power, vision systems using multiple cameras are becoming increasingly feasible and practical. Example of multi-view vision systems include the $3D$ room developed by Kanade et al; [10] and the KECK Laboratory by Davis et al; [6] These systems are able to capture multiple synchronized images of indoor scenes. This has generated a renewed interest in the computer vision community to develop efficient, scalable and robust algorithms for surface reconstruction from multiple images.

Going from binocular to multiple views has the advantage of potentially increasing the stability of the reconstruction. However, in order to be able to fully exploit this potential, the algorithm must be able to handle occlusions, especially if the views are widely separated. In this paper, we consider the problem of 3D reconstruction from multiple cameras which are placed in arbitrary but known locations. The problem is formulated as one of estimating the probability that a $3D$ point in the scene lies on the object's surface. An iterative scheme is presented which updates this probability based on the visibility constraints that exist in the images.

2. Previous Work

3D shape reconstruction from multiple images is an intensely researched area. In this section, we restrict our discussion to approaches which model and detect occlusions explicitly.

The depth map representation, which is widely used in binocular stereo, is unable to represent partially occluded background regions (due to the fact that only a single disparity value is assigned to each pixel in the reference image). Therefore, most multi-view algorithms use an explicit representation of the 3D volume of the scene. When the camera calibration is known up to a projective transformation (weakly calibrated system), Saito and Kanade [11] proposed using projective grid spaces from a large number of images. For calibrated cameras, the scene may be discretized either in equal increments of volume (voxel space) or disparity (disparity space) [4, 7, 13, 14]. The goal of reconstruction is to find voxels which lie on the surface of the objects in the scene.

The earliest approach which reconstructed a voxelated representation of the scene was the cooperative stereo algorithm proposed by Marr and Poggio [9]. The Marr and Poggio algorithm works under the assumptions that the disparity maps have unique values and are continuous almost everywhere. The Marr and Poggio approach simultaneously represented and manipulated evidence for multiple disparities. This allowed for initial consideration of several hypotheses which would eventually be pruned through

0-7695-1272-0/01 $10.00 © 2001 IEEE

subsequent competition (uniqueness assumption). Kanade and Zitnick [16] have used this cooperative algorithm and modified it to use 3D local support and continuous match likelihood values. This also allowed them to explicitly detect occluded areas as regions with low likelihood values. Although the uniqueness assumption is true for binocular stereo, as pointed out in the previous paragraph, it does not hold for multi-view stereo because of occlusions.

Along similar lines, Szeliski and Golland [15] use an iterative scheme. At each stage of the algorithm, they have a set of voxels which are known to be surface points. Using this, they compute a visibility map, which indicates whether a given camera can see a voxel. The color consistency of a voxel along views in which it is visible is used to select clear winners at each stage that are added to the set of known surface points. In addition, in order to take into account the fact that voxels on the boundaries of objects are only partially occupied, they use real valued transparencies to represent voxels which are partially occupied by opaque objects.

An approach along different lines is the photo consistent voxel coloring algorithm by Seitz and Dyer [12]. Here, the problem of detecting occlusions is solved by the scene traversal ordering used in the algorithm; the order is such that if voxel V occludes V' then V is visited before V'. However, this requires that the placement of the cameras satisfy the ordinal visibility constraint, which says that no scene point should be contained within the convex hull of the camera centers.

Kutulakos and Seitz [8] address the limitation in the placement of the cameras of the voxel coloring algorithm by using a multi-sweep implementation of voxel coloring. Multiple planes are swept through the scene volume. Each plane sweep considers only a subset of cameras from which a voxel may be visible. An additional limitation of this approach is that for each voxel it only uses a subset of the total views available. Since it ignores the information that a voxel is occluded or visible in a view which was not used, all the available views are not utilized. In fact, as pointed out by [5], this approach is likely to produce a result that includes color-*in*consistent voxels. Their generalized voxel coloring algorithm [5] addresses this limitation. The basic idea behind this approach is that carving a voxel potentially changes the visibility of other voxels. When an uncarved voxel's visibility changes, it's color consistency is reevaluated and it too is carved if it is found to be inconsistent.

A common problem underlying these three approaches [12, 8, 5] is that they make hard commitments in carving away voxels. Therefore, if a voxel is carved away in error, there is no way to recover and this leads to a cascading effect, thereby generating large errors in reconstruction. Another problem is that they reconstruct only *one* of the potentially numerous scenes consistent with the input images. Consequently, they are susceptible to aperture problems caused by image regions of near uniform color.

These two problems can be addressed in a framework that does not make hard decisions and which takes into consideration alternative hypothesis which could better explain all the available images simultaneously. This is the basis for our probabilistic framework for surface reconstruction. Recently, several authors have used a probabilistic framework for surface reconstruction [1, 3].

Broadhurst et al; [3] have proposed a probabilistic extension of the space carving algorithm[12]. This framework is applicable to the special case in which the image set can be processed in a single sweep (i.e. the cameras satisfy the ordinal visibility constraint). Here, each voxel is assigned a probability that it belongs to the surface. The voxel array is processed using the plane sweep algorithm, starting with the layer closest to the viewer. The probabilities of the layers prior to the current layer are used to determine visibility for voxels in the current layer. Our framework, in contrast, is *iterative* and works for *arbitrary* placement of cameras. In addition, we not only use the information from the views in which a voxel is visible, but also the information that a voxel is occluded in other views.

The framework developed by Bonet and Viola [1] is very similar to our formulation. However, since we restrict ourselves to opaque objects, the optimization procedure is different. They use a probabilistic framework to represent voxels with partial opacity. Their algorithm is a multi-step algorithm which progresses from an initial estimate of the volume as entirely transparent, toward a state in which much of the volume is empty, and the observations are explained by a collection of semi-transparent and opaque structures. During the initial stages of the algorithm, occlusions cannot be accurately determined. However, as the algorithm progresses, the opacity of some voxels will be realized and the occlusion caused by these voxels will be used in subsequent steps.

The organization of the rest of the paper is as follows. The basic idea behind our approach is described in section 3. Section 4 presents the algorithm details. Experimental results with both real and synthetic data are presented in section 5.

3. Multi-view Viewing Constraints

The input for multi-view reconstruction is a set of K images of a scene S. In addition, we assume that each point in the scene is visible in at least V views. Knowledge about the camera placement can be used to obtain a conservative estimate of V. In the worst case, V can be safely assigned a value of two i.e. $V \geq 2$. This reflects the fact that only those scene points which are visible in at least two images can be reconstructed. Let I_i be the ith image of this set viewed from camera C_i and M_i the corresponding projec-

tion matrix of the camera. Therefore, a point in 3D space X projects to a point x_i in image i where

$$x_i = M_i X \tag{1}$$

Figure 1 shows a surface S viewed from four cameras placed at C_0, C_1, C_2 and C_3. Point X on the surface is visible in cameras C_0 and C_1 and occluded in cameras C_2 and C_3. $C_i X A_i$, $i = 0, 1, 2, 3$ is the ray joining the camera center to the 3D point for view i i.e. it is the viewing ray of X for view i. Note that in this configuration $V = 2$.

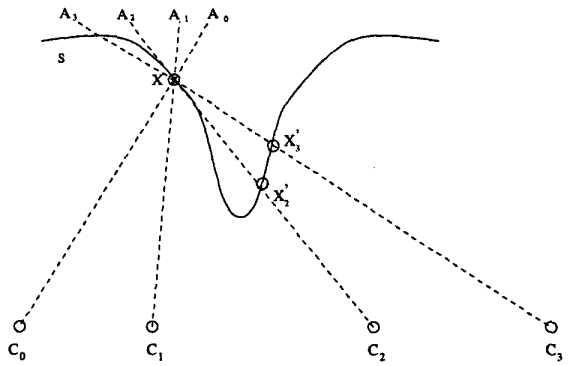


Figure 1. A scene viewed from four cameras

Under the assumption that the scene to be reconstructed is approximately Lambertian, if the $3D$ point X is not occluded in views i and j, then the pixel intensities

$$I_i(x_i) \approx I_j(x_j) \tag{2}$$

Interpreted differently, if the absolute value of the difference in pixel intensities $\|I_i(x_i) - I_j(x_j)\|$ is large, then it is highly probable that X is occluded in one of the views i and j. The converse, however is not true, unless of course each 3D point is uniquely colored. That is to say, even if $\|I_i(x_i) - I_j(x_j)\|$ is 0, it is possible that X is occluded in one of the views.

Denote by $pre(X, i)$, the set of all 3D points lying on the ray joining C_i to X which are closer to C_i than X. Let $post(X, i)$ denote all the 3D points lying on the viewing ray which are farther away from C_i than X. In figure 1 above, $pre(X, 1)$ is the set of points lying on the line segment $C_1 X$ and $post(X, 1)$ is the set of points lying on the line segment $X A_1$ (excluding the point X in both cases). At the n^{th} iteration, $P_n(X)$ denotes the probability that the 3D point X lies on the surface of an object in the scene. If X does not lie on the surface of any object, then X is said to be "free space"†. We use the following two constraints on visibility to refine $P_n(X)$.

†Note that by this definition, an interior point of an object is also a "free space" point

For a scene containing opaque objects, if X lies on the surface of an object, the following two properties must be satisfied along the viewing ray $C_i X A_i$.

1. **Constraint 1:** A point $X' \in post(X, i)$ may *either* be free space *or* it may be a surface point that is *not* visible in image i. But it can *not* be a surface point that is visible in image i.

2. **Constraint 2:** If X is visible in image i, then *all* points $X' \in pre(X, i)$ must be free space. Otherwise, if X is not visible in image i, then there must be *at least one* point $X' \in pre(X, i)$ that is a surface point.

4. Iterative Refinement of Probabilities

Ours is an iterative algorithm that works on a 3D discretized voxel space. We assume that we are given a voxel space containing the scene to be reconstructed. The size of the voxels is chosen so that the spatial resolution corresponds to the pixel resolution for most of the voxels. For voxels that project to multiple pixels or less than a pixel in a particular image, the pixel closest to the projection of the centroid of the voxel is taken as its projection. Apart from this simplistic framework, other ways of dealing with this sampling problem include a statistical consistency check proposed by Broadhurst and Cipolla [2]

At the n^{th} iteration, $P_n(X)$ represents the probability that X is a surface point. Therefore, the probability that X is free space is $(1 - P_n(X))$. Points which agree with the above two constraints are more likely to be surface points. And conversely, those that do not are unlikely to be surface points. A probabilistic measure $R(X)$ is introduced which measures how well X satisfies the above two constraints, which is then used to update the probabilities $P_{n+1}(X)$ for the next iteration.

4.1. Determination of visibility

Given that a voxel X is visible in view i, let $Pvis_i^j(X)$ denote the probability that it is visible in view j. As pointed out earlier, if the difference $\delta = \|I_i(x_i) - I_j(x_j)\|$ is large, then X is probably occluded in view j. On the other hand, if δ is small, then it may be either occluded or visible in view j. Based on this, we assign $Pvis_i^j(X)$ as

$$Pvis_i^j(X) = f(\delta) \tag{3}$$

where f is a function of δ satisfying the following two properties.

1. f should be high for small values of δ and should decrease as δ increases.

2. for small values of δ, the value of f should reflect the uncertainty that exists on whether or not X is visible if δ is small.

Let S_T be a subset consisting of T out of the K cameras and let $Pvis(S_T, X)$ denote the probability that X is simultaneously visible in all T cameras in S_T. For X to be visible in all the cameras, it must be visible in all pairs $i, j \in S_T$, so all $Pvis_i^j(X)$ $i, j \in S_T$ must be high. Therefore, the minimum value of $Pvis_i^j(X)$ $i, j \in S_T$ can be assigned to $Pvis(S_T, X)$.

$$Pvis(S_T, X) = \min_{i,j} \left(Pvis_i^j(X) \right) \text{ where } i \neq j \text{ and } i, j \in S_T \tag{4}$$

Lastly, we want to determine the probability that voxel X is visible in a view i, denoted by $Pvis(i, X)$. To determine this probability, we take advantage of the fact that every voxel is visible in at least V out of the total K views.

$$Pvis(i, X) = \max_{S_V} (Pvis(S_V, X)) \text{ where } i \in S_V \tag{5}$$

i.e. we find a subset S_V such that $i \in S_V$ and $Pvis(S_V, X)$ is maximum, and assign its probability of visibility to $Pvis(i, X)$.

4.2. Initial Probabilities

To start the process, we need to initialize the probabilities $P_0(X)$ for each voxel X. Again, taking advantage of the fact that voxel X is visible in at least V views,

$$P_0(X) = \max_{S_V} (Pvis(S_V, X)) \tag{6}$$

i.e. out of the $\binom{K}{V}$ subsets S_V, we take the subset with the maximum $Pvis(S_V, X)$ and assign its probability of visibility to $P_0(X)$. Note that while this guarantees that voxels which actually lie on the surface of an object will be assigned a high initial probability, many other voxels not lying on the surface will also be assigned high probabilities due to the fact that colors of the projected points may match in some views coincidentally. The goal of iterative refinement is to prune out those voxels which are not surface points through the use of constraints 1 and 2 along each visible ray.

4.3. Evidence Aggregation

At the n^{th} iteration, the two constraints along each viewing direction i can be used as evidence about the existence of a surface at a point X. These two constraints can be translated into probabilities as follows.

By constraint 1, all points $\hat{X} \in post(X, i)$ may be either free space or a surface point that is not visible along viewing ray i. The probability of this event $P_{post}(\hat{X})$ is given by the sum of the probabilities of these two events

$$\begin{aligned} P_{post}(\hat{X}) &= (1 - P_n(\hat{X})) + P_n(\hat{X})(1 - Pvis(i, \hat{X})) \\ &= 1 - P_n(\hat{X}) Pvis(i, \hat{X}) \end{aligned} \tag{7}$$

This simply states the fact that it cannot be the case that $\hat{X}$ is a surface point that is visible along viewing ray i.

Similarly, by constraint 2, for all points $\hat{X} \in pre(X, i)$, if X is visible in view i, then $\hat{X}$ must be free space. The probability of this event is

$$P_{pre}(\hat{X}) = Pvis(i, X)(1 - P_n(\hat{X})) \tag{8}$$

Equations 7 & 8 holds for all points in $post(X, i)$ and $pre(X, i)$ respectively. Therefore, all $P_{post}(\hat{X})$ and $P_{pre}(\hat{X})$ must be high. Let $X' \in pre(X, i)$ and $X'' \in post(X, i)$ be such that

$$P_{pre}(X') \leq P_{pre}(Y) \quad \forall Y \in pre(X, i) \tag{9}$$

$$P_{post}(X'') \leq P_{post}(Y) \quad \forall Y \in post(X, i) \tag{10}$$

i.e. X' and X'' are points with the least probability of obeying the constraints amongst $pre(X, i)$ and $post(X, i)$ respectively. Therefore, these are the two candidate points where the visibility constraint may be violated. In other words, if X preserves the constraints for those points, then we need not check for other points lying on the ray.

Both the constraints must be simultaneously satisfied along ray i for X to be a visible surface point in view i. Therefore, along ray i, the evidence for X being a visible surface point, $E_i(X)$, is simply the product of these two probabilities

$$E_i(X) = P_{pre}(X') P_{post}(X'') \tag{11}$$

4.4. Updating the Probabilities

Once the evidence $E_i(X)$ for each voxel and viewing direction are computed, they are then scaled so that the maximum $E_i(X)$ along each viewing direction is 1. This converts the absolute probabilities $E_i(X)$ to relative probabilities $R_i(X)$ and accounts for the fact that along each viewing ray, there must be one surface point that is visible along that ray.

$$R_i(X) = \frac{E_i(X)}{E_i(X_{\max})} \quad \text{and} \tag{12}$$

$$E_i(X_{\max}) = \max_Y (E_i(Y)) \quad \text{where} \tag{13}$$

$$Y \in \{pre(X, i) \cup \{X\} \cup post(X, i)\}$$

In order to combine $R_i(X)$ $i = 1, \ldots, K$, we take advantage of the fact that a voxel is visible in at least V images.

Therefore we can sort the $R_i(X)$ and multiply the V largest values to obtain $R(X)$.

$$R(X) = \prod_{i=1}^{V} R_i(X) \text{ where} \quad (14)$$
$$R_i(X) > R_j(X) \quad \forall j = i+1,\ldots,K \quad i = 1,\ldots,V$$

Intuitively, as mentioned earlier, $R(X)$ is a measure of how well the point X satisfies the two visibility constraints. The higher the value, the more is the agreement and vice-versa. Using these relative probabilities $R(X)$ and $P_n(X)$, P_{n+1} is updated using Baye's rule wherein, $P_n(X)$ is taken as the prior probability that X is a surface point.

$$P_{n+1}(X) = \frac{P_n(X)}{P_n(X)R(X) + (1 - P_n(X))} \quad (15)$$

4.5. Incorporating Smoothness

Most surfaces in the real world are smooth almost everywhere, except at surface discontinuities. At each iteration, we take this into account by considering a small $3D$ window centered at each voxel and then replacing $P_{n+1}(X)$ by the average probability in that window.

4.6. Surface Reconstruction

At each iteration of the algorithm, voxels with maximum probabilities along each viewing ray represent the reconstructed surface for that iteration and the color for each voxel is determined as the average color of its projection in all the viewing directions in which it is not occluded.

4.7. Discussion

Starting with a distribution of probability which is high for many scene points, including those which are not surface points, our algorithm uses the visibility constraints to prune down the probabilities of these false points and at the same time, boosts the probabilities of the true surface points. This is due to the fact that for true surface points, the relative probability $R(X)$ will be close to 1 and for points that are not surface points, $R(X)$ will be close to 0. So in the end, the false surface points are "carved" away. Our algorithm differs from the approach by [1] in this regards, since their approach starts with a representation of space that is initially all transparent(free space) and then adds the surface points as the iteration proceeds.

For a scene containing N voxels, during each iteration, each voxel is traversed once for every viewing direction and we need to store the relative weight for each voxel along each direction($R_i(X)$). In addition, for each viewing direction, we also need to find X' and X'' (equations 9 & 10). This can be accomplished in an efficient manner through two passes of the voxels (for each viewing direction). Therefore the time and space complexity of our algorithm is $O(NK + 2NK) = O(NK)$ during each iteration. Although, we have not done a convergence analysis of our algorithm, for all our experiments, the algorithm gave good results after 20-30 iterations. Obtaining a good estimate of V, the minimum number of views in which a voxel is visible, leads to faster convergence. Note that, in the absence of any information, as pointed out earlier, V can always be set to two. However, if in fact $V > 2$, then the algorithm converges slower by taking $V = 2$ and a larger number of iterations are required. We have also found that small overestimates of V do not hamper the results or convergence in any significant way.

5. Experimental Results

For our experiments, we have used a simple linear function for $f(\delta)$ together with a threshold. Since our cameras are color calibrated, in most cases the pixel intensity differences for visible surface points lie well within a difference of 20.

$$f(\delta) = \begin{cases} 0.55 - 0.01\delta & : \quad \delta \leq 20 \\ 0.01 & : \quad \delta > 20 \end{cases} \quad (16)$$

That is, if the absolute value of the difference in pixel intensities is less than 20, then f linearly decreases from 0.55. For $\delta = 0$, $f(\delta)$ is assigned 0.55 to reflect the fact that in this case the voxel may be either visible in both the views or it may be occluded but they have similar pixel intensities by chance. For δ greater than 20, a very low value is assigned to f. We have incorporated smoothness by using a 3x3x3 3D window.

In the first experiment, twelve synthetic images of three texture mapped spheres were created. The camera was rotated around the spheres in increments of 30 degrees. Figure 2(a) shows the first image of the input set and figure 2(b) shows the second image at 30 degrees rotation with respect to the first image. Our algorithm was run on a 100x100x100 cube enclosing the sphere. V for this set was set to two. A screen shot of the reconstructed VRML model is shown in figure 2(c)[‡]. Figure 2(d) shows the volume viewed from the top. Notice that towards the center of the volume, the sphere is not reconstructed. This is due the fact that these regions are visible in only a single camera and therefore cannot be reconstructed.

In our second experiment, fourteen images of a human subject were captured using color cameras placed on the four walls of a room, four on each wall (two of the cameras were out of synch and therefore not used). Figures 3(a)

[‡]Note that this not actually a synthesized view but a screen shot of the VRML model

and 3(b) show two images from two cameras placed on adjacent walls. For this configuration, V was taken to be 3. As in the synthetic set, the volume to be reconstructed was taken as a box of size 100x100x100 enclosing the subject. Two views of the reconstructed VRML model are shown in Figures 3(c) and 3(d).

6. Conclusion

We have presented a probabilistic framework for reconstruction of surfaces from a large number of views. The algorithm is based on two observations:

1. If a scene point is occluded from one view, then there must be another surface point along the ray joining that scene point to the camera center of that view.

2. Conversely, if a surface point is visible in a view, then there cannot be another surface point along the ray joining the camera center to that surface point.

The algorithm uses these two visibility constraints in all the views simultaneously to refine the probability that a scene point lies on the true surface, in an iterative manner. Another contribution of our paper is to use the pixel intensities of the projected images to obtain estimates for whether or not a surface point is occluded in a particular view. We have presented reconstruction results from both real and synthetic image sets.

The algorithm converges in about 20-30 iterations. We have not investigated the convergence properties of the algorithm. Intuitively, since all we are employing is the consistent visibility of the scene, the algorithm would probably converge for scenes with only opaque objects, although this remains to be demonstrated. It is also possible to adapt the algorithm for semi-transparent objects by adding a voxel opacity term. Another potential improvement that can be made is to take advantage of the knowledge of the camera placement and orientation, as in the voxel coloring algorithm [8, 12]. This is based on the fact that a voxel X that is visible in image i is more likely to be visible in all views that lie on one side of the plane passing through X and normal to camera center i. Therefore our probabilities for visibility ($Pvis_i^j(X)$ - equation 3) should be higher for all cameras, j, which lie on the same side of the plane as i. We are also currently investigating the case when the usual Lambertian assumption does not hold.

Acknowledgement

This research has been supported in part by National Science Foundation grant number NSF-EIA 01523672 and National Institute of Health under the human brain project grant number NIH-01432795

References

[1] J. D. Bonet and P. Viola. Roxels: Responsibility weighted 3D volume reconstruction. In *Proc. Seventh Int. Conf. on Computer Vision*, pages 418–425, 1999.

[2] A. Broadhurst and R. Cipolla. A statistical consistency check for the space carving algorithm. In *Proc. 11th British Machine Vision Conf.*, pages 282–291, 2000.

[3] A. Broadhurst, T. Drummond, and R. Cipolla. A probablistic framework for space carving. In *Proc. International Conference on Computer Vision(ICCV)*, volume I, pages 388–393, July 2001.

[4] R. Collins. A space-sweep approach to true multi-image matching. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 358–363, 1996.

[5] W. B. Culbertson, T. Malzbender, and G. Slabaugh. Generalized voxel coloring. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms:Theory and Practice (Proc. Int. Workshop on Vision Algorithms)*, volume 1883 of *Lecture Notes in Computer Science*, pages 100–115. Springer-Verlag, 2000.

[6] L. Davis, E. Borovikov, R. Cutler, D. Harwood, and T. Horprasert. Multi-perspective analysis of human action. In *Proceedings of Third International Workshop on Cooperative Distributed Vision*, Kyoto, Japan, November 1999.

[7] C. R. Dyer. Volumetric scene reconstruction from multiple views. In L. S. Davis, editor, *Foundations of Image Analysis*. Kluwer, 2001.

[8] K. Kutulakos and S. Seitz. What do n photographs tell us about 3d shape. In *Proceedings, 7th IEEE International Conference on Computer Vision*, pages 307–314, Corfu, Greece, 1999.

[9] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.

[10] H. Saito, S. Baba, M. Kimura, S. Vedula, and T. Kanade. Appearance-based virtual view generation of temporally-varying events from multi-camera images in the 3d room. In *Proceedings of Second International Conference on 3-D Digital Imaging and Modeling (3DIM99)*, pages 516–525, October 1999.

[11] H. Saito and T. Kanade. Shape reconstruction in projective voxel grid space from large number of images. In *CVPR99*, pages II:49–54, 1999.

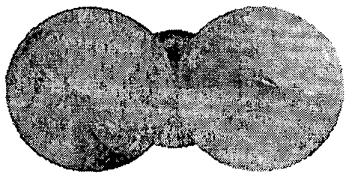
[12] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *IJCV*, 35(2):151–173, 1999.

[13] D. Snow, P. Viola, and R. Zabih. Exact voxel occupancy with graph cuts. In *Proc. Computer Vision and Pattern Recognition Conf.*, volume 1, pages 345–352, 2000.

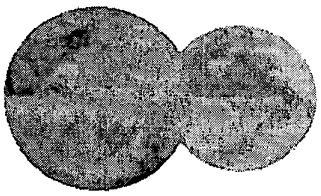
[14] R. Szeliski. Stereo algorithms and representations for image-based rendering. In *Proc. 10th British Machine Vision Conf.*, pages 314–328, 1999.

[15] R. Szeliski and P. Golland. Stereo matching with transparency and matting. In *Proc. Sixth International Conference on Computer Vision*, pages 517–524, 1998.

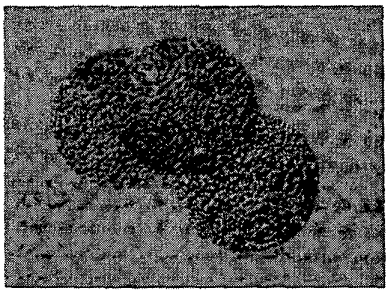
[16] C. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):675–684, July 2000.



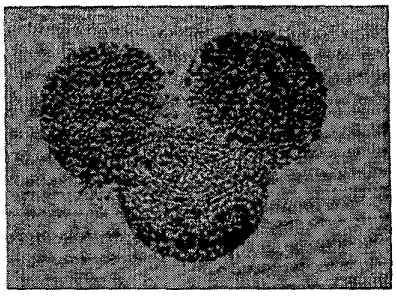
(a) First Image



(b) Second Image



(c) VRML Model



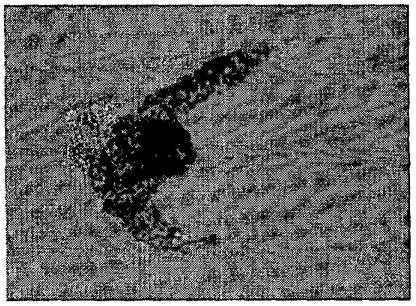
(d) VRML Model - View from top

Figure 2. Experimental results for synthetic images

(a) First Image

(b) Second Image

(c) VRML Model



(d) VRML Model - View from top

Figure 3. Experimental results for real images