# Simplified Belief Propagation for Multiple View Reconstruction

E. Scott Larsen Philippos Mordohai Marc Pollefeys
Henry Fuchs
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599 USA

## Abstract

*We address multiple-view reconstruction under an optimization approach based on belief propagation. A novel formulation of belief propagation that operates in 3-D is proposed to facilitate a true multi-image processing scheme that takes visibility into account and thus is applicable to scenes that contain significant occlusions. Visibility is not approximated but is estimated and used in a modified plane sweep stereo scheme. Optimization is performed in a simplified belief propagation framework in which messages are passed in 3-D, instead of 2-D, neighborhoods utilizing information from all available images. The information propagated from a point to one of its neighbors factors in the distance between the two points in 3-D, their difference in color. In contrast to traditional belief propagation, the observation is updated at each iteration to incorporate changes in visibility. The proposed approach is capable of enforcing smoothness on the evolving 3-D surfaces without being limited to a coarse resolution due to a volumetric representation. Moreover, our approach is applicable to both open and closed surfaces with no need for a priori knowledge of the type. We present dense reconstructions of publicly available image sets.*

## 1. Introduction

Reconstruction of a scene from sets of images or video has been one of the central themes in computer vision. As most computer vision problems, multiple-view reconstruction is an inverse problem of recovering 3-D structure from its 2-D projections. The task is further hindered by the ambiguities and difficulties associated with the detection of pixel correspondences in images. Smoothness is the most common constraint that is enforced to overcome these obstacles and states that scenes are smooth almost everywhere. It is usually enforced by allowing only small variations in a property of neighboring points, such as depth, disparity or surface normal. Extensive research has been conducted for binocular stereo [10], where the input consists of just two images. A wide range of methodologies have been applied in order to impose smoothness and other constraints in a way that simultaneously preserves the discontinuities of the scene. Recently, belief propagation [13, 12] has gained acceptance among the most successful frameworks for binocular stereo.

Here, we focus our attention to the case where more than two images are available. While more images obviously provide more information, one must be able to utilize that information correctly. By that we mean that processing the images in pairs is clearly sub-optimal since information that could have resolved some of the ambiguities remains unused. As shown in [10] and the Middlebury College Stereo Evaluation webpage (http://www.middlebury.edu/stereo), the major difficulties in establishing pixel correspondences are occlusion and lack of texture. We argue, as have other researchers, that if one attempts to match pixels in pairs of images to obtain partial reconstructions and then merges these partial reconstructions, some of the errors will not be corrected. The addition of more cameras reduces the ambiguity of uniform regions while the number of monocularly visible pixels in each image decreases for reasonable scene and camera configurations.

The practical aspect of true multiple-view processing is the need for a computational framework that allows efficient processing of potentially large amounts of data. The generalization of binocular approaches is not trivial in general. For instance the generalization of the work of Sun *et al.* [12] to more than two images would soon become computationally impractical as the number of images increases. Kutulakos and Seitz [8] proposed the space carving algorithm that computes the photo-consistent hull of the scene by carving away inconsistent voxels. Space carving inspired a large body of research but suffers from its inability to impose surface smoothness in a framework that reasons about pixels and voxels but not surfaces. Approaches that pose the problem as global optimization of an energy

Proceedings of the Third International Symposium on
3D Data Processing, Visualization, and Transmission (3DPVT'06)
0-7695-2825-2/06 $20.00 

function suffer from limitations such as the need to operate on rather coarse resolution grids or the inability to represent surfaces with sharp discontinuities.

We propose an approach that operates in 3-D and simultaneously considers information from all available images to update the positions of reconstructed points. To this end we employ the belief propagation framework that updates each point's belief (the probability distribution function for its depth) via messages from its neighbors. 3-D space is not discretized and thus resolution is determined by that of the images. Visibility is taken into account during message passing and updated after each iteration. This allows us to reconstruct scenes with large occlusions. It should be noted that current published results are typically for single objects or scenes where the foreground is relatively close to the background and visibility can be approximated without causing significant problems to the reconstruction. It should also be noted that our method does not use any foreground/background segmentation.

The paper is organized as follows: Section 2 is an overview of our approach; we review related work in Section 3; Section 4 provides the details on our novel algorithm and implementation; Section 5 describes the visibility-constrained plane sweep algorithm; Section 6 presents experimental results; and the paper concludes with Section 7.

## 2. Overview

Processing begins from a set of images with complete internal and external calibration information. The first processing step is to initialize the observation for each pixel in all images. For this purpose we use the plane sweep algorithm [3], which considers all images simultaneously to compute the cost of selecting a particular position for a 3-D point on a ray of the reference camera. We run a separate plane sweep for each camera, with the sweeping planes parallel to the image plane of the current camera. The results of the plane sweep at candidate depth values are used to initialize a pdf for each pixel, corresponding to surface likelihoods for the sampled depth values (distances from the camera center) on the ray.

We associate a single *node* with each pixel in each image. Our iterative algorithm updates the best depth estimate and confidence for each ray at each iteration. To do this, it steps along each candidate depth value and evaluates support for that depth value using its "neighbors". We define the neighbors for a candidate depth value as the nodes corresponding to the projections of that 3-D point into all other images, and the immediate neighbors of those projections (in image space). Note that this implies that the "neighborhood" for a node is a function of which of its candidate depth values is currently being evaluated. Details are in the next section but, for illustration, we consider the single node $P_d$ being updated in Figure 1. This node corresponds to pixel $p$ in image $i$. In performing the update for this node, we evaluate at all depth values, $P_1$–$P_n$, the support for a 3-D point there. For all images $j$, including $i$, we project this 3-D point into the image to get the pixel $q_j = Pr(I_j, P_d)$ - with $Pr(I, P)$ representing the projection operator on the 3D point $P$ into the image $I$. The neighborhood is the union of $q_j$ and nodes immediately adjacent to $q_j$ in image space, for projections into all images.

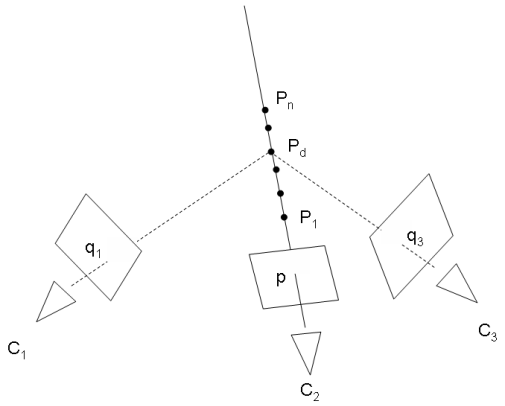


**Figure 1. Illustration of the neighborhood definition for a candidate depth along a single ray. For the ray that goes trough pixel $p$ in $C_2$, $P_d$ is a candidate 3-D point. Its neighborhood includes the four neighbors of $p$ in the reference image, as well as its projections $q_1$ and $q_3$ in the other images, rounded to the nearest pixel, along with their four-neighborhoods.**

Belief propagation [18], [9], detailed in the next section, is a message passing system that stores at each node a separate message for each of that node's neighbors. In our system, the "neighborhood" for a node is not unique for all positions along its state space (each depth candidate has a different set of neighbors), making storage and evaluation of separate messages intractable. Further, the states at each node do not directly correspond to each other (*i.e.* depth $d$ at one node is not directly comparable to depth $d$ at another node, especially from some other image with different pose). Overcoming these two difficulties gives rise to our Simplified Belief Propagation algorithm, in Section 4.

A key contribution of our approach is the visibility-constrained update mechanism for the observations. As detailed in Section 5, our approach updates visibility at every iteration and makes a fundamental change to belief propagation by allowing the observation term to vary over the iterations. In that section, we show that our technique can actually make a dramatic reduction in resource requirements,

Proceedings of the Third International Symposium on
3D Data Processing, Visualization, and Transmission (3DPVT'06)
0-7695-2825-2/06 $20.00 © 2006 IEEE

while properly addressing occlusions. It is implemented also as a plane sweep that does not include in cost computation images in which the point under consideration is invisible. The scheme is sketched in Fig. 3. Simply stated, an image is not used for the evaluation of a cost of a 3-D point if the point lies behind the current estimate of the surface for that image.

## 3. Related work

In this section we briefly review related work on multiple-view reconstruction. Koch *et al.* [6] begin by establishing binocular pixel correspondences and proceed by linking more cameras with these correspondences. When a correspondence is consistent with a new camera, taking uncertainty explicitly into account, directly preceding or following the cameras that already support it, the new camera is added to the chain and the position of the point in the scene is updated using the wider baseline. Taking a different approach, Collins [3] presents a true multi-image matching framework by evaluating the matching cost for all images simultaneously on a plane that sweeps through the scene. Kutulakos and Seitz [8] introduce the space carving algorithm which is based on the notion of "photoconsistency" to derive the visual hull of the scene. Voxels from an initial volume are progressively carved away if their projections on the images are not photoconsistent, that is if they exhibit large color variations. A limitation of the original space carving algorithm is that it cannot recover from the erroneous carving of a voxel. Some of the variations that have been proposed to address this problem include [2, 1, 17].

Kolmogorov and Zabih [7] extend the graph-cut framework, that has been very successful at binocular stereo, to multiple images. The optimization of a global energy function enforces smoothness to the solution while preserving boundaries. Smoothness is a critical limitation of the volumetric methods of the previous paragraph, which operate at the pixel-voxel correspondence level without considering the resulting surfaces. Vogiatzis *et al.* [15] also propose a volumetric algorithm using graph-cut optimization. The graph they use has an outer and an inner surface as the sink and source respectively. This allows for finer depth resolution between the two boundary surfaces but the topology of the scene has to be known a priori. Zeng *et al.* [20] offer a different combination of a volumetric representation with piece-wise graph cut optimization. The space is quantized in large voxels and a graph cut in each of them determines the validity of a potential surface patch. The trade-off between depth resolution and computational complexity is a major limitation of the graph-cut based approaches. Visibility is approximated in all these cases.

An alternative formulation of the problem includes the work of Faugeras and Keriven [4] who pose multiple view stereo in a variational framework where an initial surface evolves according to cross-correlation computed on the tangent plane of the surface, instead of the images. Yezzi and Soatto [19] address a different class of scenes in which surfaces have smooth or constant albedo. They do not rely on local correspondence, but on region similarity measures which are more effective for the types of objects they handle. Strecha *et al.* [11] present a pixel-based variational approach for recovering dense depth maps from wide-baseline views that can handle open surfaces. An anisotropic diffusion scheme that favors information from reliable points is used to guide surface evolution. Level set methods achieve excellent results, but are limited to closed surfaces, with the exception of [11], and more importantly to surfaces with smooth derivatives. Corners and other sharp features are smoothed. Visibility treatment is exact. This is an inherent property of variational approaches due to their global notion of the evolving surface.

Vogiatzis *et al.* [16] present an MRF based stereo algorithm that is applicable to multiple images given base surfaces in 3-D. If such a surface can be inferred or approximated, then a number of points can be sampled on it. These points are allowed to move on their estimated surface normals and their optimal positions are computed through belief propagation after collecting evidence from all images. Visibility is approximated. Tsin and Kanade [14] introduce kernel correlation as a robust framework that removes viewpoint induced artifacts from reconstructions. It minimizes the distance from each point to all other points with emphasis given to neighboring via a Gaussian kernel that attenuates with distance. The method is similar to ours in that it takes distance in 3-D and color similarity in all images into account as we do in Section 4. It does not model occlusion however. Zitnick *et al.* [21] are among the few researchers to address image sets with strong occlusion effects. In fact, we use the images captured by their system for some of our experiments. Their approach, however, relies heavily on segmentation and is targeted toward novel image generation and not necessarily depth accuracy.

## 4. Simplified Belief Propagation

Belief propagation algorithms can be used for, among other things, optimization with an energy function for a pairwise Markov Random Field (MRF) as

$$E(f) = -\sum_{p \in \mathcal{P}} \ln \psi(f_p) - \sum_{(p,q) \in \mathcal{N}} \ln \psi(f_p, f_q). \quad (1)$$

This energy function operates on a graph, with $\mathcal{P}$ a set of all nodes and $\mathcal{N}$ a set of node pairs, *i.e.* the set of edges. The set of nodes $q \in \mathcal{N}(p)$ with $(p, q) \in \mathcal{N}$ is termed the *neighborhood* of the node $p$. Each node takes a labeling $f_p$,

from some finite state space (*e.g.* disparity values). In these equations, $\psi(f_p)$ will be referred to as the *observation* - the data consistency term, and $\psi(f_p, f_q)$ as the *compatibility* term - the smoothness term.

Belief propagation is an iterative message passing algorithm defining the *belief* at a node $q$ at iteration $T$ as

$$b_q(f_q) = \psi(f_q) \prod_{p \in \mathcal{N}(q)} m^T_{p \to q}(f_q), \quad (2)$$

where $m^T_{p \to q}$ is the message that node $p$ sends to node $q$ at iteration $T$. Note that each message is a pdf over the state space (in our case, depth values).

There are two classes of belief propagation algorithms ([18], [9]), one termed "sum-product" and the other "max-product," denoted for their methods of updating messages. The sum-product algorithm updates the message from $p$ to $q$ at iteration $T$ via

$$m^t_{p \to q}(f_p) = \sum_{f_p} \psi(f_p, f_q)\psi(f_p) \prod_{s \in \mathcal{N}(p) \setminus q} m^{t-1}_{s \to p}(f_p) \quad (3)$$

and the max-product algorithm uses

$$m^t_{p \to q}(f_p) = \max_{f_p} \psi(f_p, f_q)\psi(f_p) \prod_{s \in \mathcal{N}(p) \setminus q} m^{t-1}_{s \to p}(f_p). \quad (4)$$

Note that max-product is often expressed and/or implemented in $-\ln$ space, in which it is min-sum and takes a form like the original energy equation (1).

In the update equations (3,4), the compatibility functions combined with the operation on it (sum or max) behave as a "filter" on the pdfs. Common implementations (and equational expressions) will combine incoming messages and filter the resulting outgoing message: "filter on output". Note that an alternative method, sometimes seen in implementations, is to instead filter incoming messages: "filter on input."

For the max-product algorithm, the two variations are not necessarily identical, but in practice are nearly the same. The first step in obtaining our algorithm is the explicit movement of the compatibility function to the incoming messages instead of the outgoing messages.

The next steps to our algorithm are using the compatibility function in two important, and uncommon, ways. The first is to actually make compatible two pdfs that would otherwise not be compatible. *E.g.* a pdf over depth values from one pixel is not directly compatible with a pdf over depth values from a pixel in some other camera, plainly seen if that camera has dramatically different orientation and position.

The second use of the compatibility function is to make compatible pdfs that are actually represented differently. E.g. one a discrete pdf over a range of depth values, and a single Gaussian distribution, from a ray with drastically different orientation and origin.

## 4.1. Representations

Each pixel in each image, identified $p$, corresponds to a node in the simplified belief propagation graph. Each pixel directly maps to a single ray (we use the pinhole camera assumption) with $P_d$ being the 3-D point corresponding to the depth value $d$ along the ray for $p$.

### 4.1.1 Observations

At each node, the observation $O_p$ is a discretely sampled pdf over candidate depth values. We initialize the observation $O(P_d)$ for each depth $d$ via plane sweeping, similar to [3], using an $n$x$n$ SAD kernel with

$$cost(P_d) = \sum_{i \neq ref} \delta_i(P_d) \| Pr(I_i, P_d) - Pr(I_{ref}, P_d) \|, \quad (5)$$

with $\delta_i$ being 0 or 1 distinguishing if $Pr(I_i, P_d)$ is outside or inside the view frustum of $I_i$. The final cost for a pixel is the usual sum across the window and across all color channels, normalized by the sum of the $\delta_i$ terms. We normalize by the number of cameras that participate in each computation in order not to favor points visible by fewer cameras.

The initial observation for a pixel (equivalently a ray) is in terms of a cost for each potential depth value. We need to convert this to a pdf through a mapping that converts high cost values to low likelihood values. We have selected a Gaussian function for this mapping. We can also define the confidence at a depth with the same scheme. We do not use the magnitude of the cost directly since the existence of a low cost for a certain depth does not preclude the existence of numerous other low costs which would make the depth ambiguous and the entire observation unreliable. The observation for a depth $P_d$ and the confidence for $P_d$ we define in terms of the cost for depth $cost(P_d)$ as follows:

$$O(P_d) = e^{-\frac{cost(P_{best})^2}{\sigma_o^2}} \quad (6)$$

$$conf(P_d) = \frac{e^{-\frac{cost(P_d)^2}{\sigma_o^2}}}{\sum_i e^{-\frac{cost(P_i)^2}{\sigma_o^2}}} \quad (7)$$

where $\sigma_o$ is selected in a way that extends the useful range of observation values.

### 4.1.2 Nodes and Messages

Due to our complex definition of a neighborhood and the significant resource requirement of storing a separate message for all neighbors, we instead store the belief at each node. We compress the belief pdf to be a single pair of values: the current best estimate depth value for that node and the confidence in that depth value, and will denote the belief

Proceedings of the Third International Symposium on
3D Data Processing, Visualization, and Transmission (3DPVT'06)
0-7695-2825-2/06 $20.00 © 2006 IEEE

at the node for pixel $p$ as a depth value and confidence in it. Nodes are initialized based on the observation, as

$$B^0(p) = \max_d O(P_d) \quad (8)$$

$$B^0_{conf}(p) = conf(B^0(p)). \quad (9)$$

### 4.2. Node Updates

The different representation form of observations and beliefs, along with the lack of separate messages for each neighbor, necessitates that we construct a variation of the node update equations. Instead of updating messages (as there are none), we update the belief at each node by reconsidering each candidate depth value via an accumulation of support from all of its neighbors. The likelihood of each node is updated by collecting messages from all neighbors in all images. For each $P_d$, a neighborhood is defined by projecting $P_d$ on all the images and adding to $P_d$'s neighborhood the nodes at the nearest integer pixel position along with the nodes at its four pixel neighbors (see Fig. 1). Messages from all nodes in this neighborhood of $P_d$ are accumulated as support for $d$ being the new best estimate depth value for the node at $p$.

The node at pixel $q$ in the neighborhood for $P_d$ sends a message to $p$ when depth $d$ is considered. This message is weighted by a 3-D distance term $d_3(P_d, q)$ and a color (radiance) similarity term $r(P_d, q)$. The compatibility function for this message is the weighting of that node, having different values for each step of $d$ when updating node $p$. This compatibility function allows this compressed pdf belief at node $q$ to be evaluated differently, in 3-D, for each candidate depth value when updating node $p$ using the discretely sampled observation pdf. Specifically, the new belief for node $p$ is updated according to

$$support(P_d) = O(P_d) + \sum_{q \in N(p)} d_3(P_d, q) r(p, q) \quad (10)$$

$$b^{t+1}(p) = \max_d support(P_d) \quad (11)$$

$$b^{t+1}_{conf}(p) = support(b^{t+1}(p)) \quad (12)$$

with

$$d_3(P_d, q) = e^{-\frac{dist(P_d, n)^2}{\sigma_d^2}} \quad (13)$$

$$r(p, q) = e^{-\frac{\sum_{\{R,G,B\}}(p-q)^2}{\sigma_r^2}}. \quad (14)$$

The distance term reduces the amount of support for points that project on neighboring pixels but are far from each other in the scene. The color distance is computed using the color of the ray under consideration $p$ in the reference image and the color of ray $q$ in its own reference image. This term ensures that a node does not receive support from a region whose colors are different from its own. Its main purpose is to preserve depth discontinuities by not allowing interference across occlusion boundaries in the images, under the assumption that different surfaces have different colors. It also enhances smoothness enforcement on uniform surfaces without over-smoothing areas with texture where stereo works well.

### 4.3. Summary

The sum-product algorithm produces a new value for state $d$ by accumulating evidence from every state value $q$ in the incoming message weighted by some compatibility function relating $d$ and $q$. This is exactly what our algorithm does: for candidate state $d$ we accumulate support from all neighboring nodes across all of their states weighting via a similarity function between $d$ and the states of each other node. So, the term

$$\sum_{f_p} \psi(f_p, f_q) m^{t-1}_{s \rightarrow p}(f_p) \quad (15)$$

is doing "filter on input" instead of "filter on output". The messages coming in are similar to delta functions, as a result of our compression method of picking the best state and only storing that and its confidence. Our compatibility function

$$\sum_{f_p} \psi(f_p, f_q) = s(P_d, q^t) d_3(P_d, q^t) r(p, q^t) \quad (16)$$

is expressed functionally and takes constant computational time, instead of linear [5] or quadratic computational time. Yet, our expression is not a perfect match to either sum-product (3) or max-product (4). We call our variation "max-sum," expressed as

$$b^t(q) = \max_{f_p} \left( \psi^{t-1}(f_q) + \sum_{p \in \mathcal{N}(q)} \psi(f_q, f_p) b^{t-1}(q) \right), \quad (17)$$

Finally, note the $t$-1 iteration notation on $\psi(f_q)$. This indicates our final contribution: a method for dealing with occlusions by updating the observation as a function of the current estimate of the system. This is detailed in the next section. Note that Vogiatzis *et al.* [16] also allowed the observation term to be updated as the iterations progressed.

## 5. Visibility-constrained Plane Sweep

The plane (or space) sweep stereo algorithm [3] is a true multi-image matching method that operates by projecting all the input images on a plane that sweeps through the

scene. When parts of the plane are placed at the locations of the actual scene surfaces, all images should agree on the color of these surfaces, as long as they are visible. Typically these planes are parallel with one of the image planes and are moved in discrete distance increments perpendicular to themselves. Each ray intersects each plane once creating a depth hypothesis. The cost for this hypothesis is computed as the sum of absolute color differences between the projections of all other images with the reference image on that point on the plane (Fig. 2).

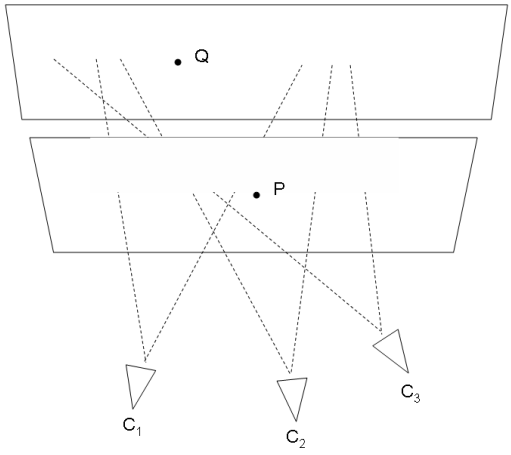


**Figure 2. Plane sweep stereo. Depth hypotheses are validated by measuring the color similarity of the projection of points on each plane to all the images. For instance if point $P$ is indeed a point of the scene, its projections on the three cameras have to be similar. On the other hand if $Q$ is not at that depth, it is more likely to project to different colors.**

While plane sweep stereo is very effective due to the use of multiple images, it does not take occlusion into account. Initially, such information is not available, but after selecting the most likely point on each ray, we can start reasoning about visibility and occlusion. We do so by choosing which images are allowed to participate in the cost evaluation for a certain hypothesis. The key observation is that a ray should not be included in cost evaluation if its current depth estimate occludes the hypothesis under evaluation. The presence of a surface does not affect what can happen behind it. On the other hand, the ray must be included if the hypothesis is at or in front of its current depth estimate. In this case, the hypothesis is assumed to be visible from the camera and the cost should be evaluated using this information. See Fig. 3 for an illustration of the proposed algorithm. The visibility-constrained plane sweep is repeated after each belief propagation iteration to refine the observations.

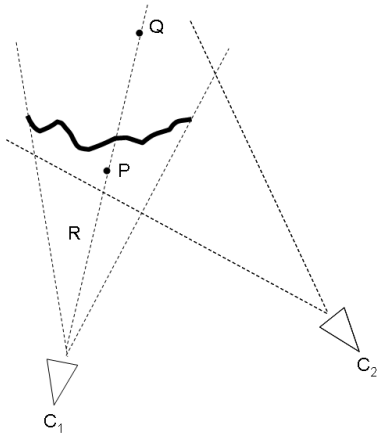


**Figure 3. Visibility-constrained plane sweep. The dark curve is a cut of the current surface estimate for camera $C_1$. The reference camera is $C_2$. Rays from $C_1$ are not used for the computation of costs if the current most likely point of the ray occludes the position under consideration. Ray $R$ is considered when computing the cost for point $P$ that is in front of the surface but not for $Q$ which is occluded in $C_1$.**

We have had the most success eliminating visibility effects by altering the $\delta_i$ term in 5. This term was originally a binary term, determined by whether or not $Pr(I, P_d)$ actually projected in the image or not. Now, we set this value to be 0 if the confidence of that node is high and it is nearer to that image's center of projection than $P_d$ - *i.e.* it is confident that it occludes. Otherwise, we set $\delta_i$ to the confidence of that node, thereby penalizing the occlusion of reliable points. This is very comparable to the probabilistic space carving method of modeling occlusions used by Broadhurst *et al.* [2].

## 6. Experimental Results

For the experiment presented here, we use images captured with a synchronized eight-camera rig by Zitnick *et al.* [21]. The datasets have been made available by the authors, to whom we are grateful, at http://www.research.microsoft.com/vision/ImageBased Realities/3DVideoDownload/. The included images illustrate the visual quality of our algorithm using 8 cameras. For parameters, we use $\sigma_d = 1.6 \cdot 10^{-3}$ (distance attenuation), $\sigma_r = 150$ (for all three color channels combined), with 50 candidate depth steps, and a 11x11 observation SAD window.

Proceedings of the Third International Symposium on
3D Data Processing, Visualization, and Transmission (3DPVT'06)
0-7695-2825-2/06 $20.00 © 2006 IEEE

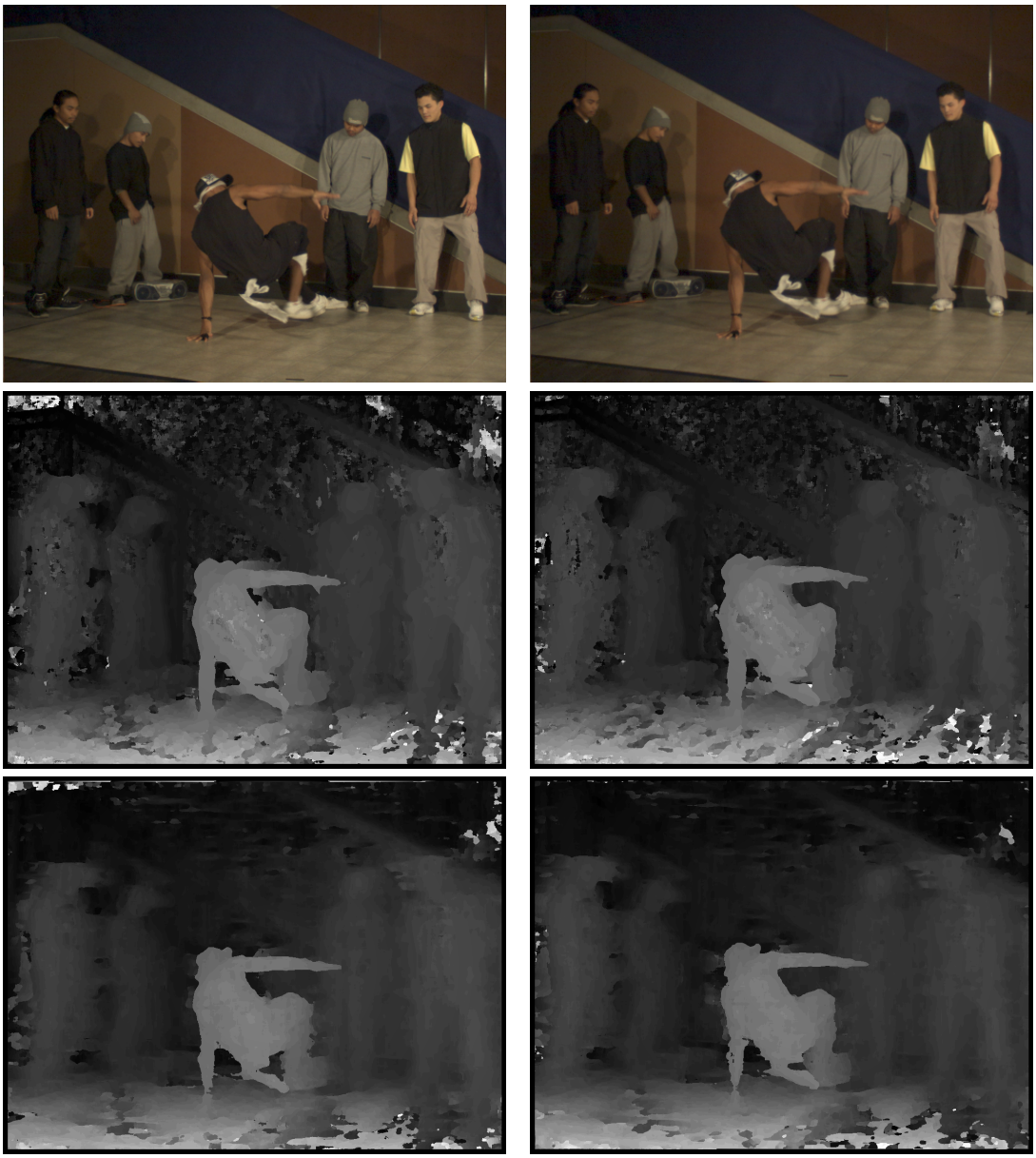

**Figure 4. Results for cameras 3 and 4. The top row is the color reference images. The second row is the initialization (from plane sweep[3]), the next is the result after nine iterations.**

Proceedings of the Third International Symposium on
3D Data Processing, Visualization, and Transmission (3DPVT'06)
0-7695-2825-2/06 $20.00 

As the observations are being re-computed at each iteration, we implement the algorithm by stepping all rays from a camera in lock-step. This allows us to not need to store the observation data explicitly. Combined with storing only the compressed beliefs, our memory requirements are dramatically reduced: for 8 images of with 1024x768 pixels each, we need less than a Gigabyte of memory, whereas just storing observations for 50 steps requires just under 4GB. Further, not only does this reduce memory requirements, but we can now increase the granularity of the depth steps we use, limited now only by computation time.

## 7. Conclusion

We have presented a novel approach for multiple view reconstruction that is capable of handling scenes with stronger occlusion effects than current methods. The key factor that allows us to achieve this is our visibility-constrained plane sweep stereo algorithm. Instead of approximating visibility or using an initial estimate throughout processing, we effectively update the visibility at every iteration by not using images from which a certain part of the scene is invisible in surface estimation for that part. Since we do not use a voxel-based representation, resolution is not limited by the size of the voxels and we do not suffer from the effects of perspective projection of the voxels on images at oblique angles. Our approach is applicable to both open and closed surfaces since we make no assumption about the surfaces and the rays are oriented toward the centers of the cameras. Further, no background segmentation is needed for our algorithm.

We have also introduced a simplified, approximate formulation of belief propagation that operates in 3-D and is able to relate nodes whose states do not correspond. Furthermore, unlike traditional belief propagation, we update the observation for each ray at every iteration as new visibility information becomes available. Maintaining observations corrupted by image measurements that do not actually observe the point under consideration does not aid convergence to the right depth.

## References


[1] M. Agrawal and L. Davis. A probabilistic framework for surface reconstruction from multiple images. In *Int. Conf. on Computer Vision and Pattern Recognition*, pages II:470–476, 2001. 3

[2] A. Broadhurst, T. Drummond, and R. Cipolla. A probabilistic framework for space carving. In *Int. Conf. on Computer Vision*, pages I: 388–393, 2001. 3, 6

[3] R. Collins. A space-sweep approach to true multi-image matching. In *Int. Conf. on Computer Vision and Pattern Recognition*, pages 358–363, 1996. 2, 3, 4, 5, 7

[4] O. Faugeras and R. Keriven. Variational principles, surface evolution, pdes, level set methods, and the stereo problem. *IEEE Trans. on Image Processing*, 7(3):336–344, March 1998. 3

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR04)*, IEEE Computer Society Conference on Computer Vision. IEEE Computer Society, 2004. 5

[6] R. Koch, M. Pollefeys, and L. Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In *European Conf. on Computer Vision*, pages I: 55–71, 1998. 3

[7] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *European Conf. on Computer Vision*, pages III: 82–96, 2002. 3

[8] K. Kutulakos and S. Seitz. A theory of shape by space carving. *Int. Journ. of Computer Vision*, 38(3):199–218, July 2000. 1, 3

[9] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Kaufmann, San Fransisco, CA, 2nd edition, 1998. 2, 4

[10] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. Journ. of Computer Vision*, 47(1-3):7–42, April 2002. 1

[11] C. Strecha, T. Tuytelaars, and L. Van Gool. Dense matching of multiple wide-baseline views. In *Int. Conf. on Computer Vision*, pages 1194–1200, 2003. 3

[12] J. Sun, Y. Li, S. Kang, and H. Shum. Symmetric stereo matching for occlusion handling. In *Int. Conf. on Computer Vision and Pattern Recognition*, pages II: 399–406, 2005. 1

[13] J. Sun, N. Zheng, and H. Shum. Stereo matching using belief propagation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(7):787–800, July 2003. 1

[14] Y. Tsin and T. Kanade. A correlation-based model prior for stereo. In *Int. Conf. on Computer Vision and Pattern Recognition*, pages I: 135–142, 2004. 3

[15] G. Vogiatzis, P. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *Int. Conf. on Computer Vision and Pattern Recognition*, pages II: 391–398, 2005. 3

[16] G. Vogiatzis, P. Torr, S. Seitz, and R. Cipolla. Reconstructing relief surfaces. In *British Machine Vision Conf.*, pages 117–126, 2004. 3, 5

[17] R. Yang, M. Pollefeys, and G. Welch. Dealing with textureless regions and specular highlights: A progressive space carving scheme using a novel photo-consistency measure. In *Int. Conf. on Computer Vision*, pages 576–584, 2003. 3

[18] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *International Joint Conference on Artificial Intelligence (IJCAI 2001)*, Distinguished Papers Track, 2001. 2, 4

[19] A. Yezzi and S. Soatto. Stereoscopic segmentation. In *Int. Conf. on Computer Vision*, pages I: 59–66, 2001. 3

[20] G. Zeng, S. Paris, L. Quan, and F. Sillion. Progressive surface reconstruction from images using a local prior. In *Int. Conf. on Computer Vision*, pages II: 1230–1237, 2005. 3

[21] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM Trans. Graph.*, 23(3):600–608, 2004. 3, 6



Proceedings of the Third International Symposium on
3D Data Processing, Visualization, and Transmission (3DPVT'06)
0-7695-2825-2/06 $20.00 © 2006 IEEE