

# Trinocular Stereo using Shortest Paths and the Ordering Constraint

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 describes a new algorithm for disparity estimation using trinocular stereo. The three cameras are placed in a right angled configuration. A graph is then constructed whose nodes represent the individual pixels and whose edges are along the epipolar lines. Using the well known uniqueness and ordering constraint for pair by pair matches simultaneously, a path with the least matching cost is found using dynamic programming and the disparity filled along the path. This process is repeated iteratively until the disparity at all the pixels are filled up. To demonstrate the effectiveness of our approach, we present results from real world images and compare it with the traditional line by line stereo using dynamic programming.*

**Keywords:** computer vision, trinocular stereo, dynamic programming, ordering constraint

## 1. Introduction

Establishing correspondences is the key problem in 3D reconstruction from stereo images. The goal of correspondence is to assign matches to each point in the reference image. This is done using a measure of similarity based on the intensities at the points. Stereo correspondence methods may be either feature based or pixel based. In feature based correspondence methods, features such as intensity edges are extracted and matched first. The similarity measure used in this case are correlation windows centered around the features [10]. Subsequently, correspondence for the remaining points is determined by interpolation and knowledge of scene geometry. Pixel based correspondence methods directly find the correspondence for each pixel using the pixel intensities. Lately, pixel based correspondence methods have gained popularity [7, 3, 14]. Pixel based methods have the advantage of giving dense depth maps. When combined with additional constraints and assumptions about the scene, it can yield very accurate results.

One such widely used and well understood constraint is the epipolar constraint. The epipolar constraint arises from the geometry of the cameras. This constraint restricts the match of a point in the reference image to lie along the epipolar line in the other image. It reduces the search space to a line but the problem still remains intractable because of the exponential size of the search space. This has made it necessary to use additional assumptions about the nature of the scene. Ohta & Kanade and Cox et al. [11, 7] make the uniqueness and ordering assumption. Uniqueness states that each point in the reference image has a unique match, and the ordering constraint requires that the order of matches is preserved along corresponding epipolar lines. Under these additional assumptions, the correspondence problem can be solved efficiently using dynamic programming for each epipolar line. The ordering constraint is based on the physical assumption that scenes are smooth along epipolar lines. Such scenes would not only be smooth along epipolar lines, but also across epipolar lines. Therefore the matches on adjacent epipolar lines are not independent. Ohta and Kanade [11] address this issue for the case of feature based correspondence by extracting connected edge segments and performing an elegant 3D dynamic programming. Agrawal et al [2] formulate the problem as a two stage dynamic programming. In the first stage, dynamic programming is used to obtain 'K' best solutions for each scanline. In the second stage another search is performed to find solutions for each row with maximum smoothness between adjacent epipolar lines.

Roy & Cox, Ishikawa & Geiger [14, 9] have generalized the one dimensional ordering constraint to a two dimensional local cohesivity constraint and formulated the problem as finding a minimum cut through a graph. The local cohesivity assumption means that the disparities tend to be locally similar in all directions and thus across epipolar lines as well. However this generalization ignores the stronger epipolar constraint. More recently, Zabih et al. [6] have used graph cuts to find local minimum (in a strong sense) of energy functions that preserves discontinuity and

applied it to a pair of stereo images.

In this paper, we address the issue of interaction between epipolar lines through the addition of one more camera. Using this trinocular camera configuration we formulate the problem as a shortest path problem and solve it efficiently using dynamic programming techniques. One of the key features of our formulation is that it allows us to consider the uniqueness and the ordering constraint in a three camera setup. Trinocular stereo, and in general multiview stereo, has been studied before. In [12], the matching is done separately on the two pairs of images and the results are merged using relaxation. [15] introduces additional trinocular constraints and combines the views using a connectionist network relaxation algorithm. In [13], feature based matching across three widely separated views is performed which simultaneously outputs the trifocal tensor[8] between the views. Point features such as corners are determined and matched across the views using a local correlation based similarity measure augmented with a local homography estimation. The homography provides the map between interest point neighborhoods for the cross correlation affinity measure and small baseline stereo can then be subsequently applied. This is then used to estimate the trifocal tensor. However, for our algorithm, we assume that the cameras are already calibrated and thereby the trifocal tensor and also the camera matrices are known.

The organization of the rest of the paper is as follows. Section 2 describes the geometry of the trinocular stereo and section 3 presents the outline of the algorithm. The formulation of trinocular stereo as a shortest path problem is presented in section 4. In section 5 we present the results of our algorithm on real images and compare it with a standard widely used stereo algorithm based on dynamic programming on single scan lines .

## 2. Trinocular Stereo Framework

Figure 1 shows the trinocular stereo configuration used. The three cameras are placed on the vertices of a right angle triangle. This ensures that the epipolar lines for the center image corresponding to the right and the top cameras are mutually perpendicular. Each point  $p$  in the center image ( $C$ ) has disparities  $d_r$  and  $d_t$  with reference to the right and top cameras respectively. Since, in this “ideal” case the baseline distances of the top and the right camera relative to the center camera are the same, the arrangement does not increase the accuracy of the disparity of the points but helps in reducing the error. Hence it does not matter whether we refer to disparity  $d_r$  or  $d_t$ . For the remainder of the paper,  $d$  will denote the disparity with reference to the right camera.

We assume that the three cameras are fully calibrated, so that each disparity value  $d$  of a pixel  $p$  in the center image corresponds to a 3D location,  $P$ , on the line joining the

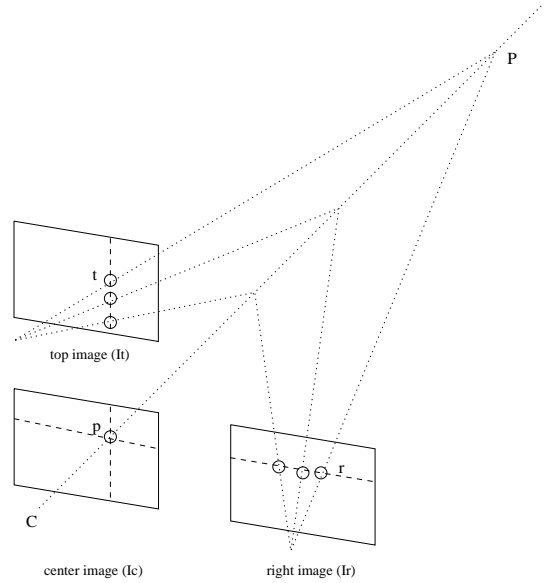


Figure 1. Trinocular Stereo Geometry

camera center to the point on the image plane. The 3D point  $P$  can then be projected in the right and top cameras to give the image points  $r$  and  $t$  respectively. In the absence of full calibration, the trifocal tensor [8] can be computed from a few known matches in the three images which can then be used to transfer point correspondences between the center and the right image to the top image. In either case, the fundamental matrices for the image pairs can then be easily recovered. Under the assumption that the image intensity is independent of the viewing direction (i.e. the surface is lambertian), the pixel intensities at points  $p$ ,  $r$  and  $t$  in the center, right and top images should be identical.

$$I_c(p) = I_r(r) = I_t(t)$$

Therefore we define the error in matching  $E(p, d)$  as

$$E(p, d) = E(p, r, t) = \max(|I_c(p) - I_r(r)|, |I_c(p) - I_t(t)|) \quad (1)$$

This error function can be used to perform the matching and ideally it will be zero for a correct match. Note that since we project the 3D point in all the three views, the cost function takes account of all the three views simultaneously. This formulation assumes that the point is visible in all the three images. If the point is occluded in one or more of the three views, then the error will be large. In that case, the point will be null matched(occluded) by this cost function. Formulation of an error function for multiple views is challenging due to the presence of occlusions[1]. In this paper, we use the simple formulation for error function above and focus on control issues in trinocular stereo.

### 3. Algorithm Outline



**Figure 2. Central Image - Test set 0**

Figure 2 is the central image from our trinocular stereo setup. The epipolar lines in the center image for the right and top cameras are also marked at the image boundaries. Although our implementation does not require the epipolar lines to be exactly horizontal or vertical (which is hard to achieve), in the discussions below, to simplify the exposition, we will assume that the epipolar lines in the center image for the right and top cameras are horizontal and vertical respectively. Suppose a point  $P$  in the center image matches a point  $R$  in the right image and let  $T$  be the corresponding point in the top image. Let  $P_l$  be a point to the left of  $P$  on the same row and  $P_r$  be to right of  $P$ . Also let  $P_t$  be a point above  $P$  in the same column and  $P_b$  be below  $P$ . Then by the ordering constraint

1. Along the horizontal epipolar line,  $P_l$  can have a match only to left of  $R$  in the right image and  $P_r$  can match to a point to the right of  $R$ .
2. Similarly for the vertical direction, matches for the points  $P_t$  and  $P_b$  must respectively lie above and below  $T$  in the top image.

Matching the horizontal or the vertical epipolar lines independently ignores the ordering constraint in the other direction. Therefore the horizontal and vertical epipolar lines should not be matched independently. Instead, we propose to match along an interleaving path through the image consisting of horizontal and vertical epipolar lines. The horizontal ordering constraint will be preserved along the horizontal portions of the path and the vertical ordering constraint will be preserved along the vertical portions. Thus there are two subproblems

1. Identify the path along which to compute disparities

2. Obtain the matches for points on this path

Since the goal of the correspondence search is to minimize the overall error in matching, this criteria can be used to recover the best path also. Therefore the best path is the path along which the error of matching is minimum. In the next section, we illustrate how the best path can be recovered using dynamic programming in 3D.

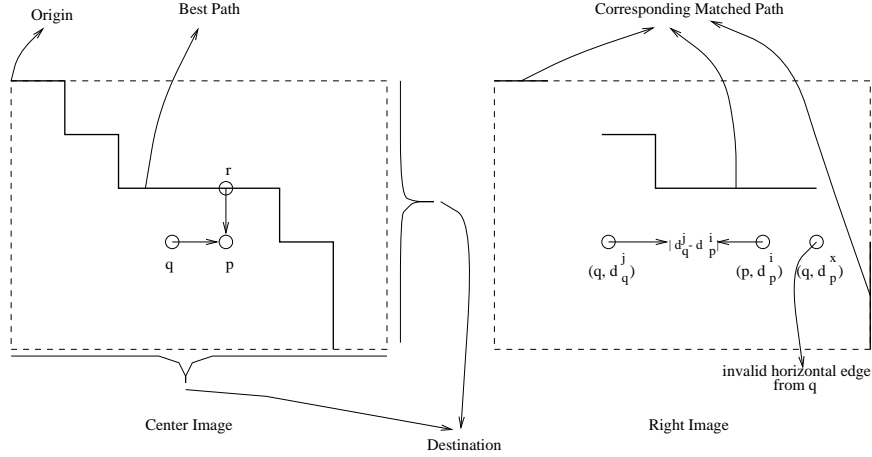
### 4. Shortest Path Formulation

In order to find the shortest path, we treat the center image as an undirected graph. The individual pixels form the nodes of this graph. The edges of this graph link each node  $(x, y)$  to its four neighbors. Note that all the three points  $(x - 1, y)$ ,  $(x, y)$  and  $(x + 1, y)$  lie on the horizontal epipolar line. Similarly the points  $(x, y - 1)$ ,  $(x, y)$  and  $(x, y + 1)$  lie on the vertical epipolar line. We also need to identify the origin and destination nodes so as to define the endpoints of the path we wish to find. The point  $(0, 0)$  is taken as the origin and all the points on the last row and last column are the destination nodes.

If there are  $D$  disparity levels, each point  $p = (x, y)$  in the center image has possible disparities  $d_p^1, d_p^2, \dots, d_p^{D-1}, d_p^D$ . Corresponding to each disparity  $d_p^i$ , we have an error  $E(p, d_p^i)$ , which is the cost of assigning disparity  $d_p^i$  to  $p$ . This error is computed using equation 1. In addition  $p$  may be occluded in one or more views, we denote this by  $d_p^0$  and the corresponding cost by  $occ$ . Similarly for points in the right and top images, let  $occ_H$  and  $occ_V$  be the cost of their being occluded. Let  $q = (x - 1, y)$  and  $r = (x, y - 1)$ . Refer to figure 3. We can arrive at  $p$  either horizontally from  $q$  or vertically from  $r$ , which corresponds to a horizontal and vertical edge respectively. Consider the horizontal edge from  $(q, d_q^j)$  to  $(p, d_p^i)$ . This edge is *valid* if it preserves the horizontal ordering constraint in the right image. i.e. the projection of the 3D point  $(q, d_q^j)$  lies to the left of the projection of the 3D point corresponding to  $(p, d_p^i)$  in the right image. In figure 3, the edge  $(q, d_q^j)$  is *invalid*. Similarly the vertical edge from  $(r, d_r^j)$  is valid if it preserves the vertical ordering constraint in the top image. Each edge has a cost associated with it. Let  $Edge_H(q, d_q^j, p, d_p^i)$  be the cost of the horizontal edge from  $(q, d_q^j)$  and  $Edge_V(r, d_r^j, p, d_p^i)$  the cost of the vertical edge. They are defined as

$$Edge_H(q, d_q^j, p, d_p^i) = \begin{cases} E(p, d_p^i) + |d_p^i - d_q^j| * occ_H & \text{valid edge} \\ \infty & \text{otherwise} \end{cases} \quad (2)$$

$$Edge_V(r, d_r^j, p, d_p^i) = \begin{cases} E(p, d_p^i) + |d_p^i - d_r^j| * occ_V & \text{valid edge} \\ \infty & \text{otherwise} \end{cases} \quad (3)$$



**Figure 3. Valid and invalid edges**

In other words, the cost of an edge which does not preserves the ordering constraint is infinite. For a valid edge, the cost is the error  $E(p, d_p^i)$  plus a term which accounts for the cost of occlusion in the other image. i.e. if  $p$  has disparity  $d_p^i$  and  $q$  has disparity  $d_q^j$ , then points between the projections of  $(p, d_p^i)$  and  $(q, d_q^j)$  in the right image will have no match in the center image as illustrated in figure 3. The optimum cost of taking a horizontal edge from  $q$  to  $(p, d_p^i)$  is denoted by  $Cost_H(p, d_p^i)$  and the optimum cost of taking a vertical edge from  $r$  to  $(p, d_p^i)$  is denoted by  $Cost_V(p, d_p^i)$ . Finally, the cost at a node,  $Cost(p, d_p^i)$  is defined as the minimum of the cost of optimum horizontal or vertical edges. They are defined recursively as follows

$$Cost_H(p, d_p^i) = \min_{j=0, \dots, D} (Cost(q, d_q^j) + Edge_H(q, d_q^j, p, d_p^i)) \quad (4)$$

$$Cost_V(p, d_p^i) = \min_{j=0, \dots, D} (Cost(r, d_r^j) + Edge_V(r, d_r^j, p, d_p^i)) \quad (5)$$

$$Cost(p, d_p^i) = \min (Cost_H(p, d_p^i), Cost_V(p, d_p^i)) \quad (6)$$

We scan the image top to bottom, left to right and at each vertex  $p$  calculate the costs for all 3D nodes  $(p, d_p^i)$ ,  $i = 0, \dots, D$ . At each node, we also keep track of the edge which gives us the minimum in equation 6 above and the corresponding horizontal or vertical node (depending on the direction of the minimum edge). This helps us backtrack to the best path and simultaneously obtain the matches for points along that path. Note that if we restrict our edges to be only vertical or only horizontal, this would correspond to the single scan line dynamic programming solution.

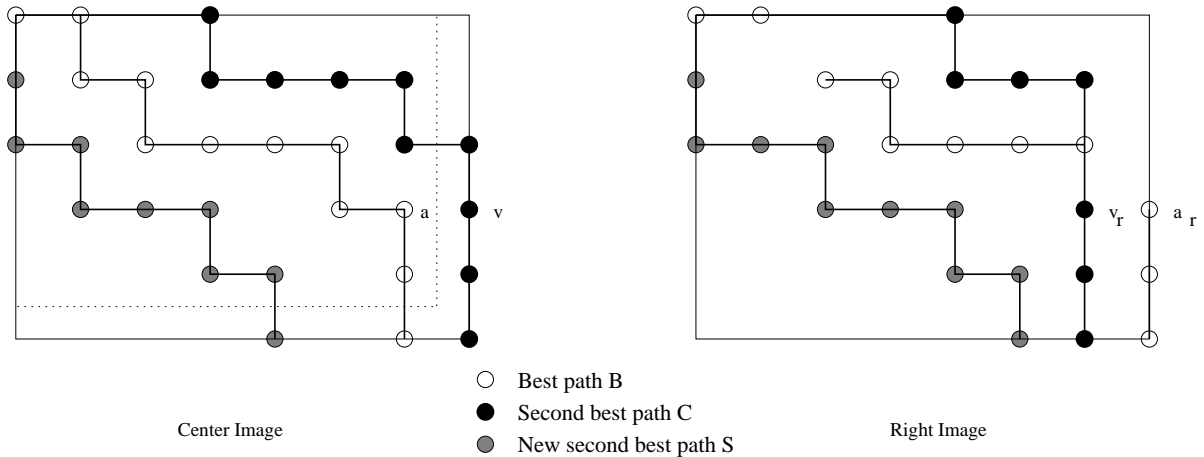
At the end of the scanning, we have the cost of all the 3D nodes for the destination points. Backtracking from the destination node with the least cost will thus simultaneously recover the best path and disparity of the points along that

path. This, however, gives undue advantage to paths which have fewer number of points. So, we scale the costs of the destination nodes by the actual number of points along that path and backtrack from the node with the minimum scaled cost.

#### 4.1. Recovering the dense disparity map

In order to recover the disparity at all points, after the path with minimum cost is found, at the next stage we find the next best path and fill disparities along that path. However, the matches obtained along the best path constrain matches in other regions by the horizontal and vertical ordering constraint. This alters the costs of all other paths. Hence the costs of all paths need to be recalculated while taking these constraints into account. In fact, the original second best cost may now even violate the ordering constraint and consequently its cost may now be infinite. In figure 4, point  $a = (x, y)$  is matched to  $a_r$  in the right and  $a_t$  in the top image. Therefore matches of all points  $(i, y)$ ,  $i = 0, \dots, x - 1$  must lie to left of  $a_r$  in the right image and matches for all points  $(x, i)$ ,  $i = y + 1, \dots, N$  must lie below  $a_t$  in the top image. The original second best path is  $C$ . Clearly,  $C$  violates the ordering constraint at node  $v$ . After the constraints from the best path are taken into account, the new second best path is  $S$  and  $S$  preserves the horizontal ordering constraint in the right camera.

This process is repeated until disparities at all points are found. When disparities at all the current destination points are determined, the points in the horizontal and vertical epipolar lines preceding the current destination points become the new destination points. This is shown by the dotted lines in figure 4.



**Figure 4. Finding next best paths**

## 4.2. Running Time

For a image with  $N$  pixels and  $D$  disparity levels at each point, the running time during each stage of the algorithm is  $O(ND^2)$ . In the worst case, we may have to go through  $N$  stages of the algorithm. (Note that for this to happen each new path that is added will find disparity at one additional new point.) Therefore the worst case running time of the algorithm using three views is  $O(N^2D^2)$ . The dynamic programming approach on separate epipolar lines requires a total running time of  $\theta(ND)$  for a pair of images. For real images of size 320x240, on a pentium 400 Mhz machine, the running time is about one hour. While this is certainly much slower than dynamic programming on separate epipolar lines, the running time is comparable to other algorithms which use maximum flows and graph cuts [5, 4, 14, 9] for interactions between epipolar lines.

## 5. Experiments & Results

In this section we compare the results of stereo obtained by our method with a “standard stereo” algorithm. This standard algorithm uses the right image for performing line by line dynamic programming as described in [7]. But since we have three images, we include the top image also for the cost function. Therefore, the cost function used for optimization in both the algorithms is the same and they differ only in the control strategy of how the disparities are filled. In our algorithm, we simultaneously enforce the ordering constraint along both directions, whereas the “standard stereo” enforces the ordering constraint only along the horizontal direction.

Figures 2, 6(a) & 7(a) show the center image for three image sequences. The epipolar lines from the top and the right camera have been overlaid for reference and define

the region within which the disparities are obtained. Figures 5(a), 6(b) & 7(b) are the disparity maps obtained by standard stereo as discussed above and figures 5(b), 6(c) & 7(c) are the results of our algorithm. Note that the tile like artifacts in the maps are due to the fact that the epipolar lines are not exactly horizontal and vertical. This made it necessary to resample the image in order to construct the graph. The range of disparities in these maps are from 10 to 70, and higher intensities correspond to larger disparities. Dark regions in both maps correspond to regions which were null matched (occlusion).

Visual comparison of the the disparity maps show that the standard stereo algorithm tends to produce a “smearing effect”, especially along edges which are perpendicular to the epipolar lines. (In this case vertical edges, since the horizontal epipolar lines have been used.) On the other hand, our algorithm is free of such artifacts. From 6(c), it is clear that our algorithm does not unduly smooth the disparity map, but preserves sharp discontinuities. Also worth noting is the fact that the occluded regions are correctly identified to lie along the contours of the persons, as expected.

Evaluation of stereo algorithms in the absence of ground truth is a difficult task. Therefore, we took four trinocular stereo sets and chose random points which were not occluded(non-null matched) by both methods and verified the matches obtained by hand matching them. Table 1 shows the accuracy ratio for each of the two methods. (A point was deemed as correctly matched if it was within a distance of two pixels from manually matched points) These figures support our claim of increased accuracy over the standard stereo method.

## 6. Conclusion

In this paper, we presented a new algorithm for trinocular stereo. Using the well known ordering constraint simultaneously in both pairs of views, we formulated the problem as a shortest path algorithm. This determines the path along which the disparity is filled first which further constrains the disparity at the other locations. The shortest paths are calculated again and the process is carried out repeatedly until disparities at all positions are determined. Experiments with real images show improved accuracy over single line by line stereo using dynamic programming.

A major direction for future research is to incorporate into the algorithm a scheme for taking into account errors in matching from the previous stages. Since our only assumption is the preservation of the ordering constraint, and at each stage we take the minimum cost path as the path along which to determine disparities, the algorithm currently has some degree of error correction built into it. But it does not guarantee that gross errors in the initial stages will not be propagated. Other areas of future research include improving the cost function and also generalizations to more than three views.

## Acknowledgement

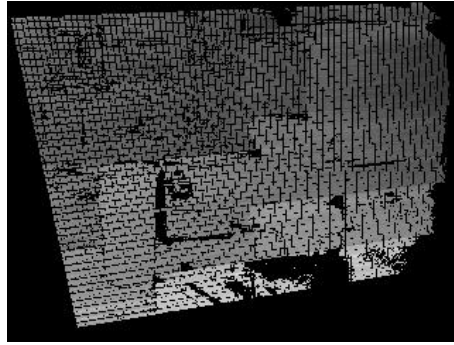
This research has been supported in part by National Science Foundation grant number NSF-EIA 01523672 and National Institute of Health under the human brain project grant number NIH-01432795. The authors would also like to thank the anonymous reviewers for their feedback.

## References

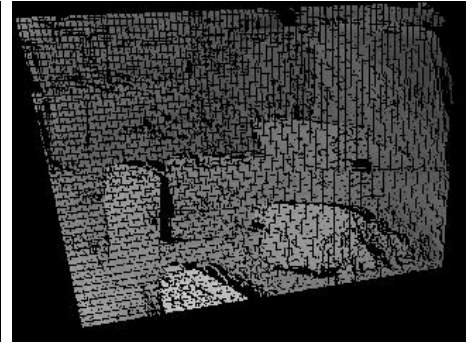
- [1] M. Agrawal and L. S. Davis. A probabilistic framework for surface reconstruction from multiple images. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, Lihue, Hawaii, December 2001.
- [2] M. Agrawal, D. Harwood, R. Duraiswami, L. Davis, and P. Luther. Three dimensional ultrastructure from transmission electron microscope tilt series. In *Proceedings Indian Conference on Vision Graphics and Image Processing*, Bangalore, India, December 2000.
- [3] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. In *Proceedings Sixth IEEE International Conference on Computer Vision*, pages 1073–1080, Mumbai, India, January 1998.
- [4] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Proceedings of the Seventh International Conference on Computer Vision*, pages 489–495, Sept. 1999.
- [5] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 648–655, 1998.
- [6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *Proc. International Conference on Computer Vision*, pages 377–384, 1999.
- [7] I. Cox, S. Hingorani, B. Maggs, and S. Rao. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, May 1996.
- [8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [9] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *Fifth European Conference on Computer Vision*, LNCS 1406, pages 232–248. Springer Verlag, June 1998.
- [10] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *PAMI*, 16(9):920–932, September 1994.
- [11] Y. Ohta and T. Kanade. Stereo by intra and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):139–154, 1985.
- [12] Y. Ohta, M. Watanabe, and K. Ikeda. Improving depth map by right-angled trinocular stereo. In *ICPR86*, pages 519–521, 1986.
- [13] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *Proc. 6th Int'l Conf. on Computer Vision*, pages 754–760, 1998.
- [14] S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proc. 6th Intl. Conference on Computer Vision*, pages 492–499, 1998.
- [15] C. V. Stewart and C. R. Dyer. The Trinocular General Support Algorithm: A Three-Camera Stereo Algorithm for Overcoming Binocular Matching Errors. In *Proc 2nd Int. Conf. on Computer Vision*, pages 134–138, 1988.

**Table 1. Accuracy Comparison of the matches obtained by the two methods**

| Index | Total Num. Points | Standard Stereo | Shortest Path Stereo |
|-------|-------------------|-----------------|----------------------|
| Set 1 | 847               | 62.0%           | 95.7%                |
| Set 2 | 862               | 80.4%           | 95.5%                |
| Set 3 | 953               | 80.6%           | 95.9%                |
| Set 4 | 603               | 84.1%           | 93.0%                |

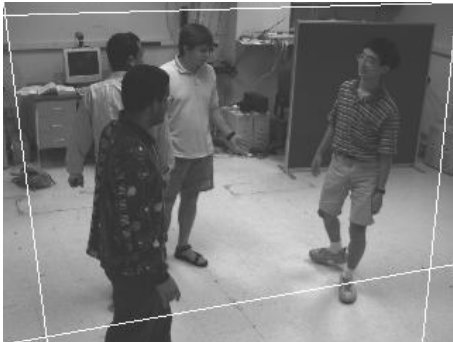


(a) Standard Stereo

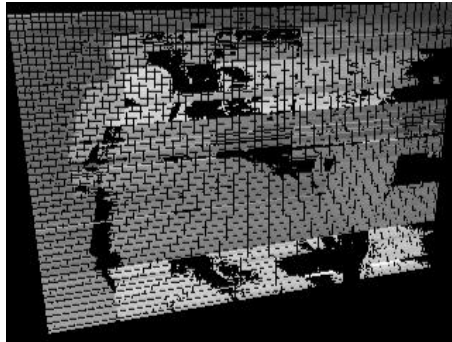


(b) Shortest Path Stereo

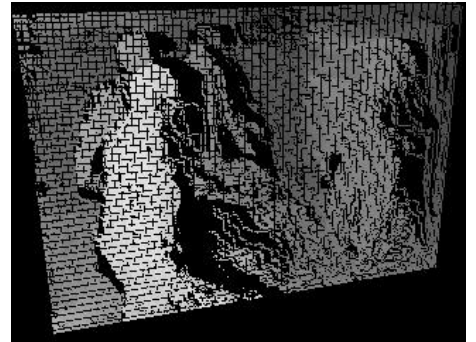
**Figure 5. Disparity maps for central image of test set 0 (Fig 2)**



(a) Center Image



(b) Standard Stereo

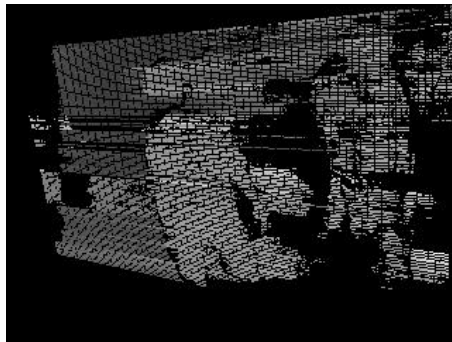


(c) Shortest Path Stereo

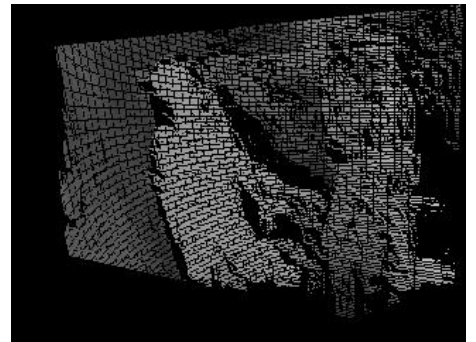
**Figure 6. Test set 1 - Disparity maps**



(a) Center Image



(b) Standard Stereo



(c) Shortest Path Stereo

**Figure 7. Test set 2 - Disparity maps**