

# The influence of shape on image correspondence

Abhijit S. Ogale and Yiannis Aloimonos

Center for Automation Research, University of Maryland, College Park, MD 20742

## Abstract

*We examine the implications of shape on the process of finding dense correspondence and half-occlusions for a stereo pair of images. The desired property of the depth map is that it should be a piecewise continuous function which is consistent with the images and which has the minimum number of discontinuities. To zeroth order, piecewise continuity becomes piecewise constancy. Using this approximation, we first discuss an approach for dealing with such a fronto-parallel shapeless world, and the problems involved therein. We then introduce horizontal and vertical slant to create a first order approximation to piecewise continuity. We highlight the fact that a horizontally slanted surface (ie. having depth variation in the direction of the separation of the two cameras) will appear horizontally stretched in one image as compared to the other image. Thus, while corresponding two images,  $N$  pixels on a scanline in one image may correspond to a different number of pixels  $M$  in the other image, which has consequences with regard to sampling and occlusion detection. We also discuss the asymmetry between vertical and horizontal slant, and the central role of non-horizontal edges in the context of vertical slant. Using experiments, we discuss cases where existing algorithms fail, and how the incorporation of new constraints provides correct results.*

## 1. Introduction

The dense correspondence problem consists of finding a unique mapping between the points belonging to two images of the same scene. If the camera geometry is known, the images can be rectified, and the problem reduces to the stereo correspondence problem, where points in one image can correspond only to points along the same scanline in the other image. If the geometry is unknown, then we have the optical flow estimation problem. In both cases, regions in one image which have no counterparts in the other image, are referred to as occlusions (or more correctly as *half-occlusions*). In this paper, we demonstrate that scene shape has profound implications for any process of establishing

point correspondence and occlusion detection. In particular, we show that correspondence, segmentation, occlusion detection, and shape estimation have to be done in concert.

### 1.1. Previous work

There exists a considerable body of work on the dense stereo correspondence problem. Scharstein and Szeliski [20] have provided an exhaustive comparison of dense stereo correspondence algorithms. Most algorithms generally utilize local measurements such as image intensity (or color) and phase, and aggregate information from multiple pixels using smoothness constraints. The simplest method of aggregation is to minimize the matching error within rectangular windows of fixed size [17]. Better approaches utilize multiple windows [9, 8], adaptive windows [11] which change their size in order to minimize the error, shiftable windows [4, 22], all of which give performance improvements at discontinuities.

Global approaches to solving the stereo correspondence problem rely on the extremization of a global cost function or energy. The energy functions which are used include terms for local property matching ('data term'), additional smoothness terms, and in some cases, penalties for occlusions. Depending on the form of the energy function, the most efficient energy minimization scheme can be chosen. These include dynamic programming [16], simulated annealing [10, 1], relaxation labeling [21], non-linear diffusion [19], maximum flow [18] and graph cuts [5, 12]. Maximum flow and graph cut methods provide better computational efficiency than simulated annealing for energy functions which possess a certain set of properties. Some of these algorithms treat the images symmetrically and explicitly deal with occlusions (eg. [12]). The uniqueness constraint [14] is often used to find regions of occlusion. Egnal and Wildes [7] provide comparisons of various approaches for finding occlusions. Recently, some algorithms [3] have explicitly incorporated the estimation of slant while performing the estimation of horizontal disparity. Lin and Tomasi [13] explicitly model the scene using smooth surface patches and also find occlusions; they initialize their disparity map with integer disparities obtained using graph cuts, after which surface fitting and segmentation are per-

formed repeatedly. Previously, Deverney and Faugeras [6] used local image deformations to obtain differential properties of 3D shapes directly.

## 1.2. Organization of the paper

In this paper, Section 2 deals with the simpler problem of estimating correspondence in Flatland, a world which consists of flat fronto-parallel surfaces only. In this shape-free world, we argue that correspondence and segmentation, which appear to be chicken-and-egg problems, can only be solved together. Section 3 addresses the problem in the presence of horizontally slanted surfaces (a portion of this work appeared in [15]). Section 4 discusses the effects of vertical slant, and Section 5 concludes by presenting experimental results and comparisons.

## 2. Correspondence in Flatland

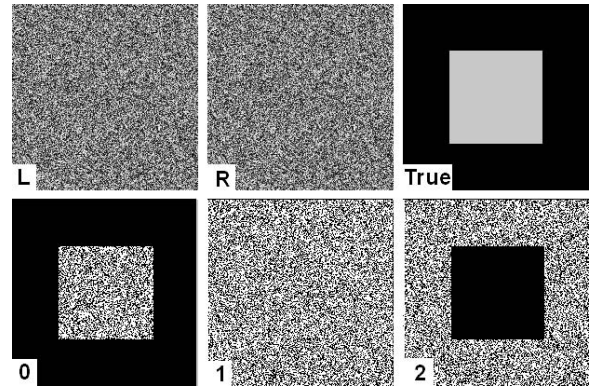
### 2.1. Chicken-and egg problems

Establishing correspondence between two images of a scene involves first selecting a local metric, such as the intensity (gray level) or color of a pixel which forms the basis for local comparisons. However, matching on the basis of such local information alone is almost impossible since many pixels have similar intensity or color. To reduce the correspondence possibilities for a pixel to a single possibility, regions around that pixel must be used along with additional continuity or smoothness assumptions about the scene depth. Thus, information around a pixel must be *aggregated* to obtain a unique match. Enforcing smoothness without a prior knowledge of depth discontinuities (segmentation) will inevitably lead to errors, especially near the discontinuities. Hence, prior knowledge of the segmentation is essential in order to correctly define regions around a pixel for information aggregation. Conversely, if exact correspondence is known, the segmentation may be easily deduced, leading to a chicken-and-egg problem. First, we study the relationship between correspondence and segmentation by working in a shapeless world called Flatland - a world which contains only fronto-parallel surfaces. Then, in Section 3, we introduce shape into the picture, and proceed to make explicit the relationship between shape and correspondence.

### 2.2. Connected matching regions

Let  $I_1(x, y)$  and  $I_2(x, y)$  be a given pair of rectified stereo images. The absolute intensity difference image is found using equation 1, where  $\delta_x$  denotes the relative horizontal shift between the two input images.

$$\Delta I(x, y, \delta_x) = |I_1(x, y) - I_2(x + \delta_x, y)| \quad (1)$$



**Figure 1. Top row: Left image, Right image, True disparity. Bottom row: Absolute intensity difference of left and right images for horizontal shift  $\delta_x = 0, 1, 2$**

The first row of Figure 1 shows a random dot pair of stereo images and the true disparity map. The second row shows the absolute intensity difference images for three horizontal shifts  $\delta_x = 0, 1, 2$ . If we observe the intensity difference images (second row), we notice that large connected regions of matching pixels (shown in black) appear for certain values of the shift. By the word ‘match’, we mean that the absolute intensity difference is below a certain threshold  $t$ , ie.  $\Delta I(x, y, \delta_x) < t$ . In this paper, we will not discuss methods for automatic selection of the threshold  $t$ .

*The appearance of large connected sets of matching pixels is the first observation of interest.*

In the case of the random-dot pair, the background matches for  $\delta_x = 0$  forming a large connected region, and the central square matches for  $\delta_x = 2$ . We know from the true disparity map that these shifts correspond to the correct disparities of the background and the square. However, we also notice that some pixels in the central square will match and form smaller connected regions even when the shift is wrong (not equal to 2). The same is true for the background pixels. *Thus, a pixel may form a part of a connected matching region even when the shift does not correspond to the true shift.* So how do we choose the correct shift for a pixel?

### 2.3. Boundaries and connectivity maximization

Recall our definition of Flatland - a world containing only fronto-parallel surfaces. Consider a uniformly colored region  $R_1$  in image  $I_1$  having a disparity  $\delta_x$ . It corresponds to a region  $R_2$  in the image  $I_2$ . Thus, if we shift image  $I_1$  by  $\delta_x$  and overlay it on  $I_2$ , then regions  $R_1$  and  $R_2$  will overlap and match perfectly and yield a connected region having an area equal to the size of  $R_1$  (and  $R_2$ ). However, if the shift is not  $\delta_x$  but has some other value, parts of  $R_1$  and  $R_2$  may

still overlap and yield some connected matching region.

The area of overlap will be maximum only when the boundaries of  $R_1$  and  $R_2$  match perfectly, which occurs only for the true shift  $\delta_x$ .

For all other shifts, the connected matching area will be less.

Similarly, in case the region  $R_1$  (and hence  $R_2$ ) is textured, connected matching regions will also be obtained for shifts other than the true shift  $\delta_x$ . For example, if the regions contain a periodic texture such as a checkerboard, with square size  $\lambda$ , then we will obtain connected matching regions even if the shifts are  $\delta_x + 2m\lambda$ , where  $m$  is an integer. Even if this happens, the largest connected region will occur only when the boundaries of  $R_1$  and  $R_2$  match, which happens only if the shift equals the true shift  $\delta_x$ . It is clear that only the knowledge of region boundaries allows us to assign correct shifts to the interior pixels. Maximizing the area of connected matching regions around a pixel is intimately related to the matching of region boundaries. Maximizing the size of matching regions is equivalent to minimizing the depth segmentation. This *principle of minimum segmentation* is a more general constraint than connectivity maximization.

Using the above arguments, it is straightforward to show that in Flatland: *for any image pixel  $(x, y)$ , the correct shift  $\delta_x$  maximizes the area  $A(x, y, \delta_x)$  of the connected matching region containing that pixel, and vice-versa.*

## 2.4. Occlusions and uniqueness

The *uniqueness constraint* enforces a one-to-one correspondence between pixels in the two images. Thus, there is a competition between pairs  $(p_L, p_R)$  of pixels, where  $p_L$  is a pixel in the left image, and  $p_R$  is a pixel in the right image. The competition is based on the principle of maximum connectivity, outlined in the earlier section. If a pair  $(p_L, p_R)$  wins, it automatically precludes the existence of all pairs of the form  $(p_L, p_{R'})$  and  $(p_{L'}, p_R)$ , such that  $p_{L'} \neq p_L$  and  $p_{R'} \neq p_R$ . Some pixels, which do not form a part of any of the winning pairs, are the occlusions. It is possible to enforce the uniqueness constraint within the correspondence search itself as it progresses: whenever we assign a new partner to a given pixel, we make sure that its previous partner (if it was previously paired) is marked as unpaired.

## 2.5. An algorithm for Flatland

Our algorithm for Flatland consists of the following steps:

1. For every shift  $\delta_x \in \{\delta_1, \delta_2, \dots, \delta_k\}$ , do
  - (a) Shift the left image  $I_L$  horizontally by  $\delta_x$  to get  $I'_L$

- (b) Match  $I'_L$  with  $I_R$ , and build connected components
- (c) For each pixel, if the connectivity increases, update left and right disparity maps, preserving uniqueness.

Thus, the process consists of matching pixels (using thresholded absolute intensity differences) for various shifts (disparity/flow candidates), finding connected components and maximizing a measure of the connectivity for each pixel. Real world scenes are quite unlike Flatland, since they rarely consist of fronto-parallel surfaces. In the following sections, we shall identify new issues which arise in the presence of slanted surfaces.

## 3. Horizontal slant

Let us leave Flatland by introducing horizontally slanted surfaces into the scene, ie. the depth on such surfaces changes as we move along the X-axis (horizontally), and does not change if we move along the Y-axis. Let us also assume for simplicity that we are using a stereo system with a parallel viewing geometry and in which the cameras are separated only by a translation along the X-axis. Our system therefore provides us with rectified images.

### 3.1. Unequal projection lengths and interval matching

Figure 2(a) shows that a horizontally slanted line  $AB$  in the scene projects onto the line segment  $a_1b_1$  in camera  $C_1$ , and  $a_2b_2$  in camera  $C_2$ . Clearly, the lengths of  $a_1b_1$  and  $a_2b_2$  are not equal. Assume that the cameras have focal length equal to 1. Let the point  $A$  have coordinates  $(X_A, Z_A)$  in space with respect to camera 1, and point  $B$  have coordinates  $(X_B, Z_B)$ , where the  $X$ -axis is along the scanline, and the  $Z$ -axis is normal to the scanline. Then, if the cameras are separated by a translation  $t$ , we can immediately find the lengths  $L_1$  and  $L_2$  of the projected line segments in the two cameras.

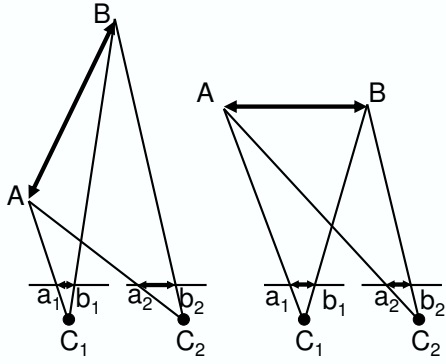
$$\begin{aligned} L_1 &= X_B/Z_B - X_A/Z_A \\ L_2 &= (X_B - t)/Z_B - (X_A - t)/Z_A \end{aligned} \quad (2)$$

Clearly, in general,  $L_1$  and  $L_2$  are not equal. For the fronto-parallel line shown in Figure 2(b),  $Z_A = Z_B = Z$ , hence

$$L_1 = L_2 = (X_B - X_A)/Z \quad (3)$$

Thus, we have the following:

- *Except for the fronto-parallel case, horizontally slanted line segments in space will always project onto segments of different lengths in the two cameras.*



**Figure 2. (a) unequal projection lengths of a horizontally slanted line (b) equal projection lengths of a fronto-parallel line**

- Consequently,  $N$  pixels on a scanline in one image can correspond to a different number of pixels  $M$  on a scanline in the other image.

We must ensure that our stereo algorithms permit unequal correspondences of this nature; hence, an interval on a scanline in one image must be matched to an interval on a scanline in the other image, where the two intervals being matched may have different lengths. Note that the scanline is treated as a continuous entity rather than a discrete pixelized entity.

*Conclusion: We must perform interval matching instead of pixel matching.*

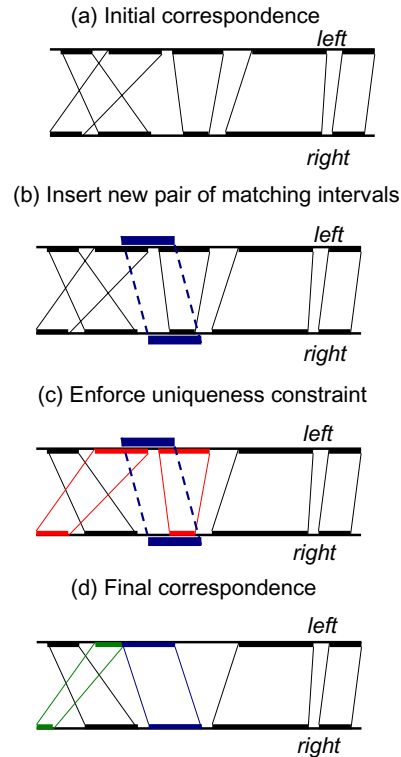
### 3.2. Slant affects Sampling

Since a horizontally slanted line segment in space has different projection lengths in the two cameras, its intensity function is also sampled differently by the two cameras. Birchfield and Tomasi [2] have provided a very useful method for matching pixel intensities, which is used by many of the best performing stereo algorithms. This method linearly interpolates the intensity function between neighboring pixels and cannot be applied directly in the presence of horizontal slant, due to the unequal sampling.

We must first resample each scanline correctly, and then apply the Birchfield-Tomasi matching method. In other words, we first stretch one of the scanlines, by an amount related to the horizontal slant we are considering, and then match this stretched scanline with the other unstretched scanline using the Birchfield-Tomasi matching method as usual. For example, if we are considering the linear correspondence function  $x_R = mx_L + d$  between points of camera L and R, then we must stretch the image of camera L by a factor  $m$  before performing the intensity based matching.

Thus, we first compute  $I_L^{min}$  and  $I_L^{max}$  for the unstretched scanline  $I_L$ , then stretch all three by a factor  $m$ , and then apply the remainder of the Birchfield-Tomasi method. As we try various values of the slant, we appropriately resample the scanlines before matching.

*Conclusion: Stretch one of the images first and then match.*



**Figure 3. The modified uniqueness constraint operates by preserving a one-to-one correspondence between intervals on the left and right scanlines, instead of pixels.**

### 3.3. Occlusions and the new interval uniqueness constraint

The uniqueness constraint [14] is often used to find occlusions. In its present form, the uniqueness constraint forces a one-to-one correspondence between pixels in the two images. In the end, the unpaired pixels are the occlusions. However, since horizontal slant allows  $N$  pixels in one image to match with a different number of pixels  $M$  in the other image, we can no longer impose a one-to-one correspondence for finding occlusions. We propose a new uniqueness constraint which enforces a *one-to-one mapping between continuous intervals* (line segments) in the two

scanlines, instead of pixels. An interval in one scanline may correspond to an interval of a different length in the other scanline, as long as the correspondence is unique. This is equivalent to enforcing uniqueness in the scene space instead of the image space, hence we may also refer to this constraint as the 3D uniqueness constraint. Figure 4 shows using a real example how intervals of different length correspond to each other, leaving behind the occlusions.

Figure 3 shows how the new uniqueness constraint can be used. Part (a) shows an existing one-to-one correspondence between intervals on the left and right scanlines. This denotes an intermediate state in the progress of a stereo matching and segmentation algorithm. Notice that the intervals may correspond in any order (ie. the ordering constraint is not needed). Now, in part (b), we wish to insert a new pair of corresponding intervals, shown in blue. (This new pair of matching intervals improves upon the existing matches according to some energy metric which depends on the stereo algorithm being used). In part (c), we see that the insertion of this pair of intervals conflicts with existing intervals (shown in red). In order to enforce uniqueness, the red pair of intervals on the right must be removed, while the red pair of intervals on the left must be resized. In part (d), we see the new correspondences. The interval pair which was resized is shown in green, and the newly inserted pair is shown in blue.

Figure 4 shows the idea of interval mapping and occlusions detection using a real example. In this figure, we see how intervals of unequal length can correspond uniquely to yield occlusions.

*Conclusion: There exists a one-to-one mapping between intervals (possibly having unequal lengths), and not between pixels.*

### 3.4. An algorithm to deal with horizontal slant

We now describe a simple scanline algorithm which implements the ideas presented above; this algorithm also uses the concept of connectivity maximization presented in Section 2 along scanlines instead of the whole image, and simultaneously searches the space of possible disparities and horizontal slants. It processes a pair of scanlines  $I_L(x)$  and  $I_R(x)$  at a time without using any vertical consistency constraints. Horizontal disparities  $\Delta_L(x)$  are assigned to the left scanline within a given range  $[\Delta_1, \Delta_2]$ , and  $\Delta_R(x)$  to the right scanline in the range  $[-\Delta_2, -\Delta_1]$ . The disparities are not assigned to pixels, but continuously over the whole scanline. The disparities are not directly estimated, but instead, we search for functions  $m_L(x)$  and  $d_L(x)$  for the left scanline, and  $m_R(x)$  and  $d_R(x)$  for the right scanline, such that given a point  $x_L$  on the left scanline, its corresponding point  $x_R$  in the right scanline would be

$$x_R = m_L(x_L) \cdot x_L + d_L(x_L) \quad (4)$$

and reciprocally:

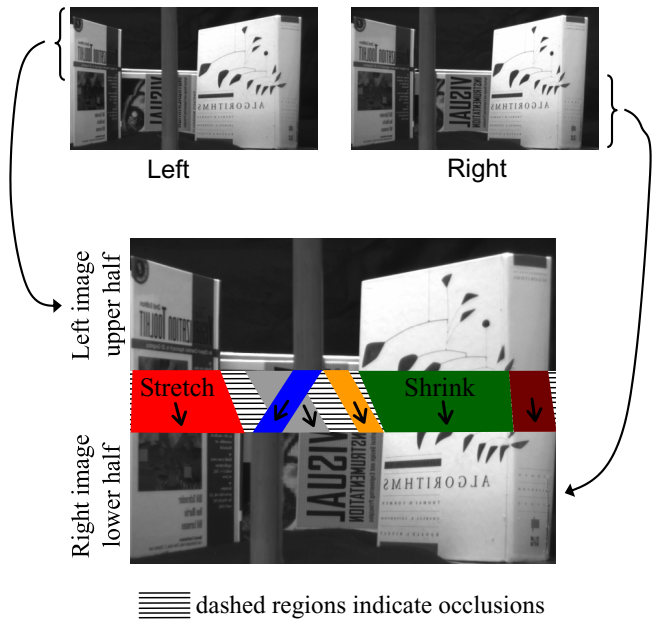
$$x_L = m_R(x_R) \cdot x_R + d_R(x_R)$$

Clearly,

$$\begin{aligned} m_R(x_R) &= 1/m_L(x_L) \\ d_R(x_R) &= -d_L(x_L)/m_L(x_L) \end{aligned}$$

The disparities are then computed as:

$$\begin{aligned} \Delta_L(x_L) &= x_R - x_L = (m_L(x_L) - 1) \cdot x_L + d_L(x_L) \\ \Delta_R(x_R) &= x_L - x_R = (m_R(x_R) - 1) \cdot x_R + d_R(x_R) \end{aligned} \quad (5)$$



**Figure 4. Top: stereo pair of images. Bottom: Corresponding intervals on the left and right scanlines can have different length. The order (left-right) of matching intervals can also change (see the blue and gray intervals).**

The functions  $m_L$  and  $m_R$  are the horizontal slants, which allow line segments of different length in the two scanlines to correspond. The scanlines are represented continuously by linearly interpolating intensities between pixel locations. Thus, if  $m_L = 2$ , then the left scanline is stretched (resampled) by a factor of 2, and then matched with the unstretched right scanline using the Birchfield-Tomasi method. Due to the stretching of one scanline before performing the intensity based matching, we are automatically modifying the traditional Birchfield-Tomasi method to properly deal with horizontal slant. For each possible  $m_L$

and  $d_L$ , absolute intensity differences between corresponding points are computed, and thresholded by a threshold  $t$ . The best value of  $m_L$  and  $d_L$  for a point is chosen such that it maximizes the size of the matching line segment containing that point (ie. the maximum connectivity approach of Section 2).

The values of the horizontal slant which are to be examined are provided as inputs, ie.  $m_L, m_R \in M$ , where  $M = \{m_1, m_2, \dots, m_k\}$ , such that  $m_1, m_2, \dots, m_k \geq 1$ . The disparity search range  $[\Delta_1, \Delta_2]$  is also provided as an input. In order to find the occlusions, we enforce the uniqueness constraint in its modified form as shown in Figure 3. We maintain a one-to-one correspondence between intervals in the two scanlines. Hence, at any stage of the process, we have a set  $S_L$  of non-overlapping intervals in the left scanline and a set  $S_R$  of non-overlapping intervals in the right scanline. An interval  $i$  is of the form  $[x_1, x_2)$ . The uniqueness constraint enforces a one-to-one mapping  $U$  between the elements of  $S_L$  and the elements of  $S_R$ . When a new corresponding pair of intervals  $i_L$  and  $i_R$  is found, the segment previously corresponding to  $i_L$  is removed if present, and the same is done for  $i_R$ . Then,  $i_L$  is added to  $S_L$ , and  $i_R$  to  $S_R$ , and the one-to-one mapping in  $U$  is updated. Thus, we always ensure that a line segment in the left scanline uniquely maps to a line segment in the right scanline. In the end, line segments which remain unmapped are the occlusions. In our implementation, we have used hash-tables to maintain the interval information and detect overlaps.

## 4. Vertical slant

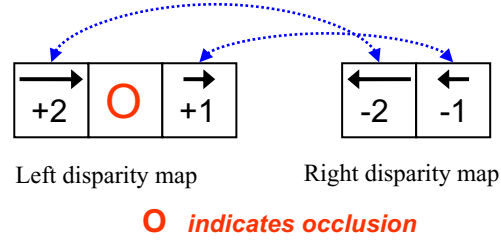
### 4.1. Fundamental differences between vertical and horizontal slant

Assume that we are given a rectified stereo pair of images. Due to the discrete (pixelized) nature of the images, changes in disparity as we move from left to right along a scanline may be caused by one of two factors:

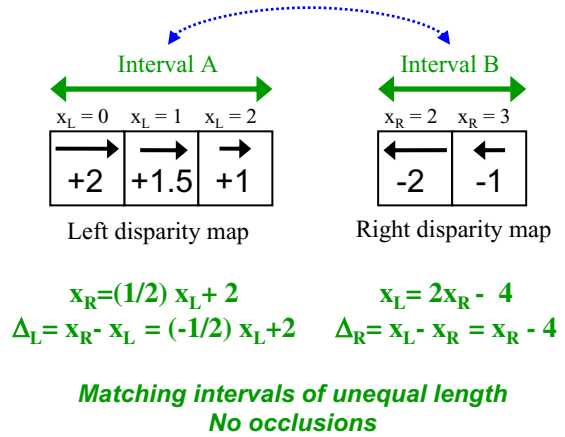
- There exists a depth discontinuity (as seen in Figure 5(a)), or
- The pixels form a part of a horizontally slanted surface (Figure 5(b)).

In this case, we can distinguish between these two possibilities because only a true depth discontinuity will cause an occlusion to appear (as seen in Figure 5(a)). However, if we move vertically and find a disparity change (as seen in Figure 5(c)), we have no way of distinguishing whether the vertical change is caused by a discontinuity or by a vertically slanted surface, since neither causes occlusions to appear. Thus, there is a fundamental difference between horizontal and vertical slant.

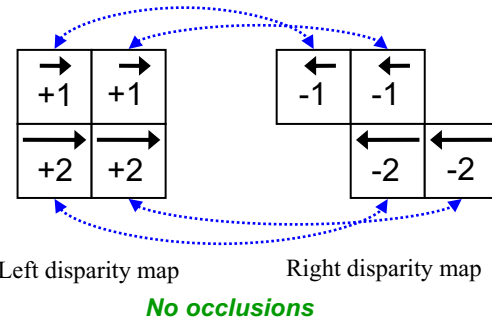
(a) Horizontal change in horizontal disparity (due to depth discontinuity; no slant)



(b) Horizontal change in horizontal disparity (due to horizontal slant)



(c) Vertical change in horizontal disparity (due to vertical slant or depth discontinuity)



**Figure 5. Top: Horizontal changes in horizontal disparity due to a discontinuity create an occlusion. Middle: Horizontal changes in horizontal disparity due to horizontal slant lead to stretching/shrinking but no occlusions. Bottom: Vertical changes in horizontal disparity due to discontinuity and vertical slant cannot be distinguished (since no occlusions occur in either case).**

## 4.2. Vertical connectivity and non-horizontal edges

Since we cannot distinguish whether purely vertical changes in disparity are due to a true discontinuity or due to a vertically slanted surface, additional assumptions must be employed if we are to enforce any vertical consistency constraints on the disparity map. Consider this example: in the image, if we have a horizontal intensity edge (a horizontal line), we have two possibilities (a) this edge corresponds to a depth discontinuity, and pixels separated by it do not lie on the same surface, or (b) the edge is just an intensity edge, and pixels separated by it lie on the same surface. If we commit the error of assuming that the pixels separated by the horizontal edge are connected and it happens to be a discontinuity, our solution will yield a disparity map without the depth discontinuity, which is clearly incorrect. It is safer to assume that such pixels are *not* connected, to allow the possibility that there may be a depth discontinuity.

*Therefore, vertical neighbors separated by a horizontal edge or no edge at all should not be connected.*

Also, if we have two images of a single-colored object, and we assume vertical connections in the interior, then we will get a single disparity in the entire interior (when maximum overlap of the images takes place) instead of a vertical gradient. Thus, we cannot assume that disparity is vertically constant even if two vertical neighbors have the same color/intensity. *Disparity can change even when there is no change in color or intensity.* (Note that we *can* assume that disparity is *continuous*, but not necessarily *constant*, if intensity or color do not vary in a region.)

However, if we have a non-horizontal edge running across the image, it will cause occlusions to appear if it is a discontinuity, and no occlusions will appear if both sides of it lie on the same surface. This distinguishing ability allows us to make the assumption that:

*Vertical neighbors lying on non-horizontal edges should be connected.*

## 5. Experiments

We have shown in the previous sections that horizontal and vertical slant play a critical role in the estimation of correspondence and occlusions. The first row of Figure 6 shows a stereo pair of images in which the blue object is horizontally slanted (ie. depth varies from left to right), and the second row shows a stereo pair in which the blue object is vertically slanted. The third column of this figure shows the results of the graph cuts [12] algorithm, while the fourth column shows our results for each of these stereo pairs. In these results, occlusions are shown in red. The graph cuts result was obtained using software kindly provided by the authors ([www.cs.cornell.edu/People/vnk/software.html](http://www.cs.cornell.edu/People/vnk/software.html)). It is clear that for both these stereo pairs, the graph cuts result

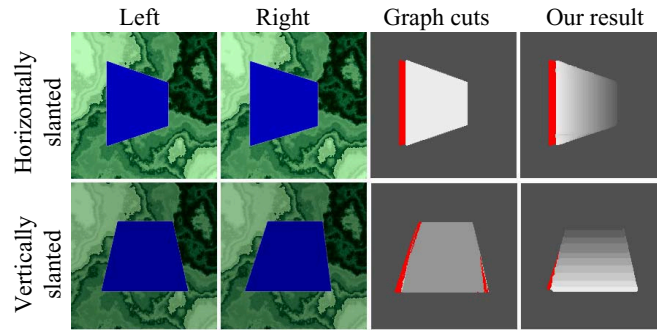
gives a constant disparity for the blue object, while our result correctly shows the slant and still finds the occlusions.

We expect that the constraints presented above will improve the results of many existing dense stereo algorithms in both qualitative and quantitative ways. However, for the sake of completeness, we compare our results with other algorithms using the test procedure created by Scharstein and Szeliski [20] available at [www.middlebury.edu/stereo](http://www.middlebury.edu/stereo). They compare the disparity map  $d_{out}$  generated by an algorithm to the true disparity  $d_{true}$ , and the pixels which deviate by more than 1 unit from their true disparity are labeled as ‘bad’ pixels. The percentage of bad pixels in the entire image, in the untextured regions and near depth discontinuities are used to compare the results of various algorithms. The percentages of bad pixels and ranks (shown in brackets) in each category are reported below, which were obtained by submitting our disparity maps (Figure 7) to the web-based evaluation program created by Scharstein and Szeliski. Our results were ranked sixth overall.

Sequence	all	untex	disc
Tsukuba	1.77 (6)	0.95 (5)	9.48 (7)
Sawtooth	0.61 (4)	0.17 (11)	5.05 (6)
Venus	3.00 (20)	5.22 (2)	7.63 (8)
Map	0.21 (2)	-	3.01 (4)

## 6. Conclusion

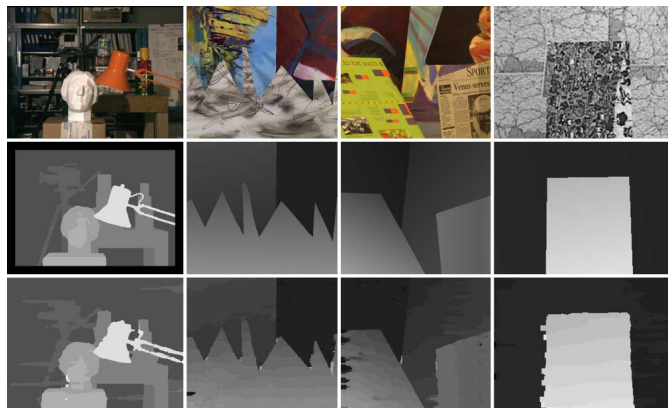
We have analyzed the effects of shape in establishing dense point correspondence between a stereo pair of images. Ideally, it is desirable to model the disparity map as a piecewise-continuous function having the minimum number of pieces, which upto zeroeth order, can be approximated by a piecewise-constant function. The idea of connectivity maximization was proposed for this flat fronto-parallel world, which is equivalent to finding the minimum segmentation. Proceeding to a first order approximation, we then examined the effects of horizontal slant: unequal projection lengths, sampling issues, and invalidation of the uniqueness constraint for finding occlusions. It was shown that interval matching and a new uniqueness constraint are required to handle horizontal slant. We then highlighted qualitative differences between horizontal and vertical slant, and discussed the importance of non-horizontal edges for the latter. Experimental evidence for these new ideas was presented in the last section. In conclusion, we have shown that correspondence, segmentation, occlusion detection, and shape estimation influence each other and have to be solved in concert.



**Figure 6. Columns 1 to 4: Left image, right image, graph cuts result for the left disparity map, our result for left disparity map. Row 1: horizontally slanted object, Row 2: vertically slanted object. Occlusions are shown in red.**

## References

- [1] S. T. Barnard. Stochastic stereo matching over scale. *IJCV*, 3(1):17–32, 1989.
- [2] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Trans. PAMI*, 20(4):401–406, 1998.
- [3] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. *ICCV*, 1:489–495, 1999.
- [4] A. F. Bobick and S. S. Intille. Large occlusion stereo. *IJCV*, 33(3):181–200, Sept 1999.
- [5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. PAMI*, 23(11):1222–1239, Nov 2001.
- [6] F. Devernay and O. Faugeras. Computing differential properties of 3-D shapes from stereoscopic images without 3-D models. *CVPR*, pages 208–213, 1994.
- [7] G. Egnal and R. Wildes. Detecting binocular half-occlusions: empirical comparisons of five approaches. *IEEE Trans. PAMI*, 24(8):1127–1133, Aug 2002.
- [8] A. Fusiello, V. Roberto, and E. Trucco. Efficient stereo with multiple windowing. *CVPR*, pages 858–863, June 1997.
- [9] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *ECCV*, pages 425–433, 1992.
- [10] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. PAMI*, 6(6):721–741, Nov 1984.
- [11] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: theory and experiment. *IEEE Trans. PAMI*, 16(9):920–932, 1994.
- [12] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. *ICCV*, pages 508–515, July 2001.
- [13] M. Lin and C. Tomasi. Surfaces with occlusions from layered stereo. *CVPR*, 1:I–710–I–717, June 2003.
- [14] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proc. Royal Soc. London B*, 204:301–328, 1979.
- [15] A. S. Ogale and Y. Aloimonos. Stereo correspondence with slanted surfaces: critical implications of horizontal slant. *CVPR*, June 2004.
- [16] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Trans. PAMI*, 7(2):139–154, March 1985.
- [17] M. Okutomi and T. Kanade. A multiple baseline stereo. *IEEE Trans. PAMI*, 15(4):353–363, April 1993.
- [18] S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. *ICCV*, pages 492–499, 1998.
- [19] D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. *IJCV*, 28(2):155–174, 1998.
- [20] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1):7–42, April 2002.
- [21] R. Szeliski. Bayesian modeling of uncertainty in low-level vision. *IJCV*, 5(3):271–302, Dec 1990.
- [22] H. Tao, H. Sawhney, and R. Kumar. A global matching framework for stereo computation. *ICCV*, 1:532–539, July 2001.



**Figure 7. Top row (Left frames), Middle row (ground truth), Bottom row (our results). Occlusions were filled in as required by the evaluation procedure.**