M.I.T. Media Lab Perceptual Computing Group Technical Report No. 246

Incorporating Intensity Edges in the Recovery of Occlusion Regions

Stephen S. Intille and Aaron F. Bobick

intille@media.mit.edu and bobick@media.mit.edu
Perceptual Computing Group
The Media Laboratory, MIT
20 Ames St., Cambridge, MA 02139

Abstract: A method for incorporating intensity edge information into the recovery of occlusion regions using a pixel-based stereo algorithm is presented. We review the construction of disparity-space images and their use to solve the stereo occlusion problem. We show the impact of intensity edges on the disparity space images, and extend our stereo technique to use information about intensity discontinuities at occlusion edges. Our approach suggests that for occlusion region analysis, two separate computations and results – one for right occlusion regions and one for left occlusion regions – may be preferable to conventional cyclopean processing.

Key words: stereo; occlusion; intensity edges; dynamic-programming stereo; disparity-space.

1 Introduction

Accurate occlusion recovery is one of the most valuable tasks a stereo algorithm can perform. Occlusion separates the world into "layers" — three-dimensionally isolated regions — which can be the basis for a natural decomposition of a scene[13]. Unfortunately, the focus of stereo research has been obtaining dense, accurate depth maps of scenes, with occlusion analysis generally treated as an afterthought, postponed until stereo matching is complete.

Recent work on stereo occlusion, however, has shown that occlusion processing can be incorporated directly into stereo matching[1, 7, 5, 9]. Stereo imagery contains both occlusion edges and occlusion *regions*[7]. Occlusion regions are spatially coherent groups of pixels that appear in one image and not in the other. These occlusion regions are caused by occluding surfaces and can be used directly in stereo and occlusion reasoning.

Monocular images, however, also contain information about occlusion. Different objects in the world have varying texture, color, and illumination. Therefore jumps between these objects or parts of the same object nearly always generate intensity edges in a monocular image. In this paper we develop a method of stereo matching and occlusion analysis that exploits the relationship between occlusion jumps and intensity edges.

We begin by briefly describing our disparity-space stereo algorithm, presented in detail elsewhere[9]. After showing the impact of intensity edges on the disparity space images, we extend our stereo technique to use information about intensity discontinuities at occlusion edges while still enforcing the occlusion constraint defined by the occlusion regions in a pixel-based matching process. This extension allows for the accurate recovery of the occlusion regions in each image of the stereo pair.

2 Disparity Space Images

A *disparity space image* (DSI) is an explicit representation of matching space; it is related to figures that have appeared in previous work[4, 7, 14, 11]. The DSI representation for i^{th} scanline in the following way: Select the i^{th} scanline of the left and right images, s_i^L and s_i^R respectively, and slide them across one another one pixel at a time. At each step, the scanlines are subtracted and the result is entered as the next line in the DSI. The DSI representation stores the result of subtracting every pixel in s_i^L with every pixel s_i^R and maintains the spatial relationship between the matched points. As such, it may considered an *(x, disparity)* matching space, with x along the horizontal, and disparity along the vertical. Given two images I_L and I_R the value of the DSI is given by:

$$\mathrm{DSI}_i^R(x,d) = I_R(x,i) - I_L(x+d,i) \qquad (1)$$

when $0 \le (x+d) < N$, $0 \le d < D$, and $0 \le x < N$. The superscript of R on DSI^R indicates the right DSI. DSI_i^L is simply a negated, skewed version of the DSI_i^R. We use these dual images, as opposed to a cyclopean representation such as [11], to make explicit the difference between occlusion regions (pixels visible only in one image) and occlusion jumps (instantaneous change in disparity in one image corresponding to an occlusion region in the other image).

To make the DSI more robust to effects of noise, we can replace subtraction with correlation using a simplified version of adaptive windows[10] that preserves sharp boundaries at occlusion jumps in the DSI[9].[1]

Figure 1a is test image pair of a sloping wedding cake whose depth profile is shown in Figure 1b. Figure 1c shows the correlation DSI for a scanline through the middle of the test image pair; near-zero values have been enhanced for human viewing. Notice the characteristic streaking pattern that results from holding one scanline still and sliding the other scanline across. When a textured region on the left scanline slides across the corresponding region in the right scanline, a line of matches can be seen in the DSI. When two textureless matching regions slide across each other, a diamond-shaped region of near-zero matches can be observed. The more homogeneous the region is, the more distinct the resulting diamond shape will be. The zero path through the DSI corresponding to the disparities representing the surface can be seen as a dark line connecting block-like segments.

2.1 Occlusion Analysis and DSI Path Constraints

In a discrete formulation of the stereo matching problem, any region with non-constant disparity must have associated unmatched pixels. Because of these occlusion regions, the matching zero path through the image cannot be continuous. The regions labeled "D" in Figure 1c mark horizontal *gaps* in the enhanced zero line in the DSI_i^L and DSI_i^R. These gaps correspond to occlusion regions in the left and right images, respectively. The gaps labeled "V" in the DSIs mark vertical jumps from disparity to disparity. These jumps correspond to left and right occlusion edges. We use this "occlusion constraint"[7] to restrict the type of matching path that can be recovered from each DSI: each time an occluded region or edge is proposed, the recovered path is forced to have the appropriate diagonal or vertical gap.

Nearly all stereo scenes obey the *ordering constraint* (or *monotonicity constraint* [7]): if object a is to the left of object b in the left image then a will be to the left of b in the right image. Thin objects with large matching disparities violate this rule, but such configurations are rare. By assuming the ordering rule we can impose a second constraint on the disparity path through the DSI that significantly reduces the complexity of the path-finding problem. In the DSI_i^L, moving from left to right, diagonal jumps can only jump forward (down and across) and vertical jumps can only jump backwards (up). In the DSI_i^R the relationship is reversed: moving left to right diagonal jumps can only jump backwards and across and vertical jumps can only jump forwards (down).

[1] If we use correlation, then we implicitly impose a certain amount of inter-scanline consisitency (e.g. [11, 5, ?]) since the windows have vertical extent. Whether this is desired depends upon the domain.

2.2 Finding the Best Path

The occlusion and ordering constraints greatly restrict the possible solution paths through disparity space. From any location in the DSI, there are only three directions a path can take – a horizontal match, a diagonal occlusion, and a vertical occlusion. This observation allows us to develop a stereo algorithm that integrates matching and occlusion analysis into a single process.

Our algorithm for finding the best path through the DSI is formulated as a dynamic programming (DP) path-finding problem in $(x, disparity)$ space. We wish to find the minimum cost traversal through the DSI when the occlusion and ordering constraints are imposed. These constraints severely limit the direction the path can take from the path's current endpoint. If we base the decision of which path to choose at any pixel only upon the cost of each possible path we can take and not on any previous moves we have made, we satisfy the DP requirements and can use DP to find the optimal path.

Our DSI analysis led us to consider the occlusion problem in a "state-like" manner. As we traverse through the DSI image finding the optimal path, we can be in any of three states: *match (M), vertical occlusion (V), or diagonal occlusion (D).* Figure 2 symbolically shows the legal transitions between each type of state. The path is further constrained at the edges of the DSI image, where several types of transitions may be invalid.

A cost is assigned to each pixel in the path depending upon the current state. We design our DP algorithm to *minimize* the cost of a path where the cost of a match is the absolute value of the DSI_i^L pixel at the match point. The better the match, the lower the cost assessed. The algorithm will attempt to maximize the number of "good" matches in the final path. Since the algorithm will also propose unmatched points — occlusion regions — we need to assign a cost for unmatched pixels in the vertical or diagonal jumps. Otherwise the "best path" would be one that matches almost no pixels.

2.3 Ground Control Points

Unfortunately, slight variations in the occlusion pixel cost can change the globally minimum path through the DSI_i^L space, particularly with noisy data[5]. Because

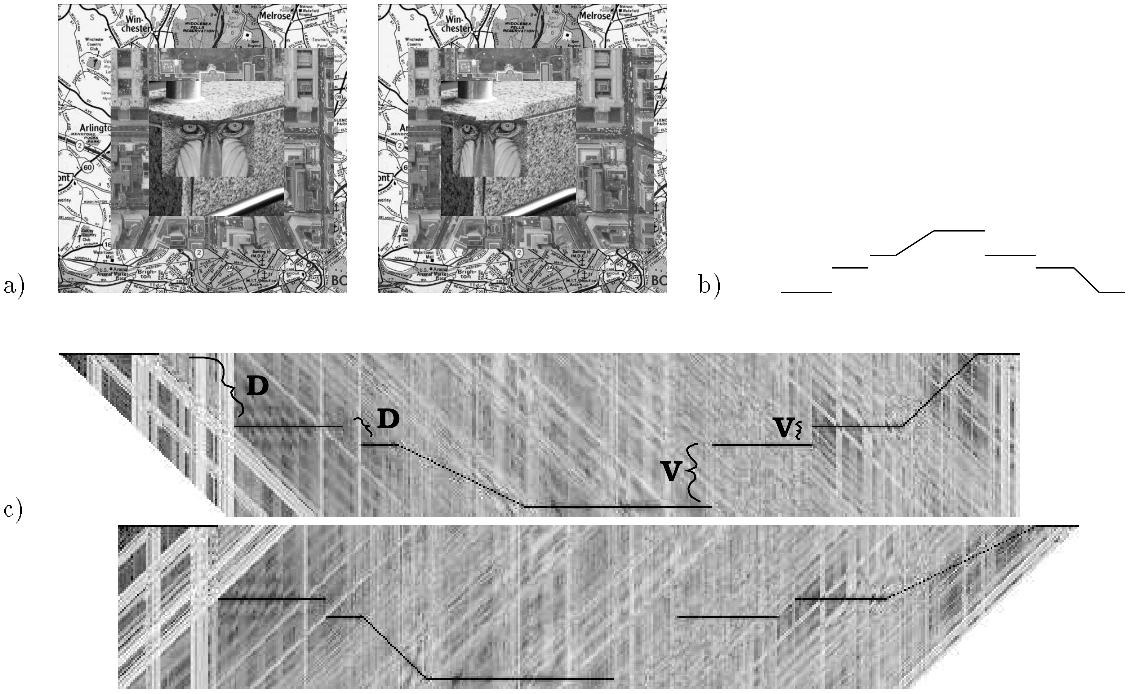


Figure 1: (a) A simulated image pair of a sloping wedding cake, whose depth profile through the center is shown in (b). (c) the enhanced, cropped DSI_i^L and DSI_i^R for one scan line.

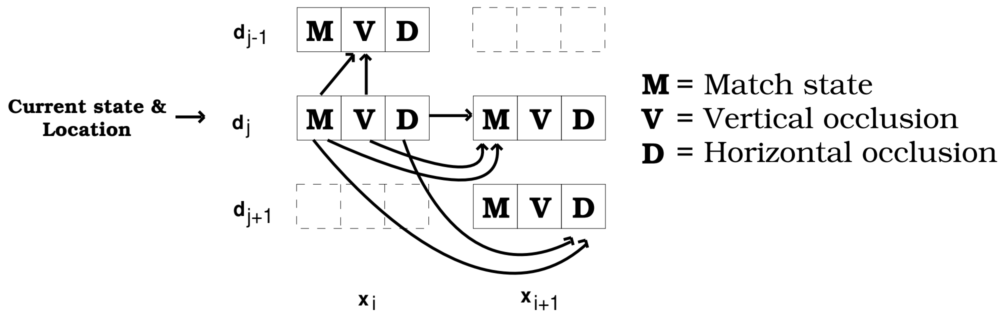


Figure 2: State diagram of moves the DP algorithm can choose through the DSI.

this cost is incurred for each proposed occluded pixel, the cost of a proposed occlusion region is linearly proportional to the width of the region.

In order to overcome this occlusion cost sensitivity, we need to impose another constraint in addition to the occlusion and ordering constraints. However, unlike previous approaches we do not want to bias the solution towards any generic property such as smoothness[7], inter-scanline consistency[11, 5], or intra-scanline "goodness"[5].

Instead, we use high confidence matching guesses: *Ground control points* (GCPs). These points are used to force the disparity path to make large disparity jumps that might otherwise have been avoided because of large occlusion costs.

Figure 3 illustrates this idea showing two GCPs and a number of possible paths between them. We note that regardless of which disparity path is chosen, the discrete lattice ensures that paths A, B, and C all require 6 occlusion pixels. Therefore, all three paths incur the same occlusion cost. Our algorithm will select the path that minimizes the cost of the proposed matches *independent of where occlusion breaks are proposed and the occlusion cost value.* If there is a single occlusion region between the GCPs in the original image, the path with the best matches is similar to A or B. On the other hand, if the region between the two GCPs is sloping gently, then a path similar to C, with tiny, interspersed occlusion jumps will be preferred. The path through *(x, disparity)* space, therefore, will be constrained solely by the occlusion and ordering constraints and the goodness of the matches between the GCPs. An exception to this situation occurs if the algorithm proposes additional occlusion regions as in path D; such solutions typically have a much higher cost than the correct one.

Figure 4-a shows the "birch" image from the JISCT stereo test set[2]. The occlusion regions in this image are difficult to recover properly because of the skinny trees, some textureless regions, and a 15 percent brightness difference between images. The skinny trees make occlusion recovery particularly sensitive to occlusion cost when GCPs are not used, since there are relatively few good matches on each skinny tree compared with the size of the occlusion jumps to and from each tree. Figure 4-b shows the results of our algorithm *without* using GCPs. The occlusion cost prevented the path on most scanlines from jumping out to some of the trees. Figure 4-c shows the algorithm run with the same occlusion cost using GCPs. Most of the occlusion regions around the trees are recovered well since GCPs on the tree surfaces eliminated the dependence on the occlusion cost. There are some errors in the image, however. Several shadow regions of the birch figure are completely washed-out with intensity values of zero. Consequently, some of these regions have led to spurious GCPs which caused incorrect disparity jumps in our final result. This problem can be minimized by changing the GCP selection algorithm to check for texture wherever GCPs are proposed[8]. On some scanlines, no GCPs were recovered on some trees which led to the scanline gaps in some of the trees.[2]

Our occlusion based stereo algorithm using GCPs performs at least as well and usually much better than other similar methods while reducing path-finding complexity and sensitivity to occlusion cost. Further, we have imposed no criteria — such as smoothness — that bias the recovered solution. However, we have not used important edge information, and we can improve our method by doing so.

[2] The processing of the birch data set used subtraction, as opposed to correlation, to create the DSI. Therefore, no inter-scanline consistency has yet been imposed.

3 Edges in the DSI

Figure 5a is a stereo pair of a man and two kids and Figure 5b is a corresponding DSI_i^L for a scanline through the man's legs. Both vertical and diagonal striations are visible in the DSI data structure. These line-like striations are formed wherever a large change in intensity (i.e. an "edge") occurs in the left or right scan line. In the DSI_i^L the vertical striations correspond to large changes in intensity in I^L and the diagonal striations correspond to changes in I^R; the converse holds for the DSI_i^R. Since the interior regions of objects tend to have less intensity variation than the edges, the subtraction of an interior region of one line from an intensity edge of the other tends to leave the edge structure in tact. The persistence of the edge traces a linear structure in the DSI. We refer to the lines in the DSI as "edge-lines."

As mentioned earlier, occlusion boundaries tend to occur at intensity edges. Recall that an occlusion is represented in the DSI by the stereo solution path containing either a diagonal or vertical jump. When an occlusion edge coincides with an intensity edge, then *the occlusion gap in the DSI stereo solution will coincide with the DSI line defined by the corresponding intensity edge.* Figures 6a and 6b show a DSI and the "edge-lines" image corresponding to the line-like striations. Figure 6c displays the solution recovered for that scanline, and Figure 6d shows the recovered solution overlayed on the lines image. The vertical and diagonal occlusions in the DSI travel along lines appearing in the DSI edge-line image.

In the next section we develop a technique for incorporating these lines into the dynamic programming solution developed in the previous section. The goal is to bias the solution so that nearly all major occlusions proposed will have a corresponding intensity edge.

Before our stereo algorithm can exploit edge information, we must first detect the DSI edge-lines. Line detection in the DSI is a relatively simple task since, in principal, an algorithm can search for diagonal and vertical lines only. For our initial experiments, we implemented such an edge finder. However, the computational inefficiencies of finding edges in the DSI for every scan line led us to seek a one pass edge detection algorithm that would approximate the explicit search for lines in every DSI.

Our heuristic is to use a standard edge-finding procedure on each image of the original image pair and use the recovered edges to generate an edge-lines image for each DSI. We have used a simplified Canny edge detector to find possible edges in the left and right image[3] and combined the vertical components of those edges to recover the edge-lines.

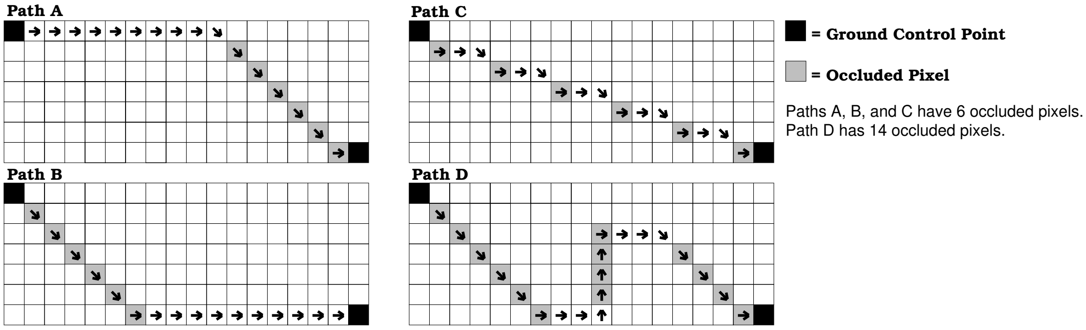
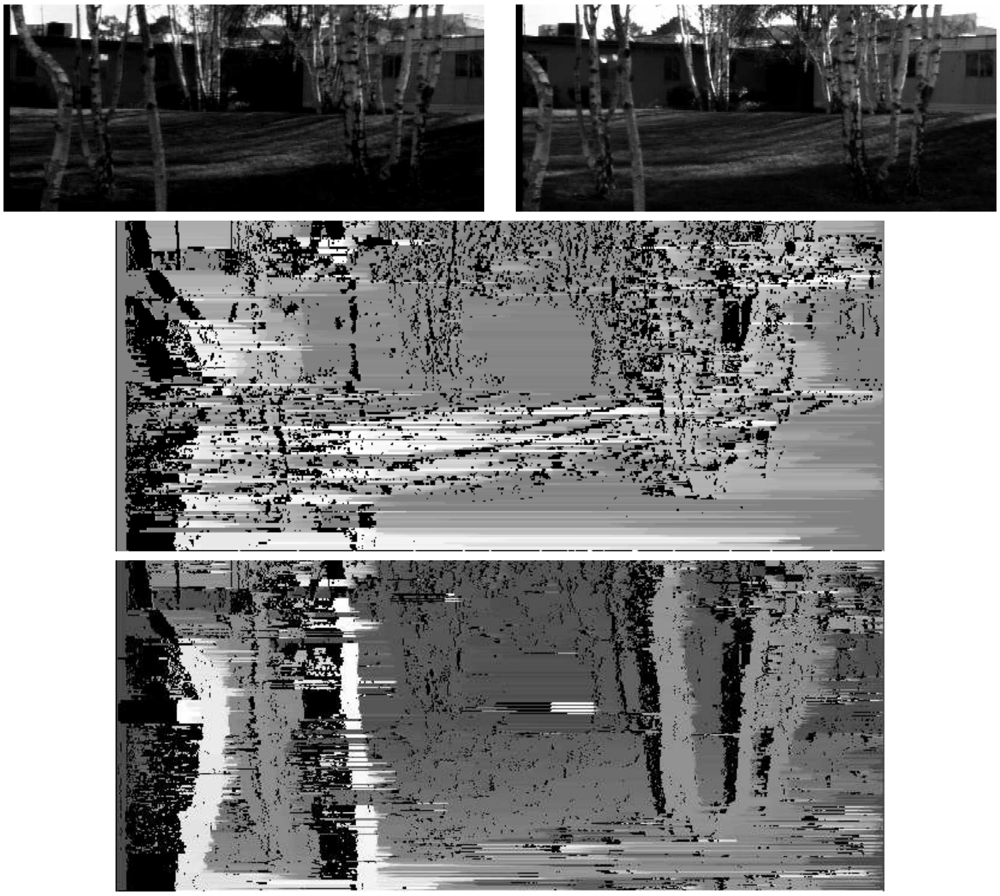


Figure 3: Four possible paths through two GCPs.

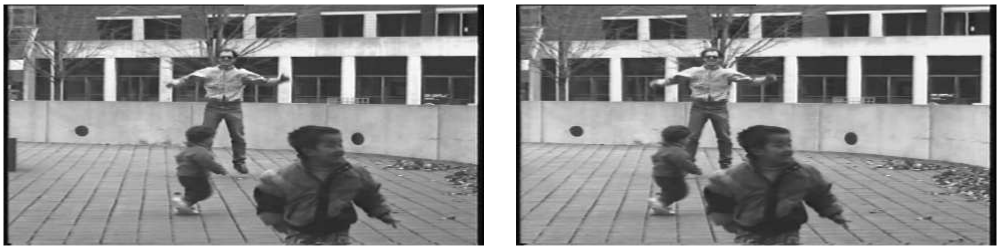
a)



b)

c)

Figure 4: (a) The "birch" stereo image pair, which is a part of the JISCT stereo test set[2], (b) Results of our stereo algorithm without using GCPs, and (c) Results of of our algorithm with GCPs.

a) 

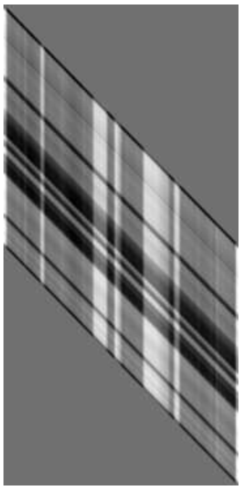
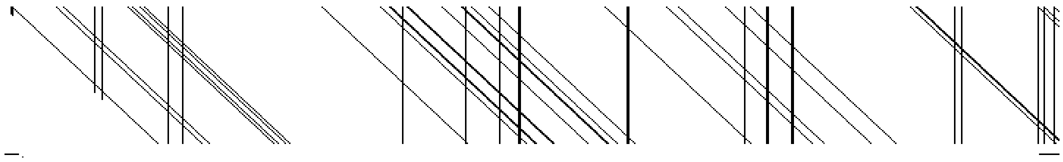
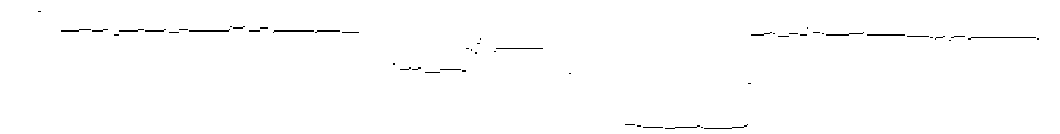
b) 

Figure 5: (a) A stereo pair of a man and two kids, and (b) a subtraction DSI_i^L for the stereo pair, where i is a scanline through the man's chest. Notice the diagonal and vertical striations that form in the DSI_i^L due to the intensity changes in the image pair. These edge-lines appear at the edges of occlusion regions.

a) 

b) 

c) 

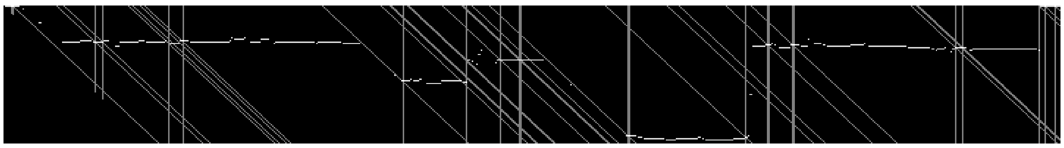
d) 

Figure 6: (a) A cropped, subtraction DSI_i^L. (b) The lines corresponding to the line-like striations in (a). (c) The recovered path. (d) The path and the image from (b) overlayed. The paths along occlusions correspond to the paths along lines.

The use of a standard edge operator introduces a constraint into the stereo solution that we purposefully excluded until now: inter-scanline consistency. Because any spatial operator will tend to find coherent edges, the result of processing one scanline will no longer be independent of its neighboring scanlines. However, since the inter-scanline consistency is only encouraged with respect to edges and occlusion, we are willing to include this bias in return the computationally efficiency of single pass edge detection.

4 Using Edges with the DSI approach

Our goal is to incorporate the DSI edge information into the dynamic programming solution in such a way as to 1) correctly bias the solution to propose occlusions at intensity edges; 2) not violate the occlusion ordering constraints developed previously; and 3) not significantly increase the computational cost of the path-finding algorithm.

As shown, occlusion segments of the solution path paths through the DSI usually occur along edge-lines of the DSI. Therefore, a simple and effective strategy for improving our occlusion finding algorithm that satisfies our three criteria above is to reduce the cost of an occlusion along paths in the DSI corresponding to the edge-lines.

Figure 7 illustrates this cost reduction. Assume that a GCP or a region of good matches is found on either side of an occlusion jump. Edge-lines in the DSI, corresponding to intensity edges in the scanlines, are shown in the diagram as dotted lines. The light solid lines show some possible paths consistent with the border boundary constraints. If the cost of an occlusion is significantly reduced along edge-lines, however, then the path indicated by the dark solid line is least expensive, and that path will place the occlusion region in the correct location.

By reducing the cost along the lines, we improve occlusion recovery without adding any additional computational cost to our algorithm other than a pre-processing computation of edges in the original image pair. Matching is still driven by pixel data but is influenced, where most appropriate, by edge information. And, ground control points prevent non-occlusion intensity edges from generating spurious occlusions in the least cost solution. The only remaining issue is how to reduce the occlusion cost along the edge-lines.

4.1 Zero cost for occlusion at edges: degenerate case

A simple method for lowering the occlusion cost along edge-lines would be simply to reduce the occlusion pixel cost if the pixel sits on either a vertical or diagonal edge-line. Clearly, reducing the cost by any amount will encourage proposing occlusions that coincide with intensity edges. However, unless the cost of occlusion along some line is free, there is a chance that somewhere along the occlusion path a stray false, but good, match will break the occlusion region. In Figure 7, the proposed path will more closely hug the dotted diagonal line, but still might wiggle between occlusion state to match state depending upon the data. More importantly, simply reducing the occlusion cost in this manner re-introduces a sensitivity to the value of that cost; the goal of the GCPs was the elimination of that sensitivity.

If the dotted path in Figure 7 were free, however, spurious good matches would not affect the recovered occlusion region. An algorithm can be defined in which any vertical or diagonal occlusion jump corresponding to an edge-line has zero cost. This method would certainly encourage occlusions to be proposed along the lines.

Unfortunately, this method is a degenerate case. The DP algorithm will find a solution that maximizes the number of occlusion jumps through the DSI and minimizes the number of matches, regardless of how good the matches may be. Figure 8a illustrates how a zero cost for both vertical and diagonal occlusion jumps leads to nearly no matches begin proposed. Figure 8b shows that this degenerate case does correspond to a potentially real camera and object configuration. The algorithm has proposed a feasible solution. The problem, however, is that the algorithm is ignoring huge amounts of well-matched data by proposing occlusion everywhere.

4.2 Focusing on occlusion regions

In the previous section we demonstrated that one cannot allow the traversal of both diagonal and vertical lines in the DSI to be cost free. Also, a compromise of simply lowering the occlusion cost along both types of edges re-introduces dependencies on that cost. Because one of the goals of our approach is the recovery of the occlusion regions, we choose to make the diagonal occlusion segments free, while the vertical segments maintain the normal occlusion pixel cost. The expected result is that the occlusion regions corresponding to the diagonal gaps in the DSI should be nicely delineated while the occlusion edges (the vertical jumps) are not changed. Furthermore, we expect no increased sensitivity to the occlusion cost.

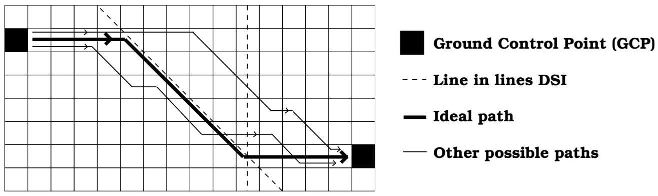


Figure 7: This figure illustrates how reducing the cost along lines that appear in the lines DSI (represented here by dotted lines) can improve occlusion recovery. Given the data between the two GCPs is noisy, the thin solid lines represent possible paths the algorithm might choose. If the cost to propose an occlusion has been reduced, however, the emphasized path will most likely be chosen. That path will locate the occlusion region cleanly with start and end points in the correct locations.

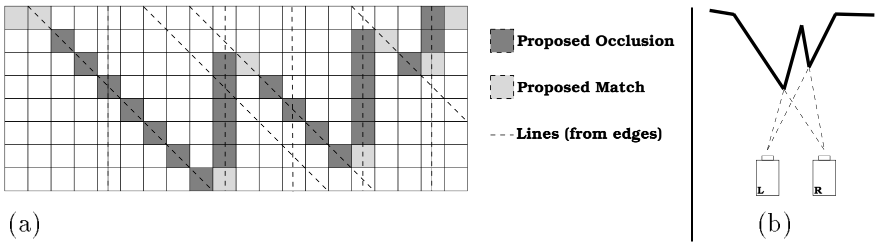


Figure 8: (a) When the occlusion cost along both vertical and diagonal edge-lines is set to zero, the recovered path will maximize the number of proposed occlusions and minimize the number of matches. Although real solutions of this nature do exist, an example of which is shown in (b), making both vertical and diagonal occlusion costs free generates these solutions even when enough matching data exists to support a more likely result.

Figure 9a shows a synthetic stereo pair from the JISCT test set[2] of some trees and rocks in a field. Figure 9b shows the occlusion regions recovered by our algorithm when line information and GCP information is not used, comparable to previous approaches (e.g. [5]). The black occlusion regions around the trees and rocks are usually found, but the boundaries of the regions are not well defined and some major errors exist. Figure 9c displays the results of using only GCPs, with no edge information included. The dramatic improvement again illustrates the power of the GCP constraint. Figure 9d shows the result when both GCPs and edges have been used. Though the improvement over GCPs alone is not nearly as dramatic, the solution is better. For example, the streaking at the left edge of the crown of the rightmost tree has been reduced. In general, the occlusion regions have been recovered almost perfectly, with little or no streaking or false matches within them. Although the overall effect of using the edges is small, it is important in that it biases the occlusion discontinuities to be proposed in exactly the right place.

4.3 Non-cyclopean views: stereo layers

Occlusion regions can be used to label object boundaries and *layers* of objects in a scene. For many, if not most, applications qualitative information that segments the scene into layers is more useful than precise, quantitative depth measurement. Object segmentation, motion analysis, object recognition, and image coding algorithms would all benefit from depth layering that preserves sharp 'silouhette" discontinuity boundaries[13].

Most previous work on detecting depth discontinuities in stereo (e.g. [6]) proposes some operation on the recovered depth map. For example, Terzopolous proposed a thin-metal plate interpolation technique[12] and fracturing the plate at areas of high strain. But a much more direct method of detecting the depth discontinuities is to make explicit use of the discovered occlusion regions.

Figures 10a and 10c are a stereo pair from a sequence of a man and two kids; this type of imagery has very large occlusion regions. For most computer vision applications such as tracking or modeling, segmentation of the man's arms and legs is more valuable information than the exact leg curvature. The outline shape of the man and his layered position with respect to the

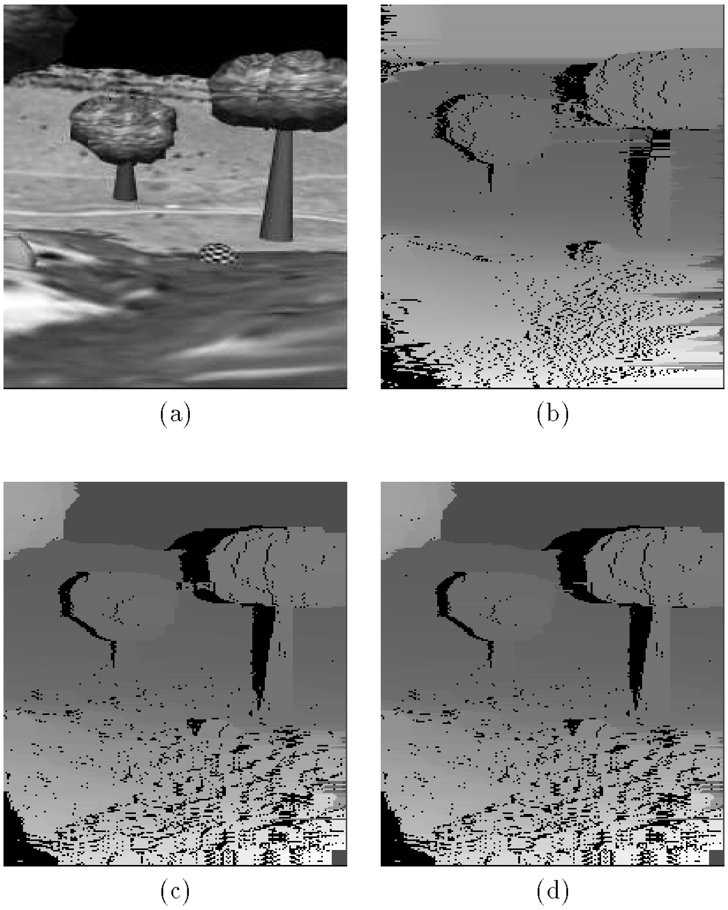
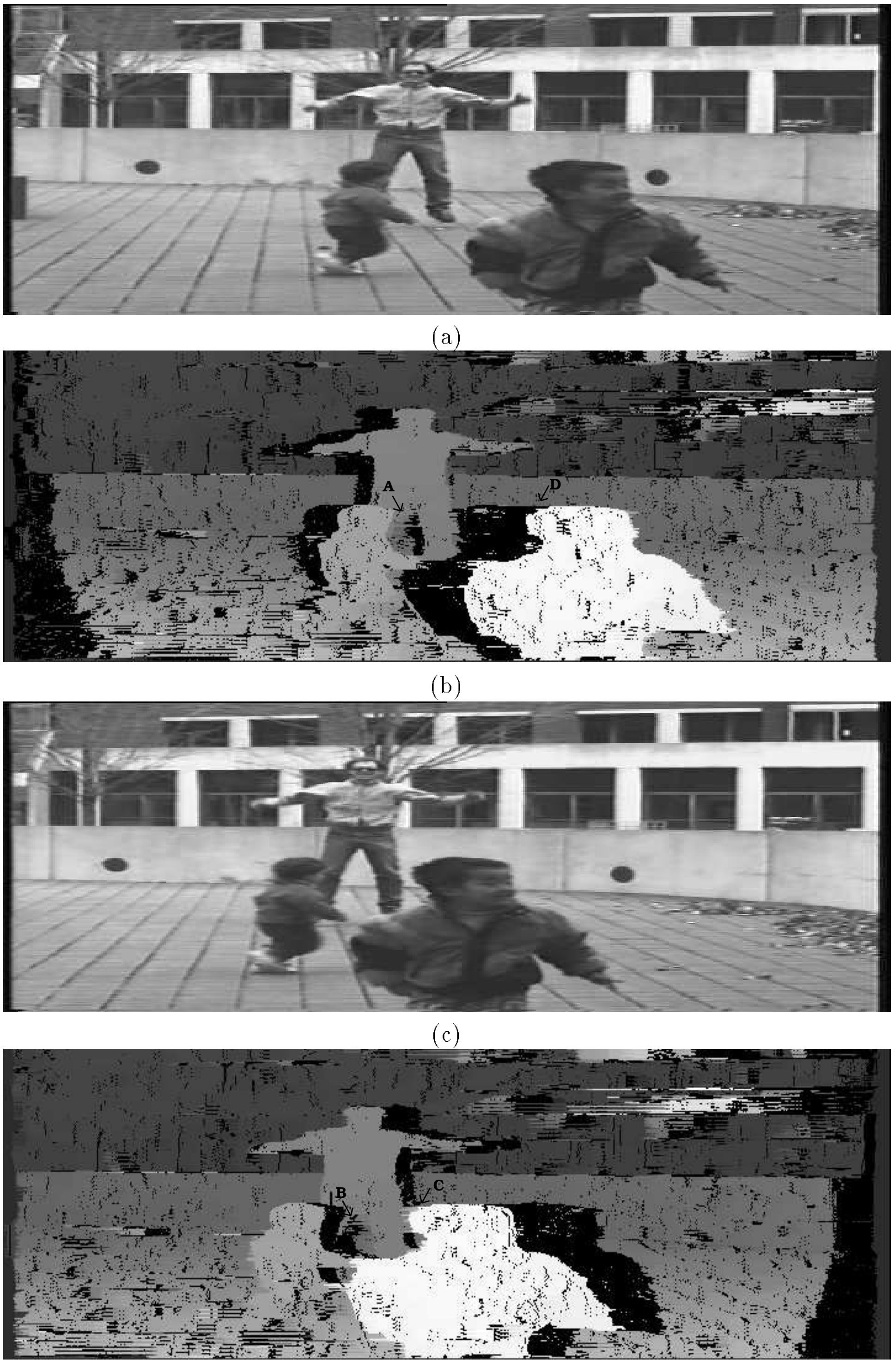


Figure 9: (a) Synthetic trees left image, (b) occlusion result without GCPs or edge-lines, (c) occlusion result with GCPs only, and (d) result with GCPs and edge-lines.

other objects in the scene (the background and two kids) is critical information that can be provided reliably by stereo algorithms that include occlusion analysis.

Our method uses edge information asymmetrically, concentrating on recovering the occlusion regions (as opposed to edges) in each image of the stereo pair. Thus, our algorithm generates *two* occlusion maps, one optimized to find each type of occlusion. Figures 10b and 10d show the stereo disparities and occlusion regions recovered using the DSI_i^L and the DSI_i^R, respectively. Studying the left and right occlusion regions in these figures, the advantage of using a dual-image solution becomes clear. In the left result (10c), the left occlusion regions are generally correct, but the right occlusion edges still have a "ragged" quality, like at the right edge of the man's left leg (marked as region A). In the dual, right solution (10d), however, the right occlusion region of the same leg is recovered more precisely (marked as region B). Other parts of the occlusion recovery are also improved using dual solutions, such as the left edge of the rightmost kids's head. The right occlusion result recovers a "ragged" left occlusion edge (marked as region C), but the left occlusion result finds the left occlusion region and the boundary of the kids's head cleanly (marked as region D). Instead of generating a single, mediocre, cyclopean reconstruction by reducing the occlusion cost by half for occlusion regions and occlusion edges, we produce two: one with near-perfect left occlusion regions and the other with near-perfect right occlusion regions.

We have just begun to explore the idea of stereo layers and have not yet constructed methods for using the occlusion maps to define the layers. Also, because the idea of layers becomes particularly useful when considering image sequences[13] we will need to consider any interaction between layers defined by motion and those defined by stereo.



(a)

(b)

(c)

(d)

Figure 10: (a), (c) Kids stereo pair, (b), (d) occlusion results using GCPs and edge-lines.

5 Summary

We have proposed an extension to our previous work on finding stereo matches and occlusion regions using a pixel-based approach. The technique is based upon finding the best path through a disparity space image. In the modified formulation, we exploit the relationship between occlusion jumps and intensity edges. Edges are used to lower the cost of proposed occlusion edges that coincide with intensity edges. The result is an algorithm that extracts large occlusion regions accurately without using criteria such as inter-scanline consistency or smoothness. A direct application of this approach is the recovery of *stereo layers* — three-dimensionally isolated regions.

References

[1] P. Belhumeur and D. Mumford. A bayseian treatment of the stereo correspondence problem using half-occluded regions. In *Proc. Comp. Vis. and Pattern Rec.*, 1992.

[2] R. Bolles, H. Baker, and M. Hannah. The JISCT stereo evaluation. In *Proc. Image Understanding Workshop*, pages 263–274, 1993.

[3] J. Canny. A computational approach to edge detection. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 8(6):679–698, 1986.

[4] S.D. Cochran and G. Medioni. 3-d surface description from binocular stereo. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 14(10):981–994, 1992.

[5] I.J. Cox, S. Hingorani, B. Maggs, and S. Rao. Stereo without regularization. NEC Research Institute Report, NEC Research Institute, October 1992.

[6] P. Fua. Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities. In *Proc. Int. Joint Conf. Art. Intel.*, pages 1292–1298, 1991.

[7] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. In *Proc. European Conf. Comp. Vis.*, pages 425–433, 1992.

[8] M.J. Hannah. A system for digital stereo image matching. *Photogrammetric Eng. and Remote Sensing*, 55(12):1765–1770, 1989.

[9] S.S. Intille and A.F. Bobick. Disparity-space images and large occlusion stereo. MIT Media Lab Perceptual Computing Group Technical Report No. 220, Massachusetts Institute of Technology, October 1993.

[10] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: theory and experiment. In *Proc. Image Understanding Workshop*, pages 383–389, 1990.

[11] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 7:139–154, 1985.

[12] D. Terzopoulos. Regularization of inverse problems involving discontinuities. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 8(4):413–424, 1986.

[13] J.Y.A. Wang and E.H. Adelson. Layered representation for motion analysis. In *Proc. Comp. Vis. and Pattern Rec.*, pages 361–366, New York City, June 1993.

[14] Y. Yang, A. Yuille, and J. Lu. Local, global, and multilevel stereo matching. In *Proc. Comp. Vis. and Pattern Rec.*, 1993.