M.I.T. Media Lab Perceptual Computing Group Technical Report No. 220
Condensed version appears in ECCV '94 proceedings, Stockholm, Sweden

Disparity-Space Images and Large Occlusion Stereo

Stephen S. Intille and Aaron F. Bobick

intille@media.mit.edu and bobick@media.mit.edu
Perceptual Computing Group
The Media Lab, Massachusetts Institute of Technology
20 Ames St., Cambridge MA 02139

Abstract: A new method for solving the stereo matching problem in the presence of large occlusion is presented. A data structure — the *disparity space image* — is defined in which we explicitly model the effects of occlusion regions on the stereo solution. We develop a dynamic programming algorithm that finds matches and occlusions simultaneously. We show that while some cost must be assigned to unmatched pixels, our algorithm's occlusion-cost sensitivity and algorithmic complexity can be significantly reduced when highly-reliable matches, or *ground control points*, are incorporated into the matching process. The use of ground control points eliminates both the need for biasing the process towards a smooth solution and the task of selecting critical prior probabilities describing image formation.

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

1 Introduction

Our world is full of occlusion. In any scene, we are likely to find several, if not several hundred, occlusion edges. In binocular imagery, we encounter occlusion times two. Stereo images contain occlusion edges that are found in monocular views and *occluded regions* that are unique to a stereo pair[5]. Occluded regions are spatially coherent groups of pixels that can be seen in one image of a stereo pair but not in the other. These regions mark discontinuity jumps and can be used to improve segmentation, motion analysis, and object identification processes, which must preserve object boundaries. There is psychophysical evidence that the human visual system uses geometrical occlusion relationships during binocular stereopsis[20][18] to reason about the spatial relationships between objects in the world. In this paper we present a stereo algorithm that does so as well.

Although absolute occlusion sizes in pixels depend upon the configuration of the imaging system, images of everyday scenes often contain occlusion regions much larger than those found in popular stereo test imagery. In our lab, common images like Figure 1 contain disparity shifts and occlusion regions over eighty pixels wide.[1] Popular stereo test images, however, like the JISCT test set[7], the "pentagon" image, the "white house" image, and the "Renault part" image have maximum occlusion disparity shifts on the order of 20 pixels wide. Regardless of camera configuration, images of the everyday world will have substantially larger occlusion regions than aerial or terrain data. Even processing images with small disparity jumps, researchers have found that occlusion regions are a major source of error[8].

2 Previous Occlusion and Stereo Work

Most stereo researchers have generally either ignored occlusion analysis entirely or treated it as a secondary process that is postponed until matching is completed and smoothing is underway[2, 13]. A few authors have proposed techniques that indirectly address the occlusion problem by minimizing spurious mismatches resulting from occluded regions and discontinuities[15, 9, 1, 19, 1, 17, 10].

Belhumeur has considered occlusion in several papers. In [5], Belhumeur and Mumford point out that occluded regions, not just occlusion boundaries, must be identified and incorporated into matching. Using this observation and Bayesian reasoning, an energy functional is derived using using pixel intensity as the matching feature and dynamic programming is used to find the minimal-energy solution. In [3] and [4] the Bayesian estimator is refined to deal with sloping and creased surfaces. Penalty terms are imposed for proposing a break in vertical and horizontal smoothness or a crease in surface slope. Belhumeur's method requires the estimation of several critical prior terms which are used to suspend smoothing operations.

[1] Typical set up is two CCD cameras, with 12mm focal length lenses, separated by a baseline of about 30cm.

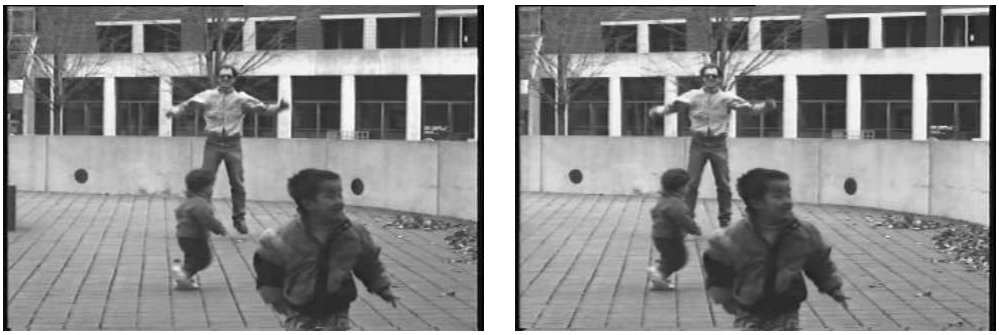

Figure 1: Noisy stereo pair of a man and kids. The largest occlusion region in this image is 93 pixels wide, or 13 percent of the image.

Geiger, Ladendorf, and Yuille[14] also directly address occlusion and occlusion regions by defining an a priori probability for the disparity field based upon a smoothness function and an occlusion constraint. For matching, two shifted windows are used in the spirit of [19] to avoid errors over discontinuity jumps. Assuming the monotonicity constraint, the matching problem is solved using dynamic programming. Unlike in Belhumeur's work, the stereo occlusion problem is formulated as a path-finding problem in a left-scanline to right-scanline matching space. Geiger *et al.* make the important observation that "a vertical break (jump) in one eye corresponds to a horizontal break (jump) in the other eye."

Finally, Cox *et al.*[12] have proposed a dynamic programming solution to stereo matching that does not require the smoothing term incorporated into Geiger and Belhumeur's work. They point out that several equally good paths can be found through matching space using only the occlusion and ordering constraints. To provide enough constraint for their system to select a single solution, they optimize a Bayesian maximum-likelihood cost function minimizing inter- and intra-scanline disparity discontinuities.

Our approach is to explicitly model occlusion edges and occlusion regions and to use them to drive the matching process. We develop a data structure which we will call the *disparity-space image* (DSI), and we use this data structure to develop a stereo algorithm that finds matches and occlusions *simultaneously*. We show that while some cost must be assigned to unmatched pixels, an algorithm's occlusion-cost sensitivity and algorithmic complexity can be significantly reduced when highly-reliable matches, or *ground control points* (GCPs), are incorporated into the matching process.

3 The DSI Representation

In this section we describe a data structure we call the *disparity-space image*, or DSI. We have used the data structure to explore the occlusion and stereo problem and it facilitated our development of a dynamic programming algorithm that uses occlusion constraints. The DSI is an explicit representation of matching space; it is related to figures that have appeared in previous work [11, 14, 21, 19].

3.1 DSI Creation for Ideal Imagery

We generate 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:

$$\mathrm{DSI}_i^R(x,d) = \begin{cases} I_R(x,i) - I_L(x+d,i) \\ \qquad \text{when } 0 \le (x+d) < N \end{cases} \tag{1}$$

where all other values are not defined and $0 \le d < N$ 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.

The above definition generates a "full" DSI where there is no limit on disparity. By considering camera geometry, we can crop the representation. In the case of parallel, front-facing cameras objects are shifted to

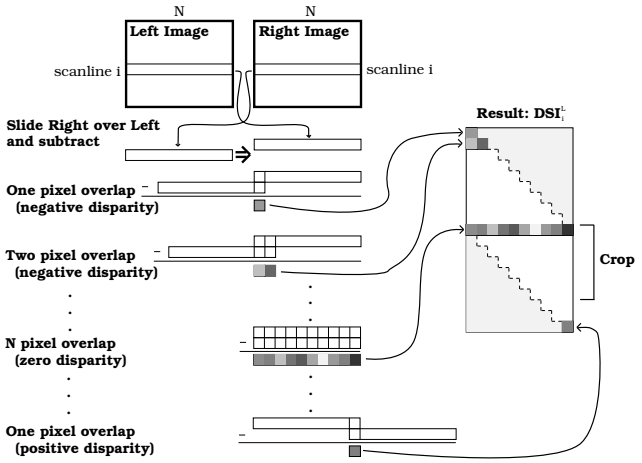

Figure 2: This figure describes how a DSI_i^L is generated. The corresponding epipolar scanlines from the left and right images are used. The scanline from the left image is held still as the scanline from the right image is shifted across. After each pixel shift, the scanlines are subtracted. The result from the overlapping pixels is placed in the resulting DSI_i^L. The DSI_i^L is then cropped, since we are only interested in disparity shifts that are zero or greater since we assume we have parallel optical axis in our imaging system.

the right in the left image. No matches will be found searching in the other direction. Further, if a maximum possible disparity, d_{max}, is known then no matches will be found by shifting right more than d_{max} pixels. These limitations permit us to crop the top N and bottom $N - d_{max}$ lines of the DSI. DSI generation is illustrated in Figure 2.

3.2 DSI Creation for Imagery with Noise

To make the DSI more robust to effects of noise, we can change the comparison function from subtraction to correlation. We define g_i^L as a group of scanlines centered around s_i^L and g_i^R as a group of scanlines centered around s_i^R . g_i^L and g_i^R are shifted across each other to generate the DSI representation for scanline i. Instead of subtracting a single pixel, however, we compare a window in g^L to a window in g^R:

$$W^L(x,d,w_x,w_y,c_x,x_y) = \sum_{s=-c_y}^{(w_y-c_y)} \sum_{t=-c_x}^{(w_x-c_x)} [(I_R(x+t,i+s) - M^L(x,i)) - (I_L(x+d+t,i+s) - M^R(x+d,i))]^2 \quad (2)$$

where $w_x \times w_y$ is the size of the window, (c_x, c_y) is the location of the center of the window, and M^L is the mean of the window in the left image:

$$M^L(x,y) = \frac{1}{w_y * w_x} \sum_{i=-c_y}^{(w_y-c_y)} \sum_{i=-c_x}^{(w_x-c_x)} I_L(i,j) \quad (3)$$

M^R is computed like M^L using the right image.

Using correlation for matching drastically reduces the effects of noise. However, windows create problems at vertical and horizontal depth discontinuities where occluded regions lead to spurious matching. We solve this problem using a simplified version of adaptive windows[16]. At every pixel location we use 9 different windows to perform the matching. The windows are shown in Figure 3. Some windows are designed so that they will match to the left, some are designed to match to the right, some are designed to match towards the top, and so on. At an occlusion boundary, some of the filters will match across the boundary and some will not. At each pixel, only the best result from matching using all 9 windows is stored. Bad matches resulting from occlusion tend to be discarded. DSI_i^L is generated by:

$$\mathrm{DSI}_i^L(x,d,w_x,w_y) = \begin{cases} \min\limits_{\substack{0 \le c_x < w_x \\ 0 \le c_y < w_y}} W_i^L(x,d,w_x,w_y) \\ \qquad \text{when } 0 \le (x-d) < N \\ \mathrm{NaN} \\ \qquad \text{otherwise} \end{cases} \quad (4)$$

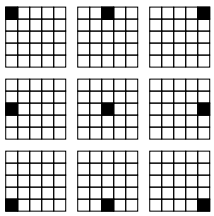

Figure 3: To reduce the effects of noise in DSI generation, we have used 9 window matching, where window centers (marked in black) are shifted to avoid spurious matches at occlusion regions and discontinuity jumps.

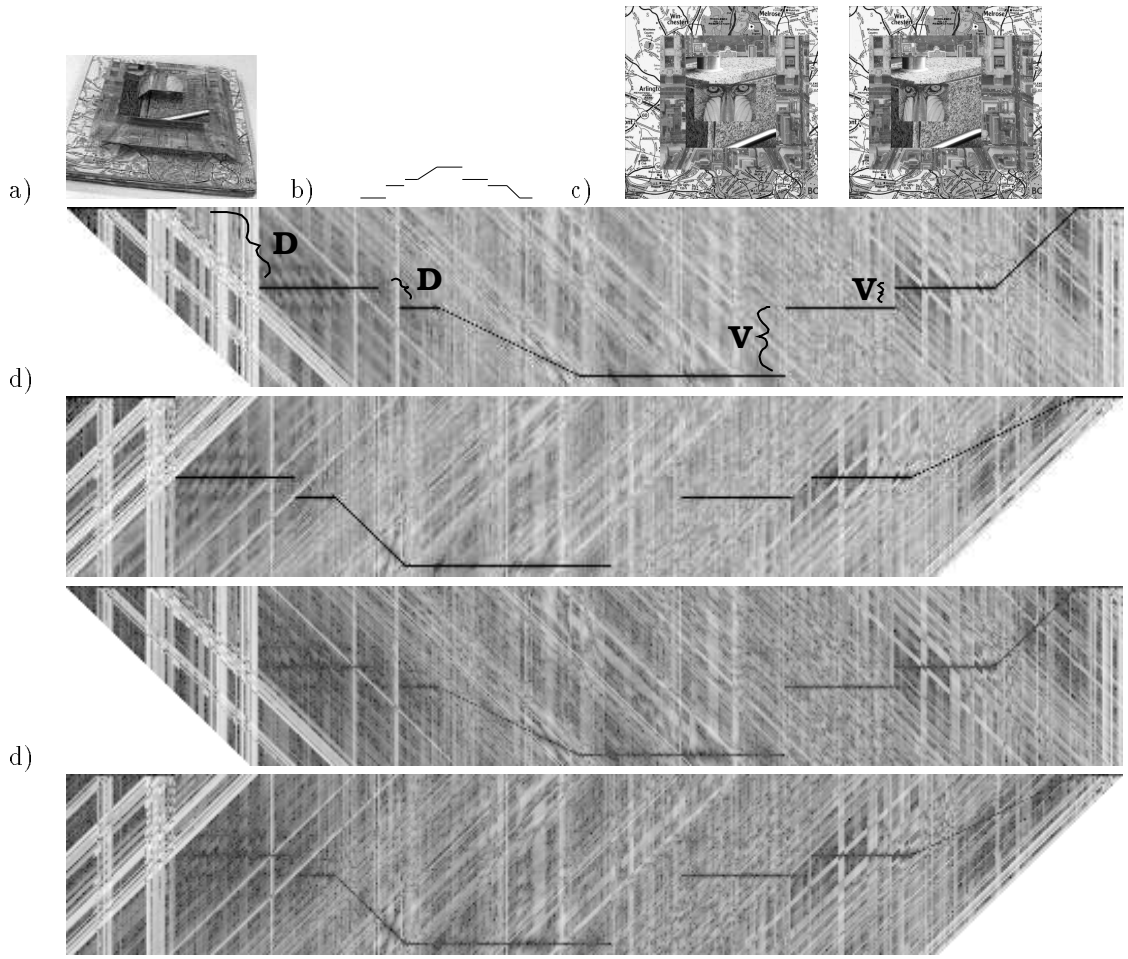

Figure 4: This figure shows (a) a model of the stereo sloping wedding cake that we will use as a test example, (b) a depth profile through the center of the sloping wedding cake, (c) a simulated, noise-free image pair of the cake, (d) the enhanced, cropped, correlation DSI representation for the image pair in (c), and (e) the enhanced, cropped, correlation DSI for a noisy sloping wedding cake (SNR = 18 dB). In the top image of (d), the regions labeled "D" mark diagonal gaps in the matching path caused by regions occluded in the left image. The regions labeled "V" mark vertical jumps in the path caused by regions occluded in the right image. In the bottom image of (d), diagonal gaps mark occluded regions in the right image and vertical jumps mark occluded regions in the left image.

To test the correlation DSI and other components of our stereo method, we have produced a more interesting version of the three-layer stereo wedding cake image frequently used by stereo researchers to assess algorithm performance. Our cake has three square layers, a square base, and two sloping sides. The cake is "iced" with textures cropped from several images. A side view of a physical model of the sloping wedding cake stereo pair is shown in Figure 4-b and a noiseless simulation of the same the wedding cake is shown in Figure 4-c. The sloping wedding cake is a challenging test example since it has textured and homogeneous regions, huge occlusion jumps, a disparity shift of 84 pixels for the top level, and flat and sloping regions. The enhanced, cropped DSI for the noiseless cake is shown in Figure 4-d.

A noisy image cake was generated with Gaussian white noise (SNR = 18 dB) The DSI generated for the noisy cake is displayed in Figure 4-e. Even with large amounts of noise, the "near-zero" dark path through the DSI disparity space is clearly visible and sharp discontinuities have been preserved.

3.3 Structure of the DSI

Figure 4-d shows the cropped, correlation DSI for a scanline through the middle of the test image pair shown in Figure 4-c. Near-zero values have been enhanced. 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_i^L. 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 correct path through DSI space can be easily seen as a dark line connecting block-like segments.

4 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. Any slope or disparity jump creates blocks of occluded pixels. Because of these occlusion regions, the matching zero path through the image cannot be continuous. The regions labeled "D" in Figure 4-d mark horizontal *gaps* in the enhanced zero line in DSI_i^L and DSI_i^R. The regions labeled "V" mark vertical jumps from disparity to disparity. These jumps correspond to left and right occlusion regions. We use this "occlusion constraint"[14] to restrict the type of matching path that can be recovered from each DSI_i^L. Each time an occluded region is proposed, the recovered path is forced to have the appropriate vertical or diagonal jump.

Nearly all stereo scenes obey the *ordering constraint* (or *monotonicity constraint* [14]): 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 they 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). If this rule is broken the ordering constraint does not hold.

5 Finding the Best Path

Using the occlusion constraint and ordering constraint, the correct disparity path is highly constrained. From any location in the DSI_i^L, 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.

However, the number of allowable paths obeying these two constraints is still huge. The possible number of paths through the DSI_i^L of size (N, D) where $D = d_{max}$ is computed using a two-parameter recurrence relation:

$$\begin{aligned} \text{Total paths} = p(N,0,D) \quad \text{where} \qquad\qquad (5)\\ p(i,j,D) = \quad p(i,j+1,D) + \\ p(i-1,j,D) + \\ p(i-1,j-1,D) \end{aligned}$$

and the boundary conditions are set using:

$$p(i,j,D) = \begin{cases} 0 & \text{when } i<0 \text{ or } j<0 \text{ or } j>i \text{ or } j>D \\ 1 & \text{when } i=0 \text{ and } j=0 \end{cases} \qquad (6)$$

where $(i = 0, j = 0)$ is the upper left corner of the s_i^L and $(i = N-1, j = 0)$ is the upper right corner.

The number of possible paths through a typical image is enormous. When the sloping wedding is reduced to a 256 pixel wide image and the maximum disparity shift is set at 45 pixels, there are 3.25e+191 legal disparity paths for each scanline! Since we cannot search all possible

paths, we adopt the strategy of previous researchers and exploit the power of dynamic programming techniques.

5.1 Dynamic Programming Constraints

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_i^L image when the occlusion constraints are imposed. We assume the epipolar scanlines have been alligned along horizontal scanlines.

DP algorithms require that the decision making process be ordered and that the decision making at any state depend only upon the current state. The occlusion constraint and ordering constraint 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 5 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 un-matched 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.

This application of dynamic programming to the stereo problem reveals the power of these techniques[6, 5, 12, 14]. When formulated as a DP problem, finding the best path through an DSI of width N and disparity range D requires considering $N * D$ DP nodes. For the 256 pixel wide version of the sloping wedding cake example, the computation considers 11,520 nodes, as opposed to 3.25e+191 paths!

5.2 Assigning occlusion cost

For the work presented here we chose a constant occlusion pixel cost. Without an additional constraint the algorithm is quite sensitive to this cost. In the next section we propose an alternative approach to reducing occlusion cost sensitivity that reduces complexity and does not artificially restrict the disparity path.

5.3 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[12]. Because this cost is incurred for each proposed occluded pixel, the cost of proposed occlusion region is linearly proportional to the width of the region. Consider the example illustrated in Figure 6. The "correct" solution is the one which starts at region A, jumps forwarded diagonally 6 pixels to region B where disparity remains constant for 4 pixels, and then jumps back vertically 6 pixels to region C. The occlusion cost for this path is $c_o * 6 * 2$ where c_o is the pixel occlusion cost. If the c_o is too great, a string of bad matches will be selected as the lower-cost path, as shown.

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[14], inter-scanline consistency[19, 12], or intra-scanline "goodness"[12].

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 7 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 path-a, path-b, and path-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 path-a or path-b. On the other hand, if the region between the two GCPs is sloping gently, then a path like path-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 cor-

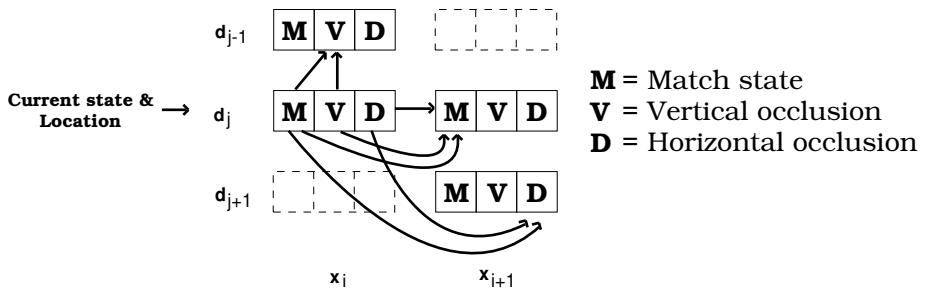

Figure 5: State diagram of legal moves the DP algorithm can make when processing the DSI_i^R. From the match state, the path can move vertically up to the vertical discontinuity state, horizontally to the match state, or diagonally to the diagonal state. From the vertical state, the path can move vertically up to the vertical state or horizontally to the match state. From the diagonal state, the path can move horizontally to the match state or diagonally to the diagonal state.

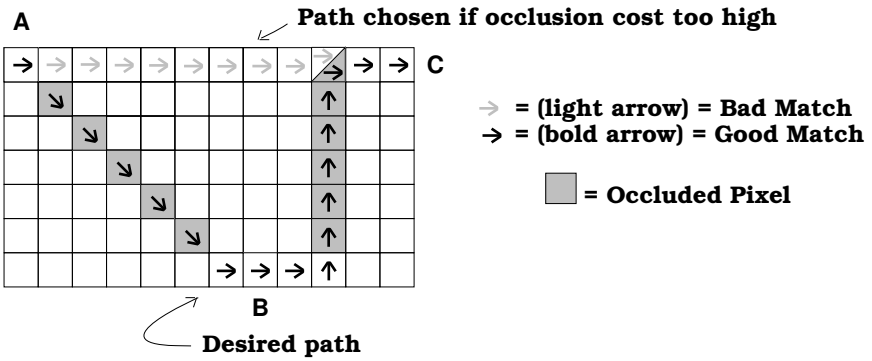

Figure 6: The total occlusion cost for an object shifted D pixels can be $cost_{occlusion} * D * 2$. If the cost becomes high, a string of bad matches may be a less expensive path. To eliminate this undesirable effect, we must impose another constraint.

rect one.

5.4 Selecting and enforcing GCPs

If we force the disparity path through GCPs, their selection must be highly reliable. We use several heuristic filters to identify GCPs before we begin the DP processing. The first heuristic requires that a control point be both the best left-to-right and best right-to-left match[15]. Second, to avoid spurious "good" matches in occlusion regions, we also require that control points have match value that is smaller than the occlusion cost. Finally, to further reduce the likelihood of a spurious match, we exclude any proposed GCPs that have no immediate neighbors that are also marked as GCPs.

Once we have a set of control points, we force our DP algorithm to choose a path through the points by assigning zero cost for matching with a control point and a very large cost to every other path through the control point's column. In the DSI_i^L, the path must pass through each column at some pixel in some state. By assigning a large cost to all paths and states in a column other than a match at the control point, we have guaranteed that the path will pass through the point.

An important feature of this approach of incorporating GCPs is that this method allows us to have more than one GCP per column. Instead of forcing the path through one GCP, we force the path through one of a few GCPs. Even using multiple windows and left-to-right, right-to-left matching, it is still possible that we will la-

bel a GCP in error if only one per column is permitted. It is unlikely, however, that none of several proposed GCPs in a column will be the correct GCP. By allowing multiple GCPs per column, we have eliminated the risk of forcing the path through a point erroneously marked as high-confidence due image noise without increasing complexity or weakening the GCP constraint.

5.5 Reducing complexity

By forcing the disparity path to hit some points, we have reduced the computational complexity of the path-finding problem by limiting the number of paths that an algorithm must consider. If a GCP is required at (x_0, d_0) the boundary conditions on the total possible paths $p(N, 0, D)$ are revised:

$$p(i,j,D) = \begin{cases} 0 & \text{when} \quad (i < 0) \text{ or } (j < 0) \text{ or } (j > i) \text{ or } (j > D) \\ 0 & \text{when} \quad (m - n) \leq (i - j) \text{ and } (i < x_0) \text{ and } (j < x_0 - 1) \\ 0 & \text{when} \quad (i \geq x_0) \text{ and } (j > d_0) \text{ and } (m - d_0) > (i - j) \\ 1 & \text{when} \quad i = 0 \text{ and } j = 0 \end{cases} \tag{7}$$

Recall that the number of possible paths in the 256 pixel wide wedding cake image is approximately 3.25e+191. However, using only twelve of the control points selected by our algorithm for that image, the legal paths are reduced to 8.41e+161. Using all of the control points would restrict the path number further. Though still large, these numbers are particularly important if a stereo algorithm tries to solve the path-finding problem without using a dynamic programming technique.

More importantly, each GCP also significantly reduces the number of DP nodes that must be considered. Without GCPs, the DP algorithm must consider one node for every point in the DSI. Specification of a GCP, however, prevents the solution path from traversing certain regions of the DSI. Because of the occlusion and monotonicity constraints, each GCP carves out two complimentary triangles in the DSI that are now not valid. Figure 8 illustrates such pairs of triangles. The total area of the two triangles, A, depends upon at what disparity d the GCP is located, but is known to lie within the range $D^{\frac{2}{4}} \leq A \leq D^{\frac{2}{2}}$ where D is the allowed disparity range. For the 256 pixel wedding cake image, $506 \leq A \leq 1012$. Since the total number of DP nodes is 11,520 each GCP whose constraint triangles do not overlap with another pair of GCP constraint triangles reduces the DP complexity by about 10%. With several GCPs the complexity is less than 25% of the original problem.

6 DP algorithm – Results

Input to our algorithm consists of a stereo pair. Epipolar lines are assumed to be known and corrected to correspond to horizontal scanlines. We assume that additive and multiplicative photometric bias between the left and right images is minimized, although the birch tree example shows our algorithm will work with significant additive differences.

The dynamic programming portion of our algorithm is quite fast; almost all time is spent in creating the correlated DSI. Generation time for each scanline depends upon the efficiency of the correlation code, the number and size of the masks, and the size of the original imagery. Running on a HP 730 workstation with a 515x512 image using nine 7x7 filters and a maximum disparity shift of 100 pixels, our current implementation takes a few seconds per scanline. However, since the most time consuming operations are simple window-based cross-correlation, the entire procedure could be made to run near real time with simple dedicated hardware.

The results generated by our algorithm using correlation with 9 masks for the noise-free wedding cake are shown in Figure 9-a. Computation was performed on the SDI_i^L but the results have been shifted to the cyclopean view. The top layer of the cake has been shifted 84 pixels. Our algorithm found the occlusion breaks at the edge of each layer, indicated by black regions. Sloping regions have been recovered as a sloping region interspersed with tiny occlusion jumps. Since we have not used any sloping or inter- or intra-scanline consistency, the solution in the sloping regions is governed only by the ground control points and the best matches in the region. There is an interesting artifact in the image that results from a high-contrast diagonally sloping line in the imagery, the shape of our filters, and the occlusion cost we have chosen.

Figure 9-b shows the results for the sloping wedding cake with noise (SNR = 18 dB). The algorithm still performs reasonably well at locating occlusion regions. Sloping regions still exhibit the tiny occlusion structure, although with less uniformity. Interestingly, once we introduce noise, the artifact appearing the the noiseless image is reduced, probably due to noise affecting the sensitivity of the occlusion cost value.

For the "kids" and "birch" results displayed in this paper, we used a subtraction DSI for our matching data. The 9-window correlation DSI was used only to find the GCPs. Since our algorithm will work properly using the subtraction DSI, any method that finds highly-reliable matches could be used to find GCPs, obviating the need for the computationally expensive cross correlation. Both the "kids" and "birch" results were generated

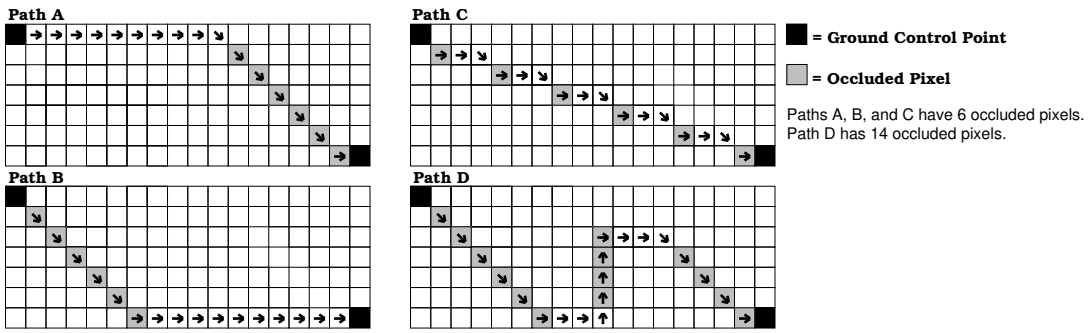

Figure 7: (a) Once a GCP has forced the disparity path through some disparity-shifted region, the occlusion will be proposed regardless of the cost of the occlusion jump. (b) The path between two GCPs will depend only upon the good matches in the path, since the occlusion cost is the same for each type of path. Path-d is the single exception, since an additional occlusion jump has been proposed. While that path is possible, it is unlikely the globally optimum path through the space will have any more occlusion jumps than necessary unless the data supporting a second occlusion jump is strong.

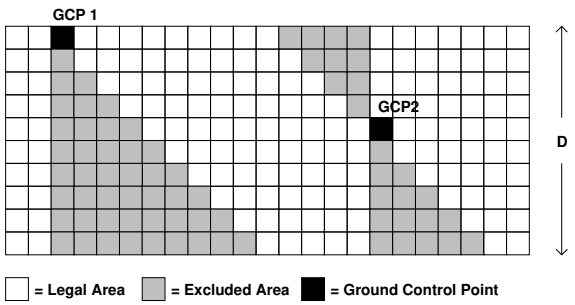

Figure 8: GCP constraint regions. Each GCP removes a pair of similar triangles from the possible solution path. If the GCP is at one extreme of the disparity range (GCP 1), then the area excluded is maximized at $D^2/2$. If the GCP is exactly in the middle of the disparity range (GCP 2) the areas is minimized at $D^2/4$.

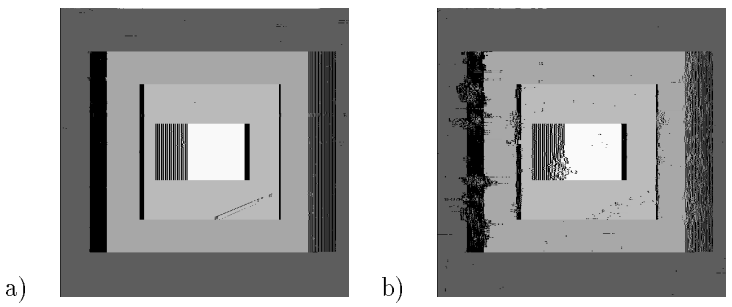

Figure 9: Results of our algorithm for the (a) noise-free and (b) noisy sloping wedding cake.

using the same occlusion cost which we chose through experimentation. This cost, however, can be varied by more than a factor of two without a major effect on the results.

Figure 10-a shows the "birch" image from the JISCT stereo test set[7]. 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 10-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 10-c shows the algorithm run with the same occlusion cost using GCPs. Most of the occlusion regions around the trees are recovered reasonably 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 might be minimized by changing the GCP selection algorithm to check for texture wherever GCPs are proposed. On some scanlines, no GCPs were recovered on some trees which led to the scanline gaps in some of the trees.

Figure 11-a is an enlarged version of the left image of Figure 1. Figure 11-b shows the results obtained by the algorithm developed by Cox *et al.*[12]. The Cox algorithm is a similar DP procedure which uses inter-scanline consistency instead of GCPs to reduce sensitivity to occlusion cost.

Figure 11-c shows our results on the same image. These images have not been converted to the cyclopean view, so black regions indicate regions occluded in the left image. The Cox algorithm does a reasonably good job at finding the major occlusion regions, although many rather large, spurious occlusion regions are proposed.

When the algorithm generates errors, the errors are more likely to propagate over adjacent lines, since inter- and intra-scanline consistency are used[12]. To be able to find the numerous occlusions, the Cox algorithm requires a relatively low occlusion cost, resulting in false occlusions. Our higher occlusion cost and use of GCPs finds the major occlusion regions cleanly. For example, the man's head is clearly recovered by our approach. The algorithm did not recover the occlusion created by the man's leg as well as hoped since it found no good control points on the bland wall between the legs. The wall behind the man was picked up well by our algorithm, and the structure of the people in the scene is quite good. Most importantly, *we did not use any smoothness or inter- and intra-scanline consistencies to generate these results.*

We should note that our algorithm does not perform as well on images that only have short match regions interspersed with many disparity jumps. In such imagery our conservative method for selecting GCPs fails to provide enough constraint to recover the proper surface. However, the results on the birch imagery illustrate that in real imagery with many occlusion jumps, there are likely to be enough stable regions to drive the computation.

7 Summary

We have presented a stereo algorithm that incorporates the detection of occlusion regions directly into the matching process. We develop an dynamic programming solution that obeys the occlusion and ordering constraints to find a best path through the disparity space image and does not use smoothness, intra- or inter-scanline consistency criteria. To eliminate sensitivity to occlusion cost we use ground control points (GCPs)—high confidence matches. These points improve results, reduce complexity, and minimize dependence on occlusion cost without arbitrarily restricting the recovered solution.

References

[1] H.H. Baker and T.O. Binford. Depth from edge and intensity based stereo. In *Proc. 7th Int. Joint Conf. Art. Intel.*, pages 631–636, 1981.

[2] F. Barnard. Computational stereo. *Computing Surveys*, 14:553–572, 1982.

[3] P. Belhumeur. Bayesian models for reconstructing the scene geometry in a pair of stereo images. In *Proc. Info. Sciences Conf.*, Johns Hopkins University, 1993.

[4] P. Belhumeur. A binocular stereo algorithm for reconstructing sloping, creased, and broken surfaces in the presence of half-occlusion. In *Proc. Int. Conf. Comp. Vis.*, 1993.

[5] 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.

[6] R.E. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

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

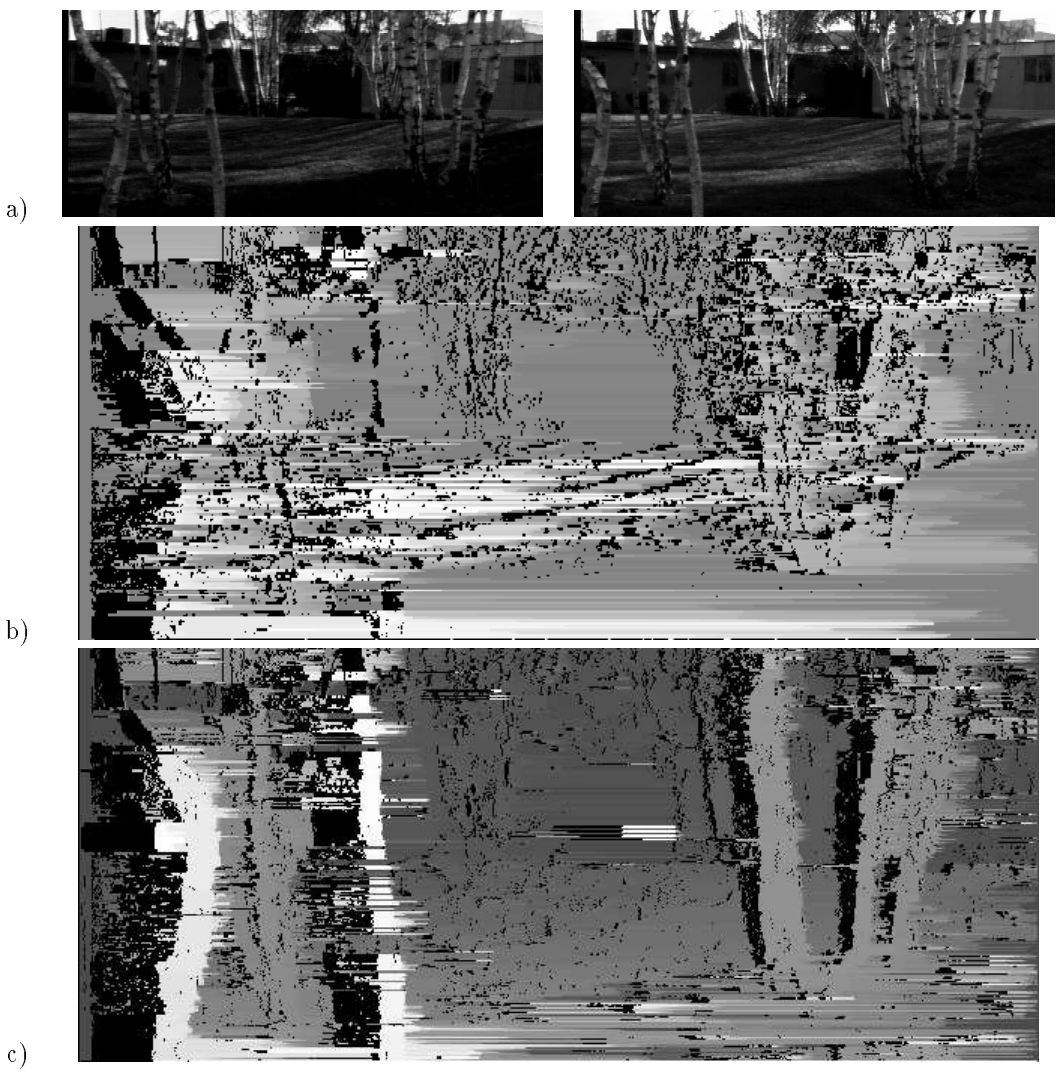

Figure 10: (a) The "birch" stereo image pair, which is a part of the JISCT stereo test set[7], (b) Results of our stereo algorithm without using GCPs, and (c) Results of of our algorithm with GCPs.

[8] R.C. Bolles and J. Woodfill. Spatiotemporal consistency checking of passive range data. SRI Technical Report – to be published, SRI International, September 1993.

[9] C. Chang, S. Catterjee, and P.R. Kube. On an analysis of static occlusion in stereo vision. In *Proc. Comp. Vis. and Pattern Rec.*, pages 722–723, 1991.

[10] R. Chung, , and R. Nevatia. Use of monocular groupings and occlusion analysis in a hierarchical stereo system. In *Proc. Comp. Vis. and Pattern Rec.*, pages 50–55, 1991.

[11] 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.

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

[13] U.R. Dhond and J.K. Aggarwal. Structure from stereo – a review. *IEEE Trans. Sys., Man and Cyber.*, 19(6):1489–1510, 1989.

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

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

[16] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: theory and experiment.

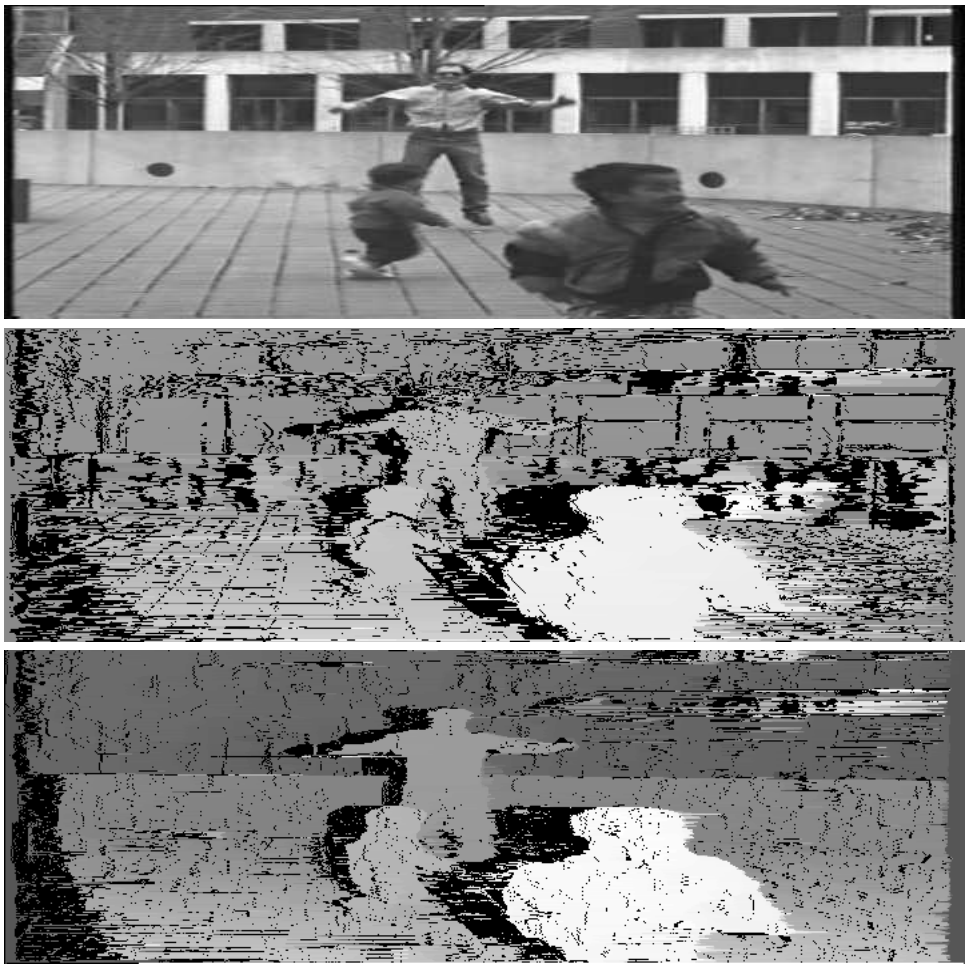

Figure 11: Results of two stereo algorithms on Figure 1. (a) Original left image. (b) Cox *et al.* algorithm[12], and (c) the algorithm described in this paper.

In *Proc. Image Understanding Workshop*, pages 383–389, 1990.

[17] J.J. Little and W.E. Gillett. Direct evidence for occlusion in stereo and motion. *Image and Vision Comp.*, 8(4):328–340, 1990.

[18] K. Nakayama and S. Shimojo. Da Vinci stereopsis: depth and subjective occluding contours from unpaired image points. *Vision Research*, 30(11):1811–1825, 1990.

[19] 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.

[20] S. Shimojo and K. Nakayama. Real world occlusion constraints and binocular rivalry. *Vision Research*, 30(1):69–80, 1990.

[21] Y. Yang, A. Yuille, and J. Lu. Local, global, and multi-level stereo matching. In *Proc. Comp. Vis. and Pattern Rec.*, 1993.