International Conference on
Computer Vision (ICCV '99),
Kerkyra, Greece, Sept. 1999

Multiway Cut for Stereo and Motion with Slanted Surfaces

Stan Birchfield
Department of Electrical Engineering
Stanford University
Stanford, California 94305
birchfield@cs.stanford.edu

Carlo Tomasi
Department of Computer Science
Stanford University
Stanford, California 94305
tomasi@cs.stanford.edu

Abstract

Slanted surfaces pose a problem for correspondence algorithms utilizing search because of the greatly increased number of possibilities, when compared with fronto-parallel surfaces. In this paper we propose an algorithm to compute correspondence between stereo images or between frames of a motion sequence by minimizing an energy functional that accounts for slanted surfaces. The energy is minimized in a greedy strategy that alternates between segmenting the image into a number of non-overlapping regions (using the multiway-cut algorithm of Boykov, Veksler, and Zabih) and finding the affine parameters describing the displacement function of each region. A follow-up step enables the algorithm to escape local minima due to oversegmentation. Experiments on real images show the algorithm's ability to find an accurate segmentation and displacement map, as well as discontinuities and creases, from a wide variety of stereo and motion imagery.

1 Introduction

The goal of *correspondence* is to determine which points in one image correspond to which points in another image taken of the same scene, i.e., to determine which image points arise from the same physical point in the world. The images may be taken by different cameras at the same time (stereo) or by the same camera at different times (motion).

The problem of correspondence is often solved by minimizing an energy functional that matches similar-looking pixels (in terms of intensity or color, for example), while penalizing the discontinuities in order to preserve piecewise-continuity. The result of such a search is the best mapping from pixels to displacements, according to the cost functional. In stereo the displacement is disparity (a scalar thanks to the epipolar constraint [10]), while in motion it is a two-element vector.

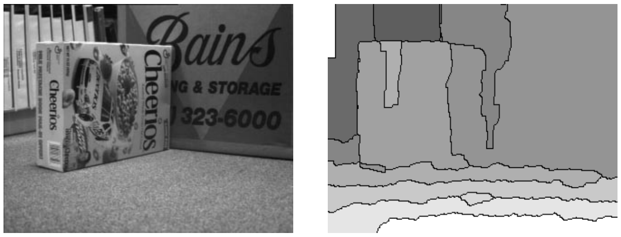

Figure 1: LEFT: An image from a stereo pair. RIGHT: The disparity map, with region boundaries overlayed, computed by the algorithm of [6], which searches over quantized disparities. The scene geometry is poorly captured by this output.

By searching over quantized disparities or motions, as is commonly done, what is preserved is actually piecewise-*constancy* rather than piecewise-continuity, thereby causing the scene geometry to be poorly captured when it contains slanted surfaces. In Figure 1, for example, the image is improperly segmented: each slanted surface is split into a series of constant-disparity regions and some regions contain more than one surface. The result, therefore, does not accurately represent the shape or orientation of the surfaces, nor are the discontinuities and creases easily recoverable from such an output. That is, differentiating and thresholding this disparity map will generate many false discontinuities because of the large jumps in disparity that occur within surfaces, and the vertical crease along the interior edge of the Cheerios box cannot be recovered because it lies in the middle of a region.

To handle slanted surfaces, we propose to minimize an energy functional that allows not just constant displacements but rather affine warpings. Our approach segments the image into a number of non-overlapping regions, each corresponding to a different surface in the world, and finds the affine parameters of the displacement function for each region. This is accomplished by alternating between two steps: (1) segmenting the image, that is, assigning a label to each pixel indicating to which region it belongs, using the multiway-cut algorithm of Boykov, Veksler, and Zabih

*Work supported by grants NSF IRI-9509149 and NSF IIS-9712833.

[6], and (2) finding the affine parameters of the displacement function for each region, using the method of Shi and Tomasi [14]. In this way, the algorithm greedily minimizes the energy functional until it converges. If the result is oversegmented, then an additional step merges adjacent regions to further reduce the energy.

After reviewing previous work and presenting our formulation in the next two sections, the two main steps of the algorithm are presented in Sections 4 and 5, respectively, followed in Section 6 by a solution to the oversegmentation problem. In the final section we present experimental results showing the algorithm's ability to find clean, accurate displacement maps and segmentations (from which discontinuities and creases can be inferred) from pairs of stereo and motion images containing severely slanted surfaces.

2 Previous Work

For years, many researchers have computed stereo correspondence by searching over all possible disparities along a scanline, which can be done efficiently using dynamic programming [2, 3, 8, 11]. These techniques, however, do not effectively incorporate information between scanlines.

Recently, stereo vision has experienced a breakthrough as maximum-flow-based techniques have been shown capable of minimizing energy functionals over the whole image, not just one scanline. Roy and Cox [12] and Ishikawa and Geiger [9] presented formulations that, with the right edge weights, can find the global minimum of such a functional.

Unfortunately their approach cannot preserve sharp discontinuities, which led Boykov, Veksler, and Zabih [6] to develop another maximum-flow-based algorithm that finds a good local minimum of a more general class of cost functionals which preserve sharp discontinuities. Additionally, their algorithm is able to minimize vector-valued functions, making it applicable to situations such as motion, although in [6] it was applied only to stereo. Vector-valued functions present a challenge, however, because of the additional computational complexity. In motion, for example, there are approximately $O(\Delta^2)$ possible displacements, compared to $O(\Delta)$ in stereo, where Δ is the maximum displacement in one direction. With slanted surfaces the number of possibilities increases dramatically to $O(n\Delta^2\sigma^2)$, where there are n pixels in the image and σ different possible orientations in one direction. In this paper we extend the work of [6] to handle vector-valued functions with large search spaces.

Our approach is similar to expectation-maximization (EM) algorithms [1, 15, 16] which iteratively segment an image into regions of affine motion. The multiway-cut algorithm performs the work of the E-step, while the affine parameters are fit in a manner similar to the M-step. Because the EM algorithms assign the labels probabilistically, however, they require suboptimal techniques for enforcing spatial consistency.

Another graph-based technique for performing image segmentation is the normalized cut algorithm [13]. In the context of stereo or motion, however, this method groups together pixels with similar profiles, where the profiles are influenced by the dissimilarities of pixels at incorrect displacements. In contrast, the multiway-cut algorithm ignores these misleading values.

3 Correspondence as segmentation

We represent correspondence between two images as a labeling $f : \mathbf{x} \rightarrow l$ for each pixel $\mathbf{x} = [\,x \quad y\,]^T$, along with a displacement function $h_l(\mathbf{x})$ for each label l. Pixels with the same label belong to the same region, so f represents a segmentation. The corresponding pixel in the other image, then, is given by $h_{f(\mathbf{x})}(\mathbf{x})$.

These displacement functions are constant if all the surfaces in the world can be assumed fronto-parallel, that is, parallel to the image plane. In [6], for example, $h_l(\mathbf{x}) = l$. With slanted surfaces, however, h_l is not constant with respect to the coordinates $\mathbf{x}$ of the pixel. In this paper we will concentrate on the affine model $h_l(\mathbf{x}) = A\mathbf{x} + \mathbf{d}$, as explained in more detail in Section 5; this framework could be extended to more sophisticated models as well.

Our goal is to find a correspondence that matches pixels of similar intensity while minimizing the number of discontinuities. We accomplish this by minimizing the following two-dimensional energy functional:

$$\gamma(f) = E_D + E_S, \tag{1}$$

where E_D is a data-dependent energy term containing the costs of assigning the labels to the pixels:

$$E_D = \sum_{\mathbf{x}} g(\mathbf{x}, f(\mathbf{x}))$$

and E_S enforces smoothness by penalizing the discontinuities:

$$E_S = \sum_{(\mathbf{x},\mathbf{x}')} \kappa(\mathbf{x},\mathbf{x}')[f(\mathbf{x}) \neq f(\mathbf{x}')].$$

The first summation is over all pixels $\mathbf{x}$ in the image, while the second summation is over every pair of neighboring pixels $\mathbf{x}$ and $\mathbf{x}'$ (using 4-neighborhood connectedness, for example). The assignment cost is the dissimilarity in image intensity: $g(\mathbf{x}, f(\mathbf{x})) = |I(\mathbf{x}) - J(h_{f(\mathbf{x})}(\mathbf{x}))|$, where I and J are the two intensity images. We set $\kappa(\mathbf{x},\mathbf{x}')$ to be proportional to the inverse of the magnitude of the gradient of intensity at that location, thresholded, in order to align the discontinuities with the intensity edges [4, 6, 7].

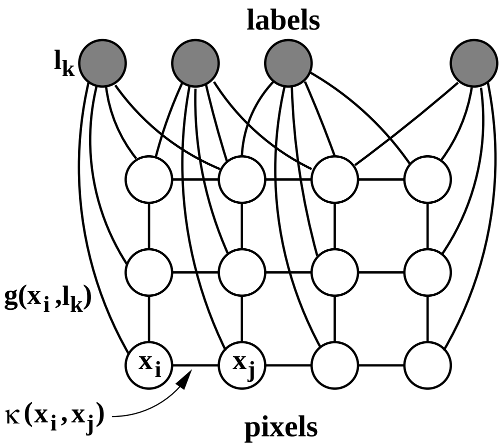

Figure 2: Minimizing Eq. (1) is equivalent to finding the minimum cost mutiway cut of this graph. Every label is connected to every pixel, although some connections have been omitted from the drawing to avoid clutter.

MULTIWAY-CUT ALGORITHM

1. Start with an initial labeling.
2. Pick two labels. Set one label-vertex as the source s and the other as the sink t, and find the minimum $s-t$ cut through the graph containing only those pixels that are already labeled either of the two labels, and only those edges connecting these pixels to each other or to one of the two labels. This minimum cut will then separate the two labels in this temporary graph, thus reassigning the pixels under consideration.
3. Repeat Step 2 for every pair of labels.
4. Repeat Steps 2 and 3 until the energy in the system does not change.

To minimize the energy functional, therefore, we alternate between segmenting the image into disjoint regions by assigning a label to every pixel and finding the affine parameters of the displacement function for every region. These two steps are discussed in the next two sections, respectively.

4 Assigning labels to pixels

Once the displacement functions are known, the segmentation, or labeling, problem is equivalent to the multiway cut problem on a certain graph [6]. This graph, shown in Figure 2, contains a vertex for every pixel in the image and a vertex for every possible label. Each pixel is connected to its four neighbors by four edges having capacities equal to the discontinuity penalty between the two appropriate pixels, and each label is connected to each pixel with an edge having a capacity equal to the cost of assigning the label to that pixel. Minimizing Eq. (1) then is the same as finding the minimum cost multiway cut of this graph, where a multiway cut is a set of edges such that the labels are not connected with each other in the induced graph (i.e., the graph with these edges removed). Once the multiway cut is found, each pixel will be connected to exactly one label.

To find an approximate solution to this problem, we use the algorithm of Boykov, Veksler, and Zabih [6], shown at the top of the page. Unfortunately, the minimum cost multiway cut problem is NP-complete [6], and, in fact, minimizing Eq. (1) has been shown to be NP-hard as well [5]. As a result, the algorithm is not guaranteed to find the global minimum. However, it does find a good local minimum, in the sense that the final energy cannot be lowered by exchanging any subset of pixels having a common label with any other subset of pixels having a common label. Moreover, the algorithm is extremely insensitive to the initial labeling, falling into local minima only when there are large untextured surfaces in the scene (in which case there is not enough local information to guide properly the search).

While there is no guarantee of the number of iterations[1] needed for convergence, in practice we have found two to be necessary initially, and only one after that (See Figure 7).

After the multiway-cut algorithm has converged, the connected components of the output are found, in order to separate regions which may be assigned the same label but are not physically connected. Then we find the displacement function for each region, as explained below.

5 Finding displacement functions

The affine model describes exactly the motion of a plane in the world viewed under orthographic projection. Under perspective projection it is usually adequate when only small motions are involved. Using this model, a point $\mathbf{x} = [\,x \quad y\,]^T$ in image I moves to $A\mathbf{x} + \mathbf{d}$ in image J, where

$$A = \begin{bmatrix} d_{xx}+1 & d_{xy} \\ d_{yx} & d_{yy}+1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{d} = \begin{bmatrix} d_x \\ d_y \end{bmatrix}.$$

The motion of each region, then, is described by a six-element vector $\mathbf{z} = [\,d_{xx} \quad d_{xy} \quad d_x \quad d_{yx} \quad d_{yy} \quad d_y\,]^T$.

[1] By *iteration,* we mean Steps 2-3, collectively.

To find the motion of a region, we minimize the dissimilarity

$$\epsilon = \int\!\!\int_W \left[J(A\mathbf{x}+\mathbf{d}) - I(\mathbf{x})\right]^2 \, d\mathbf{x}, \qquad (2)$$

where W is the set of pixels in the region. Following [14], we differentiate Eq. (2) with respect to the unknown entries in A and **d** and set the result to zero. The resulting system is then linearized about the current estimate by truncating the Taylor series expansion of $J(A\mathbf{x}+\mathbf{d})$, yielding the following linear system:

$$T\mathbf{z} = \mathbf{a}, \qquad (3)$$

where

$$\begin{aligned} T &= \int\!\!\int_W \mathbf{g}\mathbf{g}^T \, d\mathbf{x} \\ \mathbf{a} &= \int\!\!\int_W [I(\mathbf{x}) - J(\mathbf{x})]\mathbf{g} \, d\mathbf{x}. \end{aligned}$$

The motion of the region can be found by using Eq. (3) iteratively in a Newton-Raphson style minimization.

The elements of the vector **g** are image coordinates multiplied by derivatives of image intensity: $\mathbf{g} = [\mathbf{u} \quad \mathbf{v}]^T$, where $\mathbf{u} = (\partial J/\partial x)\mathbf{p}$, $\mathbf{v} = (\partial J/\partial y)\mathbf{p}$, and $\mathbf{p} = [\, x \quad y \quad 1\,]$. These equations are identical to those in [14] but with simplified notation.

In the case of rectified stereo images, $d_{yx} = d_{yy} = d_y = 0$, so the disparities in a region are described by a vector with only three elements: $\mathbf{z} = [\, d_{xx} \quad d_{xy} \quad d_x \,]$, which is found in the same manner as before but with $\mathbf{g} = \mathbf{u}^T$.

Either way, the minimization continues until either the parameters in **z** do not change significantly or the dissimilarity in the region increases.

6 Handling oversegmentation

When these two alternating steps settle on an answer for the correspondence, the result is occasionally oversegmented. An extreme example is presented in Figure 3, in which the ground plane is covered by five different regions.

Solving this problem is rather straightforward. Every pair of adjacent regions is considered, and affine parameters are fit to the union of the two (See Figure 4). If the new internal energy is less than the sum of the two individual internal energies and the cost of the discontinuity, then the regions are merged, thereby lowering the overall energy of the system. This process is repeated until no two regions can be merged to decrease the energy.

One could imagine situations in which the image is undersegmented. If such were the case, one could generate a large number of candidate affine parameters (by dividing the region and fitting parameters to the subregions, for example) and then run the multiway cut algorithm to reassign some of the pixels to some of the new labels. Generating such parameters will not be easy, however, and we have not encountered any undersegmentation in our images (according to the cost functional).

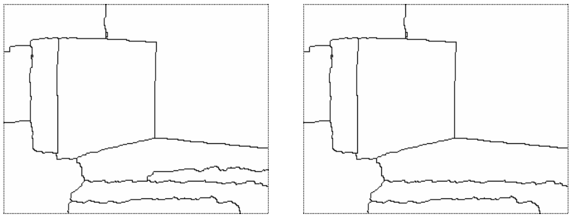

Figure 3: LEFT: Segmentation of the Cheerios image after the convergence of the multiway cut and affine-parameter fitting steps. RIGHT: Two regions on the ground plane have been merged, with more to follow.

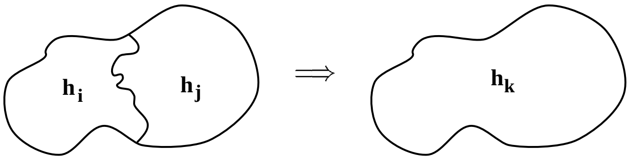

Figure 4: Oversegmentation: Two regions are merged if affine parameters for the union reduce the energy.

7 Experimental results

We present the results of the algorithm on three pairs of stereo images and two pairs of frames from motion sequences, as shown in Figure 6. These results demonstrate the algorithm's ability to find accurate dispacements and segmentations for a wide variety of imagery.[2] Notice in particular the precision with which the object contours are recovered in many cases, such as the outline of the basketball player in the last row.

In the first row, the results are nearly perfect. Each of the surfaces is properly segmented, with the only mistake being that of splitting the books, which are to the left of the Cheerios box, in two. Comparing these results with those of Figure 1, we see that the scene geometry is now accurately recovered. To help visualize the disparities computed by the algorithm, we have provided a three-dimensional reconstruction of the scene in Figure 5. From this, one can tell that the orientations of the surfaces are recovered accurately. Notice, for example, that the two faces of the Cheerios box meet along a line, the boxes meet the ground

[2] All the results were generated using the same set of parameters, except for the threshold used in the right column of the figure.

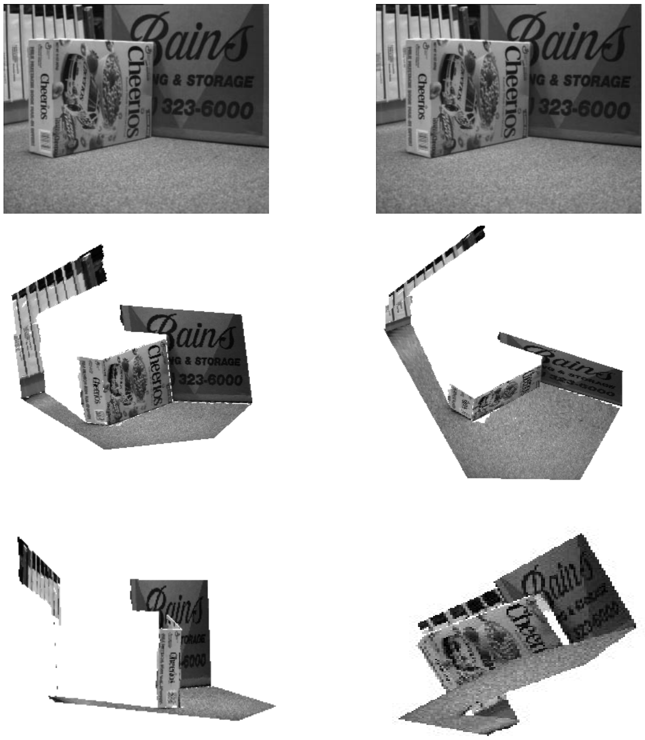

Figure 5: TOP: Stereo images, displayed for cross-eyed viewing. MIDDLE and BOTTOM: 3D reconstruction, as texture-mapped surfaces, from novel viewpoints.

plane at right angles, and the two regions corresponding to the books are, although not merged, nearly coplanar.

In the second row, whose images are from the well-known JISCT data set, the individual bushes, automobile, and two buildings are correctly segmented. Notice that the main building is correctly recovered as a single, slanted plane, not the usual pair of fronto-parallel planes. Although we may wish to have the parking meters segmented from the bushes, there is actually very little evidence in terms of disparity for such a conclusion; it takes an extremely small discontinuity penalty (which of course introduce many false discontinuities — see the results in [6]) to segment even the closer one.

The middle row shows the limitations of a simple cost functional like Eq. (1). Because there is little texture on the Clorox box and no intensity edges along most of the vertical crease, the lowest cost solution incorrectly follows the logo on the front of the box instead of the actual crease. Our algorithm does successfully minimize the functional, but the functional does not represent the world in this case. Notice, however, that much of the scene is accurately recovered, such as the creases between the floor and the boxes and many of the depth discontinuities around the Clorox box.

The fourth image contains complex motion due to the handheld camera, a person walking in the foreground, and a bicyclist peddling in the background. Nevertheless, all three planes defining the world (the ground plane and the two walls of the building) are correctly segmented from each other. The extra region under the arch appears to be caused partly by the motion of the bicyclist. Because the camera translation is rather small, there is little information to distinguish the various surfaces in the static world, which explains why the creases are in slightly incorrect locations and why the bottom of the statue is grouped with the ground plane. Notice, however, the detailed contour of the torso of the statue, as well as the outline of the pedestrian, whose lower leg is moving in a different direction from the rest of his body.

In the last row, the basketball player is accurately segmented from the crowd (even his elbow is well-preserved), and the ball is nearly completely segmented from the player. Although it is not visible in the figure, the motion of the crowd varies across the image, so that an algorithm searching over quantized motions would split it in two.

We have already seen how the final step to handle over-segmentation is key to recovering the ground plane in the Cheerios image. It also played a minor role in two other images by merging four pairs of regions to form the player's left arm and basketball, his body and right arm, and the two regions of near and far bushes (with parking meters).

After careful investigation we have concluded that none of the images is undersegmented, according to the cost functional. Specifically, we tried to find separate affine parameters for the parking meter and the bush behind it, but the resulting energy was higher than the result displayed in Figure 6. Similarly, if either arm of the basketball player is separated from the rest of its region, the energy increases.

A typical run of the algorithm is shown in Figure 7, where the energy of the system is plotted versus time. From these data we notice that the most significant iteration is the first application of the multiway-cut algorithm using quantized displacements, which reduces the energy by an amazing 80% in just one step. (Figure 1 shows the output after two iterations.) The energy is then steadily and quickly reduced by alternating between the multiway-cut segmentation and the fitting of affine parameters. Notice that many of the multiway-cut iterations shown here are not necessary: only the first two initially and the first one after every affine fitting. Thus, these same results could be achieved in just 11 iterations. The step to handle over-segmentation further reduces the energy by another 10% on this image, though its impact on other images was less noticeable.

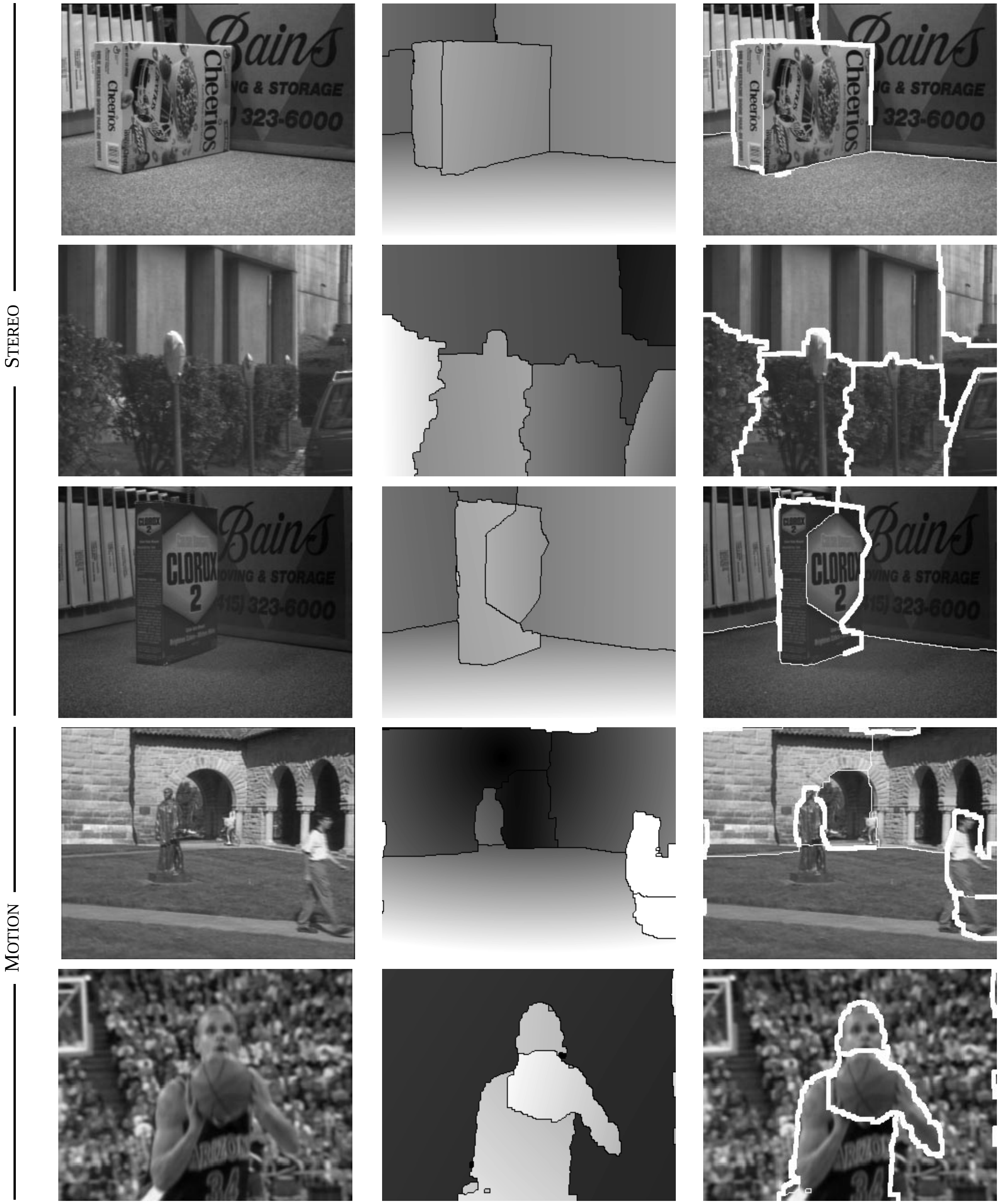

Figure 6: LEFT: An image from a stereo or motion pair. MIDDLE: The displacement map (either disparity or motion vector magnitude), with segmentation overlaid. RIGHT: The image with segmentation overlaid. Lines are thickened where the change in displacement across the boundary surpasses a threshold, thus distinguishing depth or motion discontinuities (thick lines) from creases (thin lines). These images are also available at `http://vision.stanford.edu/˜birch`.

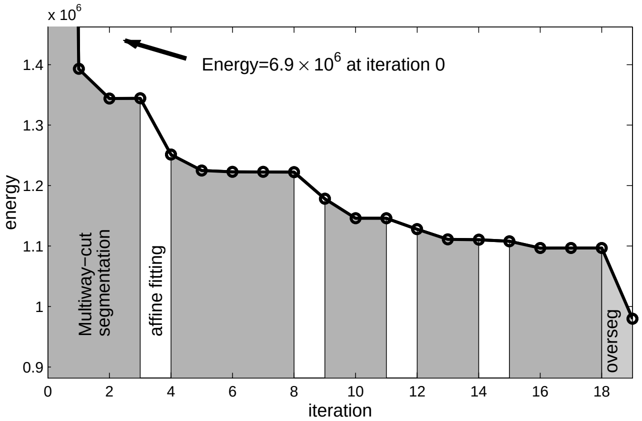

Figure 7: The algorithm greedily decreases the energy by alternating between the two steps of Sections 4 and 5, followed by a single run of the oversegmentation step. These data are from the Cheerios image.

8 Conclusion

Stereo and motion algorithms that search over all possible displacements to minimize an energy functional have traditionally assumed that all the surfaces in the world are parallel to the image plane. We have presented an algorithm to solve the correspondence problem in the presence of slanted surfaces by alternately segmenting an image into non-overlapping regions and finding the affine parameters of the displacement function of each region. An additional step enables the algorithm to recover when this alternation settles onto a suboptimal oversegmentation. This iterative, greedy algorithm is able to find clean, accurate displacement maps for a wide range of images from stereo and motion.

The main limitation of this work is the restrictiveness of the energy functional used. For example, the algorithm may become stuck in local minima if there are extremely untextured surfaces in the world, in which case it will be difficult to automatically determine their affine parameters. Moreover, it is easily distracted when intensity edges do not accompany the region boundaries, and it prefers to draw region boundaries along straight lines, thus ensuring a bias against tracing the contours of curved objects. Future work should be aimed at incorporating occlusions, the curvature of boundaries, or the shape of regions.

Acknowledgments

Thanks to Scott Cohen for his suggestions regarding the manuscript and to A. V. Goldberg for the maximum-flow code.

References

[1] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *Proc. of the 5th International Conference on Computer Vision (ICCV)*, pages 777–784, 1995.

[2] H. H. Baker and T. O. Binford. Depth from edge and intensity based stereo. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 631–636, 1981.

[3] P. N. Belhumeur. A binocular stereo algorithm for reconstructing sloping, creased, and broken surfaces in the presence of half-occlusion. In *ICCV*, pages 431–438, 1993.

[4] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. In *ICCV*, pages 1073–1080, 1998.

[5] Y. Boykov, O. Veksler, and R. Zabih. Energy minimization with discontinuities. Submitted for publication to *International Journal of Computer Vision*, 1998.

[6] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 648–655, 1998.

[7] P. Fua. Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities. In *IJCAI*, pages 1292–1298, 1991.

[8] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *International Journal of Computer Vision*, 14(3):211–226, 1995.

[9] H. Ishikawa and D. Geiger. Segmentation by grouping junctions. In *CVPR*, pages 125–131, 1998.

[10] V. S. Nalwa. *A Guided Tour of Computer Vision*. Reading, MA: Addison-Wesley, 1993.

[11] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):139–154, 1985.

[12] S. Roy and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *ICCV*, pages 492–499, 1998.

[13] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, pages 1154–1160, 1998.

[14] J. Shi and C. Tomasi. Good features to track. In *CVPR*, pages 593–600, 1994.

[15] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, 1994.

[16] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *CVPR*, pages 321–326, 1996.