Correspondence as Energy-based Segmentation

Stanley T. Birchfield and Braga Natarajan
Dept. of Electrical and Computer Engineering
Clemson University
Clemson, SC 29634
{stb,nnatara}@clemson.edu

Carlo Tomasi
Department of Computer Science
Duke University
Durham, NC 27708
tomasi@cs.duke.edu

Corresponding author:
Stanley T. Birchfield
207-A Riggs Hall
Clemson University
Clemson, SC 29634
phone: 864-656-5912
fax: 864-656-5910
email: stb@clemson.edu

Keywords: stereo, motion, correspondence, segmentation, multiway-cut, graph cuts, affine, energy minimization

Abstract

We pose the correspondence problem as one of energy-based segmentation. In this framework, correspondence assigns each pixel in an image to exactly one of several non-overlapping regions, and it also computes a displacement function for each region. The framework is better able to capture the scene geometry than the more direct formulation of matching pixels in two or more images, particularly when the surfaces in the scene are not fronto-parallel. To illustrate the framework, we present a specific correspondence algorithm that minimizes an energy functional by alternating between (1) segmenting the image into a number of non-overlapping regions using the multiway-cut algorithm of Boykov, Veksler, and Zabih; and (2) finding the affine parameters describing the displacement of the pixels in each region. After convergence, a final step escapes local minima due to over-segmentation. The basic algorithm is extended in two ways: using ground control points to detect long, thin regions; and warping segmentation results to efficiently process image sequences. Experiments on real images show the algorithm's ability to find an accurate segmentation and displacement map, as well as discontinuities and creases, on a wide variety of stereo and motion imagery.

1 Introduction

Given multiple images of a scene, the goal of visual correspondence is to determine which image points are projections of the same world point. In the case of stereo the images are taken at the same time by different cameras, while in the case of motion the images are taken by the same camera at different times. Correspondence is a crucial step in recovering 3D geometric information about a scene from multiple images. 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 binocular stereo the displacement is a scalar (the disparity), while in two-frame motion it is a two-element vector.

By searching over quantized disparities or motions, as is commonly done, what is preserved is actually piecewise-*constancy* rather than piecewise-*continuity*, thereby implicitly making the assumption that surfaces in the scene, along with their movement in the case of motion, are parallel to the camera. As a result, the scene geometry is captured poorly when this assumption is violated. In Figure 1, for example, the image from a stereo pair 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 nor 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.

A solution to such errors can be found in the insightful layers approach [32]. Because the world generally consists of a number of cohesive objects separated by boundaries, solving the correspondence problem involves finding not only the displacement of each pixel but also the surface (or layer) to which each pixel belongs. In fact, this latter task of segmentation is in some ways more fundamental than the former task of determining pixel displacements, because the segmentation provides global constraints for estimating those displacements. By separating the correspondence problem into these two separate questions, several benefits are achieved: (1) the piecewise-continuity constraint is handled naturally, (2) the resulting displacements of the pixels are more accurate because they are computed with respect to

Figure 1: LEFT: An image from a stereo pair. RIGHT: The disparity map, with region boundaries overlaid, computed by the algorithm of Boykov, Veksler, and Zabih [10], which searches over quantized disparities. The scene geometry is poorly captured by this output.

their surfaces (and hence are real-valued) rather than to a set of predetermined quantized disparities, and (3) the locations and boundaries of objects are naturally computed along with the correspondence. A much richer and more natural representation of the scene emerges.

In this paper we explore this general framework in terms of a specific algorithm 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 it 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 technique of Boykov, Veksler, and Zabih [10]; and (2) finding the affine parameters of the displacement function for each region, using the method of Shi and Tomasi [31]. After these two steps converge on a result, a final step corrects potential over-segmentation by merging adjacent regions if doing so reduces the energy further. The basic algorithm is extended to find long, thin regions using ground control points, as well as to process image sequences efficiently by warping the segmentation from the previous frame. Experimental results demonstrate the algorithm's ability to find clean, accurate segmentations and displacement maps (from which discontinuities and creases can be inferred) from pairs of stereo and motion images containing slanted surfaces and multiple moving objects.

2 Comparison with Previous Work

Many authors have formulated the correspondence problem as that of energy minimization. Early algorithms focused on stereo images and utilized the epipolar constraint to convert the 2D problem to a 1D problem that can be solved efficiently using dynamic programming [3, 4, 14, 26]. Several years ago researchers discovered that 2D energy functionals can be efficiently and effectively minimized using graph cuts. The first such work in computer vision was that of Roy and Cox [28], who demonstrated that the global minimum of a certain type of 2D cost functional could be computed with graph cuts; unfortunately their formulation does not allow for sharp discontinuities in disparity, thus yielding poor results at the boundaries of objects. Shortly thereafter, Boykov, Veksler, and Zabih [9, 10] presented an alternative formulation of graph cuts known as the *multiway cut* that, while not guaranteed to find the global minimum, nevertheless finds a provably good local minimum while preserving sharp discontinuities. This work has since been extended in a number of publications addressing occlusion, multiple cameras, computational efficiency, clustering, recognition, and non-constant intensities (e.g., contrast reversal, different camera gain and bias, and non-Lambertian surfaces) [20, 19, 21, 22, 34, 8]. The multiway-cut technique is an important tool for solving the correspondence problem as well as

other related problems, and it forms the basis for the work presented in this paper.

Another popular approach to scene understanding is that of layers [2, 32, 33, 18, 12], which like our technique formulates the correspondence problem as one of segmentation. These techniques use expectation-maximization (EM) to iteratively segment an image into regions of common (usually affine) motion. In our technique, 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. Our approach can be seen as a layered technique that enforces spatial consistency in a principled, energy-based manner using multiway cuts.

Recently, several correspondence algorithms have been proposed that match regions rather than pixels [11, 27, 24, 25, 16, 7]. In these techniques, the image is first segmented using monocular cues, then the correspondence between the regions is determined. Hong and Chen [16], for example, first segment each image independently using a color-based mean shift algorithm. Although these techniques are quite successful when applied to untextured or color images, they generally do not work on textured gray-level images because of the difficulty of obtaining a monocular segmentation in such a case. In contrast, the algorithm presented here does not simply compute the correspondence between regions after first segmenting. Rather, it simultaneously computes the segmentation as part of the correspondence. We believe that this approach of binocular segmentation, rather than monocular segmentation, is more natural.

Some researchers have attempted binocular segmentation using the profiles of pixels in the two images, where the profiles are computed by computing the dissimilarities with potential matching candidates in the other image. Shi and Malik, for example, apply their normalized cuts algorithm to motion segmentation in this manner [30]. This is similar to the mass-spring model of Blake and Zisserman [6] with one spring for each possible displacement. It is important to realize that any segmentation technique using pixel profiles suffers from the fundamental flaw that the profiles are influenced by the dissimilarities of pixels at incorrect displacements. In contrast, the multiway-cut formulation is able to ignore these misleading values because it effectively cuts the springs attached to the incorrect displacements.

3 General formulation

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 $h_{f(\mathbf{x})}(\mathbf{x})$.

If all the possible displacement functions can be enumerated *a priori*, then the problem of correspondence involves only one step: Each pixel in the image must be assigned to a region. Traditional formulations often follow such an approach, thereby assuming that the surfaces in the scene are parallel to the image plane [10]. In the case of motion, the movement of the surfaces is also assumed to be parallel to the image plane. If this assumption holds, then the displacement functions are constant ($h_l(\mathbf{x}) = l$). With a small baseline (stereo) or high frame rate (motion), the number of possible displacements is reasonably small so that their exhaustive enumeration is feasible.

When the scene becomes more complicated, however, this straightforward approach breaks down. In the case of slanted surfaces, curved surfaces, non-fronto-parallel movement, or non-rigid motion, one cannot hope to enumerate all the possible displacement functions *a priori*. With slanted motion surfaces, for example, there are approximately $O(n\Delta^2\sigma^2)$ possible displacement functions, assuming n

pixels in the image, Δ possible displacements in one direction, and σ different possible orientations in one direction. Contrasted with Δ possible displacement functions for rectified fronto-parallel stereo, this is a significant increase in computational expense. For any reasonable discretization of the space, slanted surfaces require several orders of magnitude more computation, thus rendering brute force search over all possibilities infeasible. Curved surfaces or non-rigid motion require even larger search spaces.

In what follows we describe a computationally efficient approach for handling such complicated scenes. Our goal is to find a correspondence that matches pixels of similar intensity while minimizing the number of discontinuities. This is accomplished by minimizing the following two-dimensional energy functional:

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

where

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

is a data-dependent energy term containing the costs of assigning the labels to the pixels, and

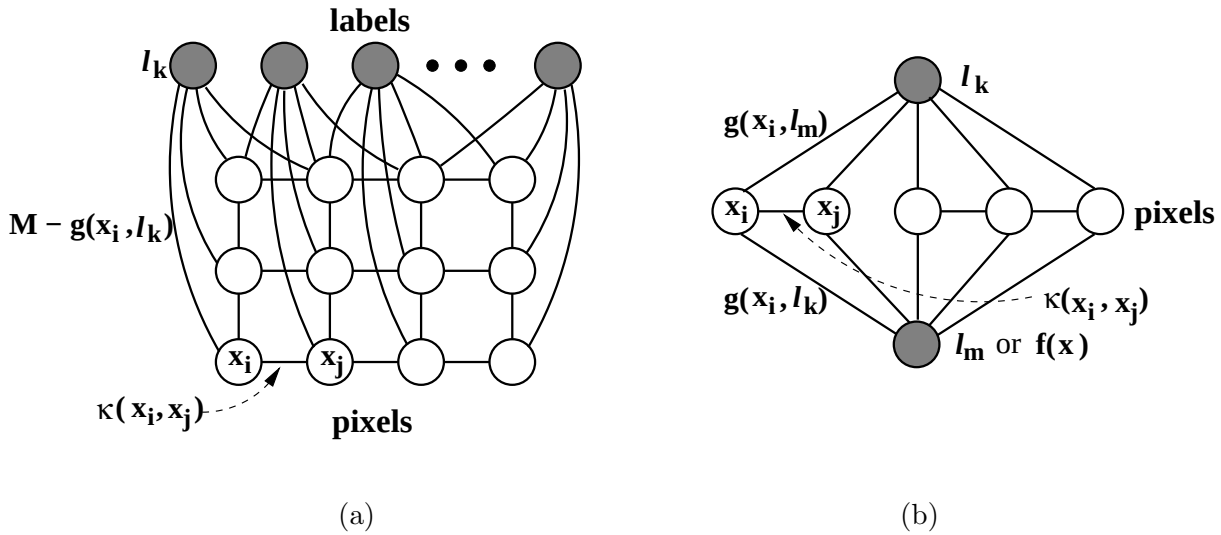
$$E_S = \sum_{(\mathbf{x},\mathbf{x}')} \kappa(\mathbf{x}, \mathbf{x}')[f(\mathbf{x}) \neq f(\mathbf{x}')]$$

enforces smoothness by penalizing the discontinuities. 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 absolute difference 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. In order to align the discontinuities with the intensity edges [5, 9, 13], the value of the discontinuity penalty depends upon the thresholded magnitude of the gradient of intensity: $\kappa(\mathbf{x}, \mathbf{x}') = \lambda_1$ if $|I(\mathbf{x}) - I(\mathbf{x}')| < \tau$, and $\kappa(\mathbf{x}, \mathbf{x}') = \lambda_2$ otherwise, where $\lambda_1 > \lambda_2$ and τ are constants. We also tried other edge detectors such as Canny, but the results were not affected.

To minimize the energy functional, two sets of parameters must be determined: the function f that describes the segmentation, and the functions $h_{f(\mathbf{x})}$ that encode the displacements of the regions. These parameters are determined by alternating between two steps: (1) segmenting the image into disjoint regions by assigning a label to every pixel, and (2) finding the affine parameters of the displacement function for each region. The first step computes f, while the second computes $h_{f(\mathbf{x})}$. A final step handles over-segmentation and, if necessary, under-segmentation, by considering the energy that would result by splitting and merging regions. These three steps are discussed in the next three sections, respectively.

4 Assigning labels to pixels

Assuming for the moment that all the possible displacement functions are known, the assignment problem (i.e., the segmentation) can be formulated using the multiway cut of the weighted graph shown in Figure 2a, which contains a vertex for every pixel in the image and a vertex for every possible label [10]. Each pixel is connected to its four neighbors by four edges with weights equal to the discontinuity penalty between the two pixels $\kappa(\mathbf{x}_i, \mathbf{x}_j)$, and each label is connected to each pixel by an edge whose weight is equal to the negative cost of assigning the label to that pixel $-g(\mathbf{x}_i, l_k)$ (to which is added a large constant M to ensure non-negative weights). Minimizing Eq. (1) is the same as finding the minimum-cost multiway cut of this graph, where a multiway cut is a set of edges such that there is no path from any label to any other label in the induced graph formed by removing these edges. As a result, once the multiway cut is found, each pixel is connected to exactly one label.



(a) (b)

Figure 2: (a) Minimizing Eq. (1) is equivalent to finding the minimum-cost multiway cut of this graph. Every label is connected to every pixel, although some connections have been omitted from the drawing to avoid clutter. (b) Computing the minimum-cost s–t cut of this graph constitutes one iteration of the multiway-cut technique. For the α-β-swap algorithm, the middle layer contains only those pixels whose current label is either l_k or l_m, and the bottom vertex is the label l_m. For the α-expansion algorithm, the middle layer contains all the pixels in the image, and the bottom vertex is not a single label but rather the current label $f(\mathbf{x})$ of each pixel $\mathbf{x}$. In either case, all the pixels in the middle layer (which is two-dimensional in reality) are connected to both labels, while each pixel is connected only to those pixels which are its neighbors in the image. Observe that the large but otherwise arbitrary constant M, which is needed in the problem formulation, is not needed in the actual implementation of either algorithm (since the dissimilarity functions $g(\cdot)$ are swapped with respect to the labels).

In their seminal work, Boykov, Veksler, and Zabih [10] describe two algorithms, α-β-swap and α-expansion, for solving the multiway-cut problem. Both algorithms find the minimum-cost multiway cut of a graph by repeatedly finding minimum-cost single cuts of graphs derived from the original graph. In both cases each single cut determines the best relabeling of a subset of the pixels using two of the many possible labeling choices, with the difference between the algorithms being the way these labeling choices are defined. The α-β-swap algorithm considers each pair of labels l_k and l_m in turn, and for each of these pairs all the pixels currently assigned to either l_k or l_m are reassigned to one of these two labels in order to minimize the overall energy of the cost functional. In contrast, the α-expansion algorithm considers each label l_k in turn, and for each such label all the pixels in the image either retain their current label or are relabeled with l_k.

The single-cut problems for the two algorithms are illustrated in Figure 2b. For α-β-swap, the graph consists of two special vertices, known as the source and the sink, corresponding to the labels l_k and l_m. In addition, there is a vertex for each pixel currently assigned to one of these two labels. Edges connect each of the pixels to both the source and the sink, and edges also connect pixels to those of their image neighbors. The former edges are assigned weights of $g(\mathbf{x}_i, l_m)$, while the latter are assigned weights of $\kappa(\mathbf{x}_i, \mathbf{x}_j)$. Once the graph is constructed, the problem is to find the minimum-cost cut that separates the source (the vertex l_k) from the sink (the vertex l_m). Several algorithms exist for finding such a cut, usually known as an s-t cut because it separates the source (s) from the sink (t). For each pixel, either its edge to the source will be cut, in which case the pixel is assigned the label l_m, or its edge to the sink

is cut, in which case it is assigned the label l_k. Notice that although a large constant M is needed in the original graph, it is not needed in this graph because the edges connecting pixels to the source or sink have been reversed.

Similarly, for α-expansion, the graph consists of a source and a sink, corresponding to the label l_k and a label meaning, "existing label", respectively. The graph also contains a vertex for every pixel, in contrast to the α-β-swap graph which has vertices for only a subset of the pixels. Edge weights are assigned in a similar manner as before, and the minimum-cost s-t cut is found in the same way. Again, for each pixel either its edge to the sink will be cut, in which case the pixel is assigned the label l_k, or its edge to the source is cut, in which case its current label is retained.

Because minimizing Eq. (1) is NP-hard [10], neither of these algorithms, which operate in an iterative, greedy manner can be guaranteed to find the global minimum. Nevertheless, both of them find a strong 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 (α-β-swap), or by assigning any subset of pixels to a particular label (α-expansion). Moreover, under certain conditions it can be proved that the minimum found is within a known constant factor of the global minimum [10]. The algorithms work well in practice, producing a local minimum that is very close to the global minimum no matter what the initial labeling (as long as the original images are reasonably textured). We have found that simply labeling all the pixels initially with l_0 works well. While there is no guaranteed bound on the number of cycles needed for convergence (By *cycle*, we mean computing the single cut of the graph in Figure 2b for all labels or pairs of labels), in practice we have found two to be necessary initially, and only one after that (See Figure 9).

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. Regions that are too small (approximately 1% of the total image area or less) are discarded. Then the displacement function is found for each remaining region, as explained in the next section.

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

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

The motion of each region, then, is described by a six-element vector $\mathbf{z} = [\, d_{xx} \quad d_{xy} \quad d_x \quad d_{yx} \quad d_{yy} \quad d_y \,]^T$. We will concentrate on the affine model $h_l(\mathbf{x}) = A\mathbf{x} + \mathbf{d}$, but this framework could be extended to other models, such as projective [15] or B-splines [23,], as well.

To find the motion of a region, the dissimilarity

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

is minimized, where W is the set of pixels in the region. Following [31], Eq. (2) is differentiated with respect to the unknown entries in A and $\mathbf{d}$, and the result is set 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:

$$T\mathbf{z} = \mathbf{a}, \tag{3}$$

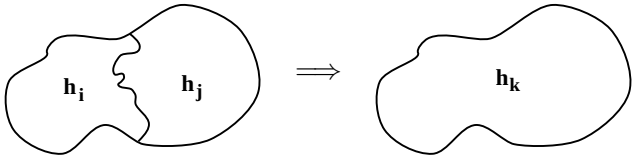


Figure 3: Over-segmentation: Two regions are merged if affine parameters for the union reduce the energy.

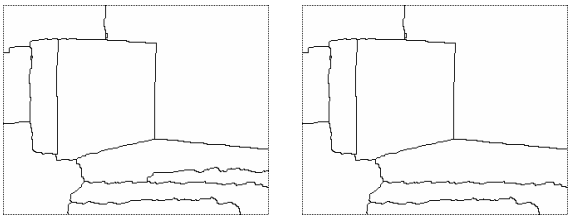


Figure 4: LEFT: Segmentation of the Cheerios image after the convergence of the multiway cut and affine-parameter fitting steps. RIGHT: Two regions on the ground plane have been merged, with more to follow.

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 $\mathbf{g}$ 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 [31] 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 $\mathbf{z}$ do not change significantly or the dissimilarity in the region increases.

6 Handling over-segmentation

Greedily alternating between the two steps just mentioned could potentially lead to a local minimum due to over- or under-segmentation. Handling over-segmentation is rather straightforward. Every pair of adjacent regions is considered, and affine parameters are fit to the union of the two (See Figure 3). 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. An extreme example of this computation in progress is presented in Figure 4, in which the ground plane is covered by five different regions.

To test and correct under-segmentation, one must divide existing regions, fit affine parameters to the subregions, and retain the subregions if the energy is lowered. Of these steps, the first one is open-ended: There are many ways to divide a region. One technique would be to select two pixels in the region

at random and to perform a floodfill operation in parallel for them until every pixel in the region is connected to one of the two initial pixels. A simpler but more restricted approach would be to divide the region in half using a line at a random angle going through the region centroid. Empirically, we have found that if enough displacement parameters are allowed initially, then the algorithm is much less likely to encounter under-segmentation as it is to encounter over-segmentation. In our experiments, we have not yet found an occurrence of under-segmentation, according to the cost functional.

7 Experimental results

In this section we present the results of the algorithm on various stereo image pairs and pairs of image frames from motion sequences. These results demonstrate the algorithm's ability to find accurate displacements and segmentations for a wide variety of imagery. We first present qualitative results, followed by quantitative results using images with ground truth.

7.1 Qualitative results

Three stereo pairs of both indoor and outdoor scenes, along with the results of our algorithm, are shown in Figure 5. In the first row, the results are quite accurate. Each of the surfaces is properly segmented, with the only mistake being that of splitting the books (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, a three-dimensional reconstruction of the scene is shown in Figure 6. 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 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, such a separation would actually increase the energy of the result.

The last 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.

More generally, Eq. (1) causes the algorithm to balance the two goals of matching similar pixels and producing a piecewise smooth disparity map. The former goal assumes that the two cameras have the same photometric properties and that the surfaces in the world are Lambertian. If these assumptions do not hold, then preprocessing of the images may be necessary. Even if the assumptions do hold, the matching will be locally ambiguous if there is not enough local variation in intensity (i.e., texture), leading to the need for the piecewise smooth prior. This latter goal is enforced using a lattice-based Markov random field [10], causing the algorithm to prefer to define segment boundaries as vertical or horizontal lines, in the absence of local intenstiy information. To improve upon the results shown here, one could include higher-order *a priori* information about the shapes of regions, such as enforcing that the edges of the box are straight. Such information would require object recognition beforehand,

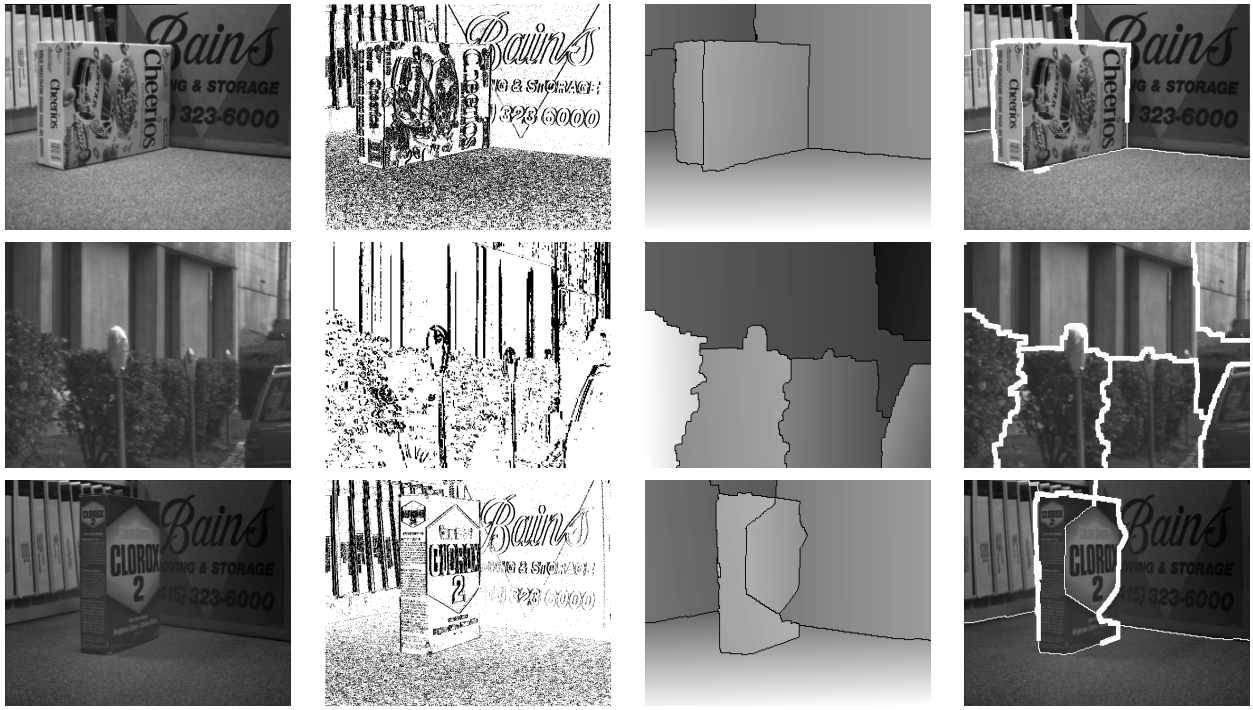


Figure 5: LEFT: An image from a stereo pair. 2ND COLUMN: The thresholded gradient of the image, with black pixels indicating large gradients. 3RD COLUMN: The disparity map, with segmentation overlaid. RIGHT: The image with segmentation overlaid. Lines are thickened where the change in displacement across the boundary surpasses a threshold of two, thus distinguishing depth discontinuities (thick lines) from creases (thin lines).

however, leading to the important but unsolved problem of how to balance bottom-up and top-down information.

Figure 7 shows the results of the algorithm on two pairs of motion frames. The first row 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 plays 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

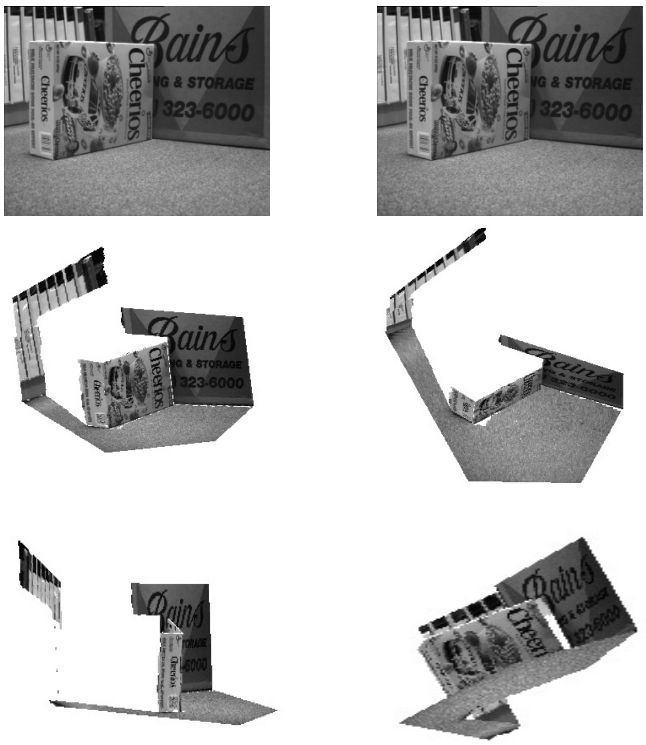


Figure 6: TOP: Stereo images, displayed for cross-eyed viewing. MIDDLE and BOTTOM: 3D reconstruction, as texture-mapped surfaces, from novel viewpoints.

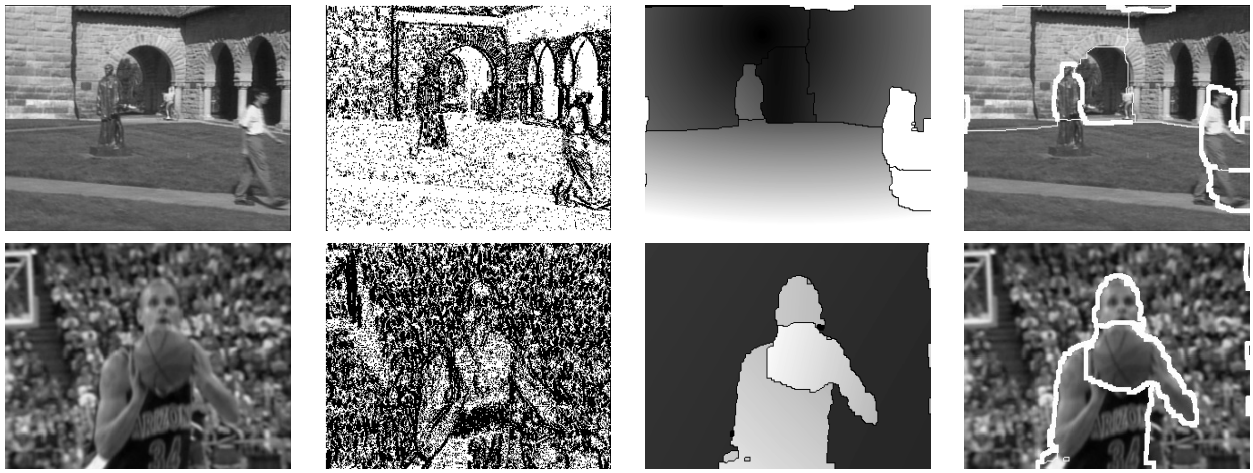


Figure 7: LEFT: An image from a pair of motion frames. 2ND COLUMN: The thresholded gradient of the image. 3RD COLUMN: The displacement map (magnitude of the motion vector), with segmentation overlaid. RIGHT: The image with segmentation overlaid.

the images is under-segmented, 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 5. Similarly, if either arm of the basketball player is separated from the rest of its region, the energy increases.

7.2 Quantitative results

In this section we present quantitative results of our algorithm on stereo images, using the database of Scharstein and Szeliski [29, 1]. Figure 8 shows the left image of each pair in the top row, the disparity map computed by our algorithm in the middle row, and the disparity map computed by the current leading algorithm of Hong and Chen [16] in the last row. The overall errors of our algorithm on these images are 0.53, 0.26, 0.61, and 8.08, respectively, while the overall errors of the Hong-Chen technique are 0.08, 1.49, 0.30, and 1.23. Although the Hong-Chen algorithm significantly outperforms ours on three of these images, our algorithm produces superior results on one of them (the Map image, second column), both qualitatively and quantitatively. Also notice that we are able to recover the shape of the video camera in the Tsukuba image, which Hong-Chen is not able to do. Keep in mind that our algorithm operates on gray-scale images, while Hong-Chen requires color images. In the original comparison our algorithm was ranked 1st, 3rd, 4th, and 17th on these images, respectively, out of 20 algorithms (see Table 5 of [29]), although more recent algorithms have changed those rankings. It is important to keep in mind that our algorithm and general framework are designed to compute real-valued disparity maps at subpixel resolution and to operate on motion images as well as stereo. These key features of our approach are not captured by Scharstein and Szeliski's original integral-disparity stereo comparison which counts as errors only those pixels whose disparity is greater than one pixel from the ground truth disparity.

7.3 Algorithm operation

A typical run of the algorithm is shown in Figure 9, 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.

7.4 Parameters

The parameters used for the experiments were $\tau = 5$ gray levels, $\lambda_1 = 12$, and $\lambda_2 = 6$. For the stereo images, the disparity values considered ranged from 0 to 30. For the motion images, the displacements ranged from $(-2, -10)$ to $(2, 2)$ for freethrow and from $(-5, -5)$ to $(15, 5)$ for statue. We have found that the range of disparities or displacements does not affect the results, although it does have a significant impact upon the running time. The images used in the experiments are listed in Table 1 along with their sizes, the number of iterations, and the time taken by the algorithm to compute the result. The number of iterations listed involves a pair of multiway cut and affine minimization as one iteration.

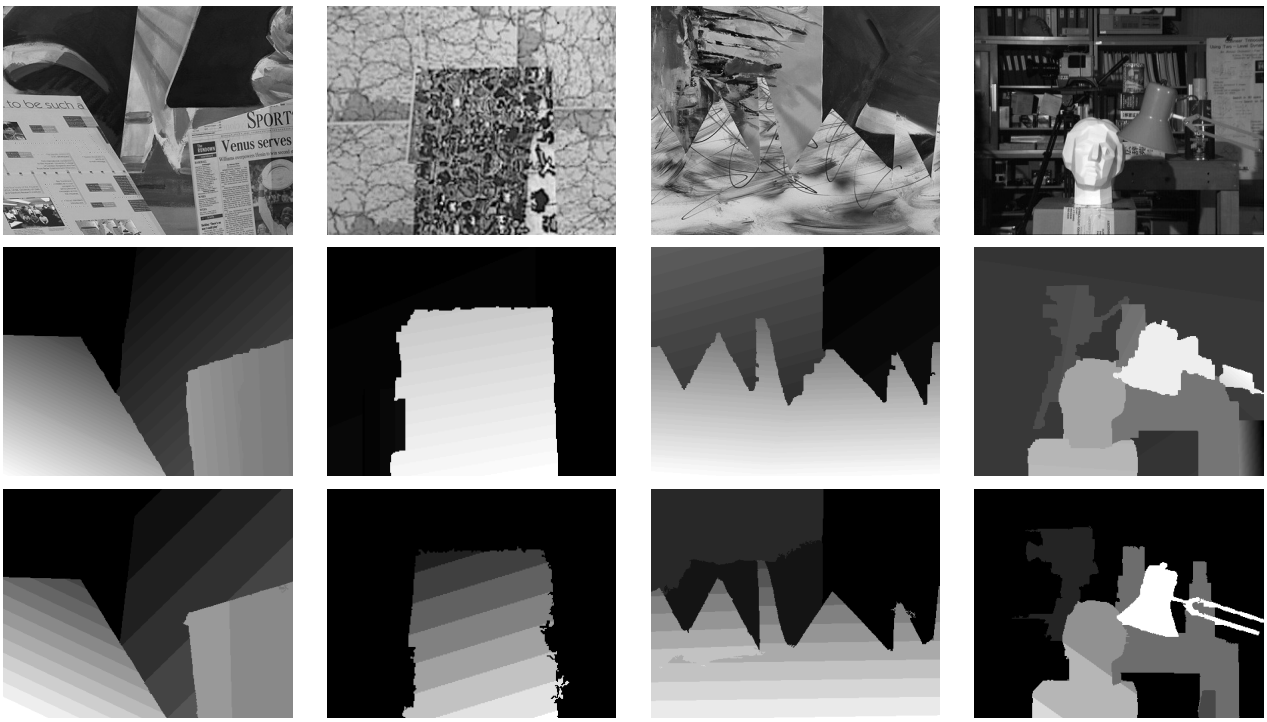


Figure 8: TOP: The left image from each stereo pair of the original Middlebury database (from left to right: Venus, Map, Sawtooth, Tsukuba). MIDDLE: Disparity map computed by our algorithm. BOTTOM: Disparity map computed by the algorithm of Hong and Chen [16].

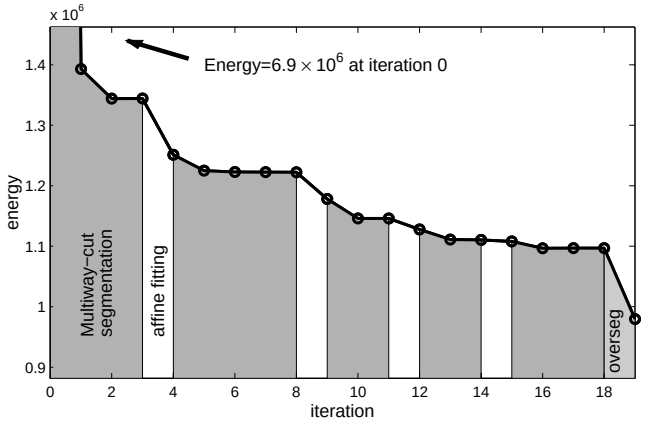


Figure 9: The algorithm greedily decreases the energy by alternating between the two steps of Sections 4 and 5, followed by a single run of the over-segmentation step. These data are from the Cheerios image.

image	size	iterations	computing time (sec.) multiway-cut	affine	overseg	total
Cheerios	640×480	8	201	33	30	264
parking meter	512×480	5	111	15	38	164
Clorox	640×480	5	116	19	24	159
Venus	434×383	4	59	7	10	76
map	284×216	2	11	1	7	19
sawtooth	434×380	3	47	5	39	91
Tsukuba	384×288	2	23	2	18	43
statue	318×243	10	109	8	10	127
freethrow	315×237	5	29	4	6	39

Table 1: The number of iterations and computing time of the algorithm on the images used in the experiments.

The computing time is based upon an unoptimized Visual C++ implementation running on a 2.8 GHz Pentium 4 computer with 256 MB RAM.

8 Extensions

In this section we present two extensions to the basic algorithm.

8.1 Using ground control points

As explained in the experimental results of the previous section, the proposed algorithm is extremely effective at minimizing the energy functional in Equation (1). The errors in the result are not due to the inability of the algorithm to find the global minimum, but rather to the inability of the simple cost functional to accurately describe the world. One place where this limitation is particularly noticeable is the case of long, thin objects. Because the cost functional simply penalizes the number of pixels along the borders of regions, the algorithm favors regions that are compact in space. Long, thin objects will not be found, because the smoothness term overwhelms the data term.

To solve this problem, we use the notion of ground control points (GCPs) [17]. Before any energy minimization is performed, the intensities of the pixels in each image are compared with all the possible matches in the other image. This yields one cross-correlation vector per pixel per image, where each element of the vector indicates the likelihood of matching one of the other pixels. If the vector has a strong, unambiguous minimum, then we declare the pixel to be a GCP. More specifically, we compute the pixels for which

$$C(\mathbf{x}; \delta^{\mathbf{x}}_{\min}) < \gamma C(\mathbf{x}; \delta), \quad \forall \delta \neq \delta^{\mathbf{x}}_{\min}, \tag{4}$$

where $C(\mathbf{x}; \delta)$ is the likelihood of matching pixel $\mathbf{x}$ at disparity δ, and $\delta^{\mathbf{x}}_{\min} = \arg\min_{\delta} C(\mathbf{x}; \delta)$. Pixels that pass this test using both the left and right cross-correlation vectors, as well as pass the left-right consistency check [13], are declared ground control points if a sufficient number of contiguous pixels agree on their disparity. The correlation is computed using a 5×5 window, and we use a threshold of 60 pixels as the minimum size of a GCP region. The constant $\gamma \in [0, 1]$ governs the amount of unambiguity needed in order to declare a GCP; it is set empirically to 0.4.

The cost functional is modified to preserve the correspondence of the GCPs. A higher cost is incurred

image GCPs disparity map

Figure 10: LEFT: Parking meter image, CENTER: Ground control points (GCPs) shown in black, RIGHT: Resulting disparity map.

if the disparity of a GCP is changed significantly from the minimum value of the cross-correlation vector:

$$g(\mathbf{x}, f(\mathbf{x})) = \begin{cases} |I(\mathbf{x}) - J(h_{f(\mathbf{x})}(\mathbf{x}))| & \text{if } |\delta_{f,h}(\mathbf{x}) - \delta^{\mathbf{x}}_{\min}| \leq 0.5 \\ \eta|I(\mathbf{x}) - J(h_{f(\mathbf{x})}(\mathbf{x}))| & \text{otherwise,} \end{cases} \tag{5}$$

where $\delta_{f,h}(\mathbf{x}) = \mathbf{x} - h_{f(\mathbf{x})}(\mathbf{x})$ is the disparity of the pixel $\mathbf{x}$, the value of η is set to 6, and the threshold of 0.5 is used to ignore roundoff error.

The results are shown in Figure 10. Compared with Figure 5, the algorithm exhibits improved behavior. The front two bushes are separated from each other, and the pole of the first parking meter is separated from the background even though there is only a one-pixel difference in their disparities. The specularities on the parking meter prevent a clear delineation of the entire meter due to a lack of GCPs on it. Notice also that one edge of the pole of the second meter is recovered, along with the side mirror of the car.

8.2 Processing an image sequence

The computational time of the algorithm is dependent upon the number of labels considered. For most images, the total number of segments found is small, ranging from approximately two to ten. Yet, because we do not know a priori how many segments exist, nor the proper displacement functions for each segment, the algorithm must search over many possibilities. For example, we search over 31 disparities for the stereo images, while for motion the search can include hundreds of displacements. By taking advantage of the temporal continuity between image frames of a sequence, the motion of objects can be found with significantly less processing.

We augment the basic algorithm in the following manner. In the first pair of frames of the sequence the algorithm is applied as before, since there is no additional information. In subsequent frames, the segmentation computed for the immediately preceding pair of frames is projected onto the images using the displacement functions of the regions. Only the labels from the previous pair of frames is used, thus dramatically reducing the computational cost. When a new object enters the scene, there is no requirement that it be labeled correctly, because the connected components step automatically generates a new label for the object, as long as the motion of the object matches one of the non-background labels. In the case that the object motion is more similar to the background than to any existing foreground object, a new label is proposed with the displacement function corresponding to the dominant motion of the new area computed efficiently by cross-correlation.

The extension was run on the 'Hamburg Taxi' sequence. Although this sequence has poor illumination and little texture, the algorithm effectively segments the three vehicles, as shown in Figure 11. Because the vehicle on the right side is severely occluded by the tree, errors result in its segmentation in several

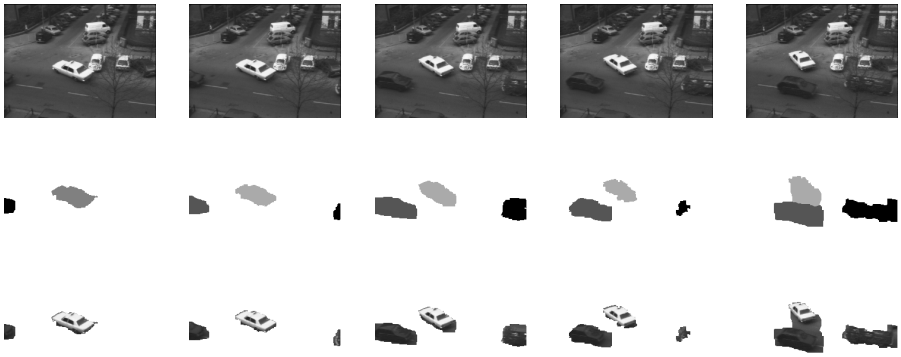


Figure 11: Segmentation results for frames 1,5,18,22, and 36 the 'taxi' sequence. TOP: original images, MIDDLE: segmentation, and BOTTOM: images overlaid on segmentation.

of the frames. The other two vehicles are segmented correctly, with a few errors in the labeling of road pixels near the end of the sequence due to the decelerating motion of the taxi. For these results, the L2 metric was used for the data cost, since L1 metric resulted in over-smoothing of regions due to the low contrast between the dark car on the left and the dark background. Use of the L1 metric necessitated squaring the smoothness parameters to $\lambda_1 = 100$ and $\lambda_2 = 40$. The algorithm is not sensitive to these parameters, with similar results being obtained when they were varied in the range of $\pm 30\%$. The sequence consists of 36 images, each of size 256×191. The execution time for the sequence was 13 seconds for the first pair of frames (using an initial search range from $(-3, -3)$ to $(3, 3)$ in the x and y directions) and an average of 2.6 seconds for each subsequent pair.

9 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. As a result, they are unable to capture the scene geometry well when the assumption is violated. In this paper we have presented a framework for solving the correspondence problem by casting it as an energy-based segmentation problem. We have described a specific algorithm that alternates between two steps: (1) segmenting an image into non-overlapping regions using the powerful multiway-cut formulation of Boykov, Veksler, and Zabih; and (2) finding the affine parameters of the displacement function of each region using Newton-Raphson minimization. An additional step enables the algorithm to recover when this alternation settles onto a suboptimal over-segmentation. This iterative, greedy algorithm is able to find clean, accurate displacement maps for a wide range of images from stereo and motion, even in the presence of slanted surfaces. In addition to these qualitative results, quantitative results on the Middlebury stereo database show that the accuracy of the algorithm is comparable to other leading algorithms.

We have also presented two extensions to the basic algorithm. First, ground control points are used to guide the energy minimization to preserve the correspondences that can be obtained reliably in a local fashion. In this manner, the bias of the Markov random field against long, thin objects is overcome. Secondly, a method is presenting for processing a multi-frame image sequence by projecting regions onto the current frame using the displacement of the previous frame. The method reduces the number

of labels that need to be searched, thus significantly reducing the processing time.

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 determine automatically 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 A. V. Goldberg and V. Kolmogorov for sharing their code, and to R. Zabih for pointing out an error in a previous version. This work was partially supported by grants NSF IRI-9509149 and NSF IIS-9712833.

References

[1] Middlebury College stereo vision research page, http://www.middlebury.edu/stereo.

[2] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In *Proceedings of the 5th International Conference on Computer Vision*, pages 777–784, June 1995.

[3] 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*, pages 631–636, 1981.

[4] P. N. Belhumeur. A binocular stereo algorithm for reconstructing sloping, creased, and broken surfaces in the presence of half-occlusion. In *Proceedings of the 4th International Conference on Computer Vision*, pages 431–438, May 1993.

[5] S. Birchfield and C. Tomasi. Depth and motion discontinuities. *International Journal of Computer Vision*, 35(3):269–293, Dec. 1999.

[6] A. Blake and A. Zisserman. *Visual Reconstruction*. Cambridge, MA: MIT Press, 1987.

[7] M. Bleyer and M. Gelautz. A layered stereo matching algorithm using image segmentation and global visibility constraints. *Photogrammetry and Remote Sensing*, 59:128–150, 2005.

[8] Y. Boykov and D. Huttenlocher. Spatially coherent matching and Bayesian recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1999.

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

[10] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[11] L. Cohen, L. Vinet, P. T. Sander, and A. Gagalowicz. Hierarchical region based stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1989.

[12] T. Darrell and A. Pentland. Cooperative robust estimation using layers of support. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):474–487, 1995.

[13] P. Fua. Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 1292–1298, 1991.

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

[15] M. Gleicher. Projective registration with difference decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[16] L. Hong and G. Chen. Segment-based stereo matching using graph cuts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[17] S. S. Intille and A. F. Bobick. Disparity-space images and large occlusion stereo. In *Proceedings of the 3rd European Conference on Computer Vision*, pages 179–186, May 1994.

[18] A. D. Jepson and M. J. Black. Mixture models for optical flow. In *Proceedings of the 4th European Conference on Computer Vision*, pages 760–761, 1993.

[19] J. Kim, V. Kolmogorov, and R. Zabih. Visual correspondence using energy minimzation and mutual information. In *Proceedings of the International Conference on Computer Vision*, 2003.

[20] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proceedings of the International Conference on Computer Vision*, 2001.

[21] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *Proceedings of the European Conference on Computer Vision*, 2002.

[22] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[23] M. Lin and C. Tomasi. Surfaces with occlusions from layered stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[24] A. Lopez, F. Pla, and J. Ribelles. 3D modeling of structured scenes through binocular stereo vision. In *Scandinavian Conference on Image Analysis*, 2001.

[25] A. Mittal and L. S. Davis. M2Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, 51(3):189–203, 2003.

[26] 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, Mar. 1985.

[27] S. Randriamasy and A. Gagalowicz. Direct cooperation between region-based stereo matching and top-down segmentation. In *Machine Vision and Applications*, 1992.

[28] S. Roy and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proceedings of the 6th International Conference on Computer Vision*, pages 492–499, 1998.

[29] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, 2002.

[30] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug. 2000.

[31] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

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

[33] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 321–326, 1996.

[34] R. Zabih and V. Kolmogorov. Spatially coherent clustering with graph cuts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.