Stereo Using Monocular Cues within the Tensor Voting Framework

Philippos Mordohai, *Member*, *IEEE*, and Gérard Medioni, *Fellow*, *IEEE*

Abstract—We address the fundamental problem of matching in two static images. The remaining challenges are related to occlusion and lack of texture. Our approach addresses these difficulties within a perceptual organization framework, considering both binocular and monocular cues. Initially, matching candidates for all pixels are generated by a combination of matching techniques. The matching candidates are then embedded in disparity space, where perceptual organization takes place in 3D neighborhoods and, thus, does not suffer from problems associated with scanline or image neighborhoods. The assumption is that correct matches produce salient, coherent surfaces, while wrong ones do not. Matching candidates that are consistent with the surfaces are kept and grouped into smooth layers. Thus, we achieve surface segmentation based on geometric and not photometric properties. Surface overextensions, which are due to occlusion, can be corrected by removing matches whose projections are not consistent in color with their neighbors of the same surface in *both* images. Finally, the projections of the refined surfaces on both images are used to obtain disparity hypotheses for unmatched pixels. The final disparities are selected after a second tensor voting stage, during which information is propagated from more reliable pixels to less reliable ones. We present results on widely used benchmark stereo pairs.

Index Terms—Stereo, occlusion, pixel correspondence, computer vision, perceptual organization, tensor voting.

1 INTRODUCTION

The premise of shape from stereo comes from the fact that, in a set of two or more images of a static scene, the same world point appears at a different position in each image. Given the images, the position of the point in the world can be determined as the intersection of at least two rays that go through the point's projections in the images and the optical centers of the cameras. Thus, two pixels that are the projections of the same point in different images and camera calibration information are sufficient for 3D reconstruction. Establishing pixel correspondences in real images, though, is far from trivial. Projective and photometric distortion, sensor noise, occlusion, lack of texture, and repetitive patterns make matching the most difficult stage of a stereo algorithm. Here, we focus on occlusion and insufficient or ambiguous texture, which are inherent difficulties of the depicted scene, and not of the sensors. We assume that camera calibration is provided to us.

To address these problems, we propose a stereo algorithm that operates as a perceptual organization process in the 3D disparity space, keeping in mind that false matches will most likely occur in textureless areas and near depth discontinuities. Since binocular processing has limitations in these areas, we use monocular information to overcome them. We begin by generating matching hypotheses for every pixel within a flexible framework that allows the use of matches generated by several matching techniques. These matches are placed in a 3D (x, y, d) space, where d denotes the disparity. In this space, the correct matches align to form surfaces, while the wrong ones do not form salient structures. We can infer a set of reliable matches based on the support they receive from their neighbors as surface inliers via tensor voting [1]. These reliable matches are grouped into layers. Note that the term layer is used interchangeably with surface since by layer we indicate a smooth, but not necessarily planar, surface in 3D disparity space. The surfaces are refined by rejecting matches that are consistent in color with their neighbors in both images. The refined, segmented surfaces serve as the "unambiguous component," defined in a way similar to [2], to guide disparity estimation for the remaining pixels.

Segmentation using geometric properties is arguably the most significant contribution of our research. It provides very rich information on the position, orientation, and appearance of the surfaces in the scene. Moreover, grouping in 3D circumvents many of the difficulties of image segmentation. It is also a process that treats both images symmetrically, unlike other approaches where only one of the two images is segmented. Candidate disparities for unmatched pixels are generated after examining the color similarity of each unmatched pixel with its nearby layers. If the color of the pixel is compatible with the color distribution of a nearby layer, disparity hypotheses are generated based on the existing layer disparities and the disparity gradient limit constraint. Tensor voting is then performed locally and votes are collected at the hypothesized locations. Only matches from the selected layer cast votes to each candidate match. The hypothesis that is the smoothest continuation of the surface is kept. In addition, assuming that the occluded surfaces are partially visible and that the occluded parts are smooth continuations of the visible ones, we are able to extrapolate and estimate the depth of monocularly visible pixels. Under this scheme, smoothness with respect to both shape, in the form of surface continuity, and appearance, in the form of color similarity, is taken into account before disparities are

- P. Mordohai is with the Department of Computer Science, University of North Carolina, Chapel Hill, NC 27599-3715.
E-mail: mordohai@cs.unc.edu.
- G. Medioni is with the Institute for Robotics and Intelligent Systems, University of Southern California, Los Angeles, CA 90083-0273.
E-mail: medioni@iris.usc.edu.

Manuscript received 11 July 2005; revised 25 Oct. 2005; accepted 7 Nov. 2005; published online 13 Apr. 2006.
Recommended for acceptance by K. Daniilidis.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0367-0705.

0162-8828/06/$20.00 © 2006 IEEE Published by the IEEE Computer Society

Fig. 1. Left images from the "Sawtooth," "Tsukuba," "Venus," "Map," "Cones," and "Teddy" stereo pairs of the Middlebury Stereo Evaluation.

assigned to unmatched pixels. We present results on widely used, benchmark stereo pairs taken from the Middlebury Stereo Evaluation Web page (http://cat.middlebury.edu/stereo/), as well as less-controlled, outdoor images. The left image of the six image pairs of the Middlebury database can be seen in Fig. 1.

The work presented here is different from that of Lee and Medioni [3] and Lee et al. [4], where the focus is on showing that the problem of stereo can be addressed from a perceptual organization perspective. The earlier tensor voting-based approach gives little attention to problem specific constraints, but rather attempts to demonstrate the capability to infer the scene surfaces from initial matches generated by a naive operator. Surface smoothness is the dominant factor in all stages, with obvious side-effects at sharp discontinuities and small objects. The use of monocular information only occurs at the last postprocessing stage. On the other hand, monocular cues are taken into account at all stages of this paper and contribute significantly in increasing the performance, especially at the most difficult parts of the input images. Specifically, we propose a way of combining various sophisticated matching techniques that take occlusion and image sampling explicitly into account. We also explicitly address the systematic matching errors that occur next to depth discontinuities and in uniform regions. Finally, disparities for unmatched pixels are generated after examining their compatibility with nearby scene surfaces and not just by surface smoothness. The entire algorithm presented in [4] is essentially the second stage of the current algorithm. A preliminary version of this research appears in [5].

The paper is organized as follows: Section 2 reviews related work. Section 3 is an overview of the algorithm. Section 4 describes the initial matching stage, Section 5 the selection of correct matches using tensor voting, Section 6 the segmentation and refinement process, and Section 7 the disparity computation for unmatched pixels. Section 8 contains experimental results. Section 9 offers an analysis of computational complexity. Section 10 concludes the paper.

2 Related Work

In this section, we review research on stereo related to ours. We focus on area-based methods since their goal is a dense disparity map. Feature-based approaches are not covered, even though the matches they produce could be used in our framework. We also focus on approaches that handle discontinuities and occlusions explicitly. The input images are assumed to be rectified.

The problem of stereo is often decomposed as the establishment of pixel correspondences followed by surface reconstruction. These two processes, however, are strongly linked since the reconstructed pixel correspondences form the scene surfaces, while, on the other hand, the positions of the surfaces dictate pixel correspondences in the images. In the remainder of this paper, we describe how surface saliency is used as the criterion for the correctness of matches, as in [3] and [4]. Arguably, the first approach where surface reconstruction does not follow but interacts with feature correspondence is that of Hoff and Ahuja [6]. Matching and surface interpolation are integrated to ensure surface smoothness, except at depth discontinuities and creases. Edge points are detected as features and matched across the two images at three resolutions. Planar and quadratic surface patches are successively fitted and possible depth or orientation discontinuities are detected at each resolution. The patches that fit the matched features best are selected while the interpolated surfaces determine the disparities of unmatched pixels.

Research on dense area-based stereo with explicit treatment of occlusion includes numerous approaches (see [7] and [8] for comprehensive reviews of stereo algorithms). They can be categorized as follows: local, global, and approaches with extended local support, such as the one we propose. Local methods attempt to solve the correspondence problem using local operators in relatively small neighborhoods. Kanade and Okutomi [9] use matching windows whose size and shape adapt according to the intensities and disparities of the pixels included in them in order to include as many pixels from the same disparity level as possible. In [10], Veksler presents a method that takes into account the average matching error per pixel, the variance of this error, and the size of the window to define new matching costs and adapt the window size. Birchfield and Tomasi [11] introduce a new pixel dissimilarity measure that alleviates the effects of image sampling, which are a major source of errors when one attempts to establish pixel correspondence. Their experiments, as those of [12] and ours, demonstrate the usefulness of this measure. Unlike all previous approaches that attempt to include in the window a large number of pixels that share the disparity of the pixel under consideration, Agrawal and Davis [13] use the matching cost of [11] for windows that can contain up to two different disparity values. The assignment of disparities to the pixels in each window is a bilabeling problem that can be efficiently solved using graph cuts.

On the other hand, global methods arrive at disparity assignments by optimizing a global cost function that usually includes penalties for pixel dissimilarity and violation of the smoothness constraint. The latter introduces a bias for constant disparity at neighboring pixels, thus favoring fronto-parallel planes. Chronologically, the first global optimization approaches for stereo were based on dynamic programming. Since dynamic programming addresses the problem as a set of 1D subproblems on each epipolar line separately, these approaches suffer from inconsistencies across epipolar lines that appear as streaking artifacts. Ohta and Kanade [14] use edges to provide interscanline constraints in order to mitigate streaking. However, the problem has not been entirely eliminated despite the attention it has received from numerous researchers, including [15], [16], [17], [18], [19], [20].

Belhumeur and Mumford [15] propose a Bayesian approach to stereo which has been extended by Belhumeur [18]. After a Bayesian formulation of image formation, the authors consider three "worlds" as prior models for the scene, with each model being more complicated than the previous one. Optimization is performed by dynamic programming, taking into account depth, surface orientation, depth discontinuities, surface creases, and occlusion. The disparity gradient limit constraint is used to determine occluded pixels. A second stage of optimization, termed "iterated stochastic dynamic programming," is necessary to achieve interscanline smoothness. Bobick and Intille [16], [20] use highly reliable matches, which they termed ground control points, to constrain the path of dynamic programming in the disparity-space image (DSI) representation. The authors observe that edges in the DSI indicate likely occlusion edges, as well as that the presence of an occluded region in the left image should correspond to an intensity edge in the right image and vice versa. Geiger et al. [17] address stereo in a Bayesian framework with a smoothness prior that models occlusion and treats both images equally. It is based on the fact that a discontinuity in disparity must correspond to an occluded region in one of the images. Two off-center matching windows that avoid discontinuities either to the left or the right of the current pixel are used and the one with the minimum cost is selected as the correct match. The solution is found in matching space using dynamic programming. Cox et al. [21] propose a maximum likelihood formulation that requires fewer assumptions and prior models than the Bayesian treatments. They also propose a novel way to avoid traditional regularization by minimizing the total number of horizontal and vertical disparity discontinuities instead of adding a term in the cost function. A postprocessing step that enforces consistency between adjacent epipolar lines is necessary to limit the appearance of streaking artifacts. Birchfield and Tomasi [19] propose an approach based on dynamic programming with disparity propagation along columns according to reliability labels that are assigned to each pixel. Since the use of matching windows and intensity preprocessing are not valid at discontinuities, the algorithm operates at the pixel level using the pixel dissimilarity measure of [11]. The novelty of the cost function is that it rewards pixel matches, while penalizing the number of occlusions and not the total number of occluded pixels. Thus, it avoids the staircase-like results that are often produced by dynamic programming.

Consistency across epipolar lines is guaranteed by using graph cuts, which operate in 2D, to optimize the objective function. Roy and Cox [22] find the disparity surface as the minimum cut of an undirected graph. In this framework, scanlines are no longer optimized independently, with interscanline coherence enforced later in a heuristic way, but smoothness is enforced globally over the entire image. Ishikawa and Geiger [23] advance graph-cut stereo by explicitly modeling occlusion and uniqueness, using a directed graph. Pixels are classified as ordinary, edges, and junctions, with the latter two categories providing additional constraints. The set of energy functions that can be optimized, however, is limited to convex functions which do not perform well at discontinuities. Kolmogorov and Zabih [24] propose an optimization technique based on graph cuts that was first published by Boykov et al. [25], which is applicable to more general objective functions. This allows a better handling of occlusion, symmetric treatment of both images, and enforcement of the uniqueness constraint. In addition, unlike the majority of the methods presented so far in this section, the ordering constraint, which is violated by certain scene configurations, is no longer necessary. The authors extend their work to multiple images and at the same time improve its binocular performance in [26].

Between these two extremes of local "winner-take-all" methods and global optimization methods are approaches that use more reliable matches to propagate information that guides disparity estimation for less reliable pixels. Following Marr and Poggio [27], Zitnick and Kanade [28] employ the support and inhibition mechanisms of cooperative stereo to ensure the propagation of correct disparities and the uniqueness of matches with respect to both images without having to rely on the ordering constraint. Reliable matches, without competitors, are used to reinforce matches that are compatible with them, while, at the same time, they eliminate the ones that contradict them, progressively disambiguating more pixels. Luo and Burkhardt [29] propose a Bayesian cooperative stereo approach based on the minimization of a nonconvex cost function by deterministic relaxation. Inhibition is implemented based on the observation that the occlusion map of one image can be derived from the disparity map of the other image and not from its own disparity map. Zhang and Kambhamettu [30] extend the cooperative framework from single pixels to image regions, segmented in the reference image. Disparities are propagated within and among image segments according to a confidence measure. The size and shape of local support areas for each match are based on image segmentation. Occlusions are detected if the converged matching score is below a threshold indicating that no good match for the pixel was found.

A different method of aggregating support is nonlinear diffusion, proposed by Scharstein and Szeliski [31], where disparity estimates are propagated to neighboring points in disparity space until convergence. The disparity space contains the matching cost for all possible disparity values for each pixel. If a diffusion operation does not increase the certainty of a match it is not performed. This avoids the oversmoothing that would be caused by effectively increasing the support region at each iteration. Sun et al. [32] formulate the problem using an MRF with explicit handling of occlusions. In the belief propagation framework, information is passed to adjacent pixels in the form of messages whose weight also takes into account image segmentation. The process is iterative and has similar properties with nonlinear diffusion. This work is extended in [33] by reformulating the problem in a way that both images are treated symmetrically and the visibility constraint, which is more general than the uniqueness and ordering constraints, is employed. Processing alternates between computing disparity maps given occlusion maps and vice versa.

Sara [2] formally defines and computes the largest unambiguous component of stereo matching, which can be used as a basis for the estimation of less reliable disparities. Other similar approaches include those of Szeliski and Scharstein [12] and Zhang and Shan [34] who start from the most reliable matches and allow the most certain disparities to guide the estimation of less certain ones, while occlusions are explicitly labeled. A different approach employing genetic algorithms is proposed by Goulermas and Liatsis [35]. The image is uniformly divided into rectangular blocks and a symbiotic genetic algorithm operates on each block. The population of each block has two objectives: its self score, which is based on image intensities and gradients as well as geometric constraints, and the symbiotic score that enforces continuity between the blocks. Processing is parallel with interactions between adjacent blocks.

The final class of methods reviewed here utilizes monocular color cues (image segmentation) to guide disparity estimation. Birchfield and Tomasi [36] cast the problem of correspondence as image segmentation followed by the estimation of affine transformations between the images for each segment. The objective energy function does not favor constant disparity and fronto-parallel surfaces, but can account for affine warping and slanted surfaces. Initially, the image is segmented and then the affine parameters are estimated for each segment. The final disparity map is produced after a few iterations. Tao et al. [37] introduce a stereo matching technique where the goal is to establish correspondence between image regions rather than pixels. It achieves outstanding results in cases where traditional stereo fails, namely, in scenes with large uniform regions that lack any meaningful intensity variations. The parameters of the affine transformation of each image segment are optimized according to the projection of the segment on the target image, taking into account possible occlusions. Both these methods are limited to planar surfaces, unlike the one of Lin and Tomasi [38], who propose a framework where 3D shape is estimated by fitting splines, while 2D support is based on image segmentation. Processing alternates between these two steps until convergence.

Recently, Wei and Quan [39] proposed a region-based progressive algorithm where reliable matches are used as ground control points to provide disparity estimates for image regions, which are obtained from color segmentation of the reference image. Since the assumption is that regions have constant disparity, they are split if they contain ground control points with multiple disparities. Then, disparities are propagated from more to less reliable regions. Hong and Chen [40] also start by performing color segmentation of the reference image. Planes are fitted to each region and their disparities and parameters are optimized within a graph cut framework that operates on regions instead of pixels. The last two methods achieve outstanding performance on the Middlebury Stereo Evaluation data sets, with the exception of the "Map," where the failure of image segmentation proves to be catastrophic, especially near depth discontinuities. This problem occurs because all these approaches, except that of [32] and [33], use image segmentation as a hard constraint, whereas segmentation itself is by no means a trivial problem.

3 Overview of Our Approach

Our approach for the derivation of dense disparity maps from rectified image pairs falls into the category of area-based stereo since we attempt to infer matches for every pixel using matching windows. It has four steps, which are illustrated in Fig. 2, for the "Sawtooth" stereo pair. The steps are:

- *Initial matching*, where matching hypotheses are generated for every pixel by a combination of different matching techniques. The data set after this stage includes multiple candidate matches for each pixel in a 3D diaprity space and can be seen in Fig. 2b.
- *Selection of correct matches*, which uses tensor voting to infer the correct matches from the unorganized point cloud of the previous stage as inliers of salient surfaces. After tensor voting, uniqueness is enforced with respect to surface saliency and the data set contains at most one candidate match per pixel. The disparity map can be seen in Fig. 2c.
- *Surface grouping and refinement*, during which the matches are grouped into smooth surfaces, using the estimated surface orientations. These surfaces are refined by removing points that are inconsistent with their color distribution resulting in the disparity map of Fig. 2d.
- *Disparity estimation for unmatched pixels*, where the goal is the assignment of disparities that ensure smoothness in terms of both surface orientation and color properties of the layers. The final disparity map and the error map can be seen in Fig. 2e and Fig. 2f.

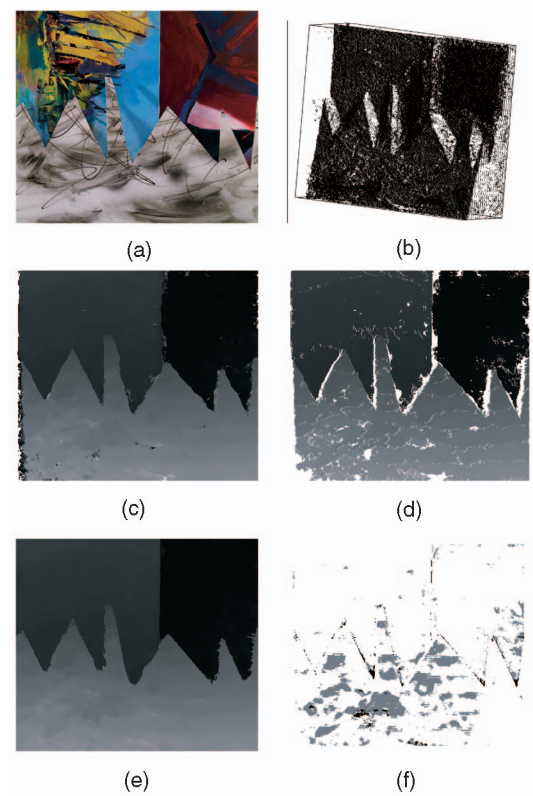


Fig. 2. Overview of the processing steps for the "Sawtooth" data set. The initial matches have been rotated so that the multiple candidates for each pixel are visible. Black pixels in the error map indicate errors greater than one disparity level, gray pixels correspond to errors between 0.5 and 1 disparity level, while white pixels are correct (or occluded and, thus, ignored). (a) Left image. (b) Initial matches in 3D. (c) Disparities after uniqueness enforcement. (d) Reliable matches. (e) Final disparities. (f) Error map.

These steps are presented in Sections 4 through 7. In the remainder of this section, we focus on important aspects of our research and how it compares to other work on stereo. A number of pixel matching techniques are reported in the literature [7], each having different strengths and weaknesses. For this reason, we propose combining them in order to maximize the number of correct candidate

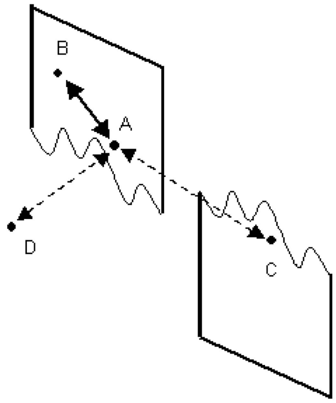


Fig. 3. Voting in 3D neighborhoods eliminates interference between adjacent pixels from different layers.

matches, which form the scene surfaces when they are reconstructed in disparity space. The combination of multiple matching techniques significantly enhances the performance of this algorithm over the work of [4].

However, the problem of stereo in its entirety, taking into account occlusions and discontinuities, cannot be fully solved at the pixel level. Support for each match has to be aggregated so that the confidence of correct matches is increased and outliers are made explicit. Aggregation in 1D neighborhoods is only motivated by computational simplicity and its shortcomings are well documented. While methods based on graph cuts and belief propagation that operate in 2D neighborhoods have achieved outstanding results, the choice of an appropriate energy function is not an easy task. Energy functions whose global minima can be computed with current optimization techniques do not necessarily model the phenomenon of stereovision in its most general form. In many cases, the disparity assignment that achieves the globally minimal energy is not necessarily associated with the lowest error rate [41]. This occurs because the energy function has to satisfy certain properties to be suitable for minimization. For instance, the penalization of disparity changes between neighboring pixels makes these approaches well suited for scenes that consist of fronto-parallel planes and prefers staircase looking solutions for slanted or curved surfaces. In this paper, following the approach of Lee et al. [4], we aggregate support in 3D neighborhoods via tensor voting. Fig. 3 shows that points A and B that are close in 3D and, therefore, are likely to belong in the same scene surface, interact strongly with each other. On the other hand, points A and C that are close in the image but not in 3D and, therefore, are most likely projections of unrelated surfaces, have very little effect on each other. Finally, point D, which is isolated in 3D and is probably generated by an error in the initial matching stage, receives no support as an inlier of a salient surface. After accumulating support by tensor voting, candidate matches that are consistent with their neighbors have high surface saliency, which validates them as correct matches.

Since the ordering constraint is violated by scene configurations that are not unlikely such as the presence of thin foreground objects, we do not enforce it. Its popularity in the literature is mostly as a requirement of specific optimization techniques. As optimization techniques have improved, most researchers have abandoned the ordering constraint. The uniqueness constraint, which states that, in the absence of transparency, there should be at most one match for each pixel, should also be enforced carefully. As Ogale and Aloimonos [42] point out, if scene surfaces exhibit horizontal slant (that is, if the epipolar line in the image is not parallel with the intersection of the epipolar plane and the scene surface), then M pixels in one image necessarily correspond to N pixels in the other image. Therefore, requiring a strict one-to-one correspondence for all pixels results in labeling $|M - N|$ pixels as occluded. These pixels that are interleaved with matched pixels, however, are perfectly visible in both images, just not at integer coordinate positions. Keeping this observation in mind, we only enforce uniqueness as a postprocessing step allowing at most one match for each pixel of the reference image in order to derive a dense disparity map. More than one pixel of the reference image is allowed to correspond to the same pixel of the target image (with integer or subpixel disparities) if the surface appears wider in the reference image. The same is true for the target image, if the surface has a wider projection there. The same "visibility constraint" that does not mark visible pixels as occluded due to surface slant is also used in [33].

A rather safe conclusion that can be drawn from the Middlebury Stereo Evaluation is that the use of monocular information, such as color, contributes to improvements in the performance of a stereo algorithm. In [5], we proposed a novel way of integrating monocular information that requires very few assumptions about the scene and does not fail when image segmentation fails. Candidate matches that were retained after tensor voting are grouped into smooth surfaces based on their positions and estimated surface normals. Then, these surfaces are reprojected to both images and points that are inconsistent with the other points of the surface in terms of color distribution in either image are rejected. This step removes erroneous matches for pixels in areas where one of the surfaces, usually the foreground, overextends and covers the other surface, even if it is binocularly visible [43]. The problem is more pronounced at occluded pixels that are likely to be assigned the disparity of the occluding surface. The wrong matches are removed since they do not project to the same surface in both images and, thus, the color distributions are inconsistent. Under this scheme, both images are treated symmetrically, unlike most segmentation-based methods, where only the reference image is segmented. Furthermore, we do not attempt to segment the image, but instead solve a simpler problem: grouping points, with surface orientation estimates, into smooth 3D surfaces.

The final step is the assignment of disparities to unmatched pixels. One can view the retained matches from the previous stage as the "reliable" matches of a progressive scheme since they are both consistent geometrically with their 3D neighbors and color-consistent with their neighboring pixels of the same surface in both images. Disparities are propagated from these pixels to unmatched ones, ensuring smoothness in terms of both geometry and color properties. We are also able to obtain disparity estimates for occluded pixels by enforcing smoothness with respect to surface orientation and color consistency with respect to the image in which they are visible. These estimates are accurate as long as there are no abrupt changes in the monocularly visible parts of each surface.

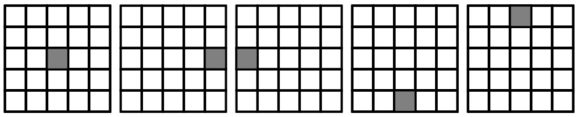


Fig. 4. The five shiftable windows applied for each disparity choice at every pixel. The shaded square corresponds to the pixel under consideration. The same window is applied to the target image.

4 Initial Matching

In this section, we propose a scheme for combining a variety of matching techniques, thus taking advantage of their combined strengths. For the results presented in Section 8, four matching techniques are used, but any type of matching operator can be integrated in the framework. These techniques are:

- A small (typically, 5×5) normalized cross correlation window, which is small enough to capture details and only assumes constant disparity over small windows of the image. This technique is referred to as the "correlation window" in the remainder of the paper.
- A *shiftable* normalized cross correlation window of the same size as the above. The fact that it is shiftable improves performance near discontinuities. It is referred to as the "shiftable window."
- A 25×25 normalized cross correlation window, which is applied only at pixels where the standard deviation of the three color channels is less than 20. The use of such a large window over the entire image would be catastrophic, but it is effective when applied only in virtually textureless regions, where smaller windows completely fail to detect correct matches. This technique is referred to as the "large window."
- A *symmetric interval* matching window (typically, 7×7) with truncated cost function as in [12]. This is referred to as the "interval window."

Note that the typical window sizes are for image resolutions similar to those of the Middlebury image pairs, which range from 284×216 to 450×375. Larger window sizes would most likely be necessary for higher resolution images.

4.1 Correlation Windows

This is one of the most common approaches for the establishment of pixel correspondences. It performs well over a wide range of scene types and imaging conditions. We choose it over alternatives such as the sum of absolute or squared differences because it is invariant to camera gain and, thus, more general. The correlation coefficients for all possible disparity values of each pixel are computed and all peaks of the correlation function are kept if their magnitude is comparable to the maximum for the pixel since they are good candidates for correct pixel correspondences. They are used as inputs to the tensor voting stage, where the decisions are made based on surface saliency and not the correlation coefficient itself since it can be affected by factors, such as repetitive patterns or the degree of texture of one surface over the other. These factors may cause wrong matches if decisions are made based solely on the correlation coefficients. See [44], [43] for an analysis of the effects of texture in correlation-based matching.

4.2 Shiftable Windows

We also use shiftable correlation windows [16] due to their superior performance near depth discontinuities. The limitation of window-based matching is that, no matter how small the window is, pixels from two or more surfaces are included in it at discontinuities. By not centering the window on the pixel under consideration, we can find a shift that includes as many pixels from the same surface as the pixel under consideration as possible. See Fig. 4 for the five windows used here. Given a pixel in the reference image, we compute cross correlation for each disparity level for five different window shifts around the pixel under consideration and keep the one with the maximum correlation coefficient as the score for that disparity level. As with correlation windows, we keep all significant peaks of the score function as candidate matches. Table 1 shows the performance of regular and shiftable correlation windows on the four initial Middlebury image pairs. The performance metric used is the number of correct matches, up to one disparity level off from the ground truth, over the total number of matching candidates. The same metric is reported for pixels at discontinuities based on the discontinuity maps provided by the authors of the Web page.

TABLE 1
Percentage of Good Matches Generated by Regular and Shiftable Correlation Windows over All Unoccluded Pixels and Discontinuities

Image Pair	Size	Regular		Shiftable	
		Total (%)	Disc. (%)	Total (%)	Disc. (%)
Tsukuba	5×5	62.1	57.2	62.0	62.1
	7×7	66.9	55.6	66.9	62.3
	9×9	70.3	50.0	69.4	60.8
Sawtooth	5×5	88.3	51.2	90.0	69.8
	7×7	92.4	63.1	94.4	68.8
	9×9	93.7	59.6	95.7	68.3
Venus	5×5	75.7	61.7	75.4	64.7
	7×7	81.9	58.2	81.9	62.0
	9×9	85.9	54.9	85.7	60.3
Map	5×5	96.9	68.6	98.0	70.9
	7×7	98.4	67.7	98.9	70.2
	9×9	98.2	64.4	98.8	68.7

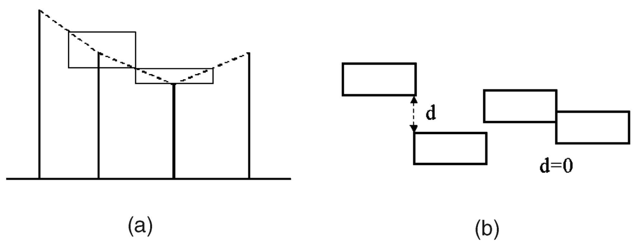


Fig. 5. Symmetric interval matching. Both images are interpolated and color distance is computed between the left and right interval (and not an interval and a pixel). Vertical lines in (a) mark the original pixel positions on the scanline. (a) Interval representation. (b) Distance between intervals.

Even though shiftable windows perform better, correlation windows also add valuable candidate matches and reinforce the ones on which both operators agree. Our experiments show that using both types improves performance.

4.3 Large Windows

This is a different correlation-based matching technique that aims at producing correct disparity estimates for untextured areas. It is only applied to parts of the image where color variance is below a certain threshold and, thus, usually not near discontinuities, where color variance is high. For the experiments presented here, a 25×25 window was applied to pixels where the standard deviation of the three color channels, within the 25×25 window, was less than 20 intensity levels. A final step is required for the rejection of unreliable matches. This is especially important here since we are specifically targeting the most ambiguous pixels of the image. We perform a simple test to determine the reliability of each match. The correlation coefficients for each disparity of a given pixel are divided by the sum of all correlation coefficients for that pixel to give the "normalized score" of each disparity. This allows us to detect matching candidates with high uncertainty. For instance, if the disparity range is 20 and all correlation coefficients are equal, the normalized score for all disparity levels would be 5 percent. If the normalized score of a matching candidate is not significantly larger than 5 percent, the matching candidate is unreliable. For the experiments in the remainder of the paper, we reject matching candidates with normalized scores below 20 percent of the average normalized score over all matching candidates. What should be noted is that multiple matches for each pixel are still allowed and often occur.

4.4 Interval Windows

The final matching technique is very different from the above, not only because we use the matching cost of [11], but mostly because of the truncation of the cost for each pixel at a certain level. That makes the behavior robust against pixels from different surfaces that have been included in the window. Our implementation is that of Szeliski and Scharstein [12]. Both images are linearly interpolated along the x axis so that samples are created at half-pixel positions. The intensity of each pixel in each of the three color channels is now represented as the interval between the minimum and maximum value of the intensity at the integer pixel position and the half-pixel positions before and after it on the scanline, as shown in Fig. 5.

Numerically, the cost for matching pixel (x_L, y) in the left image with pixel (x_R, y) in the right image is the minimum distance between the two intervals, which is given by the following equation and is zero if they overlap:

$$C(x_L, x_R, y) = \sum_{c \in \{R,G,B\}} min\Big\{dist(I_{Lc}(x_i, y), I_{Rc}(x_j, y)), \\ c_{trunc} : x_i \in \Big[x_L - \frac{1}{2} \quad x_L + \frac{1}{2}\Big], x_j \in \Big[x_R - \frac{1}{2} \quad x_R + \frac{1}{2}\Big]\Big\}. \tag{1}$$

The summation is over the three color channels and $dist()$ is the Euclidean distance between the value of a color channel I_{Lc} in the left image and I_{Rc} in the right image. If the distance for any channel exceeds the truncation parameter c_{trunc}, the total cost is set to $3c_{trunc}$. Typical values for c_{trunc} are between 3 and 10. For the experiments presented in Section 8, a value of 5 was used. Even though, statistically, the performance of interval windows is slightly worse than that of the shiftable windows, both overall and at discontinuities, and worse overall than the correlation windows, they are useful because they produce correct disparity estimates for pixels where the other windows fail due to the different nature of the dissimilarity measure and the robust formulation we use.

Each matching technique is repeated using the right image as reference and the left as target. This increases the true positive rate especially near discontinuities, where the presence of occluded pixels in the reference window affects the results of matching. When the other image is used as reference, these pixels do not appear in the reference window. A simple parabolic fit [7] is used for subpixel accuracy, which makes slanted or curved surfaces appear continuous and not staircase-like. We have found the parabolic fit to work well, in practice, even if it is only justified for quadratic cost functions. Computational complexity is not affected since the number of matching hypotheses is unchanged and it is independent of the number of permissible disparity levels. Besides the increased number of correct detections, the combination of these matching techniques offers the advantage that the failures of a particular technique are not detrimental to the success of the algorithm, as long as the majority of the operators do not produce the same erroneous disparities. Our experiments have also shown that the errors produced by small windows, such as the 5×5 and 7×7 used here, are randomly spread in space and do not usually align to form nonexistent structures. This property is important for our methodology that is based on the perceptual organization, due to good alignment, of candidate matches in space. Note that we avoid applying the large window, which is more susceptible to systematic errors, near discontinuities and, thus, it does not cause any problems there.

5 Selection of Correct Matches

This section describes how correct matches can be selected among the candidates of the previous stage by examining how they can be grouped with their neighboring candidate matches to form smooth 3D surfaces. This is accomplished by tensor voting, which also allows us to infer the orientation of these surfaces.

5.1 Overview of Tensor Voting

The use of a voting process for structure inference from sparse and noisy data was presented in [1]. The methodology is noniterative and robust to considerable amounts of outlier

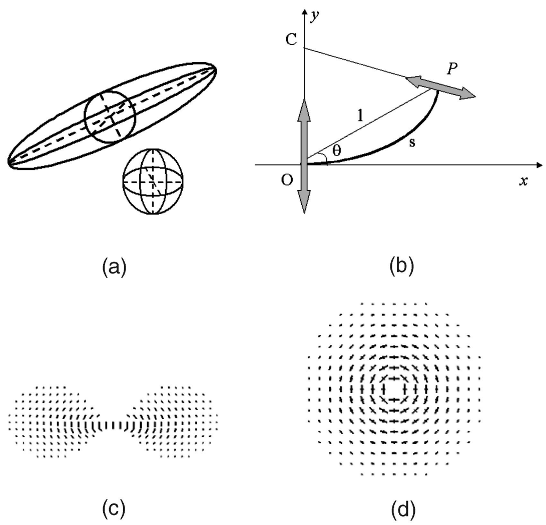


Fig. 6. Tensor voting. (a) The shape of the tensor indicates if there is a preferred orientation, while its size the confidence of this information. The top tensor has a strong preference of orientation and is more salient than the bottom tensor, which is smaller and unoriented. (b) Vote generation for a stick voter as a function of the distance and curvature of the arc and the orientation of the voter. (c) and (d) cuts of the stick and ball voting fields. Shown is the normal orientation of the propagated vote. The size of the vote is proportional to its magnitude.

noise. It has one free parameter: the scale of voting σ, which essentially defines the size of the neighborhood of each point. The input data is encoded as second-order symmetric tensors, and constraints, such as proximity, colinearity, and cocurvilinearity are propagated by voting within neighborhoods. The tensors allow the representation of points on smooth surfaces, surface intersections, curves, and junctions, without having to keep each type in separate spaces. In 3D, a second-order tensor has the form of an ellipsoid or, equivalently, of a 3×3 matrix. Its shape encodes the type of feature that it represents, while its size the *saliency* or the confidence we have in this information (Fig. 6a). The same information in algebraic form is contained in the eigenvalues and eigenvectors of the tensor.

During the voting process, each input site casts votes to its neighboring input sites that contain data represented with tensors. The votes are also second-order symmetric tensors. Their shape corresponds to the orientation the receiver would have if the voter and receiver were in the same structure. We first describe the case of a *stick tensor*, which encodes a surface orientation with perfect certainty. A stick tensor has one nonzero eigenvalue which is associated with an eigenvector that is *normal* to the surface. The saliency (strength) of the vote decays with respect to the length of the smooth circular path connecting the voter and receiver, according to the following equation (see, also, Fig. 6b):

$$\mathbf{S}_{SO}(l,\theta,\sigma) = e^{-\left(\frac{s^2+c\kappa^2}{\sigma^2}\right)} \begin{bmatrix} -sin(2\theta) \\ cos(2\theta) \end{bmatrix} [-sin(2\theta)\ cos(2\theta)],$$
$$s = \frac{\theta l}{sin(\theta)}, \qquad (2)$$
$$\kappa = \frac{2sin(\theta)}{l},$$

where s is the length of the arc between the voter and receiver and κ is its curvature, σ is the scale of voting, and c is a constant. The votes cast by stick tensors are also stick tensors. The votes cast by unoriented voters, which are represented by *ball tensors* and are equivalent to identity matrices, can be derived from the above equation, but this is beyond the scope of this paper. We refer interested readers to [1], [45] for more details. They only attenuate with distance since nothing suggests nonstraight continuation and encode the normal orientation of a line or, equivalently, a pencil of planes, passing through the voter and receiver. The votes also include an uncertainty component. The accumulation at each point of votes from numerous voters on the same surface results in a strong preference for that surface. 2D cuts of the stick and the ball voting field can be seen in Figs. 6c and 6d, which show the normal orientations propagated by the voter. In case we have no orientation information about the inputs, they are encoded as unit unoriented tensors.

Vote accumulation is performed by tensor addition, which is equivalent to the addition of 3×3 matrices. After voting is completed, the eigensystem of each tensor is computed and the tensor is decomposed as in:

$$\begin{aligned} T &= \lambda_1 \hat{e}_1 \hat{e}_1^T + \lambda_2 \hat{e}_2 \hat{e}_2^T + \lambda_3 \hat{e}_3 \hat{e}_3^T \\ &= (\lambda_1 - \lambda_2)\hat{e}_1 \hat{e}_1^T + (\lambda_2 - \lambda_3)\big(\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T\big) \\ &\quad + \lambda_3\big(\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T + \hat{e}_3 \hat{e}_3^T\big), \end{aligned} \qquad (3)$$

where λ_i are the eigenvalues in decreasing order and $\hat{e}_i$ are the corresponding eigenvectors. The three components in which we analyze the tensor are: the stick component $\hat{e}_1 \hat{e}_1^T$, which is large for points in surfaces, the plate component $\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T$, which is large for points in curves or surface intersections, and the ball component $\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T + \hat{e}_3 \hat{e}_3^T$, which is large for points that have no preference of orientation, such as junctions. We define surface saliency as the difference between the two largest eigenvalues, curve saliency as the difference between the second and third eigenvalue, and junction saliency as the smallest eigenvalue. If surface saliency is maximum, the point most likely belongs to a surface and $\hat{e}_1$ is the surface normal. Outliers receive little and inconsistent support from their neighborhood and are identified by their low saliency and the lack of a dominant orientation. In the case of stereo, we assume that that all inliers lie on surfaces that reflect light toward the cameras and, therefore, we do not consider curves and junctions. In practice, the resulting tensors, at least at the locations of the correct matches, exhibit surface saliency values much higher than the other types.

5.2 Selection of Matches as Surface Inliers

This section describes how correct matches are inferred from the point cloud of candidate matches generated by the initial matching stage by examining how much support they receive from their neighboring candidate matches after tensor voting. The goal here is to address stereo as a perceptual organization problem, based on the premise that the correct matches should form coherent surfaces in disparity space. This is the only part of our approach that is based on [4]. The input is a cloud of points in a 3D space $(x, y, z_{scale} \times d)$, where z_{scale} is a constant used to make the input less flat with respect to the d axis since disparity has a narrower dynamic range than the spatial domain. The typical value of z_{scale} is 8 and the sensitivity to it is

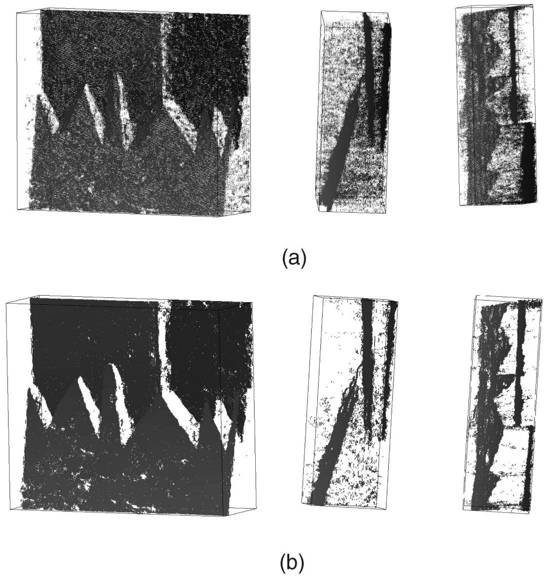


Fig. 7. Rotated views (front, side, and top) of the set of candidate matches for the "Sawtooth" image pair, before and after tensor voting and uniqueness enforcement. Gray levels encode different disparity values. (a) Initial matches. (b) Matches after tensor voting and uniqueness.

extremely low for a reasonable range such as 4 to 20. The quantitative matching scores are disregarded and all candidate matches are initialized as unoriented tensors with all eigenvalues equal to one. If two or more matches fall within the same $(x, y, z_{scale} \times d)$ voxel, their initial saliencies are added, thus increasing the confidence of candidate matches confirmed by multiple matching techniques. Since d is estimated with subpixel accuracy, each integer disparity level has z_{scale} possible subpixel levels.

The inputs are encoded as unoriented, ball tensors and cast votes to their neighbors. What should be pointed out here is the fact that, since information propagation is performed in 3D, there is very little interference between candidate matches for pixels that are adjacent in the image but come from different surfaces, as shown in Fig. 3. This is a major advantage over information propagation between adjacent pixels, even when it is mitigated by some dissimilarity measure. Rotated views of the input cloud of points can be seen in Fig. 7a.

When voting is completed, the surface saliency of each candidate match can be computed as the difference between the two largest eigenvalues of the tensor. Uniqueness is enforced with respect to the left image by retaining the candidate match with the highest surface saliency for each pixel. The same can be done for all pixels of the right image to obtain the right disparity map, if desired. We do not enforce uniqueness in the strict one-to-one sense since it is violated by slanted surfaces which project to a different number of pixels on each image. The definition of uniqueness we use is that each pixel can have at most one disparity value or be occluded [42], [33]. This allows more than one, but, typically, not more than two, pixels in one image to match to the same pixel in the other image, at subpixel disparities. Surface saliency is a more reliable criterion for the selection of correct matches than the score of a local matching operator because it requires that candidate matches, identified as such by local operators, should also form coherent surfaces in 3D. This scheme is capable of rejecting false positive responses of the local operators, which is not possible at the local level. The resulting data sets still contain errors, mostly near discontinuities and the borders of the image, which are corrected at the next stage. Rotated views of the matching candidates after this stage can be seen in Fig. 7 b. Note that we have eliminated a free parameter from the algorithm of [5] by not thresholding with respect to surface saliency, but, instead, feeding all matching candidates after uniqueness enforcement to the next stage.

6 Surface Grouping and Refinement

Candidate matches that have not been rejected are grouped in layers using a simple growing scheme. By layers, we mean surfaces with smooth variation of surface normal. They do not have to be planar and the points that belong to them do not have to form one connected component.

Labeling starts from seed matches that have maximum surface saliency. Since the input to this stage includes candidate matches for almost all pixels, we only examine the eight nearest neighbors of the seed in the reference image. If they are smooth continuations of the growing surface, they are added to it and their neighbors are also considered for addition. We adhere to the disparity gradient limit constraint, which dictates that the maximum disparity jump between two pixels of the same surface is one and, thus, stop growing the surfaces when we encounter disparity jumps over one disparity level. When no more matching candidates can be added to the surface, the unlabeled point with maximum surface saliency is selected as the next seed. Small surfaces comprised of less than 0.5 percent of the image pixels are removed since they are probably noisy patches, unless they are compatible with a larger nearby surface. If a small surface patch is less than 10 percent of the maximum image dimension away from a large surface with compatible position and orientation, it is not removed from the data set. Support from a larger surface means that the small part is most likely correct, but, due to occlusion or failure of the matching operators, is not connected to the main part of the surface. After this step, the data set consists of a set of labeled surfaces which contain errors mostly due to foreground overextension. A number of candidate matches that survived uniqueness enforcement while not being parts of large salient surfaces are removed here. These are typically the "best" candidates for pixels for which the correct match has not been found by any of the matching operators and, thus, all matching candidates are wrong. This occurs more often for untextured pixels.

The next step is the refinement of the layers. The goal is to remove the overextensions of the foreground by ensuring that the color properties of the pixels, which are the projections of the grouped points, are *locally* consistent within each layer. The color consistency of a pixel is verified by computing the ratio of pixels of the same layer with similar color to the current pixel over the total number of pixels of the layer within the neighborhood. This is repeated in the target image and, if the current label assignment does not correspond to the maximum ratio *in both images*, then the pixel is removed from the layer. The color similarity ratio for pixel (x_0, y_0) in the left image with layer i can be computed according to the following equation:

TABLE 2
Total Matches and Error Rate for Each Image Pair before and after Surface Grouping and Refinement

Image Pair	Total before	Error rate before	Total after	Error rate after
Tsukuba	84810	5.31%	69666	1.33%
Sawtooth	144808	2.95%	136894	1.08%
Venus	147320	6.16%	132480	1.24%
Map	48657	0.44%	45985	0.05%
Cones	132856	4.27%	126599	3.41%
Teddy	135862	7.24%	121951	4.97%

$$R_i(x_0, y_0) = \frac{\sum_{(x,y)\in N} T(l(x,y) = i \wedge dist(I_L(x,y), I_L(x_0,y_0) < c_{thr}))}{\sum_{(x,y)\in N} T(lab(x,y) = i)}, \quad (4)$$

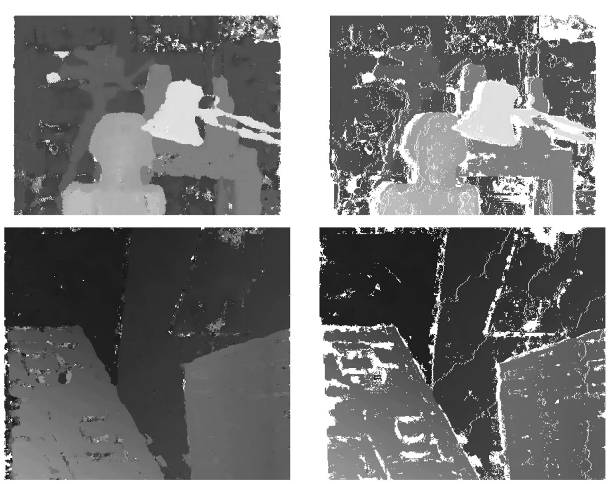
where $T()$ is a test function that is 1 if its argument is true, $l()$ is the label of a pixel, and c_{thr} is a color distance threshold in RGB space, typically equal to the c_{trunc} parameter of the interval windows. If both these conditions are true, pixel (x, y) is counted as consistent in color with pixel (x_0, y_0) for label i. The same is applied in the right image for pixel $(x_0 - d_0, y_0)$. The size of the neighborhood is the second and final parameter of this stage. It can be set equal to the range of the voting field during tensor voting.

This step corrects surface overextensions that occur near occlusions since the overextensions are usually not color-consistent in both images and are thus detected and removed. Table 2 shows the total number of candidate matches and the error rates before and after refinement for the four Middlebury image pairs. The disparity maps for the "Sawtooth" example before and after grouping and refinement can be seen in Figs. 2c and 2d. The same for the "Tsukuba" and "Venus" can be seen in Fig. 8.

7 Disparity Estimation for Unmatched Pixels

The goal of this stage is to generate candidate matches for the remaining unmatched pixels. Given the already estimated disparities and labels for a large set of the pixels, there is more information available now that can enhance our ability to estimate the missing disparities. We opt for a progressive approach under which only the most reliable correspondences are allowed in the beginning. These are correspondences that satisfy strict geometric and color requirements in both images. The requirements become less strict as we proceed.

Given an unmatched pixel in the reference image, we examine its neighborhood for layers to which the pixel can be assigned. Color similarity ratios are computed for the pixel with respect to these layers as in (4). The layer with the maximum ratio is selected as the potential layer for the pixel. Then, we need to generate a range of disparities for the pixel. This is done by examining the disparity values of the selected layer's pixels in the neighborhood. The range is extended according to the disparity gradient limit constraint, which holds perfectly in the case of rectified parallel stereo pairs. Disparity hypotheses (d_h) are verified one by one in the target image by computing similarity ratios for all potential corresponding pixels $(x - d_h, y)$ and rejecting those that are not consistent with the selected layer. This is not done if the disparity hypothesis indicates that the new match is occluded by existing reliable matches, in which case, we allow occluded surfaces to grow underneath the occluding ones. On the other hand, we do not allow new matches to occlude existing consistent matches. Votes are collected at valid potential matches in disparity space, as before, with the only difference being that only matches from the appropriate layer cast votes (see Fig. 9). The most salient among the potential



(a) (b)

Fig. 8. Disparity maps after uniqueness (left column) and after surface grouping and refinement for the "Tsukuba" and "Venus" image pairs from the Middlebury Stereo Evaluation. (a) Disparities after uniqueness. (b) Results of surface refinement.

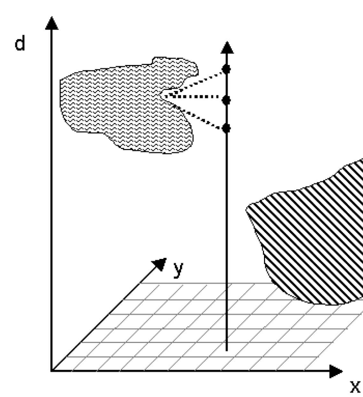


Fig. 9. Candidate generation for unmatched pixels based on segmented layers. The unmatched pixel is compatible with the left surface only, thus votes are collected at disparity hypotheses generated by matches of the left surface. Also note that only matches from the appropriate layer vote at each candidate.

TABLE 3
Error Rates for the Original Middlebury Image Pairs

Image pair	Unoccluded (%)	Rank	Discont. (%)	Rank	Textureless (%)	Rank
Tsukuba	1.51	11	7.96	12	2.02	24
Sawtooth	0.70	12	4.35	11	0.50	26
Venus	1.09	12	13.95	26	1.39	16
Map	1.31	24	11.47	26	-	-

TABLE 4
Quantitative Evaluation for the New Middlebury Image Pairs (Acceptable Error at 1.0 Disparity Level)

Image pair	Unoccluded (%)	Rank	All (%)	Rank	Discont. (%)	Rank
Tsukuba	3.79	9	4.79	9	8.86	6
Venus	1.23	4	1.88	5	11.5	9
Teddy	9.76	5	17.0	5	24.0	8
Cones	4.38	3	11.4	4	12.2	5

matches is selected and added to the layer since it is the one that ensures the smoothest surface continuation.

For the results presented here, we applied the following progressive growing scheme, which has two parameters: c_{thr}, which is the color threshold used for computing the similarity ratios, and σ_3, the scale of voting, which also defines the size of the neighborhood in which similarity ratios are computed. For the first iteration, we initialize the parameters with $c_{thr} = 1$ and $\sigma_3^2 = 20$. These are very strict requirements and have to be satisfied on both images for a disparity hypothesis to be valid. Votes are accumulated on valid hypotheses which also do not occlude any existing matches and the most salient continuation is selected. We then repeat the process without requiring consistency with the target image and add more matches, which usually are for occluded pixels that are very similar to their unoccluded neighbors. The added matches are generally correct, but valid hypotheses cannot be generated for all pixels. In the second iteration, we increment both c_{thr} and σ_3^2 by their initial values and repeat the same process. The choice of parameters here is not critical. For instance, maintaining a constant σ_3 produces very similar results. For the experiments shown here, both parameters are increased by constant increments at each iteration until convergence.

Typically, there are a few pixels that cannot be resolved because they exhibit low similarity to all layers or because they are specular or in shadows. Candidates for these pixels are generated based on the disparities of all neighboring pixels and votes are collected at the candidate locations in disparity space. Again, the most salient ones are selected. We opt to use surface smoothness at this stage instead of image correlation or other image-based criteria since we are dealing with pixels where the initial matching and color consistency failed to produce a consistent match.

8 Experimental Results

This section contains results on the color versions of the four image pairs of [7] and the two proposed in [46], which are available online at http://cat.middlebury.edu/stereo/. *All six examples were processed with identical parameters*. The initial matching in all cases was done using the four matching operators presented in Section 4 using both the left and right image as reference. The correlation and shiftable windows were 5×5, the interval windows were 7×7 with the truncation parameter set at 5, and the large window was 25×25, applied at pixels with intensity variance less than 20. For the large windows only, pixels with normalized score below 20 percent of the average were rejected. The scale of the voting field for the detection of correct matches was $\sigma^2 = 50$, which corresponds to a voting radius of 14 or a neighborhood of $29 \times 29 \times 29$. Refinement was performed with a voting radius of 18 and c_{thr} equal to 5. In the final stage, c_{thr} was initialized as 1 and incremented by 1 for 25 iterations, while σ_3^2 was initialized as 20 and incremented by 20 at each iteration.

A second surface refinement operation was performed to remove errors around the surface discontinuities. This time, the voting radius was significantly smaller, set equal to 7, since we are only interested in correcting the borders of each surface. The value of c_{thr}, on the other hand, was equal to 40, to allow larger color variation within each surface. The parameters for the final stage were identical with those of the previous paragraph.

The error metric reported in the tables is the one proposed in [7], where matches are considered erroneous if they correspond to unoccluded image pixels and their disparity error is greater than one integer disparity level. Table 3 contains the error rates we achieved, as well as the rank our algorithm would achieve among the 38 algorithms in the evaluation. The error rates reflect the number of errors larger than one disparity level for all unoccluded pixels, for pixels near discontinuities, and for textureless pixels. We have rounded the disparities to integer values for this evaluation. We refer readers to the Middlebury Stereo Evaluation Web page for results obtained by other methods. Based on the results for all unoccluded pixels, our algorithm would rank 15th in the evaluation as of 5 July 2005. As with all methods that take color explicitly into account, performance on the "Map" is not as good as that achieved by methods that do not use monocular information due to the random textures in the image.

Tables 4 and 5 report our results for the new version of the Middlebury Stereo Evaluation that includes "Tsukuba," "Venus," and the two image pairs introduced in [46]. The new image pairs contain curved and slanted surfaces, with

TABLE 5
Quantitative Evaluation for the New Middlebury Image Pairs (Acceptable Error at 0.5 Disparity Level)

Image pair	Unoccluded (%)	Rank	All (%)	Rank	Discont. (%)	Rank
Tsukuba	25.5	11	26.2	11	21.2	8
Venus	3.32	1	4.12	1	14.6	2
Teddy	14.6	3	21.8	4	33.3	4
Cones	7.05	2	14.5	3	17.4	3

different degrees of detail and texture, and are, thus, more challenging. This is more pronounced for methods that make the assumption that scene surfaces are planar and parallel to the image plane. This assumption is explicitly made when one penalizes disparity differences between neighboring pixels. This demonstrates the capability of the algorithms to estimate precise subpixel disparities. We have not rounded the disparities in this case. For the new evaluation, the error rate over all pixels, including the occluded ones, has replaced the evaluation over textureless pixels. The ranks are among the 12 algorithms included in the evaluation, as of 5 July 2005. Considering performance at unoccluded pixels, our results are tied at the fourth place when the acceptable error is one and rank third when it is 0.5.

Figs. 10 and 11 show the final disparity map and the error map for the "Venus," "Tsukuba," "Map," "Cones," and "Teddy" image pairs. The results for "Sawtooth" appear in Fig. 2. White in the error maps indicates an error less than one half of a disparity level or occluded pixel, gray indicates an error between one half and one disparity level (acceptable), and black indicates large errors above one disparity level.

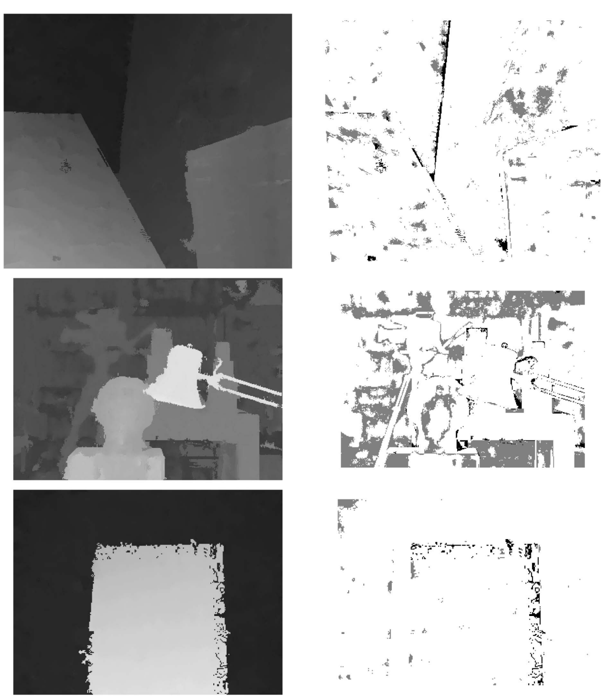


Fig. 10. Final disparity maps and error maps for the "Venus," "Tsukuba," and "Map" image pairs from the Middlebury Stereo Evaluation (see Fig. 1 for the input images).

8.1 Results on Aerial Images

The final results are on less controlled images, taken from an airplane under sunlight. The left and right images are shown in Figs. 12a and 12b. They are grayscale, with different camera gains, and contain large shadows. The same matching techniques, with the exception of the interval windows, which are not very effective for grayscale images, are used to produce the initial matches and voting is performed to infer surface saliencies. The results after tensor voting and uniqueness enforcement can be seen in Fig. 12c and the final disparity map in Fig. 12d. The outlines of the buildings have been superimposed manually to show the accuracy of the reconstruction. This experiment demonstrates the effectiveness of the proposed nonparametric color representation for images that are very different than the ones of the Middlebury Stereo Evaluation. The images here are grayscale and they lack the vivid and distinctive colors of the previous examples. The surface refinement and final disparity estimation stages of our approach, however, still perform well.

9 Computational Complexity

In this section, we provide an analysis of the computational complexity of each step as a function of the number of pixels N, the number of possible disparities D, and the number of pixels included in the matching windows W. Processing times for unoptimized C++ code refer to a Pentium 4 processor at 2.8MHZ.

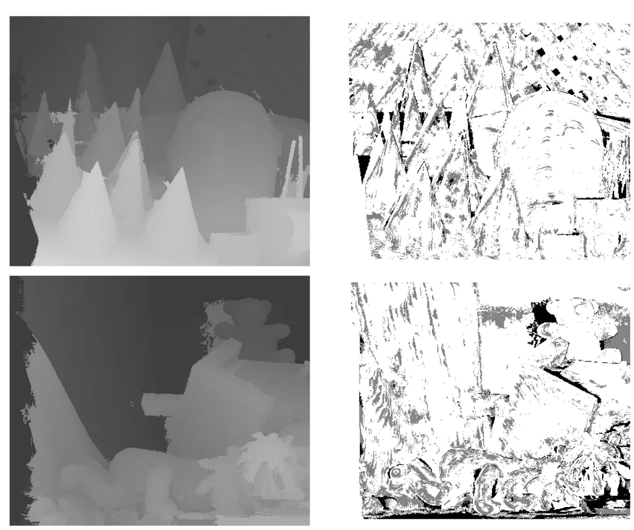


Fig. 11. Final disparity maps and error maps for the "Cones" and "Teddy" image pairs from the Middlebury Stereo Evaluation (see Fig. 1 for the input images).

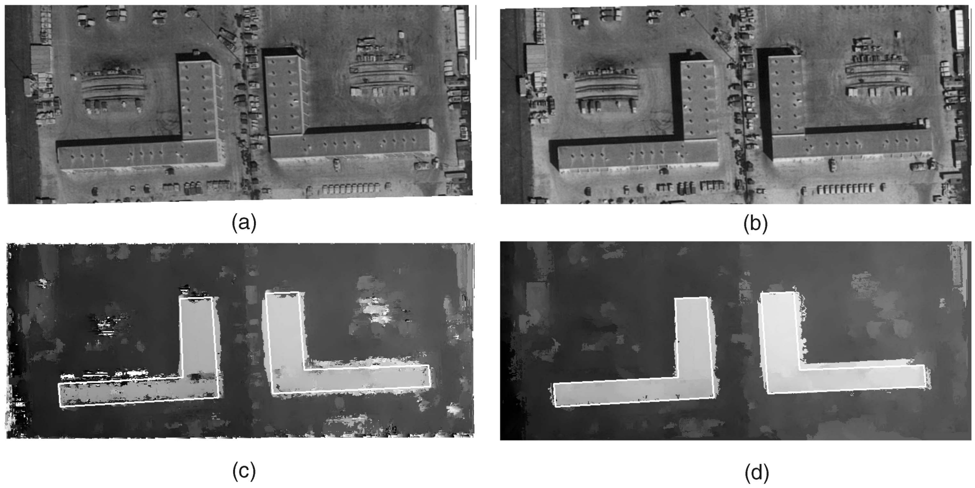


Fig. 12. Input aerial images, results of surface grouping and refinement, and final disparity map. The outlines of the buildings have been superimposed manually. (a) Left image. (b) Right image. (c) Refined surfaces. (d) Final disparity map.

In the initial matching stage, the complexity is linear with respect to the number of pixels, possible disparities, and the window size since the score or cost is computed over all pixels of the window for each disparity of each pixel of the reference image. Complexity is $O(NDW)$. Keep in mind, however, that more efficient implementations to avoid repeating computations are possible. For instance, Veksler [10] proposes a method for computing matching costs that is independent of the window size by using the integral image. The execution time for the 5×5 cross-correlation window on the Tsukuba image, which is 384×288 with 20 possible disparity levels, is 4 seconds. Execution times for other techniques are similar and scale linearly with the three parameters.

The complexity of the tensor voting stage is $O(NlogN)$. It is independent of the number of possible disparities since there is at most a small fixed number of candidate matches for each pixel. The operations that need to be performed is an initial sorting of the data and searches for the neighbors of each candidate match in 3D. Tensor voting for 517,819 matching candidates for the Tsukuba image pair takes 2 minutes and 30 seconds.

Uniqueness enforcement is performed at the pixel level and is virtually instantaneous. Surface grouping is linear in the number of pixels since it is performed as a single pass over the matching candidates, which are one or none for each pixel, and only the 8-neighbors of the corresponding pixels are examined. Subsequent operations, such as the rejection of groups with too few members, are linear in the number of groups and, thus, negligible. During surface refinement, pixels within the neighborhood of the reprojections of the grouped matching candidates on both images are examined. The process takes 6 seconds for the Tsukuba image pair with a neighborhood radius of 18 pixels. All these steps are $O(N)$ and take very few seconds, dominated by the surface refinement step.

Finally, disparity estimation for unmatched pixels is linear in the number of unmatched pixels, which, typically, are a small subset of all pixels, as well as the number of allowable disparities for each of them, as indicated by their neighbors that belong to the most similar surface. Complexity for the worst case is $O(NDlogN)$. In general, it is a function of the number of unmatched pixels. The processing time for 24,551 unmatched pixels, which is the maximum percentage of unmatched pixels among the six image pairs, of the Tsukuba image pair is 5 minutes and 22 seconds.

10 Discussion

We have presented a novel stereo algorithm that addresses the limitations of binocular matching by incorporating monocular information. We use tensor voting to infer surface saliency and use it as a criterion for deciding on the correctness of matches as in [3] and [4]. We are, however, able to significantly improve the performance of the algorithm mainly for two reasons: the initial matching stage that provides better inputs to the following stages and the combination of geometric and photometric cues in all phases of processing. Textured pixels away from depth discontinuities can be easily resolved by even naive stereo algorithms. As stated in the introduction, we aim at reducing the errors at untextured parts of the image and near depth discontinuities. Under our approach, the typical phenomenon of the overextension of foreground surfaces over occluded pixels is mitigated by removing from the data set candidate matches that are not consistent in color with their neighboring pixels in both images. On the other hand, surface smoothness is the main factor that guides the matching of uniform pixels.

Arguably, the most significant contribution is the segmentation into layers based on geometric properties and not appearance. We claim that this is advantageous over other methods that use color-based segmentation since it utilizes the already computed disparities which are powerful cues for grouping. In fact, grouping candidate matches in 3D based on good continuation is a considerably easier problem than image segmentation. This scheme allows us to treat both images symmetrically and provides estimates for the color distribution of the layers, even if it varies significantly throughout each layer. The choice of a local, nonparametric color representation allows us to handle surfaces with texture or heterogeneous and varying color distributions, such as the ones in the "Venus" images, in which image segmentation may be hard. Surface grouping serves as the basis of the refinement stage that eliminates surface overextensions.

A second significant contribution is the initial matching stage that allows the integration of any matching technique without any modification to subsequent modules. The use of a large number of matching operators, applied to both images, can be viewed as another form of consensus. While all operators fail for certain pixels, the same failures are usually not repeated, with the same disparity values, by other operators. Our experiments show that the results of combining the four techniques we used over all the image pairs are superior to those generated by using a smaller set of them.

We employ a least commitment strategy and avoid the use of constraints that are violated by usual scene configurations. One such constraint is the requirement that adjacent pixels should have the same disparity to avoid incurring some penalty. While this constraint aids the optimization process of many approaches, it becomes an approximation for scenes that do not consist of fronto-parallel surfaces. Processing in 3D via tensor voting enforces the more general constraint of good continuation and eliminates interference between adjacent pixels from different world surfaces without having to assess penalties on them. In our work, the assumption that scene surfaces are fronto-parallel is only made in the initial matching stage, when all pixels in a small window are assumed to have the same disparity. After this point, the surfaces are never assumed to be anything other than continuous. We also do not use the ordering constraint, which was introduced to facilitate dynamic programming. The uniqueness constraint is applied as described in Section 3 to allow one-to-many correspondences. Thus, no unnecessary difficulties are introduced for slanted or curved surfaces.

Our algorithm fails when surfaces are entirely missed at the initial matching stage or when they are entirely removed at the surface refinement stage. We are not able to grow surfaces that are not included in the data before the final stage. On the other hand, we are able to smoothly extend partially visible surfaces to infer the disparities of occluded pixels, assuming that occluded surfaces do not abruptly change orientation. A limitation of our work is that one cannot predict the usefulness of intermediate results based on the error rate. "Cleaner" data sets after layer refinement may not contain enough information to guide correct disparity estimation for the unmatched pixels. In our future work, we intend to derive a set of criteria that adapt the refinement stage according to both surface orientation and color distribution of the layers.

ACKNOWLEDGMENTS

This research has been supported in part by US National Science Foundation grant IIS 03 29247. The authors are grateful to Ammar Chinoy and Lily Cheng for their contributions in software development and experimental evaluation.

REFERENCES

[1] G. Medioni, M.S. Lee, and C.K. Tang, *A Computational Framework for Segmentation and Grouping.* Elsevier, 2000.

[2] R. Sara, "Finding the Largest Unambiguous Component of Stereo Matching," *Proc. European Conf. Computer Vision,* vol. 3, pp. 900-914, 2002.

[3] M.S. Lee and G. Medioni, "Inferring Segmented Surface Description from Stereo Data," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* pp. 346-352, 1998.

[4] M.S. Lee, G. Medioni, and P. Mordohai, "Inference of Segmented Overlapping Surfaces from Binocular Stereo," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 6, pp. 824-837, June 2002.

[5] P. Mordohai and G. Medioni, "Stereo Using Monocular Cues within the Tensor Voting Framework," *Proc. European Conf. Computer Vision,* pp. 588-601, 2004.

[6] W. Hoff and N. Ahuja, "Surfaces from Stereo: Integrating Feature Matching, Disparity Estimation, and Contour Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 2, pp. 121-136, Feb. 1989.

[7] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *Int'l J. Computer Vision,* vol. 47, nos. 1-3, pp. 7-42, Apr. 2002.

[8] M.Z. Brown, D. Burschka, and G.D. Hager, "Advances in Computational Stereo," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 8, pp. 993-1008, Aug. 2003.

[9] T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 9, pp. 920-932, Sept. 1994.

[10] O. Veksler, "Fast Variable Window for Stereo Correspondence Using Integral Images," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 556-561, 2003.

[11] S. Birchfield and C. Tomasi, "A Pixel Dissimilarity Measure that Is Insensitive to Image Sampling," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 4, pp. 401-406, Apr. 1998.

[12] R. Szeliski and D. Scharstein, "Symmetric Subpixel Stereo Matching," *Proc. European Conf. Computer Vision,* vol. 2, pp. 525-540, 2002.

[13] M. Agrawal and L.S. Davis, "Window-Based, Discontinuity Preserving Stereo," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 66-73, 2004.

[14] Y. Ohta and T. Kanade, "Stereo by Intra- and Interscanline Search Using Dynamic Programming," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 7, no. 2, pp. 139-154, Mar. 1985.

[15] P.N. Belhumeur and D. Mumford, "A Bayesian Treatment of the Stereo Correspondence Problem Using Half-Occluded Regions," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* pp. 506-512, 1992.

[16] S.S. Intille and A.F. Bobick, "Disparity-Space Images and Large Occlusion Stereo," *Proc. European Conf. Computer Vision,* vol. 2, pp. 179-186, 1994.

[17] D. Geiger, B. Ladendorf, and A. Yuille, "Occlusions and Binocular Stereo," *Int'l J. Computer Vision,* vol. 14, no. 3, pp. 211-226, Apr. 1995.

[18] P.N. Belhumeur, "A Bayesian-Approach to Binocular Stereopsis," *Int'l J. Computer Vision,* vol. 19, no. 3, pp. 237-260, Aug. 1996.

[19] S. Birchfield and C. Tomasi, "Depth Discontinuities by Pixel-to-Pixel Stereo," *Proc. Int'l Conf. Computer Vision,* pp. 1073-1080, 1998.

[20] A.F. Bobick and S.S. Intille, "Large Occlusion Stereo," *Int'l J. Computer Vision,* vol. 33, no. 3, pp. 1-20, Sept. 1999.

[21] I.J. Cox, S.L. Hingorani, S.B. Rao, and B.M. Maggs, "A Maximum-Likelihood Stereo Algorithm," *Computer Vision and Image Understanding,* vol. 63, no. 3, pp. 542-567, May 1996.

[22] S. Roy and I.J. Cox, "A Maximum-Flow Formulation of the N-Camera Stereo Correspondence Problem," *Proc. Int'l Conf. Computer Vision,* pp. 492-499, 1998.

[23] H. Ishikawa and D. Geiger, "Occlusions, Discontinuities, and Epipolar Lines in Stereo," *Proc. European Conf. Computer Vision,* vol. 1, pp. 232-248, 1998.

[24] V. Kolmogorov and R. Zabih, "Computing Visual Correspondence with Occlusions via Graph Cuts," *Proc. Int'l Conf. Computer Vision,* vol. 2, pp. 508-515, 2001.

[25] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 11, pp. 1222-1239, Nov. 2001.

[26] V. Kolmogorov and R. Zabih, "Multi-Camera Scene Reconstruction via Graph Cuts," *Proc. European Conf. Computer Vision,* vol. 3, pp. 82-96, 2002.

[27] D. Marr and T.A. Poggio, "Cooperative Computation of Stereo Disparity," *Science,* vol. 194, no. 4262, pp. 283-287, Oct. 1976.

[28] C.L. Zitnick and T. Kanade, "A Cooperative Algorithm for Stereo Matching and Occlusion Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 7, pp. 675-684, July 2000.

[29] A. Luo and H. Burkhardt, "An Intensity-Based Cooperative Bidirectional Stereo Matching with Simultaneous Detection of Discontinuities and Occlusions," *Int'l J. Computer Vision,* vol. 15, no. 3, pp. 171-188, July 1995.

[30] Y. Zhang and C. Kambhamettu, "Stereo Matching with Segmentation-Based Cooperation," *Proc. European Conf. Computer Vision,* vol. 2, pp. 556-571, 2002.

[31] D. Scharstein and R. Szeliski, "Stereo Matching with Nonlinear Diffusion," *Int'l J. Computer Vision,* vol. 28, no. 2, pp. 155-174, 1998.

[32] J. Sun, N.N. Zheng, and H.Y. Shum, "Stereo Matching Using Belief Propagation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 7, pp. 787-800, July 2003.

[33] J. Sun, Y. Li, S.B. Kang, and H.Y. Shum, "Symmetric Stereo Matching for Occlusion Handling," *Proc. Int'l Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 399-406, 2005.

[34] Z. Zhang and Y. Shan, "A Progressive Scheme for Stereo Matching," *Lecture Notes in Computer Science,* vol. 2018, pp. 68-85, 2001.

[35] J.Y. Goulermas and P. Liatsis, "A Collective-Based Adaptive Symbiotic Model for Surface Reconstruction in Area-Based Stereo," *IEEE Trans. Evolutionary Computation,* vol. 7, no. 5, pp. 482-502, 2003.

[36] S. Birchfield and C. Tomasi, "Multiway Cut for Stereo and Motion with Slanted Surfaces," *Proc. Int'l Conf. Computer Vision,* pp. 489-495, 1999.

[37] H. Tao, H.S. Sawhney, and R. Kumar, "A Global Matching Framework for Stereo Computation," *Proc. Int'l Conf. Computer Vision,* vol. 1, pp. 532-539, 2001.

[38] M.H. Lin and C. Tomasi, "Surfaces with Occlusions from Layered Stereo," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 710-717, 2003.

[39] Y. Wei and L. Quan, "Region-Based Progressive Stereo Matching," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 106-113, 2004.

[40] L. Hong and G. Chen, "Segment-Based Stereo Matching Using Graph Cuts," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 74-81, 2004.

[41] M.F. Tappen and W.T. Freeman, "Comparison of Graph Cuts with Belief Propagation for Stereo, Using Identical MRF Parameters," *Proc. Int'l Conf. Computer Vision,* pp. 900-907, 2003.

[42] A.S. Ogale and Y. Aloimonos, "Stereo Correspondence with Slanted Surfaces: Critical Implications of Horizontal Slant," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 568-573, 2004.

[43] M. Okutomi, Y. Katayama, and S. Oka, "A Simple Stereo Algorithm to Recover Precise Object Boundaries and Smooth Surfaces," *Int'l J. Computer Vision,* vol. 47, nos. 1-3, pp. 261-273, Apr. 2002.

[44] R. Sara and R. Bajcsy, "On Occluding Contour Artifacts in Stereo Vision," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* pp. 852-857, 1997.

[45] C.K. Tang and G. Medioni, "Inference of Integrated Surface, Curve, and Junction Descriptions from Sparse 3D Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 11, pp. 1206-1223, Nov. 1998.

[46] D. Scharstein and R. Szeliski, "High-Accuracy Stereo Depth Maps Using Structured Light," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 195-202, 2003.

Philippos Mordohai received the Diploma in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 1998, and the MS and PhD degrees, both in electrical engineering, from the University of Southern California, Los Angeles, in 2000 and 2005, respectively. He is currently a postdoctoral research associate in the Department of Computer Science of the University of North Carolina at Chapel Hill. His doctoral dissertation work focused on the development of perceptual organization approaches for computer vision and machine learning problems. The topics he has worked on include feature inference in images, figure completion, binocular and multiple-view stereo, instance-based learning, dimensionality estimation, and function approximation. His current research is on the reconstruction of urban environments from multiple video-cameras mounted on a moving vehicle. He is a member of the IEEE and the IEEE Computer Society and a reviewer for both the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and the *IEEE Transactions on Neural Networks*.

Gérard Medioni received the Diplome d'Ingénieur Civil from the École Nationale Supérieure des Télécommunications, Paris, France, in 1977, and the MS and PhD degrees in computer science from the University of Southern California (USC), Los Angeles, in 1980 and 1983, respectively. He has been with (USC) since 1983, where he is currently a professor of computer science and electrical engineering, codirector of the Computer Vision Laboratory, and chairman of the Computer Science Department. He was a visiting scientist at INRIA Sophia Antipolis in 1993 and Chief Technical Officer of Geometrix, Inc. during his sabbatical leave in 2000. His research interests cover a broad spectrum of the computer vision field and he has studied techniques for edge detection, perceptual grouping, shape description, stereo analysis, range image understanding, image to map correspondence, object recognition, and image sequence analysis. He has published more than 100 papers in conference proceedings and journals. Dr. Medioni is a fellow of the IEEE and a fellow of the IAPR. He has served on the program committees of many major vision conferences and was program chairman of the 1991 IEEE Computer Vision and Pattern Recognition Conference in Maui, program cochairman of the 1995 IEEE Symposium on Computer Vision held in Coral Gables, Florida, general cochair of the 1997 IEEE Computer Vision and Pattern Recognition Conference in Puerto Rico, program cochair of the 1998 International Conference on Pattern Recognition held in Brisbane, Australia, and general cochairman of the 2001 IEEE Computer Vision and Pattern Recognition Conference in Kauai. He is on the editorial board of the *Pattern Recognition and Image Analysis* journal and the *International Journal of Computer Vision* and one of the North American editors for the *Image and Vision Computing* journal.