Stereo Matching with Segmentation-Based Cooperation

Ye Zhang and Chandra Kambhamettu

Video/Image Modeling and Synthesis Lab
University of Delware, Newark DE 19716, USA,
{zhangye,chandra}@cis.udel.edu,
http://www.cis.udel.edu/∼vims

Abstract. In this paper we present a new stereo matching algorithm that produces accurate dense disparity maps and explicitly detects occluded areas. This algorithm extends the original cooperative algorithms in two ways. First, we design a method of adjusting the initial matching score volume to guarantee that correct matches have high matching scores. This method propagates "good" disparity information within or among image segments based on certain disparity confidence measurement criterion, thus improving the robustness of the algorithm. Second, we develop a scheme of choosing local support areas by enforcing the image segmentation information. This scheme sees that the depth discontinuities coincide with the color or intensity boundaries. As a result, the foreground fattening errors are drastically reduced. Extensive experimental results demonstrate the effectiveness of our algorithm, both quantitatively and qualitatively. Comparison between our algorithm and some other representative algorithms is also reported.

Keywords. Stereoscopic Vision, Occlusion Detection, Cooperative Algorithm.

1 Introduction

Stereo matching has long been one of the central research problems and thus one of the most heavily studied areas in computer vision. Traditional stereo matching algorithms, also known as *feature-based* methods, only match points with a certain amount of local information (such as zero-crossings or intensity edges), with the disadvantage of producing only sparse disparity maps. However, most modern applications (such as view synthesis, image-based rendering, z-keying, and virtual reality) require dense, accurate disparity maps. Therefore, we focus on dense stereo matching approaches in this paper.

Numerous dense stereo matching algorithms, including local matching (e.g., [16,10]), global optimization (e.g., [23,7]), dynamic programming (e.g., [2,14]), and cooperative algorithms (e.g., [18,28]), have been proposed over the past decades. An excellent taxonomy and evaluation of dense stereo algorithms can be

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2351, pp. 556–571, 2002.

found in [25]. According to the requirements of modern applications in computer graphics and virtual reality, the disparity maps recovered by a stereo matching algorithm should be smooth and detailed, i.e., continuous and even surfaces should produce a region of smooth disparities with their boundaries precisely delineated. Unfortunately, the disparity maps produced by most stereo matching algorithms have *foreground fattening* errors due to disparity discontinuities. Adaptive window [16] and iterative evidence aggregation [24] may sometimes mitigate these errors to some extent. But they do not explicitly handle depth discontinuities and are computationally expensive. The segmentation based method proposed in [27] assumes that the disparities are piecewise smooth and embed this assumption into the planar representation of the disparities within individual image segments. This method is able to produce results with less fattening errors. However, the planar assumption may be an oversimplification of a real scene. Another difficult but critical problem in stereo matching is the handling of occlusion. Some algorithms [3,15,5] have been proposed to use the *ordering constraint* to detect occlusions. However, this constraint may not be valid in a real scene containing thin vertical foreground objects. Textureless areas pose another challenge to stereo matching. Without enough local color/intensity variations, local matching methods tend to generate arbitrarily wrong results. In this case, global optimization is preferred since there are chances that the information from correct matches can be propagated to the textureless areas.

Generally speaking, accurate stereo matching remains difficult due to depth discontinuities, occluded and textureless areas, to name a few. In this paper, we propose a new global stereo algorithm, segmentation-based cooperation, that produces accurate dense disparity maps and explicitly detects occluded areas. In our algorithm, the reference image is first segmented into homogeneous regions and each image segment is labeled with a confidence level by using cross validation. We then extend the original cooperative algorithms [18,28] in two ways. First, we design a method of adjusting the initial matching score volume. This method introduces a new concept, "feature disparity", of a local patch within an image segment. The feature disparity can be thought of as "good" disparity information propagated within/among the image segments based on certain confidence measurement criterion. The initial matching scores of the feature disparity are set to a relatively large value so as to guarantee that correct matches have high initial matching scores. This technique raises the chances for the following update (inhibition) process to locate the correct matches. Second, we develop a scheme of choosing local support areas by enforcing image segmentation information. In this scheme, two different kinds of local support areas, matching support area and smoothing support area, are clearly distinguished. This scheme sees that the depth discontinuities coincide with the color/intensity boundaries. As a result, the foreground fattening errors are drastically reduced.

The rest of this paper is organized as follows. Section 2 discusses the general assumptions in our algorithm. Section 3 presents the segmentation-based cooperative algorithm. Extensive experimental results are reported in Section 4. Section 5 concludes this paper.

2 General Assumptions

All vision algorithms, explicitly or implicitly, embrace certain assumptions. For stereo matching problem, the most widely adopted assumptions include *uniqueness* and *smoothness*, i.e., the disparity maps have unique values and are continuous. In fact, these two assumptions are made not only in stereo matching but also in motion analysis, where the motion displacements are assumed to be unique and smooth. Some attempts [1,26] have been made to relax the uniqueness assumption when transparent surfaces exist in the scene. However, dealing with transparency is very difficult and the proposed methods [1,26] only work in some simple situations. In this paper, we only consider the more usual case: opaque scenes. Therefore, we still make the uniqueness assumption. The concept of "inhibition area" proposed in cooperative algorithms [18] explicitly reflects the uniqueness assumption. For two stereo images that are horizontally rectified, Fig. 1 [28] illustrates the inhibition area of a point (x, y) on the reference image when assigned a disparity d. It is easy to see that the inhibition area consists of all the possible 3D points that are projected to (x, y) on the reference image and to $(x + d, y)$ on the other view. Since the inhibition area is explicitly considered in the update functions, cooperative algorithms possess a global optimization behavior.

Generally, the smoothness assumption is valid for the projected image areas of continuous and even surfaces. But at surface or object boundaries, this assumption is often broken. If the matching algorithm is not aware of this, the resultant disparity maps tend to be oversmooth, i.e., the details may be lost. A lot of efforts [10,13,5,27] have been made to intelligently enforce the smoothness constraint so that the disparity discontinuities can be well preserved. Inspired by [27], we assume that the disparities vary smoothly within a homogeneous image segment. However, unlike [27], we do not assume image segments as the projected areas of planes, thus making our approach more general.

Finally, without loss of generality, we assume that the input images are well rectified, i.e., the disparities are purely along one dimension.

3 Algorithm and Implementation

3.1 Initial Matching Score Volume

The image coordinate x, y and the disparity d defines a 3D disparity space. To compute the stereo match for a point $p(x, y)$ on the reference image, we need to compute the matching score $E_0(x, y, d)$ at each disparity level. Therefore all the matching scores for different points at different disparity levels form a 3D volume [1] in this disparity space. This volume can be defined as

$$E_0(x, y, d) = \rho(I_r, I_l, x, y, d), \tag{1}$$

[1] This volume is also called Disparity-Space Image (DSI) in [25] and [5].

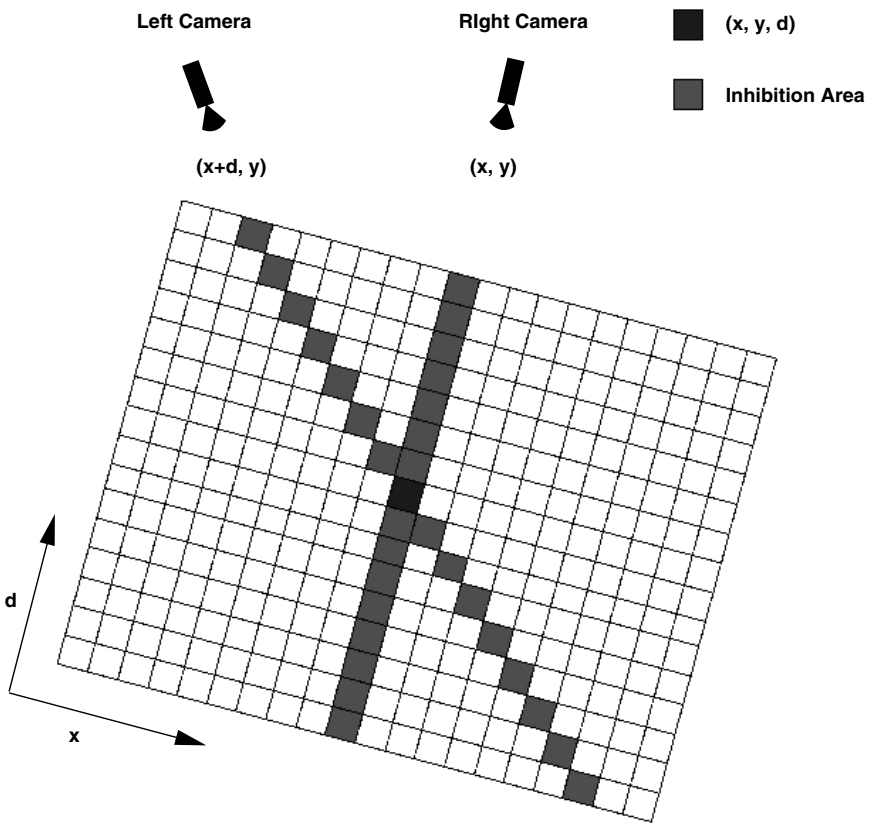

Fig. 1. The inhibition area illustrated on a slice of matching volume (y coordinate is held constant). This illustration is based on a well-rectified image pair where the disparity is purely along x dimension.

where I_l, I_r are the intensity functions of the left and right images, respectively, and ρ is the similarity measurement function (e.g., sum-of-squared-difference (SSD), sum-of-absolute-difference(SAD), or normalized correlation). Although we only discuss the two-frame case, it is straightforward to extend the matching score volume to the multiple-frame case by simply summing up the scores from other views since the matching score volume is associated with a fixed reference image. For example, [20], [17] and [19] exploited this idea to compute multiple baseline stereo by using sum-of-SSD (SSSD) or sum-of-SAD (SSAD).

3.2 Initial Matching Score Adjustment

Cooperative algorithms require that the correct matches produce high initial matching scores. However, the opposite does not need to be true [28]. This is because cooperative algorithms make decisions "globally" through inhibition

and local support, thus possessing a good tolerance to false high-score matches. However, due to projective distortion and inappropriate window sizes or image noise, some correct matches may produce low initial matching scores. Therefore, we need to adjust the initial matching score volume so that we can make sure that correct matches are indeed labeled with high initial scores.

We exploit the ideas of *cross validation* and *image segmentation* to adjust the initial matching scores. In this paper, we adopt the image segmentation algorithm proposed in [8]. According to our assumptions, the disparities should change smoothly within an image segment. In [27], these disparities have been modeled as a plane, or a plane-plus-parallax. More advanced models for image segments, such as a variable-order parametric model [4], have also been proposed to represent the displacements during motion analysis. However, by using model-based representation, the accuracy of the recovered disparities is limited by the model's ability to approximate the real surfaces. For example, if the scene contains a spherical surface that is approximated by a plane model, the results will not be accurate. In our work, we do not use any *a priori* model to represent the disparities in an image segment. Instead, we adopt a "multiresolution" strategy to adjust the initial matching scores so that the update process can be attracted towards the correct matches.

Image segmentation, in fact, is an information-based scale-space filtering. In the segmentation map, the image portions with similar color or intensity, and normally within the same neighborhood, are aggregated as a segment, thus reducing the resolution. However, similar colors do not always mean similar depths. For example, the projected image area of a very slanted Lambertian surface with uniform texture tends to be classified as one image segment, while the depths may change a lot within this segment. Based on this observation, to adjust the matching volume, we first split each image segment into small local patches (segments smaller than the pre-defined local patch size are deemed as one patch. If the final remainder of a large segment after splitting is smaller than the patch size, it is deemed as one patch). The splitting process is illustrated in Fig. 2. Splitting a large segment into small local patches equals to increasing the resolution of the disparities within this segment. We assume that, within each small local patch, the disparities are very similar, i.e., if we assign an appropriate disparity to all the points in the local patch, the corresponding patch in the other view should be very similar to the original patch. We call this disparity "feature disparity" of this local patch. We can find the feature disparity d_i of patch p_i by solving

$$d_i = \underset{d \in D}{argmax}\ Sim(i, d), \tag{2}$$

where $Sim(i, d)$ is a similarity measurement when patch p_i is assigned a disparity d, $D = (D_{min}, D_{max})$ is the set of all the possible disparities.

However, performing exhaustive search in D to find the feature disparity of a patch is not a good idea. The reasons are two-fold: one is that exhaustive search is time-consuming; the other is that large search scope increases the chances for image noise to overwhelm the correct results, as typically the local patch does

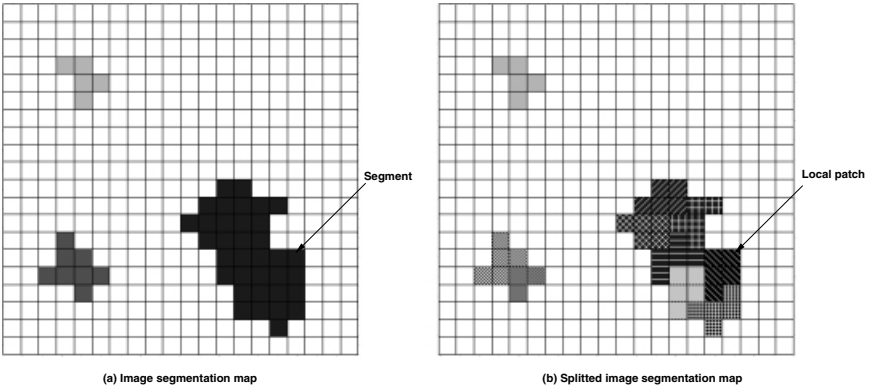

Fig. 2. The process of splitting image segments.

not have much texture information. In our algorithm, we design a method to reduce the search scope based on a confidence measurement of each segment. The confidence measurements are delivered via cross validation[2]. First, a matching score volume $E_0(x, y, d)$ is computed between a stereo pair I_0 and I_1 along the epipolar line. The measured disparity is the one with the largest matching score. We perform the similarity computation twice by reversing the roles of the two images and consider as valid only those matches for which we measure the same depth at corresponding points when matching from I_0 to I_1 and from I_1 to I_0[3]. To further increase the $Signal/Noise$ ratio, we filter out those valid points that are either isolated or have very large standard deviation in a small neighborhood. Finally, we get an initial disparity map with few errors. Fua [11] pointed out that as the $Signal/Noise$ ratio decrease, the performance of cross validation degrades gracefully in the sense that the density of matches decreases accordingly but the ratio of correct to false matches remains high. In other words, a relatively *dense* disparity map is almost a *guarantee* that the matches are correct (up to the precision allowed by the resolution being used). In fact, to further guarantee the correctness of the valid matches, we can adopt a simplified version of adaptive windows [16] to perform the cross validation: We can perform cross validation by using different sizes of local windows (e.g., 5×5 and 3×3) and consider as valid only those matches for which we measure the same depth by using different window sizes. Based on these observations, if we divide the reference image into segments with homogeneous color/intensity, we can label the confidence level

[2] It is also interesting to notice that from the perspective of cooperative algorithms, cross validation is actually performed in the inhibition area.

[3] Following this method, it is straightforward to extend cross validation to more than two frames: Compute the valid disparities between I_0 and $I_2, I_3, ..., I_{N-1}$, respectively. Then merge the results together and get a sparse initial "valid" disparity map. If two sets of views produce different valid disparities, the one with higher matching score wins. Also, the matching score in the matching score volume is updated accordingly.

of the disparities in each image segment according to the density of the valid matches within the corresponding segment. That is,

$$L(s) = \begin{cases} VALID & if \ \ r \geq \alpha_1; \\ SEMIVALID & if \ \ \alpha_2 \leq r < \alpha_1; \\ INVALID & if \ \ r < \alpha_2, \end{cases} \tag{3}$$

where r is the ratio of valid disparity points in segment s, α_1 and α_2 are positive thresholds, and $VALID, SEMIVALID$ and $INVALID$ are all symbolic values. $VALID$ means that we have high confidence on the disparity map within segment s. $INVALID$ means low confidence, and $SEMIVALID$ means medium confidence. This labeling method reflects an assumption we have made: image segments where the valid disparity points are dense are more reliable. Again, Fua's experiments [12] have shown that this assumption holds in most cases.

Once we have labeled the confidence level of image segments, we can compute the feature disparity of each local patch by

1. If patch p_i belongs to a $VALID$ segment s, find the minimum ($dmin_i$) and maximum ($dmax_i$) disparities of all the valid points in s. Then solve

$$d_i = \underset{dmin_i - \delta < d < dmax_i + \delta}{argmax} Sim(i, d), \tag{4}$$

   where δ is a small positive number, and d_i is the feature disparity of p_i.
2. If patch p_i belongs to a $SEMIVALID$ segment s, find the minimum ($dmin_i$) and maximum ($dmax_i$) disparities of all the valid points in s and all its neighboring segments. Solve Eq. 4, and d_i is the feature disparity of p_i. If patch p_i belongs to an $INVALID$ segment s, d_i is undefined.

From above, we can see that for $VALID$ and $SEMIVALID$ segments, the disparity range for searching feature disparity has been reduced. This will not only improve the efficiency but also improve the robustness of the algorithm. Notice that we do not compute the feature disparity for $INVALID$ segments. This is because it is very possible that the $INVALID$ segments may be occluded areas and no reliable information is available. The feature disparities computed for Tsukuba head scene are illustrated in Fig. 3.

After we find the feature disparities for some patches, the initial matching score volume is adjusted as

$$E_0(x, y, d) = \begin{cases} C & if \ \ \exists i \ \ (x, y) \in p_i \ \ and \ \ d = d_i, \\ E_0(x, y, d) & else, \end{cases} \tag{5}$$

where d_i is the feature disparity for local patch p_i and C is a relatively large matching score. In our work, it is set as the maximum value of all the initial match scores at point (x, y).

Compared with [27] where disparities in one segment are always iteratively hypothesized according to the neighboring segments (we call this *inter-segment* hypothesis), our method should be more efficient. For a segment with high confidence, we search the feature disparity only within the valid disparity range of

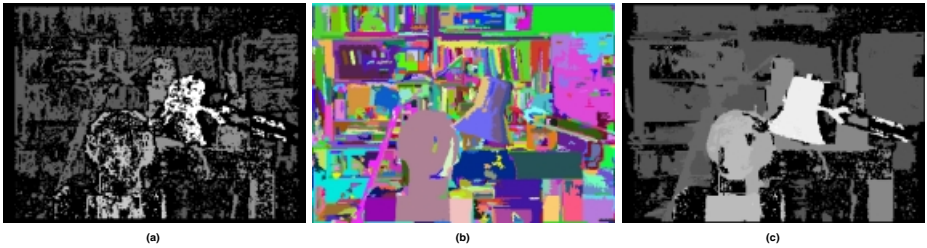

Fig. 3. Feature disparities computation: (a) initial valid disparity map, (b) image segmentation map, (c) computed feature disparity map (black areas mean no definition).

this segment (we call this *intra-segment* hypothesis). Only for those segments without enough confidence, we utilize the information from the neighbors. It is also important to remember that the feature disparity is *not* the final disparity value assigned to every point in one local patch. However, we believe that it should be close to the correct disparities. Therefore, the feature disparity serves as a force to drag the update process towards the correct disparities. It is still the global cooperation that makes the final decision on the disparity values.

3.3 Segmentation-Based Local Support

Many stereo matching algorithms require a concept of "local support", i.e., aggregation of evidence from the neighboring pixels. This is because stereo matching is in general ambiguous: there may be multiple equally good matches if the matching score is computed independently at each point. Normally a support region is a two-dimensional $(x - y)$ or three-dimensional $(x - y - d)$ neighborhood of the current pixel. Traditional local support assumptions include: the depth constancy assumption by Marr and Poggio [18,9], the disparity gradient limit by Pollard, Mayhew and Frisby (PMF) [21], the disparity similarity function by Prazdny [22]. Kanade and Okutomi [16] presented a detailed analysis of the relationship and differences among them. They also proposed a method to adaptively adjust the SSD or correlation window size and shape according to the variation of the local intensities and disparities.

However, it is important to realize that there are two kinds of "local support". The first one is used to compute the matching score of a point given a disparity. The matching score of a point is normally the similarity measurement between a local area of the interested point and the local area in the other view corresponding to the assigned disparity. We call this local area, often in the form of a $m \times n$ window centered on the interested point, a "matching support" area. The matching support area needs to be large enough to contain enough color/intensity variations (texture information) for reliable matching, and be small enough to avoid the effects of projective distortion. For example, the local window used to compute the matching volume and the adaptive window in [16] belong to the matching support. The underlying assumptions of this support are Lambertian surfaces, i.e., surfaces whose appearance does not vary with

viewpoint. The other kind of local support is called the "smoothing support". The local support used in cooperative algorithm [28] belongs to this category. The purpose of the smoothing support is mainly to propagate disparity information within a neighborhood and make the resultant disparity maps smooth. The underlying assumption is that the disparities do not vary much within the smoothing support area of the interested point. In smoothing support area, we do not need rich color/intensity variations (texture information). The only concern is to make sure disparities actually do not change much within this area.

For simplicity, Zitnick and Kanade [28] chose a box-shaped 3D local smoothing support area. The problem of this simple strategy is that the depth discontinuities may be blurred because the smoothing support area may be applied across the depth boundaries. In our work, we propose a scheme to choose the smoothing support area by utilizing the image segmentation information. Specifically, we define the 3D smoothing support area of a point (x, y) as

$$\Phi(x, y, d) = \{(x', y', d') \| (x, y) \in p_i \wedge (x', y') \in p_i \wedge d' \in [d - r_d, d + r_d]\}, \quad (6)$$

where p_i is the local patch within an image segment that contains (x, y), and r_d is a small positive number that defines the support along the d dimension. Since the local patch within an image segment is used to define the $x - y$ support, the image segmentation information is explicitly enforced.

Then, we can define an aggregated matching score volume by averaging the matching scores within the smoothing support areas:

$$A_n(x, y, d) = \frac{1}{N(x, y, d)} \sum_{(x', y', d') \in \Phi(x, y, d)} E_n(x', y', d'), \quad (7)$$

where E_n is the matching volume, n is the iteration number, and $N(x, y, d)$ is the number of points in $\Phi(x, y, d)$.

Observing the definition of Φ carefully, we can notice that it has some interesting characteristics. First, unlike common local support definitions, smoothing support areas Φ of different points do not overlap with each other in the image plane. Second, all the points in patch p_i have the same aggregated matching score. This means that the aggregation process can be implemented in a very fast way. However, the drawback is that the disparity propagation within one image segment may not be enough. Therefore, a substitute smoothing support area of (x, y, d) can be defined as (x, y)'s local patch p_i and p_i's neighbors, i.e.,

$$\begin{aligned} \Phi'(x, y, d) = \{(x', y', d') \| \ & (x, y) \in p_i \wedge \\ & ((x', y') \in p_i \vee ((x', y') \in p_j \wedge p_j \textit{ is a neighbor of } p_i)) \wedge \\ & d' \in [d - r_d, d + r_d]\}. \end{aligned} \quad (8)$$

Then the aggregated matching volume is defined as

$$A'_n(x, y, d) = \frac{1}{N'(x, y, d)} \sum_{(x', y', d') \in \Phi'(x, y, d)} E_n(x', y', d'), \quad (9)$$

where $N'(x, y, d)$ is the number of points in $\Phi'(x, y, d)$.

$A'_n(x, y, d)$ has the advantage of propagating more information through an image segment because Φ' overlap with each other in an image segment, thus making the disparities within the image segment more smooth. In our implementation, we use Φ' as local support areas. It is also worth mentioning that weighted summation may be used in Eq. 9. For simplicity and efficiency reasons, we use simple summation in our implementation.

3.4 Matching Score Update

The uniqueness assumption states that one pixel in the reference image has only one match within a set of elements that project to the same pixel in the other view. As illustrated in Fig. 1, the inhibition area of a point (x, y, d) can be defined as all the elements that overlap this point when projected to an image. This means that the inhibition area consists of two lines of sight. In other words, the inhibition area Ψ of point (x, y) when assigned disparity d can be defined as

$$\Psi(x, y, d) = \{(x', y', d') \| \ (x', y', d') \ \textit{projected to} \ (x, y) \ \textit{in the reference view or} \ (x + d, y) \ \textit{in the other view}\}. \tag{10}$$

Many inhibition functions are available. Here we choose the one used by Zitnick and Kanade [28] for it's simplicity:

$$R_n(x, y, d) = \left(\frac{A_n(x, y, d)}{\sum_{(x', y', d') \in \Psi(x, y, d)} A_n(x', y', d')} \right)^{\alpha}, \tag{11}$$

where $R_n(x, y, d)$ denotes the amount of inhibition at (x, y, d) and α is a positive constant called the "inhibition constant". This constant controls the speed of convergence. To guarantee that there is a single element within Ψ that will converge to 1, α must be greater than 1. Then the update function can be defined as

$$A_{n+1}(x, y, d) = A_0(x, y, d) * R_n(x, y, d). \tag{12}$$

To prevent oversmoothing to some extent, the initial aggregated match values A_0 are introduced in this update function to restrict the current match values. Zitnick and Kanade [28] compared this update function with the original Marr and Poggio [18] update function. Three advantages have been claimed: First, the Marr and Poggio function used discrete match values, and was not well defined for real scenes, while Eq. 12 uses continuous matching values and is well defined for real scenes. Second, use of A_0 maintains better disparity details. Third, Eq. 12 is much simpler and computationally efficient. In our work, we used this update function and found that it worked excellently in our experiments. Fig. 4 illustrates an initial versus converged slice of the matching volume.

Fig. 4. The convergence of a slice in the matching score volume by applying Eq. 12: (a) the initial slice, (b) the converged slice.

3.5 Occlusion, Confidence Measurement, and Subpixel Accuracy

Real scenes almost always contain occluded areas. Unfortunately, most stereo algorithms are not able to handle occlusion explicitly. Instead, most of them hypothesize disparities in occluded areas based on the disparities in the neighborhood and may produce errors. Increasing the number of cameras is a natural way to reduce occlusion, but it is not always feasible. Some research (e.g., [5]) have proposed finding occlusion and matches simultaneously by imposing the ordering constraint, which states that the objects maintain the same left-to-right order in different views. However, the ordering constraint may mislabel visible pixels as occluded, and this constraint may not be valid when there exist thin vertical objects in the foreground.

In cooperative algorithms, the converged match values can be used as a natural criterion for occlusion detection [28]. Because no correct matches exist for the occluded areas, the converged match values corresponding to occluded pixels should be small. Furthermore, the update (inhibition) process decreases the match values of occluded pixels. However, for mutually occluded areas within the disparity range, higher match values may occur in occluded areas. So, as long as the mutually occluded areas do not have similar colors/intensities, all the converged match scores at occluded pixels should be small. Thus, in our algorithm, if a converged matching score is less than a threshold, the corresponding pixel is labeled as occluded. Following the same logic, the converged match scores of the resultant disparity map can be directly used as its confidence measurements.

In our algorithm, subpixel accuracy can be achieved via two ways. One is to split the initial matching volume into half-pixel or quarter-pixel levels. The matching scores at subpixel levels can be interpolated by fitting a curve (e.g., a quadric) to the neighboring scores. The other way is to directly fitting a curve to the final matching scores after the update process converges. We adopt both methods in our implementation.

4 Experimental Results

We have implemented our segmentation-based cooperative (SBC) algorithm under a PC platform. The algorithm takes about four seconds per iteration with 256×256 images on a Pentium III 800MHz machine. We have applied this algorithm to real imagery. Fig. 5 illustrates the results on the head scene from

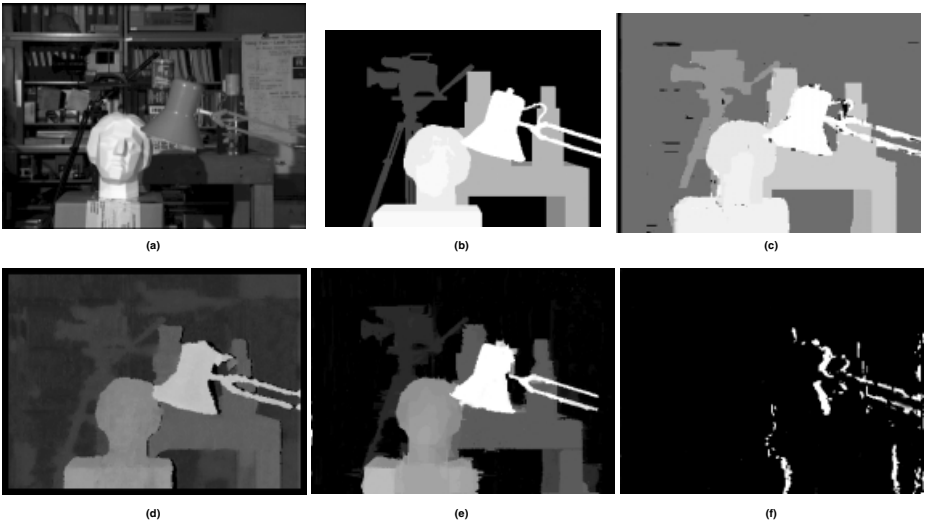

Fig. 5. Results on Tsukuba heads cene: (a) is the reference image. (b) is the ground truth. (c), (d) and (e) are the disparity maps computed by using the GPM-MRF algorithm [6], the cooperative algorithm [28], and the SBC algorithm, respectively. (f) is the occlusion map computed by using the SBC algorithm.

University of Tsukuba. This data set consists of 25 images taken from a 5×5 camera array. Only two images along a horizontal row are used as the input data to our algorithm. The head scene contains textureless areas such as the table and the lampshade. It also contains thin structures such as the rods of the lamp. Figures 5 (a) and (b) show the reference image and the ground truth, respectively. For comparison purpose, Figures 5 (c), (d), (e) show the results produced by using the GPM-MRF algorithm [6], the cooperative algorithm [28], and the SBC algorithm, respectively. Fig. 5 (f) shows the detected occluded areas by the SBC algorithm. We can see that the rods of the lamp, the shape of the head, the outline of the desk and the profile of the camera are clearly preserved in our result. We can also see that for thin structures (such as the rods of the lamp), our algorithm produces least fattening errors. Our algorithm also correctly reports the occluded areas, i.e., the right sides of the lampshade, the desk and the head, the upper sides of the rods.

Since the head scene is provided with dense ground truth disparities, we can quantitatively evaluate the SBC algorithm. Table 1 shows the comparison between the SBC algorithm and some other representative algorithms. The error rate is defined as the percentage of those disparity values with absolute errors greater than one pixel compared with the ground truth. From the table we can see that our algorithm produces very accurate results.

Fig. 6 presents the results on the CMU shrub scene and coal mine scene. The disparity maps are smooth and maintain clear depth boundaries at the same time. For the CMU shrub scene, the parking sign and the shrub boundaries are

Table 1. Comparison of SBC and other algorithms on Tsukuba head scene

Algorithms	$Error > 1$
SBC	1.2
Zitnick and Kanade[28]	1.4
GPM-MRF [6]	2.8
LOG-filtered L_1 [6]	9.0
Normalized Correlation [6]	10.0

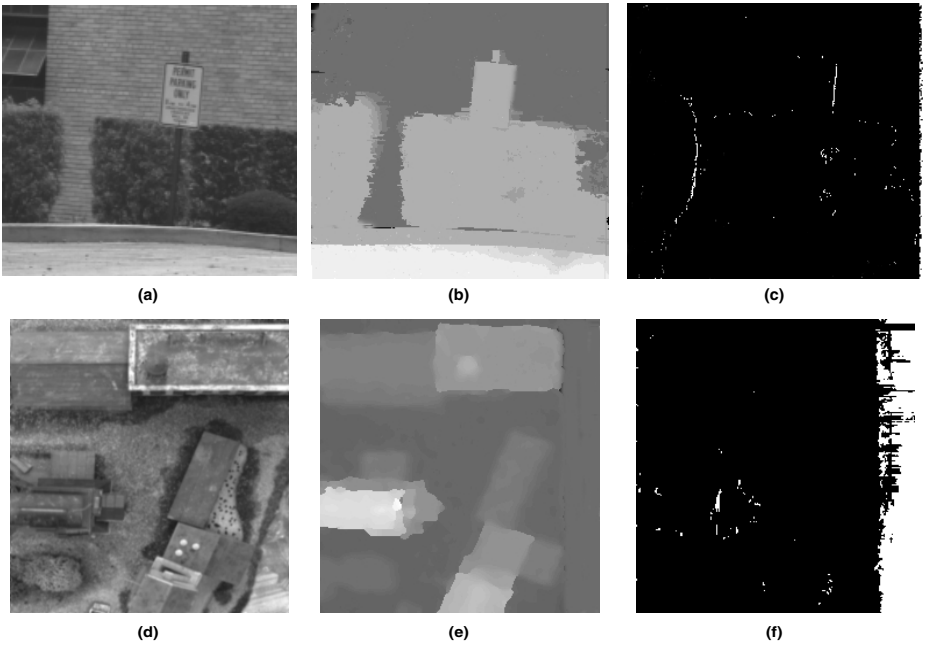

Fig. 6. Results on CMU shrub and coal mine scenes: (a) and (d) are the reference images of shrub and coal mine scenes, respectively. (b) and (e) are the disparity maps computed by using the SBC algorithm. (c) and (f) are the computed occlusion maps.

clearly preserved in the produced disparity map. Although the background (the brick wall) contains a lot of repetitive patterns, our algorithm still recovered the disparities correctly. For the CMU coal mine scene, the shapes of the buildings are precisely delineated in the produced disparity map.

We also extended our algorithm to three-frame stereo matching by accumulating the matching scores from all views. Fig. 7 shows the results by applying our algorithm on a Triclops snapshot. The Triclops consists of three calibrated camera heads configured as a "L" shape. Figures 7 (a), (c) and (d) show the images acquired by the top, the left and the right camera, respectively. Fig. 7 (b) shows the computed disparity map. We can see that the outline of the person

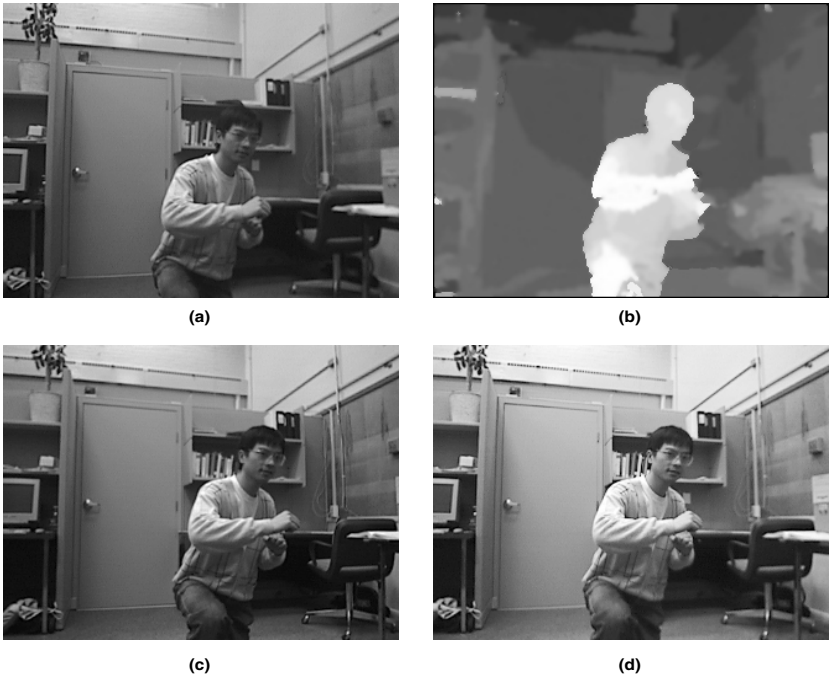

Fig. 7. Results on a three-view lab scene by using SBC algorithm: (a) top image, (b) disparity map, (c) left image, (d) right image.

is clearly maintained. The depth relationships between the body, the hand and the leg are accurately recovered.

We have further performed extensive experiments on other benchmark data (such as the pentagon scene, the meter machine scene, etc.) and also on stereo data produced in our lab. Our system has consistently produced accurate disparity maps.

5 Conclusion

We have presented a new segmentation-based cooperative algorithm for stereo matching. This algorithm extends the earlier cooperative algorithms [18,28] in two ways. First, we designed a method of adjusting the initial matching score volume to guarantee that correct matches have high matching scores. This method propagates reliable disparity information among/within image segments based on the confidence labels of image segments, thus improving the robustness of the algorithm. Second, we developed a scheme for choosing local support areas by enforcing the image segmentation information. This scheme sees that the depth discontinuities coincide with the color/intensity boundaries. As a result, the foreground fattening errors are drastically reduced. We also show that the converged

matching scores can be used as the confidence measurements and occluded areas can be easily detected by setting a threshold on the converged matching scores. Through extensive experiments, we demonstrate the effectiveness of our SBC algorithm.

Our algorithm may produce oversmooth results when depth discontinuities appear in a homogeneous image segment. One possible solution is to enforce not only color/intensity segmentation information, but also depth segmentation information. By doing so we can make sure that the smoothing support areas seldom overlap depth discontinuities, thus maintaining more detailed depth boundaries.

Acknowledgments. Research funding was provided by the National Science Foundation Grants CAREER IRI-9984842 and CISE CDA-9703088.

References

1. E.H. Adelson and P. Anandan. Perceptual organization and the judgment of brightness. *Science*, 262:2042–2044, 1993.
2. P.N. Belhumeur. A bayesian-approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3):237–260, August 1996.
3. P.N. Belhumeur and D. Mumford. A bayesian treatment of the stereo correspondence problem using half-occluded regions. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 506–512, 1992.
4. M.J. Black and A.D. Jepson. Estimating optical-flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):972–986, October 1996.
5. A.F. Bobick and S.S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3):1–20, September 1999.
6. Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 648–655, 1998.
7. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *Proceedings of IEEE Computer Society International Conference on Computer Vision*, pages 377–384, 1999.
8. D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 750–755, 1997.
9. M. Drumheller and T.A. Poggio. On parallel stereo. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1439–1448, 1986.
10. P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6(1):35–49, 1993.
11. P.V. Fua. Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1292–1298, 1991.
12. P.V. Fua. Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1292–1298, 1991.

13. E.B. Gamble and T. Poggio. Visual integration and detection of discontinuities: The key role of intensity edges. In *MIT AI Memo*, 1987.
14. D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. In *Proceedings of European Conference on Computer Vision*, pages 425–433, 1992.
15. H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *Proceedings of European Conference on Computer Vision*, pages xx–yy, 1998.
16. T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, September 1994.
17. S.B. Kang, J. Webb, C.L. Zitnick, and T. Kanade. A multibaseline stereo system with active illumination and real-time image acquisition. In *Proceedings of IEEE Computer Society International Conference on Computer Vision*, pages 88–93, 1995.
18. D. Marr and T.A. Poggio. Cooperative computation of stereo disparity. *Science*, 194(4262):283–287, October 15 1976.
19. Y. Nakamura, T. Matsura, K. Satoh, and Y. Ohta. Occlusion detectable stereo – occlusion patterns in camera matrix. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 371–378, 1996.
20. M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, April 1993.
21. S.B. Pollard, J.E.W. Mayhew, and J.P. Frisby. Pmf: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.
22. K. Prazdny. Detection of binocular disparities. *BioCyber*, 52:93–99, 1985.
23. S. Roy and I.J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proceedings of IEEE Computer Society International Conference on Computer Vision*, pages 492–499, 1998.
24. D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. *International Journal of Computer Vision*, 28(2):155–174, 1998.
25. D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo methods. In *Proceedings of IEEE Workshop on Stereo and Multi-Baseline Vision*, pages 131–140, 2001.
26. R. Szeliski and P. Golland. Stereo matching with transparency and matting. *International Journal of Computer Vision*, 32(1):45–61, August 1999.
27. H. Tao, H.S. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *Proceedings of IEEE Computer Society International Conference on Computer Vision*, pages I: 532–539, 2001.
28. C.L. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):675–684, July 2000.