

Stereo Correspondence with Occlusion Handling in a Symmetric Patch-Based Graph-Cuts Model

Yi Deng, Qiong Yang, *Member, IEEE*, Xueyin Lin, and Xiaou Tang, *Senior Member, IEEE*

Abstract—A novel patch-based correspondence model is presented in this paper. Many segment-based correspondence approaches have been proposed in recent years. Untextured pixels and boundaries of discontinuities are imposed with hard constraints by the discontinuity assumption that large disparity variation only happens at the boundaries of segments in the above approaches. Significant improvements on performance of untextured and discontinuity area have been reported. But, the performance near occlusion is not satisfactory because a segmented region in one image may be only partially visible in the other one. To solve this problem, we utilize the observation that the shared edge of a visible area and an occluded area corresponds to the discontinuity in the other image. So, the proposed model conducts color segmentation on both images first and then a segment in one image is further cut into smaller patches corresponding to the boundaries of segments in the other when it is assigned with a disparity. Different visibility of patches in one segment is allowed. The uniqueness constraint in a segment level is used to compute the occlusions. An energy minimization framework using graph-cuts is proposed to find a global optimal configuration including both disparities and occlusions. Besides, some measurements are taken to make our segment-based algorithm suffer less from violation of the discontinuity assumption. Experimental results have shown superior performance of the proposed approach, especially on occlusions, untextured areas, and near discontinuities.

Index Terms—Stereo, correspondence, segmentation, graph-cuts, occlusion, energy minimization.



1 INTRODUCTION

STEREO correspondence is considered as a classical difficult problem due to its significance in computer vision and inherent ambiguity. It takes two or more images simultaneously captured by cameras from different viewpoints as its input. The resultant output is a dense disparity map that represents the correspondence between points in different images. The obtained disparity map can be used to recover the three-dimensional structure in the scene.

Two of the main challenges in stereo are *discontinuity* and *occlusion* problems. The discontinuity issue stems from a *smoothness* assumption, which is explicitly or implicitly used in many dense stereo approaches [1]. It assumes the disparity map to be smooth *almost* everywhere. However, this is violated at the boundary of the object. The convex smoothness function entails a significant penalty for large discontinuity and, therefore, leads to poor object boundary results [2], [3]. To cure this, some *discontinuity-preserving* smoothness functions are designed to improve the accuracy at discontinuity areas [1]. Common discontinuity preserving smoothness functions include the Potts function [4] and the truncated function [5]. A fixed amount of penalty is imposed for large

discontinuity in these methods. Moreover, the intensity differences between neighboring pixels are also used to guide the smoothness criteria [6], [5] so that neighboring pixels with similar colors are given harder smoothness constraints because they are more likely to have similar disparities. Recently, several *segment-based* methods have been proposed [7], [8], [9], [10], [11], [12]. Tao et al. [7] provided a global matching framework using image segmentation information. Hong and Chen [8] used graph-cuts to provide a global solution for segment matching, whereas a region-growing strategy was used by Wei and Quan [9]. Bleyer and Gelautz [10] formulated the correspondence problem in combination with the pixel and segment levels. The correspondence problem is modeled in the segment level and the occlusion is detected in the pixel level using the *uniqueness* constraint. In all these algorithms, a color segmentation process initially separates the reference image into several regions with uniform (or similar) colors and each region is assumed to correspond to a plane in the scene. With this polyhedral approximation of the scene, matching is performed using a segment as a unit. The discontinuity is constrained to be at the boundaries of a segment. The untextured area is matched as a large unit, so more information than as individual pixels can be gathered and improved performance can be obtained when processing images from a natural scene. Although impressive results are reported, only the segmentation information in the left (or reference) image is used and the occlusion result is still not accurate. Moreover, the violation of the discontinuity assumption still causes obvious artifacts in the result.

The second challenge in stereo correspondence is *occlusion* handling. Due to the structure of the scene, some parts of an object within it may be visible in only one of the

• Y. Deng and X. Lin are with Tsinghua University, 3-524, FIT Building, Beijing 100084, P.R. China.

E-mail: dengyi00@mails.tsinghua.edu.cn, lxy-dcs@mail.tsinghua.edu.cn.

• Q. Yang and X. Tang are with the Visual Computing Group, Microsoft Research Asia, Beijing Sigma Center No. 49, Zhichun Road, Hai Dian District, Beijing 100080, P.R. China.

E-mail: {qyang, xitang}@microsoft.com.

Manuscript received 13 Dec. 2005; revised 29 June 2006; accepted 6 Sept. 2006; published online 18 Jan. 2007.

Recommended for acceptance by M. Pollefeys.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0701-1205. Digital Object Identifier no. 10.1109/TPAMI.2007.1043.

cameras. These points are called *half-occlusion* points [13], and their projection onto the image is known as *occluded* points or *occlusion*, since their corresponding points in other images are not visible. The main difficulty for the occlusion problem is that occluded points cannot be detected directly, and we can only use the correspondence of *visible* (opposite of occluded) points with other assumptions to detect them. Methods using *ordering* and *uniqueness* constraints are two traditional ones for occlusion handling. The *ordering* constraint inhibits the ordering change of corresponding points in different images. It is often used in a dynamic programming framework [13] because it can reduce the solution space and allow for a more efficient algorithm. But, it is often violated when thin, front objects exist in the scene. The *uniqueness* constraint, however, only prevents a point in one image from being matched with more than one point in the other image and ordering change is allowed. Zitnic and Kanade [14] used the uniqueness as the inhibition in their cooperative framework, while Ishikawa and Geiger [3] imposed it in a max-flow framework. Kolmogorov and Zabih [6] used the pixel assignment formulation for the correspondence problem and tried to find an optically *unique* configuration using graph-cuts. Sun et al. [15] used a variant version of uniqueness constraint, the *visibility* constraint, to detect occlusions in an iterative belief propagation framework. The visibility constraint can avoid some problems raised from the sampling problem pointed out by Ogale and Aloimonos [16] when horizontally slanted planes exist in the scene. Promising improvements on occlusion results are reported in the above papers. For other occlusion handling techniques, readers can refer to surveys by Egnal and Wildes [13] and by Brown et al. [17].

The method proposed in this paper is an improvement to our previous work in [18]. Our motivation is to use segmentation information to improve the occlusion results. Under the observation that the discontinuity in one image corresponds to the occlusion border in the other [13], we use the border of the segment in one image to “cut-out” the potential occlusion area in the other image. Here, we take the advantage of the assumption used in segment-based approaches, which assumes that discontinuity only happens at the boundaries of a segment separated by a color segmentation algorithm. We call this technique the *discontinuity* assumption. Both images are first segmented and the segment of one image is warped to the other by an assumed disparity. We then introduce the concept of a *patch*. The warped segment is divided into several small patches by the segment boundaries in the other image. We preserve the uniqueness in a segment level and constrain the boundaries of occlusions to be the boundaries of patches. A symmetric global framework using graph cuts is constructed to find the disparity and occlusions embodied by the patch segmentation. The new correspondence approach gives a proper constraint for occlusions, which leads to better results near occlusions and inherits the advantage of segment-based approaches on slanted planes, as well as untextured areas and areas near discontinuity. To alleviate the effect of violating the discontinuity assumption, we propose two new methods, called soft-constraints and resegmenting, along with the basic algorithm in this paper. We also show some performance improvements that have been found in our experiments.

The rest of the paper is organized in the following way: Section 2 presents the concept of a patch for occlusion

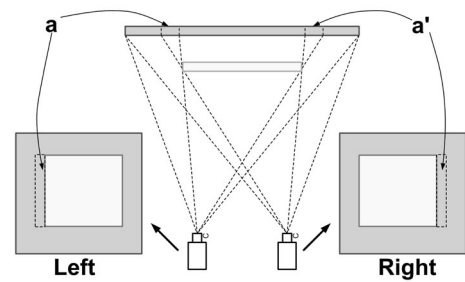


Fig. 1. A simple situation with two parallel planes in the scene. The images captured by cameras are shown on the left and the right to the cameras, respectively. Occluded areas are enclosed by dashed lines.

handling. The energy minimization framework with α -expansion and graph-cuts we use is provided in Section 3. Section 4 shows further improvements on the basic algorithm. Implementation details are given in Section 5. We provide experimental results and analysis in Section 6 and some comparisons of our algorithm and some others in Section 7. Finally, we conclude with Section 8.

2 OCCLUSION HANDLING FOR SEGMENTS

2.1 Occlusion and Discontinuity

We start to describe our idea by analyzing a simple situation shown in Fig. 1.

The two parallel planes form a typical scene with occlusion and discontinuity. The areas a and a' are visible in just one of the cameras, respectively. Their projections to the corresponding cameras where they are visible are called *occlusion areas* enclosed by dashed lines. One phenomenon we want to point out is that the projection of the left border of a on the right camera is a border separating the images of two planes with different disparities, i.e., it is a *discontinuity* border. A similar situation happens for a' . Using the *discontinuity assumption*, we can find this discontinuity border in the right image by color segmentation. However, in the left image, we may not segment out the corresponding occlusion area. Our idea is to use the segmentation results of the right image to help in segmenting the occlusion area in the left image and vice versa. The relationship between discontinuity and occlusion borders was also mentioned in some early occlusion handling approaches [13], but color segmentation information was not used.

2.2 Definition of Patch

Since part of a segment in one image may be invisible in the other image, the segment should be further separated into smaller units, which is defined as a *patch*. The points in a segment can have no more than one disparity and the visibilities of points are constrained to be the same within a patch.

The concept of a patch is further illustrated by an example here. Two input images are first separated into several nonoverlapping segments by color segmentation. We consider an arbitrary segment in the left image, denoted as r . Supposing its disparity is d , we can warp every point in r into the right image. The warped points, however, may overlay on several different segments in the right image, e.g., two segments s' and t' in Fig. 2. We can then define two patches q_1 and q_2 in r according to whether the points within it are warped to s' or t' . The shared border between q_1 and q_2 (i.e., e)

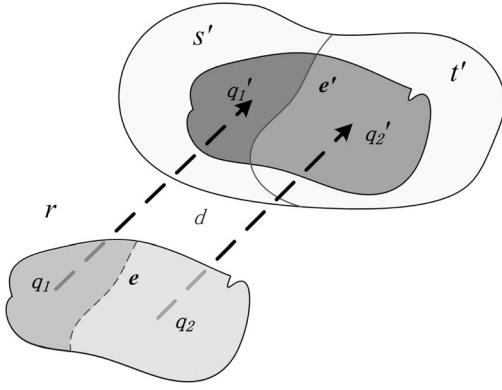


Fig. 2. This figure shows the definition of a patch. The segment r in the left image is warped to the right image by its disparity d , overlapping segment s' and t' . r is thus segmented into q_1 and q_2 , which are defined to be two patches. The shared border e' between s' and t' , which is a potential discontinuity border, is thus warped to the shared border e between q_1 and q_2 , which is a potential occlusion border. q_1' and q_2' are also symmetrically defined as two patches. See the text for more details.

corresponds to part of the shared border between s' and t' (e'). Since e' is a potential discontinuity edge, e is a potential occlusion border, i.e., q_1 may be visible when q_2 is occluded, or on the contrary.

Note that the concept of patch is defined with an assumed disparity (d for r in the above example), so it is not simply a smaller segment, but a region-version of the assignment in Kolmogorov and Zabih's model [6]. In their job, an assignment is defined under an assumed disparity for a single pixel.

2.3 Uniqueness within a Segment

Uniqueness was used to detect occlusions in several publications [14], [6], [3]. It claims that each point in the image, either the left or the right one, has no more than one correspondence in the other image. This is equivalent to saying that each point in the images can have no more than one disparity.

If images are segmented and represented by segments, we hope the uniqueness assumption is held in the segment level. As mentioned in the previous section, however, the visibility of points within a segment may be different. The patch thus defined, however, can solve this problem, i.e., assuming each patch as a unit is either fully visible or invisible.

2.4 Formulation of the Correspondence Problem in a Patch Level

Suppose the input images are denoted as I_L and I_R , respectively. The color segmentation will segment each image into a set of nonoverlapping segments, denoted as $S_L = \{s(i) | s(i) \subset I_L, \forall i \neq j, s(i) \cap s(j) = \emptyset \text{ and } \bigcup_i s(i) = I_L\}$ and similar S_R . The set of all segments is denoted as $S = S_L \cup S_R$. The goal of stereo correspondence is to decide the disparity of each segment and the visibility of the different parts in each segment. In other words, to decide the componential patches of each segment whenever it is partially occluded. We denote the disparity of a segment s as $d_s \in \mathcal{D}$, where \mathcal{D} is the set of all possible disparities. Being fully occluded is a special disparity in \mathcal{D} denoted as φ . Like other segment-based approaches [7], [8], each of the elements in \mathcal{D}

(except φ) is a set of disparity plane parameter $\langle c_1, c_2, c_3 \rangle$, i.e., the disparity of a pixel (x, y) is

$$d = c_1x + c_2y + c_3.$$

The i th patch of a segment s under disparity d_s is denoted as $p_s^{d_s}(i)$, $i = 1 \dots N_s^{d_s}$, where $N_s^{d_s}$ is the number of patches of s under disparity d_s .¹ An occlusion variable $o_s^{d_s}(i)$ for each patch is used to indicate whether this patch is occluded in the other image. If $o_s^{d_s}(i)$ equals 1, the i th patch is occluded, otherwise it is visible.² Thus, the whole correspondence problem can be formulated as computing an optimal configuration f , which includes the disparity of each segment and visibility of each of the patches, i.e.,

$$f = \{D_L, O_L, D_R, O_R\},$$

where $D_L = \{d_s | s \in S_L\}$, $O_L = \{o_s^{d_s}(i) | s \in S_L, i = 1 \dots N_s^{d_s}\}$ and D_R and O_R are similarly defined.

All the parts of quadruple f are not independent because the disparity of the left and right image must be consistent with each other. For example, if a point p in the left image has a disparity of d_s , its corresponding point p' in the right image needs to share the same disparity. This consistency requirement induces the uniqueness constraints, so we call each consistent quadruple f a *unique* configuration.

3 ENERGY MINIMIZATION FRAMEWORK

3.1 Energy Function

The optimal configuration is computed under an energy minimization framework, i.e.,

$$f_{opt} = \arg \min_f E(f),$$

where $E(f)$ is the energy function. In order to prohibit a nonunique configuration, we design an energy function, which takes finite energy only if its argument is a unique configuration. The energy function is in the form of:

$$E(f) = \begin{cases} E_{data}(f) + E_{smooth}(f) + E_{occl}(f) & f \text{ is unique,} \\ \infty & \text{otherwise.} \end{cases}$$

$E_{data}(f)$ is the energy of matching errors for all visible points. It is the sum of the matching errors for all visible patches:

$$E_{data}(f) = \sum_{s \in S, d_s \neq \varphi} \sum_{i=1}^{N_s^{d_s}} (1 - o_s^{d_s}(i)) \epsilon_{patch}(p_s^{d_s}(i)),$$

where $\epsilon_{patch}(p_s^{d_s}(i))$ is the matching errors of $p_s^{d_s}(i)$.

$E_{smooth}(f)$ imposes the smoothness constraint. If two neighboring patches have different disparities, we impose a nonzero penalty. Since selection of this energy function affects whether the energy can be efficiently minimized by graph-cuts, we will give its definition later.

$E_{occl}(f)$ is introduced to penalize occluded points, in case of the trivial configuration with all segments occluded that

1. For occluded segments, i.e., $d_s = \varphi$, we define the whole segment as one patch and, thus, $N_s^\varphi = 1$.

2. For convenience of expression and implementation, we also define an occlusion variable for a fully occluded segment ($d_s = \varphi$), but the value of the occlusion variable is meaningless.

- 1) Start with an initial configuration $f = \{\forall s \in S, d_s = \varphi\}$ (i.e. all segments are occluded).
- 2) Set success $\leftarrow 0$
- 3) For each disparity $d \in \mathcal{D}$
 - a) Find $\hat{f} = \arg \min E(f')$ among f' within one α -expansion of f
 - b) If $E(\hat{f}) < E(f)$, set $f \leftarrow \hat{f}$ and success $\leftarrow 1$
- 4) If success = 1 goto 2
- 5) Return f

Fig. 3. The framework of the α -expansion algorithm [4] used in our problem.

will minimize the whole energy function. Its value is proportional to the number of occluded points and is defined as:

$$E_{occl}(f) = C_o \sum_{s \in S} \sum_{i=1}^{N_s^{d_s}} T(d_s = \varphi \vee o_s^{d_s}(i) = 1) A(p_s^{d_s}(i)),$$

where $T(\cdot)$ is a function which takes value 1 if its argument holds, and 0 otherwise, and $A(p)$ is the number of points (area) of the patch p .

3.2 Minimizing with the α -Expansion Algorithm

The state space of f is huge, so direct minimization of $E(f)$ is intractable. Therefore, we use the α -expansion framework proposed by Boykov et al. [4] to obtain the approximated minimization. This is an iterative algorithm. In each iteration, a subspace near the solution of the last iteration is defined by α -expansion. The graph-cuts technique [19] can be used to give an exact optimal solution over this subspace and the total energy is guaranteed not to increase after each iteration. The final solution upon convergence is an approximation of the global optimization. Fig. 3 shows a detailed description of the α -expansion framework [4].

Each α expansion step for our problem is defined as follows: An α is a disparity chosen from \mathcal{D} . Segments (i.e., the variables) are classified as *active* ones and *inactive* ones. A segment s is called an *active* segment if its disparity before an expansion, denoted as \bar{d}_s , is neither α nor φ (fully occluded). Other segments are called *inactive* segments. In each α -expansion step, the disparity of each active segment can: 1) change to α , 2) keep unchanged as \bar{d}_s , or 3) become fully occluded (i.e., become φ). Only visibility of the patches of the inactive segment can change.³

3.3 α -Expansion Minimized by Graph-Cuts

The optimal configuration within an α expansion is computed using graph-cuts, but we need not directly design the graph for cutting. In Kolmogorov and Zabih's work [19], a general method of creating a graph for minimizing a *binary* function is proposed. All that is required is to rewrite the energy function to be minimized into a binary function constrained by α -expansion. The form of the energy function and the binary variables need to be well-considered so that the binary function obeys a required *regularity* condition [19].⁴

3. If all the patches of a segment become occluded, the segment becomes fully occluded. Likewise, if one of the patches of a segment which was fully occluded before expansion becomes visible, this segment's disparity becomes α .

4. This condition can also be represented as the *submodular* condition in combinatorial optimization area, as noted in [19].

For active and inactive segments, we design binary variables as follows:

1. For an *inactive* segment s , we use the *occlusion* variable for each patch defined under disparity α , denoted as $o_s^\alpha(i)$, which has the same meaning as the occlusion variable defined above.
2. For an *active* segment s , the situation is more complicated. Three kinds of variables are defined:
 - a. A *holding* variable, denoted as l_s , is used for each segment. If in the optimal configuration, l_s takes value 0, we change the disparity of s into α and otherwise keep it unchanged.
 - b. A set of *occlusion* variables $o_s^\alpha(i) (i = 1, \dots, N_s^\alpha)$, corresponding to each patch of s under disparity α , are used. If $o_s^\alpha(i)$ equals one, the i th patch of s under disparity α is occluded and otherwise visible.
 - c. A set of *visibility* variables $v_s^{\bar{d}_s}(i) (i = 1, \dots, N_s^{\bar{d}_s})$, corresponding to each patch under the disparity of s before α -expansion, are used. If $v_s^{\bar{d}_s}(i)$ equals one, the i th patch of s is visible and otherwise occluded.

We denote the set of all holding variables as L , the set of all occlusion variables as O , all visibility variables as V , and $F = \{L, O, V\}$ as a triplet including the above three sets. We call an F to be *legal*, only if, for each active segment s , both of the following conditions hold:

1. if $l_s = 1$, $\forall i, o_s^\alpha(i) = 1$, and
2. if $l_s = 0$, $\forall i, v_s^{\bar{d}_s}(i) = 0$.

In the first situation, the disparity of s holds (not equal to α), thus the occlusion variables for patches under disparity α need to take the value 1. In the second situation, the disparity of s changes to α and, thus, the visibility variables for patches under disparity \bar{d}_s need to take the value 0.

Within the α -expansion space, each legal F corresponds to a unique disparity configuration, denoted as $f'(F)$. Thus, an α -expansion step becomes the minimization of a binary function:

$$\begin{aligned} \hat{f} &= \arg \min_f E(f') \\ &= f' \left(\arg \min_F E(f'(F)) \right) \\ &= f' \left(\arg \min_F E^b(F) \right), \end{aligned}$$

where f' ranges within the α -expansion, and $E^b(F)$ is defined as:

$$E^b(F) = \begin{cases} E(f'(F)) & \text{for legal } F \\ \infty & \text{otherwise.} \end{cases}$$

We can separate $E^b(F)$ into four parts:

$$E^b(F) = E_{legal}^b(F) + E_{data}^b(F) + E_{smooth}^b(F) + E_{occl}^b(F).$$

$E_{legal}^b(F)$ takes the value of zero if F is legal and the value of infinity otherwise. We can directly write its definition as:

$$E_{legal}^b(F) = \sum_{s \in S} \sum_i T_{01}(l_s, o_s^\alpha(i)) + \sum_{active\ s} \sum_i T_{01}(v_s(i), l_s), \quad (1)$$

where $T_{01}(X, Y)$ is a 01-banning function, which takes the values of infinity if $X = 0$ and $Y = 1$ and the zero value for other inputs.

The data energy can be trivially achieved from $E_{data}(f)$ as:

$$E_{data}^b(F) = \sum_{s \in S} \sum_i (1 - o_s^\alpha(i)) \epsilon_{patch}(p_s^\alpha(i), \alpha) \\ + \sum_{active\ s} \sum_i v_s(i) \epsilon_{patch}(p_s^\alpha(i), \bar{d}_s).$$

The occlusion energy is defined as:

$$E_{occl}^b(F) = \sum_{inactive\ s} e_{inact}(F, s) + \sum_{active\ s} e_{act}(F, s), \\ e_{inact}(F, s) = \sum_i o_s^\alpha(i) A(p_s^\alpha), \\ e_{act}(F, s) = \sum_i T_{01}(v_s(i), l_s) A(p_s^{\bar{d}}) \\ + \sum_i T_{01}(l_s, o_s(i)) A(p_s^\alpha), \quad (2)$$

where $T_{01}(X, Y)$ returns 1 if $X = 0$ and $Y = 1$, and 0 otherwise.

$E_{smooth}^b(F)$ is chosen to be a function similar to what is used in [6], which punishes neighboring patches sharing the same disparity but with different visibility:

$$E_{smooth}^b(F) = C_s \sum_{p^\alpha} \sum_{p' \in \mathcal{N}_p} L(p, p') T(o_p \neq o_{p'}) \\ + C_s \sum_{p_s^{\bar{d}}} \sum_{p' \in \mathcal{N}_p} L(p, p') T(v_p \neq v_{p'}), \quad (3)$$

where \mathcal{N}_p is the set of neighboring patches with the same disparities with p (they may belong to a neighboring segment), and $L(p, p')$ is the length of the shared border between p and p' . It is not difficult to prove that this energy function on binary variables leads to the smoothness energy function $E_{smooth}(f)$ as follows:

$$E_{smooth}(f) = L(p, p') \begin{cases} 0 & d_p = d_{p'} \\ C_s & d_p = \varphi \vee d_{p'} = \varphi \\ 2C_s & otherwise. \end{cases}$$

We can see that all terms of $E^b(F)$ are at most of two variables, i.e., it is in the following form:

$$E^b(F) = \sum_i E^i(x_i) + \sum_{i < j} E^{i,j}(x_i, x_j)$$

and, for each of the pairs of energy terms⁵ $E^{i,j}(x_i, x_j)$,

$$E^{i,j}(0, 0) = E^{i,j}(1, 1) = 0, \\ E^{i,j}(0, 1) \geq 0, E^{i,j}(1, 0) \geq 0,$$

so the following *regularity* condition required in [19] holds:

$$E^{i,j}(0, 0) + E^{i,j}(1, 1) \leq E^{i,j}(0, 1) + E^{i,j}(1, 0).$$

We can then use graph-cuts to minimize $E^b(F)$.

5. They are T_{01} in (1), T_{01} in (2), and $T(o_p \neq o_{p'})$ and $T(v_p \neq v_{p'})$ in (3).

4 FURTHER IMPROVEMENTS

4.1 Soft Constraints

To generally satisfy the discontinuity assumption, an over-segmenting result is preferred. But, the small size of regions may affect the accuracy of untextured areas. To trade off this contradiction, we segment the reference image in two levels: the coarse and fine levels. Patches of fine levels are used as the matching units and the smoothness coefficients are controlled by the relation of the fine and coarse segmentation. If two segments in the fine level are originated to the same segment in the coarse level, we impose stronger smoothness constraints on patches between these two segments so that they have a bias of choosing the same disparity and vice versa.

Suppose the coarse and fine segmentation results of the left image are S_L^c and S_L^f , respectively. We give each of the segments an index and denote the indices of each pixel x of the segment in two levels as $l^c(x)$ and $l^f(x)$, respectively. We give an empirical formula for the smoothness coefficient, i.e., C_s in (3), between two neighboring patches p_1 and p_2 (with the same disparity) at a fine level as:

$$C_s(p_1, p_2) = C_s^0 \left(1 + \lambda \frac{|s_1^*| + |s_2^*|}{|s_1| + |s_2|} \right), p_1 \subseteq s_1, p_2 \subseteq s_2,$$

where C_s^0 is a smoothness constant, s_1 and s_2 are segments in fine level to which the two patches belong (note that s_1 and s_2 may be the same segment), λ is a scalar constant balancing the intension of the soft constraint (in our experiment, it is set to 1.4), s_1^* is the portion of pixels in s_1 which shares the same segment in coarse level with s_2 , i.e.,

$$s_1^* = \{x_1 \in s_1 | \exists x_2 \in s_2, l^c(x_1) = l^c(x_2)\}.$$

s_2^* is similarly defined.

4.2 Resegment Strategy

One common problem of all segment-based approaches is the violation of the *discontinuity* assumption. This assumption assumes that discontinuity always generates a visual border in the captured images. However, if two objects with different ranges contain similar colors, the discontinuity in the image may not be visually distinguishable, i.e., some segment may contain more than one disparity corresponding to the different ranges of the objects. We call this kind of segment a *mixed segment*. This problem is not easy to solve, whereas we provide here a method to remedy some of the situations when the discontinuity border, which cannot be seen in one image, is visible in the other image. This situation can be found when the occluded object is thin or is only occluded from one viewpoint near its bounds so that, from another viewpoint, it is not occluded anymore.

Our idea is based on this observation: Many of the mixed segments contain two kinds of pixels, the nearer and the farther ones, and after the α -expansion iterations are over, one of the disparities is assigned to the mixed segment. Some of the points that are of different disparities from the one assigned to the segment will be set to occlusion. So, if we extract the occluded parts from the entire mixed segment and make it a new segment, we expect that a new matching

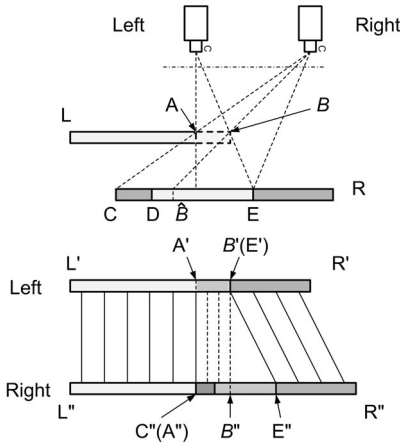


Fig. 4. The matching of a mixed segment. The upper part is the structure of the nearer and the farther objects. The lower part is the images captured by the left and right camera, respectively. For distinguishing, the projection of the points in the left and right image is prefixed by primes and double primes, respectively (e.g., A' is the project of A in the left image, and A'' is the project in the right one, etc.). Please see the main text for more details.

process can make the correct disparity assigned to the new segment.

We first illustrate the situation in which the points with different disparities will be considered to be occluded. Fig. 4 shows a portion of a scene in which a discontinuity is evident in the left image and a corresponding occlusion occurs in the right one. The segment LA is a nearer segment and CR is a farther one. If the colors of LA and DE are so similar that we cannot distinguish the projection of A in the left image, $L'E'$ becomes a mixed segment. In the right image, if CD contains different colors from LA , C'' can be identified as a segment border by color segmentation. Then, $L'A'$ and $A'B'$ can be separated into two patches of the segment $L'B'$ corresponding to the disparity of the nearer plane. Suppose the area of $L'A'$ (in the 2D image) is so much larger than $A'B'$ that its disparity is correctly assigned in the optimized configuration, and the disparity of $E'R'$ is also correctly assigned. We now analyze the visibility of $A'B'$ by analyzing the energies under two situations:

1. If $A'B'$ is *occluded*, the energy is

$$E_1 = E_{occl}(A'B') + C_s L(A') + C_s L(B') + E_{rest}, \quad (4)$$

where $L(A')$ and $L(B')$ are the lengths of the borders, and E_{rest} is the energy of other parts.

2. If $A'B'$ is *not occluded*, its disparity is assigned to be the same as $L'A'$, and it is matched with the segment $C'D'$. The energy is thus:

$$E_2 = E_{data}(A'B', C''B'') + 2C_s L(B') + E_{rest}. \quad (5)$$

Therefore, we can see that, when the matching of $A'B'$ to $C''B''$ contains a matching error so large that the following condition holds:

$$E_{data}(A'B', C''B'') > E_{occl}(A'B') + C_s(L(A') - L(B')),$$

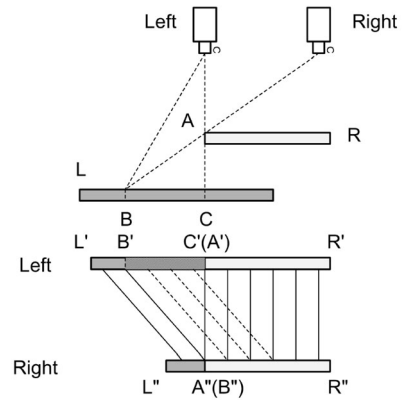


Fig. 5. Real occlusion. Unlike Fig. 4, the occlusion patch $B'C'$ will be warped to an area within $B'R''$ which is a visible segment, because the corresponding surface AR in the scene is in front of the surface BC and occludes the latter from the viewpoint of right camera.

E_1 in (4) has lower energy. Then, $A'B'$ will be considered to be occluded. Our goal is to identify this kind of occlusion area and separate $A'B'$ into a new segment. In the second α -expansion iteration, the new segment $A'B'$ is no longer constrained to have the same disparity as $L'A'$ and may be matched correctly.

In the example we just analyzed, if $A'B'$ is considered to be occluded, the corresponding patch $C''B''$ is also considered to be occluded. So, we call the occlusion patch $A'B'$ a *fake* occlusion because it is not really occluded by some nearer object in the scene. Opposite to the *fake* occlusion, we call those occlusion areas *real* occlusion when they are really occluded by some nearer parts. An example of such a situation is shown in Fig. 5. If we warp the points of patch $B'C'$ to the right image by the disparity of the segment $L'A'$, they will fall in the segment $A''R''$, which is unoccluded. The reason is that the surface BC is occluded by the nearer surface AR from the right camera's viewpoint. From the analysis above, by checking the visibility of warped points, we can identify the fake occlusion points and separate them into a new segment (we call *resegment* operation). After this, a similar algorithm is performed on the new segment sets and gives the final result.

5 IMPLEMENTATION

5.1 Algorithm Flow

The flowchart of the whole correspondence algorithm is shown in Fig. 6. The input is two images captured by the left and right cameras, respectively (Fig. 7a and Fig. 7b). Our algorithm first segments the left image at a coarse level using a mean-shift segmentation algorithm [20] (Fig. 7c). A trimmed Sum-of-Absolute-Difference (SAD) algorithm with Birchfield and Tomasi's dissimilarity algorithm [21] plus the cross-checking algorithm is used to find disparities of reliable points [8] (Fig. 7d). A robust plane fitting, like M-estimate [22], is performed on reliable points to find possible disparity planes \mathcal{D} similar to [8]. Input images are then segmented using the same algorithm at a fine level (Fig. 7e and Fig. 7f). The symmetric patch-based algorithm is used to compute an

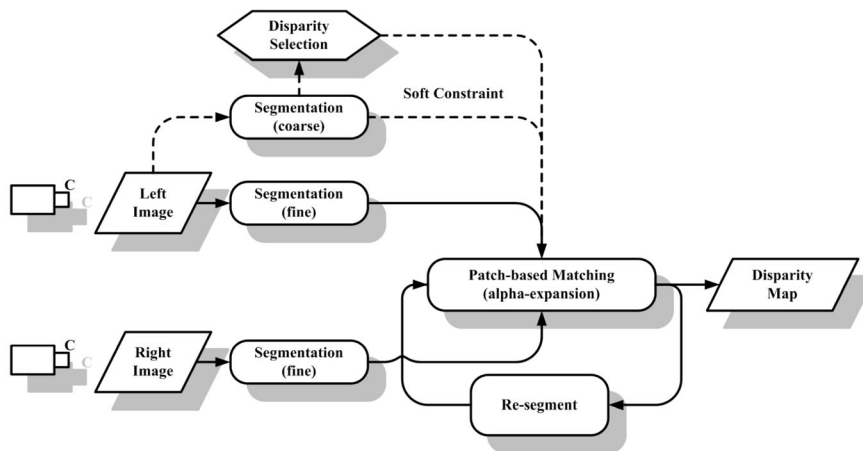


Fig. 6. The flowchart of the whole patch-based stereo correspondence algorithm.

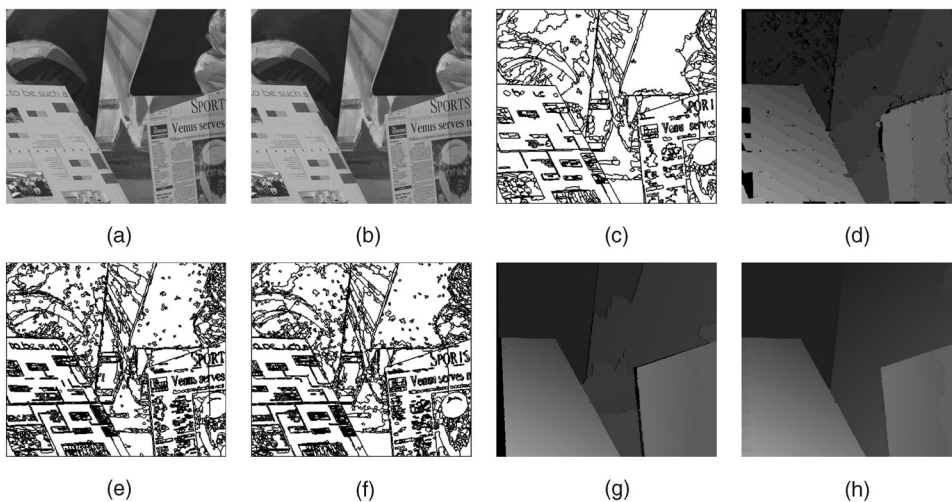


Fig. 7. Intermediate results of our algorithm for the “Venus” image pair. (a) and (b) are left and right input images, respectively. (c) is the segmentation results of the left image at a coarse level for labeling selection and (d) is the labeling selection results. (e) and (f) are segmentation results of both images at a fine level. (g) and (h) are disparity maps with and without occlusions.

optimal solution under the soft constraints. Then, a resegmenting algorithm is performed to cut some mixed segments into smaller segments, and the α -expansion minimization is rerun. The final results with and without occlusions can be obtained (Fig. 7g and Fig. 7h).

5.2 Parameter Selection

We find that two parameters in our algorithm are somehow sensitive to input images. They are the smoothness constant C_s and occlusion constant C_o . We propose a method to automatically select the values so that our algorithm can be adaptive to the input images.

Our selection strategy is designed with the following analysis: The correctly matched patch pairs contain the least SAD error in a noise-free situation. So, they can be selected by minimizing the data error energy. Noise may cause a wrong patch to have smaller matching energy than the correct one, but the wrong patch is often inconsistent with its neighbors. Smoothness energy is used to punish the inconsistency and reject the incorrect match. So, when there is greater noise, a stronger smoothness energy is needed and vice versa.

The noise level is estimated using the disparity map of reliable points in the label selection step. After cross-checking to the local matching of both images, we find some relatively reliable disparities. For each such reliable point, we compute a matching error ϵ and take the average of all matching errors $\bar{\epsilon}$ as the average noise level. C_s and C_o are set by values proportional to $\bar{\epsilon}$.

6 EXPERIMENTAL RESULTS

6.1 Occlusion Results

To evaluate the performance of detecting occlusion and the effectiveness of modeling the occlusion of our algorithm, we first compare the occlusion result with several recent approaches that handle occlusions or use segmentation information:

1. “GC + occl” algorithm by Kolmogorov and Zabih [6]: a pixel-based approach using a uniqueness and a symmetric graph-cuts framework to handle occlusion,
2. “Layer” algorithm by Lin and Tomasi [23]: a combination of pixel-based and segment-based approaches,

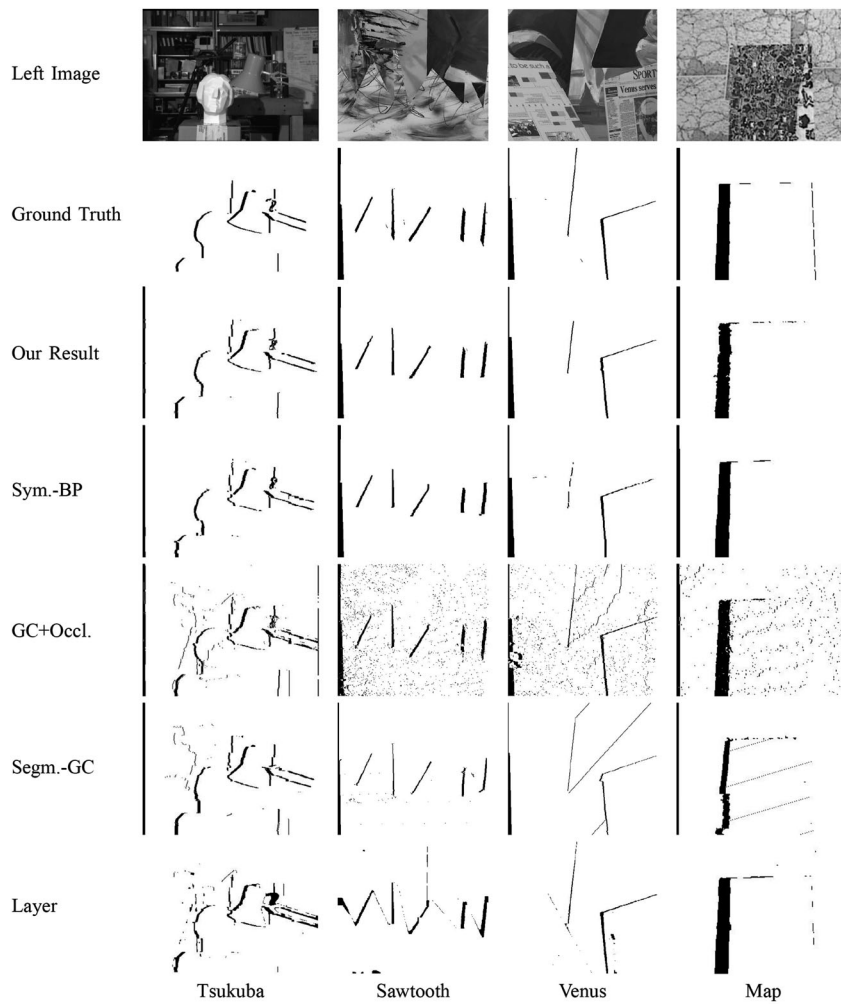


Fig. 8. Ground truth of occlusions and the results of several algorithms.

3. “Seg + GC” algorithm by Hong and Chen [8]: a segment-based asymmetric graph-cut approach that does not explicitly detect occlusion, and
4. a recently published “Sym-BP” approach by Sun et al. [15]: a pixel-based approach using visibility constraints for occlusion detection and imposing segmentation in a soft way.

Four image pairs of the benchmark site by Scharstein and Szeliski [1], [24], named “Tsukuba,” “Sawtooth,” “Venus,” and “Map,” respectively, are used. Same parameters are selected for all pairs. We use the source code by Kolmogorov to compute results of “GC+occl.” The result of “Layer” is from the authors’ Web site [25]. The results are shown and compared in Fig. 8. Table 1 gives error statistics for the four respective pairs. They are quantitatively evaluated by three criteria, which are the percentages of: *false positive*, *false negative*, and bad points *near occlusion*. A bad point is a point whose absolute disparity error is greater than one [1]. We make a near occlusion mask by dilating the occlusion area to 10 pixels and excluding the occlusion area.

From Table 1, we can see that, our result is slightly better than the *Sym.-BP* algorithm and obviously better than the others in the first three image pairs. The result of “Map” is

worse than *Sym.-BP*, but is still better than the others. Most errors on *map* lie on the boundary of the nearer plane because of the violation of the discontinuity constraint. In the scene of “Map” images, the border of the nearer image has similar colors to the farther one, which leads to a number of mixed segments. From Fig. 8, we can see that the border of our detected occlusion is much cleaner than those of others and is more similar to the ground truth. The reason is that the border is cutting by the segment in the right image. This experiment shows the power of our patch-based approaches on occlusion handling.

6.2 Disparity Results

We use the benchmark for dense two-frame stereo algorithms [1] to compare our overall performance with other algorithms. There are two versions of a stereo evaluation list. In the first version [24], the four image pairs used in the occlusion experiments are used. Three criteria are used for each image pair. They include the percentages of bad points at: nonocclusion (marked by “all”), untextured (“untext.”), and the discontinuity (“disc.”) area.⁶ The list including our

6. An exception for “Map” is that almost all of the area of this image pair is textured, so no “untext.” evaluation is used for “Map.”

TABLE 1
Errors of Occlusions of Different Algorithms

	false positive	false negative	near occl.
<i>Tsukuba</i>			
OUR METHOD	0.63%	26.97%	4.17%
Sym.-BP [15]	0.70%	29.90%	6.17%
KZ [6]	1.50%	32.68%	6.39%
Segm.-GC [8]	1.19%	32.51%	7.72%
Layered [23]	1.77%	27.92%	13.45%
<i>Sawtooth</i>			
OUR METHOD	0.10%	16.57%	0.46%
Sym.-BP	0.20%	17.00%	1.64%
KZ	1.54%	25.77%	3.76%
Segm.-GC	0.26%	20.48%	2.40%
Layered	0.77%	37.74%	5.48%
<i>Venus</i>			
OUR METHOD	0.13%	18.31%	0.33%
Sym.-BP	0.20%	25.40%	1.50%
KZ	1.89%	32.90%	5.91%
Segm.-GC	0.55%	17.73%	0.67%
Layered	0.37%	50.63%	0.90%
<i>Map</i>			
OUR METHOD	0.36%	11.15%	2.45%
Sym.-BP	0.30%	8.70%	1.18%
KZ	1.44%	10.27%	6.31%
Segm.-GC	1.06%	15.42%	15.46%
Layered	0.51%	7.39%	3.55%

algorithm is shown in Table 2. Our result is obtained by setting the same parameters for all image pairs. The algorithm is ordered by the average rank over the ranks of all criteria.

In version 2 [26], “Tsukuba” and “Venus” are kept and two new image pairs, named “Teddy” and “Cones,” respectively, are appended. New images are more challenging because they contain more complicated structures and nonplanar surfaces. Error rates at untextured areas are no longer counted in the final rank. The error rate of all areas,

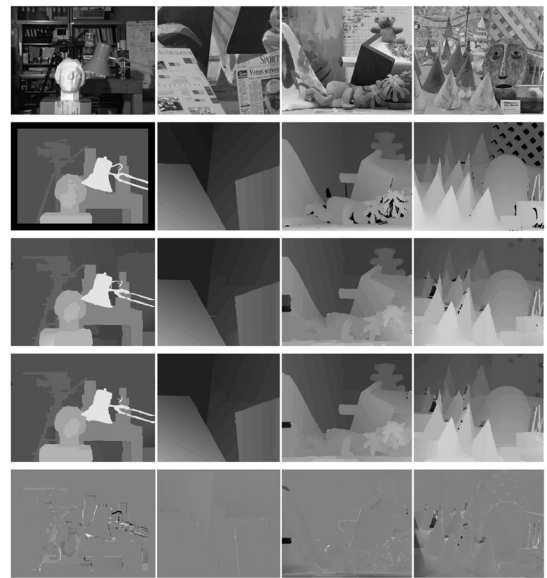


Fig. 9. Results of the middlebury data sets with automatically chosen parameters. The first row is the left image and the second row is the ground truth. The third and fourth rows are the results of our algorithm without resegment and after resegmenting, respectively. The bottom row is the signed disparity error map of our final results.

including pixels at the occlusion area, is used instead, and is marked as “all.” This new criteria contains the accuracy of the guessing to disparities at the occlusion area. The error rate at nonocclusion area is marked as “nonocc.” The error at the discontinuity is still used and marked as “disc.,” but the mask includes the area near occlusion. We set the same parameters for all the image pairs, whereas this parameter is different from the ones used in version 1. Disparity maps of our algorithm are shown in Fig. 9, including the results with and without resegmenting. Quantitative evaluations and rankings are shown in Table 3.

TABLE 2
Evaluation Results on the Middlebury Stereo Test-Bed: Version 1

Algorithm	Tsukuba			Sawtooth			Venus			Map	
	all	untext.	disc.	all	untext.	disc.	all	untext.	disc.	all	disc.
OUR METHOD	0.80 ₁	0.14 ₁	4.68 ₁	0.16 ₁	0.00 ₁	1.78 ₁	0.06 ₁	0.00 ₁	1.12 ₁	0.25 ₅	3.59 ₇
Sym.BP+occl	0.97 ₂	0.28 ₃	5.45 ₂	0.19 ₂	0.00 ₁	2.09 ₂	0.16 ₄	0.02 ₄	2.77 ₆	0.16 ₁	2.20 ₁
Segm.-based GC	1.23 ₅	0.29 ₅	6.94 ₆	0.30 ₆	0.00 ₁	3.24 ₆	0.08 ₂	0.01 ₂	1.39 ₂	1.49 ₂₄	15.46 ₂₉
Graph+segm.	1.39 ₉	0.28 ₃	7.17 ₈	0.25 ₅	0.00 ₁	2.56 ₄	0.11 ₃	0.02 ₃	2.04 ₃	2.35 ₂₉	20.87 ₃₃
Segm.+glob.vis.	1.30 ₇	0.48 ₈	7.50 ₁₀	0.20 ₃	0.00 ₁	2.30 ₃	0.79 ₇	0.81 ₈	6.37 ₁₁	1.63 ₂₆	16.07 ₃₁
Layered	1.58 ₁₂	1.06 ₁₄	8.82 ₁₃	0.34 ₇	0.00 ₁	3.35 ₇	1.52 ₁₅	2.96 ₂₅	2.62 ₅	0.37 ₁₂	5.24 ₁₂
Belief prop.	1.15 ₃	0.42 ₆	6.31 ₃	0.98 ₁₄	0.30 ₁₉	4.83 ₁₂	1.00 ₁₀	0.76 ₇	9.13 ₁₇	0.84 ₂₁	5.27 ₁₃
MultiCam GC	1.85 ₁₅	1.94 ₂₀	6.99 ₇	0.62 ₁₂	0.00 ₁	6.86 ₁₇	1.21 ₁₂	1.96 ₁₅	5.71 ₉	0.31 ₉	4.34 ₁₁
Region-Progress.	1.44 ₁₀	0.55 ₉	8.18 ₁₁	0.24 ₄	0.00 ₁	2.64 ₅	0.99 ₉	1.37 ₁₃	6.40 ₁₂	1.49 ₂₅	17.11 ₃₂
2-pass DP	1.53 ₁₁	0.66 ₁₀	8.25 ₁₂	0.61 ₁₀	0.02 ₁₀	5.25 ₁₃	0.94 ₈	0.95 ₉	5.72 ₁₀	0.70 ₁₉	9.32 ₂₀
GC+occl.	1.19 ₄	0.23 ₂	6.71 ₄	0.73 ₁₃	0.11 ₁₃	5.71 ₁₅	1.64 ₁₈	2.75 ₂₃	5.41 ₈	0.61 ₁₇	6.05 ₁₅
⋮											
Max. surf.	11.10 ₄₀	10.70 ₃₈	41.99 ₄₀	5.51 ₃₉	5.56 ₄₀	27.39 ₃₉	4.36 ₃₄	4.78 ₃₁	41.13 ₃₉	4.17 ₃₈	27.88 ₃₉

TABLE 3
Evaluation Results on Middlebury Stereo Test-Bed: Version 2

Algorithm	Ave. Rank	Tsukuba			Venus			Teddy			Cones		
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
Double-BP	1.5	0.88 ₁	1.29 ₁	4.76 ₁	0.14 ₂	0.60 ₃	2.00 ₂	3.55 ₁	8.71 ₂	9.70 ₁	2.90 ₁	9.24 ₂	7.80 ₁
Segm-visb	3.7	1.30 ₆	1.57 ₂	6.92 ₇	0.79 ₅	1.06 ₄	6.76 ₇	5.00 ₂	6.54 ₁	12.3 ₂	3.72 ₃	8.62 ₁	10.2 ₄
Sym.BP-occl	3.7	0.97 ₂	1.75 ₃	5.09 ₂	0.16 ₃	0.33 ₂	2.19 ₃	6.47 ₄	10.7 ₃	17.0 ₄	4.79 ₇	10.7 ₆	10.9 ₅
AdaptiveWeight	5.2	1.38 ₈	1.85 ₄	6.90 ₆	0.71 ₅	1.19 ₅	6.13 ₄	7.88 ₆	13.3 ₆	18.6 ₇	3.97 ₅	9.79 ₄	8.26 ₂
OUR METHOD	6.3	1.25 ₄	1.95 ₆	6.45 ₄	0.10 ₁	0.32 ₁	1.36 ₁	8.87 ₉	12.0 ₄	20.2 ₁₀	8.16 ₁₃	14.6 ₁₁	14.9 ₁₁
SemiGlob	6.8	3.26 ₁₃	3.96 ₁₁	12.8 ₁₆	1.00 ₆	1.57 ₆	11.3 ₁₁	6.02 ₃	12.2 ₅	16.3 ₃	3.06 ₂	9.75 ₃	8.90 ₃
Layered	8.3	1.57 ₉	1.87 ₅	8.28 ₉	1.34 ₈	1.85 ₇	6.85 ₈	8.64 ₈	14.3 ₇	18.5 ₆	6.59 ₁₁	14.7 ₁₂	14.4 ₁₀
GC-occ	8.4	1.19 ₃	2.01 ₈	6.24 ₃	1.64 ₁₁	2.19 ₁₀	6.75 ₆	11.2 ₁₂	17.4 ₁₂	19.8 ₉	5.36 ₉	12.4 ₉	13.0 ₉
MultiCamGC	9.2	1.27 ₅	1.99 ₇	6.48 ₅	2.79 ₁₅	3.13 ₁₃	3.60 ₄	12.0 ₁₃	17.6 ₁₃	22.0 ₁₂	4.89 ₈	11.8 ₈	12.1 ₇
TensorVoting	10.1	3.79 ₁₄	4.79 ₁₄	8.86 ₁₀	1.23 ₇	1.88 ₈	11.5 ₁₂	9.76 ₁₀	17.0 ₁₁	24.0 ₁₄	4.38 ₆	11.4 ₇	12.2 ₈
CostRelax	10.8	4.76 ₁₆	6.08 ₁₆	20.3 ₁₇	1.41 ₁₀	2.48 ₁₁	18.5 ₁₅	8.18 ₇	15.9 ₉	23.8 ₁₃	3.91 ₄	10.2 ₅	11.8 ₆
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Infection	19.1	7.95 ₂₀	9.54 ₂₀	28.9 ₂₀	4.41 ₁₇	5.53 ₁₇	31.7 ₁₉	17.7 ₁₉	25.1 ₁₉	44.4 ₂₀	14.3 ₂₀	21.3 ₁₉	38.0 ₂₀

From the result, we can see that our patch-based formulation inherits the advantage of segment-based approaches on discontinuity and untextured areas. When the scene can be estimated by several planar surfaces, like the image pairs in version 1, our algorithm provides outstanding results. Except the “Map” pair, our algorithm obtains all top criteria for the other image pairs. The untextured error is especially small. Even in version 2, our algorithm ranks at *fifth* position over 20 algorithms altogether (some results with low ranks are omitted in Table 3).

Like the result reported in our earlier work [18] and other segment-based algorithms [8], [9], the result of “Map” is still worse than others. The reason is that the colors of different disparities near discontinuity are similar. Our new result is better because of using the resegment framework, although the problem still needs more effort. There are also some artifacts in the results for “Teddy” and “Cones.” The reason is that some of the heavily slanted disparity planes are not fitted and the cone’s surfaces are not well-estimated by planes.

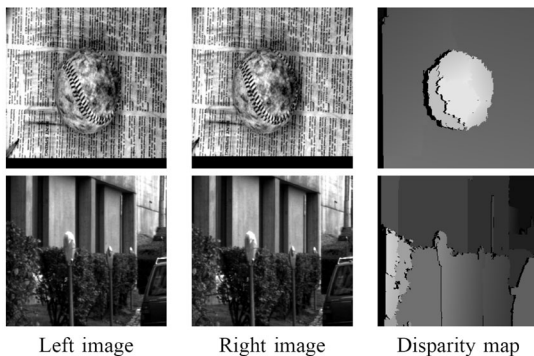


Fig. 10. Some other results (called “ball” and “parkmeter,” respectively) by our algorithm. Occlusion is shown in black.

6.3 Some Other Results

In order to further investigate the performance of the proposed algorithm, we employed it on some other data sets, and the results are shown in Fig. 10.

Some structures more complicated than planes appear in these new data sets, e.g., the sphere shape of the ball. Color differences in a corresponding area can also be found. Our algorithm gives sound results. But, some artifacts can be found at the boundary of the ball and the disparity of the park-meter can also be further improved.

7 COMPARISON WITH OTHER ALGORITHMS

In this section, we compare our algorithm and other correspondence approaches.

7.1 Disparity Space

We first consider the differences in disparity space. In most pixel-based algorithms, the search is directly performed on 1D disparity space. A piecewise smoothness assumption (explicitly or implicitly imposed) gives a bias of the same disparities for neighboring pixels. This affects the performance of those algorithms when greatly slanted planes exist in the scene. So, Ogale and Aloimonos [16] used a 2D linear parameter space for horizontally slanted planes, and most of the segment-based algorithms used a 3D disparity plane space which can model generally slanted planes. In Table 2, we find that the pixel-based algorithm can easily achieve sound results in “Tsukuba” and “Map,” which only contain fronto⁷ or near-fronto planes. Segment-based approaches perform better in “Sawtooth” and “Venus,” which contain horizontally (in “Venus”) and vertically (in both of them) slanted planes. But the 3D linear space is much larger than 1D disparity space, so a label selection algorithm is often used to select all possible linear parameters before matching.

7. A *fronto* plane is a plane orthogonal to the optical axis of the reference image.

Besides, using the segment as the matching unit can significantly reduce the scale of the state space. This is why segment-based approaches use larger 3D linear disparity plane parameter space than pixel-based approaches.

7.2 Use of Segmentation Information

Another comparison is between the different levels of segmentation information used. In most previous pixel-based algorithms, texture information is used to control the smoothness intention of neighboring points, e.g., in [6], the intensity difference between neighboring pixels is used to adjust the smoothness coefficient. This method can lead to some limited improvement on accuracy. The segment-based algorithms use the color segmentation results as a *hard* constraint for disparities. The points in a segment are considered as a single matching unit, and the number of matching units is greatly reduced. But, the discontinuity assumption is generally not correct and may sometimes cause errors. Sun et al. [15] used the segmentation information in a soft style. The fitted planes give a bias to pixel-based matching, and the results on "Sawtooth" and "Venus" are much better than other pixel-based algorithms. Their approach does not suffer from segmentation error directly, but the plane fitting is in a local style and the fronto bias still exists. Our approach belongs to the segment-based category, but we use more segmentation information for occlusion handling. The *soft constraints* and *resegmenting* method also alleviate the effect of violation on the discontinuity assumption.

8 CONCLUSION

We proposed a novel patch-based matching model and a correspondence algorithm using graph-cuts with occlusion handling. Unlike traditional segment-based approaches, both images are segmented and segmented regions are further split into a smaller unit, the patch. The occlusion and visible boundary are aligned with the edge of segments in the other image. Performance near occlusion is saliently improved over other segment-based approaches. The advantage of using segmentation information remains, which makes better results, especially in untextured areas and discontinuity for natural scenes with slanted planes. Two methods are proposed to increase the performance when the discontinuity assumption is violated. The experimental results demonstrate that our approach corresponds to the state-of-the-art.

ACKNOWLEDGMENTS

This work was performed at Microsoft Research Asia.

REFERENCES

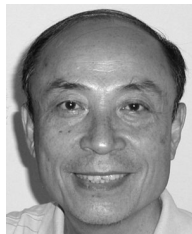
- [1] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *Int'l J. Computer Vision*, vol. 47, no. 1, pp. 7-42, May 2002.
- [2] S. Roy, "Stereo without Epipolar Lines: A Maximum-Flow Formulation," *Int'l J. Computer Vision*, vol. 34, nos. 2/3, pp. 147-161, 1999.
- [3] H. Ishikawa and D. Geiger, "Occlusions, Discontinuities, and Epipolar Lines in Stereo," *Proc. European Conf. Computer Vision*, pp. 232-249, 1998.
- [4] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization Via Graph Cuts," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 1, pp. 532-539, 2001.
- [5] 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.
- [6] V. Kolmogorov and R. Zabih, "Computing Visual Correspondence with Occlusions Using Graph Cuts," *Proc. IEEE Int'l Conf. Computer Vision*, 2001.
- [7] H. Tao, H.S. Sawhney, and R. Kumar, "A Global Matching Framework for Stereo Computation," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 1, pp. 532-539, 2001.
- [8] L. Hong and G. Chen, "Segment-Based Stereo Matching Using Graph Cuts," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, vol. 1, 2004.
- [9] Y. Wei and L. Quan, "Region-Based Progressive Stereo Matching," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 106-113, 2004.
- [10] M. Bleyer and M. Gelautz, "Graph-Based Surface Reconstruction from Stereo Pairs Using Image Segmentation," *Proc. SPIE*, vol. 5656, Jan. 2005.
- [11] Y. Zhang and C. Kambhampettu, "Stereo Matching with Segmentation-Based Cooperation," *Proc. European Conf. Computer Vision*, pp. 556-571, 2002.
- [12] C. Baillard and H. Maistre, "3-D Reconstruction of Urban Scenes from Aerial Stereo Imagery: A Focusing Strategy," *Computer Vision and Image Understanding*, vol. 76, no. 3, pp. 244-258, Dec. 1999.
- [13] G. Egnal and R.P. Wildes, "Detecting Binocular Half-Occlusions: Empirical Comparisons of Five Approaches," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1127-1133, Aug. 2002.
- [14] C.L. Zitnic 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.
- [15] J. Sun, Y. Li, S.-B. Kang, and H.-Y. Shum, "Symmetric Stereo Matching for Occlusion Handling," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 399-406, June 2005.
- [16] A.S. Ogale and Y. Aloimonos, "Stereo Correspondence with Slanted Surface: Critical Implication of Horizontal Slant," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 568-573, 2004.
- [17] 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.
- [18] Y. Deng, Q. Yang, X. Lin, and X. Tang, "A Symmetric Patch-Based Correspondence Model for Occlusion Handling," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 2, pp. 1316-1322, Oct. 2005.
- [19] V. Kolmogorov and R. Zabih, "What Energy Functions Can Be Minimized via Graph Cuts?" *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147-159, Feb. 2002.
- [20] C. Christoudias, B. Georgescu, and P. Meer, "Synergism in Low-Level Vision," *Proc. Int'l Conf. Pattern Recognition*, vol. 4, pp. 150-155, 2002.
- [21] 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.
- [22] C.V. Stewart, "Robust Parameter Estimation in Computer Vision," *SIAM Rev.*, vol. 41, no. 3, pp. 513-537, Sept. 1999.
- [23] M. Lin and C. Tomasi, "Surfaces with Occlusions from Layered Stereo," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1073-1078, Aug. 2004.
- [24] <http://bj.middlebury.edu/~schar/stereo/web/results.php>, Oct. 2005.
- [25] http://robotics.stanford.edu/~michelin/layered_stereo, Jan. 2003.
- [26] <http://benjerry.middlebury.edu/~schar/stereo/newEval/php/results.php>, Jan. 2007.



Yi Deng received the BS degree in computer science and technology from Tsinghua University in 2000. He is now a PhD candidate in the media lab of the Computer Science and Technology Department, Tsinghua University, Beijing, P.R. China. His current fields of research interest include stereo correspondence, robot navigation, image processing, and artificial intelligence. In these fields, he has published several papers in top conference proceedings on computer vision.



Qiong Yang received the PhD degree from Tsinghua University in 2004. Now, she is working at Microsoft Research Asia as an associate researcher. Her research interests include face recognition and verification, face search and image retrieval, image fusion/visualization, image decomposition/segmentation, visual tracking and recognition, machine learning, and interactive human-computer interfaces. She is a member of the IEEE.



Xueyin Lin received the BS degree in automatic control from Tsinghua University, Beijing, China, in 1962. He is a professor with the Department of Computer Science and Technology, Tsinghua University. He joined Tsinghua University in 1962 as a faculty member in the Department of Automatic Control and then went to the Department of Electronics. He was a visiting professor with the University of Cincinnati, Cincinnati, Ohio, from 1983 to 1985 and from 1993 to

1994, respectively. His main research interests are in the areas of computer vision, image processing, and pattern recognition. He has published more than 100 papers in these areas.



Xiaou Tang (S '93-M '96-SM '02) received the BS degree from the University of Science and Technology of China, Hefei, in 1990, and the MS degree from the University of Rochester, Rochester, New York, in 1991. He received the PhD degree from the Massachusetts Institute of Technology, Cambridge, in 1996. He is a professor in the Department of Information Engineering, the Chinese University of Hong Kong and the group manager of the Visual

Computing Group at the Microsoft Research Asia. He is a local chair of the IEEE International Conference on Computer Vision (ICCV) 2005, an area chair of CVPR '07, a program chair of ICCV '09, and a general chair of the IEEE ICCV International Workshop on Analysis and Modeling of Faces and Gestures 2005 and 2007. He was a guest editor for the IEEE *Journal of Oceanic Engineering* and *IEEE Transactions on Circuits and Systems for Video Technology*. He is an associate editor of *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* and *Pattern Recognition Journal*. His research interests include computer vision, pattern recognition, and video processing. He is a senior member of the IEEE.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.