

Improving Border Localization of Multi-Baseline Stereo Using Border-Cut

Marc-Antoine Drouin

Martin trudeau

Sébastien Roy

DIRO, Université de Montréal, Canada
{drouim,trudeaum,roys}@iro.umontreal.ca

Abstract

This paper presents a novel algorithm that improves the localization of disparity discontinuities of disparity maps obtained by multi-baseline stereo. Rather than associating a disparity label to every pixel of a disparity map, it associates a position to every disparity discontinuity. This formulation allows us to find an approximate solution to a 2D labeling problem with robust smoothing term by minimizing multiple 1D problems, thus making possible the use of dynamic programming. Dynamic programming allows the efficient computation of the visibility of most of the cameras during the minimization. Whilst the proposed minimization strategy is particularly suitable for stereo with occlusion, it may be used with other labeling problems.

1. Introduction

The goal of binocular stereo is to reconstruct the 3D structure of a scene from two views. Occlusion occurs when part of a scene is visible in the reference but not in the supporting camera. The difficulty of detecting occlusion comes from the fact that it is induced by the 3D structure of the scene, which is unknown until the correspondence is established. When extending binocular stereo to multiple cameras, the amount of occlusion increases since each pixel of the reference camera can be hidden in more than one supporting camera. This is particularly true when going from a single to a multiple-baseline configuration. We propose a novel algorithm that improves the localization of disparity discontinuities of disparity maps obtained by multi-baseline stereo. For some applications such as augmented reality, well-localized borders are indeed very important. The algorithm computes simultaneously the border localization and most of the visibility information. Heuristics may be required to establish camera visibility for a small subset of the cameras.

The rest of this paper is divided as follows: in Section 2, previous work is presented. Section 3 describes our algorithm. Visibility is discussed in Section 4. Experimental results are presented in Section 5.

2. Previous work

In Egnal [8], five basic strategies to overcome occlusion for two cameras are presented: left-right checking, bimodality test, goodness Jumps constraint, duality of depth discontinuity and occlusion, and uniqueness constraint. Some algorithms rely on one or more of these strategies, and are often based on varying a correlation window position or size [14, 10, 32, 15]. Other algorithms use dynamic programming [21, 12, 5]. Two methods using graphs [13, 16] have been proposed. In [29], visibility and disparity are iteratively minimized using belief propagation. Most of these methods are binocular in nature and do not generalize well to the case of multiple cameras. Some researchers have proposed specially designed algorithms to cope with occlusion in multiple camera configurations. They can be coarsely divided into three categories. Some approaches are based on the heuristic that a low matching cost function implies the absence of occlusion [15, 20, 25, 23]. Others guarantee a solution that is geo-consistent [7, 18, 28, 9, 17]. These approaches preserve the consistency between the recovered visibility and the geometry [7]. Finally, some algorithms are mixes between heuristic and geo-consistent algorithms [6, 34, 11].

Many discrete optimization methods have been applied to stereo [4, 19, 30]. Our minimization strategy is similar to [4, 19]. All these methods find iteratively a disparity for a subset of the pixels while keeping the disparity of the others fixed. Nevertheless, our approach does not work directly on the disparity map; it works on the localization of the discontinuities. The two main differences between our approach and active contour ones are the discrete formulation and the absence of constraints on the smoothing term [22].

3. Formulation

We have a set \mathcal{P} of reference pixels, for which we want to compute disparity, and a set \mathcal{D} of disparity labels. A \mathcal{D} -configuration $f : \mathcal{P} \mapsto \mathcal{D}$ associates a disparity label to every pixel. When occlusion is not modeled, the energy

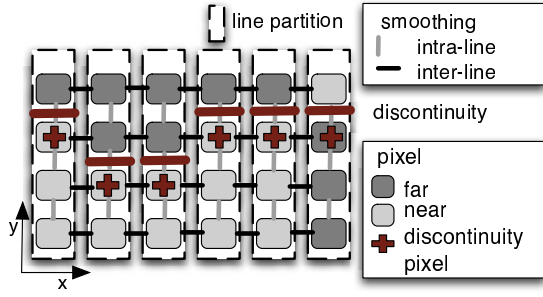


Figure 1. Representation of a disparity map with vertical line partitioning.

function to minimize is

$$E(f) = \underbrace{\sum_{\mathbf{p} \in \mathcal{P}} e(\mathbf{p}, f(\mathbf{p}))}_{\text{pixel likelihood}} + \underbrace{\sum_{\mathbf{p} \in \mathcal{P}} \sum_{\mathbf{p}' \in \mathcal{N}_{\mathbf{p}}} s(\mathbf{p}, \mathbf{p}', f(\mathbf{p}), f(\mathbf{p}'))}_{\text{pixel smoothing}} \quad (1)$$

where $\mathcal{N}_{\mathbf{p}}$ is pixel \mathbf{p} 's neighborhood, which can be 4- or 8-connected. When the smoothing term is robust, the minimization of Eq. 1 is an NP-hard problem [4].

We partition the reference image into the set \mathcal{L} of its vertical or horizontal lines, where each line λ is a vector of pixels $(p_1^\lambda, \dots, p_N^\lambda)$, N being the number of pixels on each line. Two lines are neighbors if two of their pixels are. The neighborhood of line λ is denoted \mathcal{N}_λ and is 2-connected when $\mathcal{N}_{\mathbf{p}}$ is 4- or 8-connected. If there are only 2 different disparity values and a single disparity discontinuity in a line then there is only one *discontinuity* pixel p_d^λ in the line λ preceding a pixel with a different disparity (see Fig. 1). A \mathcal{B} -configuration $b : \mathcal{L} \mapsto \mathcal{B}$ associates a *discontinuity* pixel to every line. The set \mathcal{B} is simply $\{1, 2, \dots, N\}$ and represents the index of the *discontinuity* pixel. There are $N - 1$ possible discontinuity locations and it is possible for a line to have no discontinuity at all, which correspond to the label N .

We define \mathcal{E} to be the set containing the two end pixels of every line. If we have an endpoint disparity map f' that associates a disparity to every pixel in \mathcal{E} and a \mathcal{B} -configuration b , then we can define $F_{b, f'}$ to be the corresponding \mathcal{D} -configuration. Similarly, B_f is the \mathcal{B} -configuration associated with f (since f may have more than one discontinuity on one of its lines, B_f is not always well-defined). For an endpoint disparity map f' , allowing only one discontinuity

per line, the energy to minimize is

$$E_{f'}(b) = \underbrace{\sum_{\lambda \in \mathcal{L}} e_{f'}(\lambda, b(\lambda))}_{\text{discontinuity likelihood}} + \underbrace{\sum_{\lambda \in \mathcal{L}} \sum_{\lambda' \in \mathcal{N}_\lambda} s_{f'}(\lambda, \lambda', b(\lambda), b(\lambda'))}_{\text{inter-line smoothing}} \quad (2)$$

with respect to b , where $e_{f'}$ contains the likelihood of pixels on the same line and intra-line smoothing, i.e. the smoothing of neighboring pixels belonging to the same line (see Fig. 1). The inter-line smoothing (see Fig. 1) is the pixel smoothing of neighboring pixels belonging to different lines (see Fig. 1). Note that $E_{f'}(b) = E(F_{b, f'})$. The neighborhood used in Eq. 2 is 2-connected making the minimization easy assuming f' known and B_f well-defined. Standard stereo algorithms find sharp discontinuities but often they are not well localized in areas with occlusion [6]. Nevertheless, disparities on each side of these discontinuities are generally well recovered and provide reasonable endpoint disparity maps f' for the minimization of $E_{f'}(b)$ to yield a good global disparity map.

BORDER-CUT(f)

```

1  repeat
2      change ← 0
3      for  $\mathcal{R} \subset \mathcal{P}$  such that  $B_{f|\mathcal{R}}$  is well-defined
4      do Compute  $f'$  from  $f|\mathcal{R}$ 
5           $E \leftarrow \min_b [E_{f'}(b) + R_{\mathcal{R}}(f|\mathcal{R}^C, F_{b, f'})]$ 
6           $b^* \leftarrow$  the minimum in the above formula
7          if  $E < E_{f'}(B_{f|\mathcal{R}}) + R_{\mathcal{R}}(f|\mathcal{R}^C, f|\mathcal{R})$ 
8              then change ← 1
9          Update  $f$  using  $F_{b^*, f'}$ 
10     until change = 0
11     return  $f$ 

```

Figure 2. Overview of the Border-Cut algorithm

3.1. The Border Cut Algorithm

In a disparity map, there is generally more than one discontinuity on a line of pixels. Our algorithm does not work on whole lines but only on segments containing one *discontinuity* and centered around it. Two segments are neighbors if two of their pixels are and if they belong to different lines. A line sweeping strategy is used to choose the segments whose pixels will form the *active* subset \mathcal{R} of the set of pixels \mathcal{P} , for which the localization of borders will be improved. This is done while keeping the disparity of other pixels (called *passive*) fixed. The smoothing between *active* and *passive* pixels is denoted $R_{\mathcal{R}}(f|\mathcal{R}^C, F_{b, f'})$, where \mathcal{R}^C is the complement of \mathcal{R} , i.e. $\mathcal{P} - \mathcal{R}$. Let us describe this process for horizontal lines swept from left to right. We first pick a disparity δ and only consider *discontinuities* between a pixel with disparity smaller than δ and one with

disparity greater than or equal to δ . We call such a *discontinuity* a δ -discontinuity. We find the leftmost horizontal δ -discontinuity. We then recursively collect neighboring segments with δ -discontinuities to form \mathcal{R} , making sure there is at most one segment on each line. Having an initial disparity map f , we can minimize Eq. 2 on this set. This minimization, which we call *border move*, occurs on line 5 of the algorithm outlined in Fig. 2. The disparity map is then updated (line 9) and we start the above procedure again from the next leftmost unprocessed δ -discontinuity. This is repeated until all δ -discontinuities have been processed. This line sweeping strategy is also repeated in the other three directions with the same δ . Finally, this process is done for all values of δ .

We call the execution of the outer loop (line 1 to 10) a *cycle*. Note that each time we update f on line 9, the energy $E_{f'}$ decreases or remains the same. At each *cycle* before the last, the energy decreases by at least the minimum cost of changing the label of a pixel, thus ensuring convergence. In practice, convergence is obtained after only a few cycles. After the last, a local minimum with respect to a *border move* is found.

To compute the value of the *discontinuity* likelihood term for all possible discontinuity locations, we only need to scan all the pixels on a line once. Since only 2 pixels have their disparity changed when the discontinuity is moved by 1 pixel, the likelihood term e_f can be efficiently recursively computed. The same technique can be used for the computation of the smoothing term. Consequently, each *border move* is in $\Theta(\#\mathcal{L} \cdot N^2)$ when Eq. 2 is minimized using dynamic programming. In our tests, we used segments of up to 19 pixels.

The algorithm may remove disparity discontinuities, but never adds new ones. Discussion about initialization is postponed until section 5.1.

4. Visibility

To model occlusion, we must compute the volumetric visibility $V_c(\mathbf{p}, \delta, f)$ of a 3D reference point formed from the pixel \mathbf{p} at disparity δ from the point of view of a camera c , given a disparity configuration f . It is set to 1 if the point is visible, and 0 otherwise. The visibility information is collected into a vector, the *visibility mask*

$$V(\mathbf{p}, \delta, f) = (V_1(\mathbf{p}, \delta, f), \dots, V_C(\mathbf{p}, \delta, f))$$

where C is the number of cameras outside the reference; a vector $(1, \dots, 1)$ means that the 3D point is visible in all supporting cameras. We call \mathcal{M} the set of all possible visibility masks; an \mathcal{M} -configuration $g : \mathcal{P} \mapsto \mathcal{M}$ associates a mask to every pixel. The visibility masks are *correct* when g satisfies

$$g(\mathbf{p}) = V(\mathbf{p}, f(\mathbf{p}), f) \quad (3)$$

for all $\mathbf{p} \in \mathcal{P}$. As in [7], we transform Eq. 1 into an energy function with masks

$$E(f, g) = \sum_{\mathbf{p} \in \mathcal{P}} e(\mathbf{p}, f(\mathbf{p}), g(\mathbf{p})) + \text{smoothing}$$

and Eq. 2 into

$$E_{f'}(b, g) = \sum_{\lambda \in \mathcal{L}} e_{f'}(\lambda, b(\lambda), g|_{\lambda}) + \text{smoothing}. \quad (4)$$

Mutatis mutandis, our new $e_{f'}$ is the *discontinuity* likelihood term that uses the pixel likelihood with masks. Since we expect masks belonging to adjacent pixels to be spatially coherent, we can add a visibility smoothing term that penalizes the change of visibility status between adjacent pixels. However, no visibility smoothing was used in the results presented in this paper.

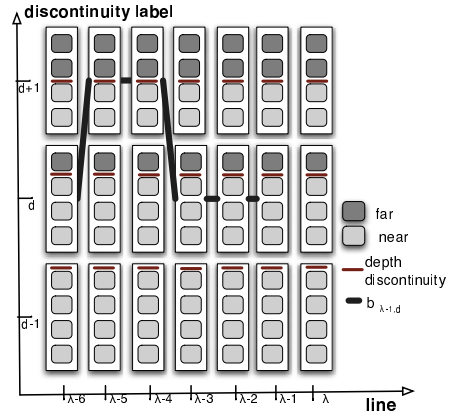


Figure 3. DP matching process. To determine the best discontinuity map up to line λ with discontinuity at d , for the various discontinuity locations on $\lambda - 1$, we look at the best solution up to $\lambda - 1$, available by construction. An example of a $b_{\lambda-1,d}$ is illustrated.

4.1. Optimizing discontinuity and visibility

The algorithm works by using dynamic programming (DP) applied perpendicularly to the orientation of the lines in \mathcal{L} . For instance, if the lines are vertical then DP proceeds horizontally either from left to right or right to left. We illustrate this for vertical lines processed from left to right. The cameras are assumed to be in a cross-shaped configuration, with the reference in the center and the 4 supporting images at an equal distance from it. The order of the cameras in the mask is left, right, top and bottom.

When dynamic programming attempts to compute the location of the discontinuity of line λ , it can rely on knowledge of the location of the discontinuities of all preceding lines (Fig. 3). Because of this, the visibility between any camera to the left of the reference and the 3D point formed by pixels \mathbf{p}_i^λ at any disparity is also known (Fig. 4). Two

similar strategies for binocular and multi-camera stereo were presented in [1, 6] respectively. These approaches minimize Eq. 1 directly, whilst we minimize Eq. 2, allowing the use of much more visibility information. Since the disparity of *passive* pixels is fixed during the minimization process, the visibility of the bottom and top cameras is known for each pixel of the line being examined. Figure 4 illustrates the visibility information available when processing a disparity map with our Border-Cut algorithm.

When solving the discontinuity localization problem, two 2-dimensional tables t and t' are filled out; $t(\lambda, d)$ is the lowest energy of all discontinuity maps of lines 0 to λ with line λ having its *discontinuity* at d ; $t'(\lambda, d)$ is the index of the *discontinuity* of $\lambda - 1$ given by this map of lowest energy, denoted $b_{\lambda,d}$. A sample discontinuity map $b_{\lambda-1,d}$ is highlighted in Fig. 3 and is also shown in Fig. 4. The table t' is used to compute the different $b_{\lambda,d}$'s.

Explicitly, the tables t and t' are defined inductively as

$$\begin{aligned} t(0, d) &= e_{f'}(0, d, g|_{\lambda_0}) \\ t'(0, d) &= d \end{aligned}$$

and for $\lambda > 0$

$$t(\lambda, d) = \min_{d' \in \mathcal{B}} \begin{pmatrix} e_{f'}(\lambda, d, g|_{\lambda}) \\ + s_{f'}(\lambda - 1, \lambda, d', d) \\ + r_{\mathcal{R}, f'}(\lambda, d) \\ + t(\lambda - 1, d') \end{pmatrix}$$

$t'(\lambda, d)$ is the index of the minimum in the above formula

where $r_{\mathcal{R}, f'}(\lambda, d)$ is the smoothing between the pixels of line λ having its *discontinuity* at d and *passive* pixels surrounding it. The visibility masks are

$$g(\mathbf{p}_i^\lambda) = (O_1(i, \lambda, d, d'), \dots, O_4(i, \lambda, d, d'))$$

where $O_c(i, \lambda, d, d')$ is a visibility function that is equal to 1 iff the camera c is visible from pixel \mathbf{p}_i^λ (see next section). For most c 's, it only requires the knowledge of $b_{\lambda-1, d'}(\lambda')$ for $\lambda' < \lambda$ and of *passive* pixels. The $b_{\lambda, d}$'s can be computed with the relations

$$\begin{aligned} b_{\lambda, d}(\lambda) &= d \\ b_{\lambda, d}(\lambda') &= t'(\lambda' + 1, b_{\lambda, d}(\lambda' + 1)) \quad \text{for } 0 \leq \lambda' < \lambda. \end{aligned}$$

It is thus possible to compute $b_{\lambda-1, d'}(\lambda')$ for all $\lambda' < \lambda$ and $d' \in \mathcal{B}$. This allows us to compute visibility $O_c(i, \lambda, d, d')$ for all d' and for most of the c 's and finally $t(\lambda, d)$.

4.2. Computing visibility

We define $O_c(i, \lambda, d, d')$ as the visibility of camera c for pixel \mathbf{p}_i^λ , for the *discontinuity* map defined on lines 0 to λ having its *discontinuity* on line λ at d and having the lowest

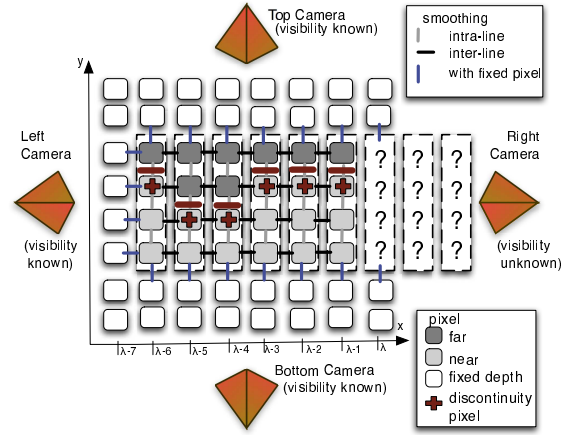


Figure 4. Visibility information available when searching for the location of the *discontinuity* pixel of vertical line λ using DP from left to right. The visibility of the left camera is available by construction whilst right visibility is unknown since border localization for lines to the right of λ is not yet computed. Top and bottom visibilities are always available since they only depend on disparity of pixels on the line being examined and of *passive* pixels.

energy amongst those with *discontinuity* d' on $\lambda - 1$, i.e. the disparity map $b_{\lambda', d'}$ extended to λ by placing its *discontinuity* at d . We show how to compute the visibility function O_1 for the left camera assuming that all segments are perfectly aligned, i.e. there is no *passive* pixel in the bounding rectangle of the region \mathcal{R} .

We first introduce the auxiliary function $S_1(i, \lambda, d)$ that represents how far the *shadow* of all the pixels to the left of \mathbf{p}_i^λ and \mathbf{p}_i^λ itself goes, for the *discontinuity* map of lowest energy defined on lines up to λ with a *discontinuity* on λ at d . Explicitly,

$$\begin{aligned} S_1(i, 0, d) &= \max \left\{ \max_{\lambda' < 0} \left(f|_{\mathcal{R}^c}(\mathbf{p}_i^{\lambda'}) + \lambda' \right), F_{b_{0, d}, f'}(\mathbf{p}_i^0) \right\} \\ S_1(i, \lambda, d) &= \max \left\{ S_1(i, \lambda - 1, b_{\lambda, d}(\lambda - 1)), F_{b_{\lambda, d}, f'}(\mathbf{p}_i^\lambda) + \lambda \right\} \quad \text{for } \lambda > 0 \end{aligned}$$

where the negative λ' 's are the lines of *passive* pixels to the left of the first (zeroth!) active one (of course $F_{b_{\lambda, d}, f'}$ is only defined on lines 0 to λ). When the segments are not aligned, the disparity of *passive* pixels can be needed to compute S_1 even for λ greater than 0. We leave the modifications to the reader. This recursive computation of S_1 used a continuous mesh representation of the disparity map. A consequence of this is that the ordering constraint is applied in the selection of visibility masks, but not on the disparity map itself. Now, O_1 is simply defined as

$$O_1(i, \lambda, d, d') = \begin{cases} 1 & \text{if } \delta_{f'}(i, \lambda, d) + \lambda > S_1(i, \lambda - 1, d') \\ 0 & \text{otherwise} \end{cases}$$

where $\delta_{f'}(i, \lambda, d)$ is the disparity of \mathbf{p}_i^λ when the discontinuity of λ is at d , given f' . O_1 is equal to 1 when the left camera is visible. The corresponding relations for the top and bottom ones are obtained similarly except that they only use disparity of the pixels of the current line and of *passive* pixels. For the right camera, no visibility information is available. Because it has been noticed that removing a camera that is visible is far less damaging than keeping a camera that is not [20], the right camera is only used when no others are. In multi-baseline stereo, it is reasonable to expect every pixel of the reference to be visible by at least one supporting camera. This use of a heuristic can make the Border-Cut algorithm oscillate, but after one or two cycles, changes are negligible.

Note that when the cameras are located on only 2 non opposite sides of the reference (for example top and left), the direction for which DP processes horizontal and vertical lines can always be chosen so that the visibility is handled *correctly*. With these camera configurations, the Border-Cut algorithm finds a local minimum, with respect to a *border move*, to energy function 4 respecting the constraint 3. The pixel likelihood term takes a user-defined occlusion cost when no camera is visible. Note that during a *border move*, the masks of *passive* pixels may change. This should be considered in the term $r_{\mathcal{R}, f'}$. However in our tests, ignoring this did not have an impact on the quality of the solutions. The explanation comes from the fact that close objects are generally enlarged by stereo matchers [7]. Since *border moves* usually correct this, *passive* pixels have a tendency to regain cameras rather than the opposite, a phenomenon that has little impact as mentioned above [20].

5. Experimental results

In all our experiments, the matching cost function used came from [17] which is based on [2]. We used color images but only the references in gray scale are shown here. As for the smoothing term, we used the experimentally defined smoothing function that also comes from [17]:

$$s(\mathbf{p}, \mathbf{p}', f(\mathbf{p}), f(\mathbf{p}')) = \gamma h(\mathbf{p}, \mathbf{p}') l(f(\mathbf{p}) - f(\mathbf{p}'))$$

where h is defined as

$$h(\mathbf{p}, \mathbf{p}') = \begin{cases} 3 & \text{if } |I_{ref}(\mathbf{p}) - I_{ref}(\mathbf{p}')| < 5 \\ 1 & \text{otherwise} \end{cases}$$

where l is 1 at 0 and 0 elsewhere. $I_{ref}(\mathbf{p})$ is the intensity of pixel \mathbf{p} in the reference image. The parameter γ is user-defined. A pixel disparity is considered erroneous if it differs by more than one step from the ground truth. This error measurement was used in two comparative studies [31, 27].

5.1. Tsukuba Head and Lamp

This dataset is from the Multiview Image Database from the University of Tsukuba (see Fig. 5). It is composed of a

5×5 image grid. Each image has a resolution of 384×288 . The search interval was between 0 and 15 pixels and we used 16 disparity steps. The 5-camera configuration is a

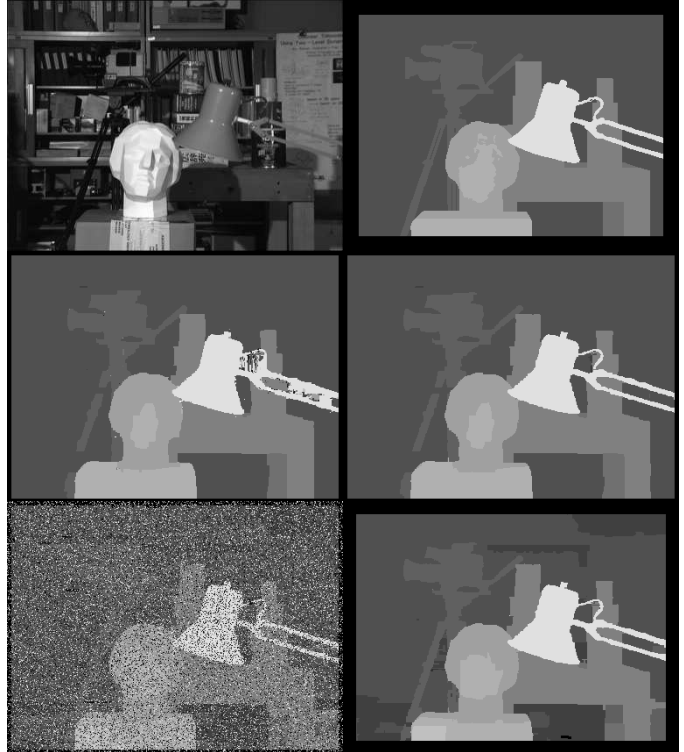


Figure 5. Reference images for the Head and Lamp (top left) and ground truth (top right) from the Multiview Images database of the University of Tsukuba. Disparity map obtained from GEO-BNV (middle left) and the improvement obtained after applying our algorithm (middle right). Disparity map of Hybrid-IDP with 36 % corruption (bottom left) and result after applying our algorithm (bottom right).

cross with 4 cameras equidistant to the center one (the reference). The 3-camera configuration is obtained from the previous one by removing the top and bottom cameras. Some disparity maps are shown in Fig. 5 and error percentages are given in Table 6. The entries ASYM-KZ1 and KZ1' come directly from [34]. HYBRID-IDP and NAKA-BNV come from [6]. The disparity map for the latter was not available, so we could not use it to initialize our algorithm. GEO-BNV and REL-DP come from [7] and [11] respectively. All these occlusion modeling methods are multi-baseline and used 5 cameras. Our algorithm lowered the error rate with all initializations. We also initialized with the current best two algorithms (SEG+VIS [3], SYMBP+OCC [29]) of the Middlebury Stereo Evaluation - (Version 2) [26]. The fast TREEDP based on dynamic programming was used as well [33]. We provide results obtained with a fixed smoothing parameter for all the different initializations, with the 3- and 5-camera configurations. We also provide those computed

with the best parameter for each disparity map computation. Of interest is the fact that results are comparable no matter which initial stereo matcher was used, even when this matcher only utilized two cameras without occlusion modeling. Only one *cycle* was performed with segments of up to 11 pixels; the running time was below 19 seconds on an AMD Athlon(tm) 64 Processor 3500+, even though we did not compute $e_{f'}$ and $s_{f'}$ recursively as described at the end of section 3.1. Note in Fig. 5 how a few pixels with correct disparity between the branches of the arm of the lamp were enough to allow Border-Cut to recover the disparities of this region correctly. When using three cameras with the fixed parameter, we could not reduce the error rate of SEG+VIS and KZ1'. Nevertheless, we saw a significant improvement when the best parameter was used. On this dataset, the sensitivity to initialization and to change of the smoothing parameter is reduced when the number of cameras increases from three to five. The reduced sensitivity of multi-baseline stereo to change of the smoothing parameter was also mentioned in [34].

Initialization algorithm with number of cameras used	Before B-C	After B-C			
		5 cameras		3 cameras	
		fixed γ	best γ	fixed γ	best γ
NAKA-BNV(5 cam's)	1.70	-	-	-	-
GEO-BNV(5 cam's)	2.23	1.17	1.07	1.48	1.44
HYBRID-IDP(5 cam's)	1.67	1.09	1.09	1.17	1.17
REL-DP(5 cam's)	1.86	1.13	1.07	1.24	1.11
KZ1'(5 cam's)	1.28	1.06	1.04	1.77	1.19
ASYM-KZ1(5 cam's)	1.30	1.11	1.08	1.09	1.08
SEG+VIS(2 cam's)	1.57	1.11	1.07	1.69	1.08
SYMBP+OCC(2 cam's)	1.75	0.96	0.96	1.02	1.02
TREEDP(2 cam's)	2.84	1.04	1.04	1.05	0.96
Average	-	1.08	1.05	1.31	1.13

Figure 6. Percentage of error in disparity for all pixels of the different algorithms for Head and Lamp scene, before and after applying Border-Cut, using 3 and 5 images.

Figure 7 shows the sensitivity to change of the smoothing parameter and to the quality of the initial disparity map of our algorithm, using HYBRID-IDP for initialization. The probability of a pixel to be *corrupted* (i.e. having its disparity changed to one picked at random from a uniform distribution) ranges from 0 to 80%. When used with multi-baseline camera configurations, our algorithm is very stable to change of the smoothing parameter. Without *corruption*, the error stays below 3.7% for γ ranging from 0 to 100. Moreover, even when pixels have a 36% chance of being *corrupted*, the error rate remains below 1.40% (see Fig. 5). It is thus very resistant to *corruption* of the initial disparity map even when a single *cycle* is performed. Since our approach does not add new discontinuities but may remove existing ones and since it is robust to *corruption*, we can start with an under-smoothed initial disparity map in order to avoid missing small objects. In our tests, this was not necessary.

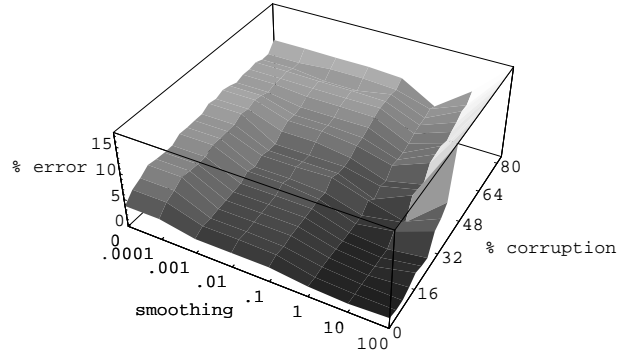


Figure 7. Variation of the error rate as a function of smoothing and pixel *corruption*.

5.2. City scene

This dataset from the Multiview Image Database of the University of Tsukuba (Fig. 8) contains 81 images of 640×480 in a 9×9 grid. We only used 5 images in a cross configuration. Each disparity map was computed using 44 disparity steps and the search interval was between 0 and 43 pixels. Two *cycles* were performed with segments of up to 11 pixels; the running time was below 92 seconds on an AMD Athlon(tm) 64 Processor 3500+. Some disparity discontinuities present in the reference were manually segmented to create a partial *discontinuity* ground truth. A discontinuity location is considered erroneous if it differs by more than one pixel from the ground truth. This is motivated by the fact that discontinuities in a scene are not perfectly aligned with the pixel grid. The different 5-camera algorithms used to initialize Border-Cut come from [7] and [6], whilst the 2-camera algorithms BNV and MF came from [4] and [24] respectively. The results are presented in Table 9 and some disparity maps are shown in Fig. 8. The percentage of pixels with a change in disparity greater than one for GEO-MF before and after Border-Cut is only 0.71%. Nevertheless, the improvement in border localization after applying Border-Cut is significant. Note that a correct discontinuity location does not imply that the disparities on both sides are correct as well.

5.3. Middlebury Binocular Comparative Study

To improve border localization, our Border-Cut algorithm is more appropriate for multi-baseline stereo configurations, since in binocular stereo, occlusion affects the localization of depth discontinuities on only one side of objects. Nevertheless, we provide results for the stereo pairs of the new Middlebury comparative study. We use as ini-

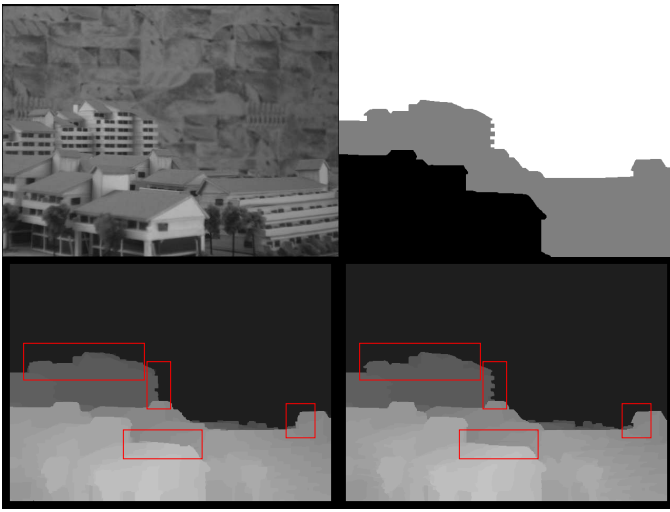


Figure 8. Reference image for the City scene (top left) from the Multiview Images database of the University of Tsukuba. Some manually segmented disparity discontinuities (top right). Disparity map obtained from GEO-MF (bottom left) and improved by Border-Cut (bottom right). Regions of interest are framed (see high resolution images in the electronic version of this paper for small details).

tialization the results obtained from BNV-OCC, KZ1, BNV and TREEDP coming from [16],[17], [27]-c and [33] respectively. The first two methods model occlusion, whilst the others do not. They all use the same smoothing model and similar cost function as that of our Border-Cut algorithm. Results are shown in Fig. 11. Border-Cut, when used on binocular stereo, is more sensitive to change of the smoothing parameter and occlusion cost (the latter is not even used in multi-baseline Border-Cut). The algorithm is also more sensitive to the initialization. Nevertheless, the average error reduction for BNV-OCC and KZ1 are respectively 0.48 and 0.21. For BNV and TREEDP, the error reduction is 1.61 and 1.88 respectively. As expected, the error reduction is more significant for algorithms that do not model occlusion. For TREEDP the error in disparity in discontinuity regions (as defined by the Middlebury Stereo Evaluation - Version 2) for the Venus scene was reduced from 7.74% to 2.22% (see Fig. 10). The improvement in discontinuity regions is even greater for algorithms known for having trouble with them, such as the one of [24]. For this particular case, the error rate was reduced from 21.5% to 4.61%. The increase of the error rate on the Tsukuba scene for KZ1 is caused by the fact that the same smoothing parameter was used for all four scenes, the one minimizing the average error.

6. Conclusion

We have presented a new algorithm to improve discontinuity localization of the disparity maps obtained from

Algorithm	Before B-C (best γ)	After B-C (fixed γ)
FULL-MF (5 cam's)	22.3	11.7
FULL-BNV (5 cam's)	19.8	11.1
GEO-MF (5 cam's)	23.2	10.4
GEO-BNV (5 cam's)	15.4	11.4
Hybrid-IDP (5 cam's)	28.2	14.7
BNV (2 cam's)	21.4	9.8
MF (2 cam's)	30.5	9.8

Figure 9. Percentage of error in the discontinuity location of the different algorithms for City scene, before and after Border-Cut, using 5 images.

multi-baseline stereo. From an initial disparity map, the algorithm associates a position to every disparity discontinuity, instead of a disparity label to every pixel. This formulation allows to find an approximated solution to a 2D labeling problem with a robust smoothing term by minimizing multiple 1D problems. We showed how to include efficiently visibility computation into each 1D minimization using dynamic programming. Our method obtains sharp and well-located disparity discontinuities starting from the output of a wide range of stereo matchers. For binocular algorithms and those that do not model occlusion, the improvement after applying *Border-Cut* is significant to the point where the final results are indistinguishable from those obtained after initializing with multi-baseline methods with occlusion modeling. Moreover, our framework, when used with certain camera configurations, guarantees the *correct* visibility. The validity of our framework was demonstrated on standard datasets with ground truth and was compared to other state-of-the-art multi-baseline stereo matchers. For future work, better ways of defining and extracting borders should be found. Applications of the Border-Cut algorithm to other vision problems should also be investigated.

References

- [1] P. N. Belhumeur. A Bayesian approach to binocular stereopsis. *Int. J. Computer Vision*, 19(3):237–260, 1996.
- [2] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(4):401–406, 1998.
- [3] M. Bleyer and M. Gelautz. A layered stereo algorithm using image segmentation and global visibility constraints. In *Proc. of Int. Conf. on Image Processing*, 2004.
- [4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cut. In *Proc. Int. Conference on Computer Vision*, pages 377–384, 1999.
- [5] I. J. Cox, S. Hingorani, B. M. Maggs, and S. B. Rao. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, 1996.
- [6] M.-A. Drouin, M. Trudeau, and S. Roy. Fast multiple-baseline stereo with occlusion. In *Proc. of the IEEE Conf. on 3-D Digital Imaging and Modeling*, pages 540–548, June 2005.
- [7] M.-A. Drouin, M. Trudeau, and S. Roy. Geo-consistency for wide multi-camera stereo. In *Proc. of IEEE Conference on*

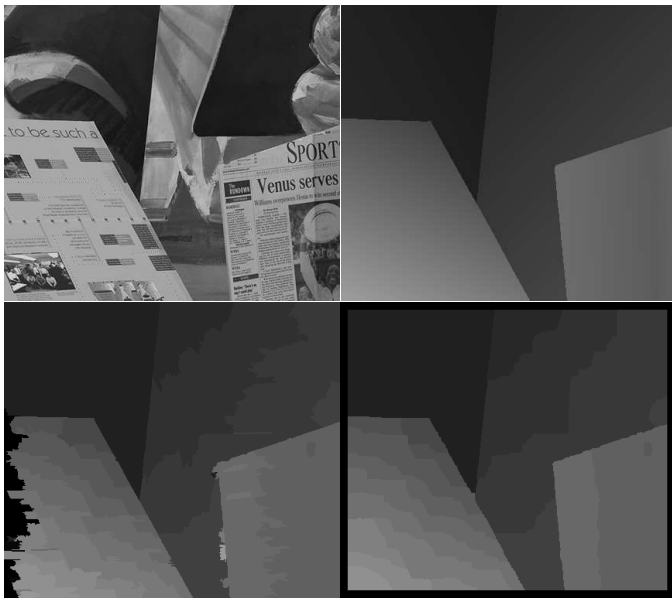


Figure 10. Reference image for the Venus scene (top left) from the Middlebury Comparative study. Ground truth (top right). Disparity map obtained by TREEDP (bottom left) and improved by Border-Cut using 2 cameras (bottom right).

Algorithm	Tsukuba		Venus		Teddy		Cones	
	B	A	B	A	B	A	B	A
BNV-OCC	2.01	1.97	2.19	1.52	17.4	17.1	12.4	11.5
KZ1	1.99	2.31	3.13	3.09	17.6	17.6	11.8	10.7
BNV	4.12	2.62	4.33	2.81	25.0	23.9	18.2	15.9
TREEDP	2.84	1.92	2.10	0.81	23.9	20.9	18.3	16.0

Figure 11. Percentage of error in disparity for all pixels of the different algorithms from the new Middlebury comparative study, before (B) and after (A) Border-Cut, using 2 images and fixed parameters.

Computer Vision and Pattern Recognition, pages 351–359, June 2005.

[8] G. Egnal and R. P. Wildes. Detecting binocular half-occlusions: Empirical comparisons of five approaches. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(8):1127–1133, 2002.

[9] O. D. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *Proc. European Conference on Computer Vision*, pages 379–393, 1998.

[10] A. Fusiello, V. Roberto, and E. Trucco. Efficient stereo with multiple windowing. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[11] M. Gong and Y.-H. Yang. Fast stereo matching using reliability-based dynamic programming and consistency constraint. In *Proc. Int. Conference on Computer Vision*, 2003.

[12] S. Intille and A. F. Bobick. Disparity-space images and large occlusion stereo. In *Proc. European Conference on Computer Vision*, pages 179–186, 2002.

[13] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *Fifth European Conference on Computer Vision*, pages 232–248, 1998.

[14] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment.

IEEE Trans. Pattern Analysis and Machine Intelligence, 16(9):920–932, 1994.

[15] S. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multiview stereo. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[16] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions via graph cuts. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 508–515, 2001.

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

[18] K. Kutulakos and S. Seitz. A theory of shape by space carving. *Int. J. Computer Vision*, 38(3):133–144, 2000.

[19] C. Leung, B. Appleton, and C. Sun. Fast stereo matching by iterated dynamic programming and quadtree subregioning. In *Proc. of the IEEE Conf. on British Computer Vision*, September 2004.

[20] Y. Nakamura, T. Matsuura, K. Satoh, and Y. Ohta. Occlusion detectable stereo -occlusion patterns in camera matrix-. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 1996.

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

[22] S. J. Osher and R. P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, Berlin, 2002.

[23] J. Park and S. Inoue. Acquisition of sharp depth map from multiple cameras. *Signal Processing: Image Commun.*, 14:7–19, 1998.

[24] S. Roy. Stereo without epipolar lines : A maximum-flow formulation. *Int. J. Computer Vision*, 34(2/3):147–162, 1999.

[25] M. Sanfourche, G. L. Besnerais, and F. Champagant. On the choice of the correlation term for multi-baseline stereovision. In *Proc. of the IEEE Conf. on British Computer Vision*, September 2004.

[26] D. Scharstein and R. Szeliski. www.middlebury.edu/stereo.

[27] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV* 47(1/2/3):7–42, April-June 2002., 47, 2002.

[28] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *Int. J. Computer Vision*, 35(2):151–173, 1999.

[29] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 399–406, June 2005.

[30] J. Sun, N. Zheng, and H. Shum. Stereo matching using belief propagation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(7):787–800, July 2003.

[31] R. Szeliski and R. Zabih. An experimental comparison of stereo algorithms. In *Vision Algorithms: Theory and Practice*, pages 1–19. Springer-Verlag, 1999.

[32] O. Veksler. Fast variable window for stereo correspondence using integral images. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[33] O. Veksler. Stereo correspondence by dynamic programming on a tree. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, June 2005.

[34] Y. Wei and L. Quan. Asymmetrical occlusion handling using graph cut for multi-view stereo. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 902–909, June 2005.