Fast Multiple-baseline Stereo with Occlusion

Marc-Antoine Drouin Martin Trudeau Sébastien Roy
DIRO
Université de Montréal
{drouim,trudeaum,roys}@iro.umontreal.ca

Abstract

This paper presents a new and fast algorithm for multi-baseline stereo designed to handle the occlusion problem. The algorithm is a hybrid between fast heuristic occlusion overcoming algorithms that precompute an approximate visibility and slower methods that use correct visibility handling. Our approach is based on iterative dynamic programming and computes simultaneously disparity and camera visibility. Interestingly, dynamic programming makes it possible to compute exactly part of the visibility information. The remainder is obtained through heuristics. The validity of our scheme is established using real imagery with ground truth and compares favorably with other state-of-the-art multi-baseline stereo algorithms.

1. Introduction

The goal of multi-baseline stereo is to reconstruct the 3D structure of a scene from multiple views with optical centers located in a two-dimensional grid configuration. In this paper, we used a configuration of five rectified images, equally spaced and arranged in a cross (Fig. 1). The disparity map is reconstructed from the point of view of the center camera which we call the reference. Occlusion occurs when part of a scene is visible in the reference but not in some supporting camera (Fig. 1). 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. This paper proposes a novel multiple-baseline stereo algorithm that computes simultaneously the disparity and part of the visibility information. The remaining visibility information cannot be computed and is found using heuristics. The proposed approach uses iterative dynamic programming (IDP) [4], a fast method for computing disparity maps. When applied to ordinary stereo, it minimizes the same energy function as Graph Cut [3] but obtains slightly higher error rates.

We use a unique property of dynamic programming that allows the application of IDP to mutiple-baseline stereo in a way that is impossible to do with Graph Cut. In this paper, we assume that the images are already rectified. For more details about image rectification, see [25].

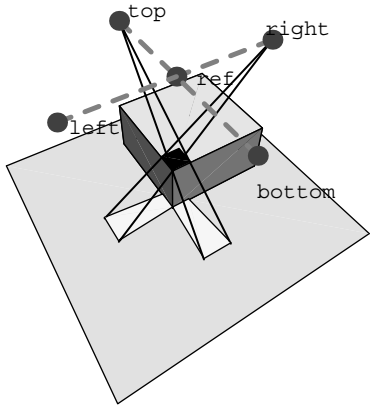

Figure 1. A cross-shaped camera configuration. The reference and the supporting cameras are labeled *ref*, *left*, *right*, *top* and *bottom* respectively. An example of occluder and occluded pixels are shown in black and white respectively.

To compute a disparity map for a reference image, we use a set of reference pixels $\mathcal{P}$ and a set of disparity labels $\mathcal{D}$. A $\mathcal{D}$-configuration $f : \mathcal{P} \mapsto \mathcal{D}$ associates a disparity label to every pixel. When occlusion is not modeled, the energy function to minimize typically is

$$E(f) = \underbrace{\sum_{\mathbf{p}\in\mathcal{P}} e(\mathbf{p}, f(\mathbf{p}))}_{\substack{\textit{pointwise} \\ \textit{likelihood}}} + \underbrace{\sum_{\mathbf{p}\in\mathcal{P}} \sum_{\mathbf{r}\in\mathcal{N}_\mathbf{p}} s(\mathbf{p}, \mathbf{r}, f(\mathbf{p}), f(\mathbf{r}))}_{\textit{smoothing}} \quad (1)$$

where e uses all the cameras for each pixel and $\mathcal{N}_\mathbf{p}$ is a neighborhood of pixel $\mathbf{p}$. This can be solved because the likelihood term $e(\mathbf{p}, f(\mathbf{p}))$ is pointwise independent and the

smoothing uses a 2-site clique form.

To model occlusion, we must compute the volumetric visibility $V_i(\mathbf{q}, d, f)$ of a reference pixel $\mathbf{q}$ located at disparity d from the point of view of a camera i, given a disparity configuration f defined for all other pixels. It is set to 1 if the point is visible, and 0 otherwise. The visibility information is collected into a vector as a *visibility mask*

$$V(\mathbf{q}, d, f) = (V_1(\mathbf{q}, d, f), \ldots, V_N(\mathbf{q}, d, f))$$

where N is the number of cameras outside the reference; a vector $(1, \ldots, 1)$ means that the 3D point is visible in all supporting cameras and $(0, \ldots, 0)$ means that it is invisible. 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 of the reference image. Using this, we transform Eq. 1 into an occlusion-aware energy function

$$E(f, g) = \sum_{\mathbf{p} \in \mathcal{P}} e(\mathbf{p}, f(\mathbf{p}), g(\mathbf{p})) + \textit{smoothing}. \quad (2)$$

Ideally, we would like to use $g(\mathbf{p}) = V(\mathbf{p}, f(\mathbf{p}), f)$ everywhere. Since this introduces a dependency between f and g, thus making the problem too hard, we will use instead for some cases a *correct* visibility, i.e. a g satisfying

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

for each component of these vectors and all $\mathbf{p} \in \mathcal{P}$. For the others, we will say that the handling of visibility is *heuristic*. Typically, we define the pointwise likelihood

$$e(\mathbf{p}, d, \mathbf{m}) = \frac{\mathbf{m} \cdot C(\mathbf{p}, d)}{|\mathbf{m}|} \quad \text{for } \mathbf{p} \in \mathcal{P},\ d \in \mathcal{D},\ \mathbf{m} \in \mathcal{M} \quad (4)$$

where $C(\mathbf{q}, d) = (c_1(\mathbf{q}, d), \ldots, c_N(\mathbf{q}, d))$ is the vector of matching costs of the pixel $\mathbf{q}$ at disparity d for each camera. We use $|\mathbf{m}|$ to represent the l_1-norm which is just the number of cameras visible from $\mathbf{q}$ at d. In a multi-baseline configuration, we can make the hypothesis that every point of the reference is seen by at least one supporting camera. So the case where $|\mathbf{m}| = 0$ cannot occur. For example, a simple cost function would be $c_i(\mathbf{q}, d) = (I_{ref}(\mathbf{q}) - I_i(T_i(\mathbf{q}, d)))^2$ where I_{ref} and I_i are respectively the reference and the supporting image i. $\mathbf{T}_i$ transforms pixel $\mathbf{q}$ at disparity d into the corresponding pixel of image i. In order to simplify the discussion, we will always consider the disparity as a positive value independently of the supporting camera used, and the T_i's take this into account.

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

2. Previous work

In a recent empirical comparison of strategies to overcome occlusion for 2 cameras, Egnal [6] enumerates 5 basic ones: left-right checking, bimodality test, goodness Jumps constraint, duality of depth discontinuity and occlusion, and uniqueness constraint. Some algorithms that have been proposed rely on one or more of the these strategies, and are often based on varying a correlation window position or size [11, 8, 26, 12]. Other algorithms use dynamic programming [18, 9, 1, 5] because of its ability to efficiently solve more complex matching costs and smoothing terms. Two methods using graph theoretical approaches [10, 13] have been proposed, but again they do not generalize well to multiple-camera configurations. Okutomi and Kanade have proposed a matching cost function designed to reduce ambiguity in stereo with multiple cameras having collinear optical centers[19]. However, their approach does not model occlusion.

When extending binocular stereo to multiple baselines, the amount of occlusion increases since each pixel of the reference camera can be hidden in more than one supporting camera. Some researchers have proposed specially designed algorithms based on pre-computed visibility masks to cope with this. A subset $\mathcal{M}_h$ of the most likely visibility masks of $\mathcal{M}$ is selected based on knowledge of the camera configuration. In order to determine the mask for a pixel $\mathbf{p}$ at disparity $f(\mathbf{p})$, the most photo-consistent one $g_f^*(\mathbf{p})$ is selected, that is

$$g_f^*(\mathbf{p}) = \arg \min_{m \in \mathcal{M}_h} e(\mathbf{p}, f(\mathbf{p}), m)\, w(m)$$

where $w(m)$ is a weight function favoring certain masks over others [17]. The problem thus becomes the minimization of $E(f, g_f^*)$ in f. Since e is pointwise independent, the new problem is reduced to the original formulation of Eq. 1 and is easily solved using standard algorithms. This technique is used in [17, 16, 20, 12]. Since the selected masks and the disparity map do not always respect Eq. 3, these methods are *heuristic*. A survey paper by Scharstein and Szeliski compares various binocular standard algorithms [22].

Other approaches try to minimize directly Eq. 2 in f and g, subject to the constraint of Eq. 3. Such *correct* methods have to solve a substantially more difficult problem than *heuristic* ones. In [15, 23], visibility-based methods are introduced. The matching cost incorporates the visibility information as a photo-consistency matching criteria, thereby implicitly modeling occlusion in the reconstruction process. Space carving can be seen as a greedy algorithm that minimizes Eq. 2 without smoothing. Similarly, a level-set method [7] uses the visibility information from the evolving reconstructed surface to explicitly model occlusion. In this case, depths are continuous and the problem is difficult to cast in the discrete setting of Eq. 2. Nevertheless, the idea is similar. In [14], a stereo algorithm based on graph cuts is presented. It strictly enforces visibility constraints to guide the matching process and ensures that all visibility masks

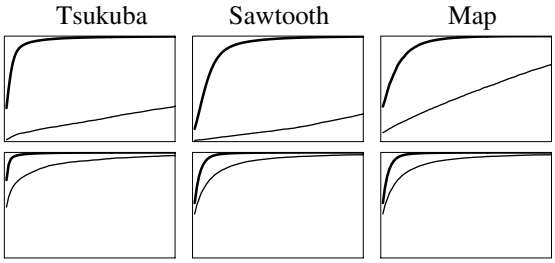

Figure 2. Cumulative histograms of matching cost values of occluded (thin line) and non occluded (thick line) pixels for 3 of the 4 test sequences of the Middlebury comparative study [22], for the **top)** depths from ground truth **bottom)** most likely depths (direct search). The x-axis' range is $[0, 80]$ and the y-axis' is $[0, 1]$.

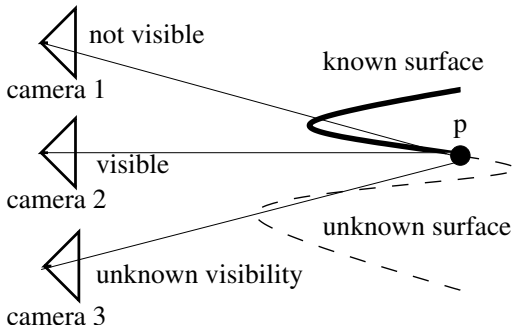

Figure 3. Example of the use of partial disparity map information. It is known that the point p is not visible from camera 1 but it is visible from camera 2. For camera 3, the partial disparity map does not allow us to know the visibility.

are consistent with the recovered surface. This algorithm jumps from one configuration respecting Eq. 3 to another. The formulation imposes strict constraints on the form of the smoothing term, constraints that will not apply to our method.

3. A hybrid algorithm

Heuristic approaches rely on the hypothesis that photo-consistency implies visibility. The figure 2 suggests that this is not always true. Using the matching cost function and images from the Middlebury comparative study [22], we computed the cumulative histograms of cost values for pixels classified as occluded and non occluded, based first on the ground truth and then on the computed disparity maps using direct search. The histograms are very different when the ground truth is used, but not when a direct search is. This indicates that many occluded pixels have a low cost and illustrates the fact that photo-consistency does not imply visibility. Ideally, we would like to benefit from the speed and simplicity of heuristics without being affected by the similarity between the matching cost distribution of occluded and non occluded pixels.

The algorithm we propose goes through the reference image pixel by pixel, building potential disparity maps for all the pixels up to the current one. At all time, the algorithm has access to the correct visibility information for a subset $\mathcal{C}_c$ of the set $\mathcal{C}$ of all supporting cameras. This visibility information comes from the partial knowledge of the disparity map (Fig. 3).

We can build a set $\mathcal{M}_c$ of masks using only cameras in $\mathcal{C}_c$. Given that each camera can be used or not, and discarding the empty mask, we are left with $2^{\#\mathcal{C}_c} - 1$ masks ($\#$ denotes the cardinality as usual). A mask from $\mathcal{M}_c$ is *correct*, independently of the visibility status of the cameras not in $\mathcal{C}_c$ (it has been noticed that removing a camera that is visible is far less damaging than keeping a camera that is not visible [17]). For the cases where no camera in $\mathcal{C}_c$ is visible, we must select a mask in another set $\mathcal{M}_h$ that only uses cameras in the subset $\mathcal{C}_h = \mathcal{C} - \mathcal{C}_c$. We contruct $\mathcal{M}_h$ so it only contains masks with one supporting camera. We thus have $\#\mathcal{M}_h = \#\mathcal{C}_h$. Since the set $\mathcal{M}_h$ can contain more than one mask, we use the heuristic that photo-consistency implies visibility to select the visibility mask. When a mask in $\mathcal{M}_h$ is selected, it is known that all cameras in the subset $\mathcal{C}_c$ are not visible. Since they are not used in $\mathcal{M}_h$, the heuristic mask is likely to be in fact correct. We expect the choice between $\mathcal{M}_c$ and $\mathcal{M}_h$ to be spacially coherent, we can thus add a visibility smoothing term that penalizes the use of masks belonging to differents sets for adjacent pixels.

Explicitly, for a pixel $\mathbf{p}$ and a certain disparity map f constructed up to the previous pixel, if $\mathbf{p}$ is visible by at least one camera in $\mathcal{C}_c$, the mask of $\mathbf{p}$ at d is set to a partial visibility $V'(\mathbf{p}, d, f)$ with each component defined as

$$V'_i(\mathbf{p}, d, f) = \begin{cases} V_i(\mathbf{p}, d, f) & \text{if } i \in \mathcal{C}_c \\ 0 & \text{otherwise.} \end{cases}$$

If $\mathbf{p}$ is not visible by any camera in $\mathcal{C}_c$, its mask is defined as $\arg\min_{m \in \mathcal{M}_h} e(\mathbf{p}, d, m)$. Note that the mask which minimizes the previous energy function would also minimize it if other masks containing more than one camera were added to the set $\mathcal{M}_h$. This comes from the matching cost function of Eq. 4 and the fact that the mean of multiple values is always greater than the smallest of these values.

Our algorithm finds an f and a g having a low energy according to Eq. 2 (not necessarily a global minimum), respecting the constraint

$$g(\mathbf{p}) = \begin{cases} V'(\mathbf{p}, f(\mathbf{p}), f) & \text{if } \exists i \in \mathcal{C}_c : V_i(\mathbf{p}, f(\mathbf{p}), f) = 1 \\ \arg\min\limits_{m \in \mathcal{M}_h} e(\mathbf{p}, f(\mathbf{p}), m) & \text{otherwise.} \end{cases}$$

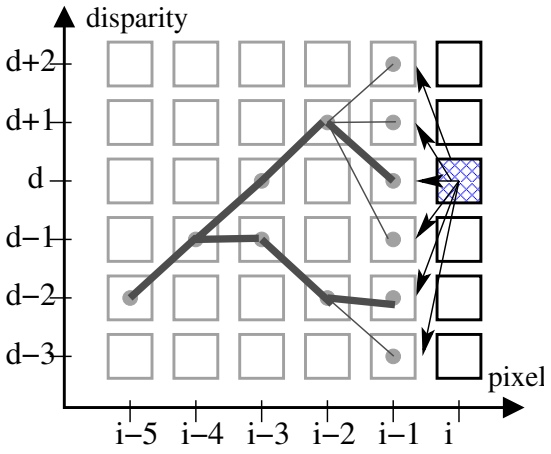

Figure 4. DP matching process. To determine the best disparity map up to pixel i with disparity d, for the different disparity values of $i-1$, we look at the best solution up to $i-1$, available by construction.

Disparity and visibility will be solved simultaneously using dynamic programming, taking into account long range visibility interactions.

3.1. Optimizing disparity and visibility

The stereo matching proceeds using Dynamic Programming (DP) applied along epipolar lines, which can be horizontal or vertical for our camera configuration. We illustrate the process for a left-right epipolar line, with a center reference image and left and right supporting images. When dynamic programming proceeds along, the computation of the disparity at pixel i can rely on knowledge of the disparities of all preceding pixels (Fig. 4). In the following discussion, a left to right order is assumed, but a right to left one is possible as well. Because of this, the visibility between any camera to the left of the reference and the 3D point formed by pixel i at disparity d is also known (Fig. 3). A similar strategy for binocular stereo was presented in [1]. When going from left to right, $\mathcal{C}_c$ consists of the left camera and $\mathcal{C}_h$ of the right one. Consequently, $\mathcal{M}_c$ contains only the mask consisting of the left camera, namely $(1,0)$. Similarly, $\mathcal{M}_h$ is simply $\{(0,1)\}$. When solving the correspondence problem along an epipolar line, two 2-dimensional tables t and t' are filled out; $t(i,d)$ is the lowest energy of all disparity maps of pixels 0 to i with pixel i at disparity d; $t'(i,d)$ is the disparity of pixel $i-1$ given by this map of lowest energy, denoted $f_{i,d}(i')$. Two sample disparity maps, $f_{i-1,d}$ and $f_{i-1,d-2}$, are highlighted in Fig. 4. The table t' is used to compute the different $f_{i,d}$'s.

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

$$\begin{aligned} t(0,d) &= e(0,d,(1,0)) \\ t'(0,d) &= d \\ t(i,d) &= \min_{d'\in\mathcal{D}} \begin{pmatrix} & e_v(i,d,d') \\ + & s(i-1,i,d',d) \\ + & t(i-1,d') \end{pmatrix} \\ t'(i,d) & \quad \text{is the index of the minimum in the above formula} \end{aligned}$$

where

$$e_v(i,d,d') = \begin{cases} e(i,d,(1,0)) & \text{if } O(i,d,d') < 0 \\ e(i,d,(0,1)) & \text{otherwise} \end{cases}$$

and $O(i,d,d')$ is a visibility function that is smaller than 0 iff the left camera is visible. It only requires the knowledge of $f_{i-1,d'}(j)$ for $j < i$. The $f_{i,d}$'s can be computed with the relations

$$\begin{aligned} f_{i,d}(i) &= d \\ f_{i,d}(j) &= t'(j+1, f_{i,d}(j+1)) \quad \text{for} \quad 0 \le j < i. \end{aligned}$$

It is thus possible to compute $f_{i-1,d'}(j)$ for all $j < i$ and $d' \in \mathcal{D}$. This allows us to compute visibility $O(i,d,d')$ for all d' and finally $t(i,d)$. Note that if for some $j \le i' \le i$ and $d, d' \in \mathcal{D}$ we have $f_{i,d}(j) = f_{i',d'}(j)$ then $f_{i,d}(k) = f_{i',d'}(k)$ for all $k \le j$. Moreover, the likelihood term e of a pixel i is not pointwise independent but depends on every pixel located to its left. As mentioned before, s may include visibility as well as disparity smoothing.

3.2. Computing visibility

When computing the left visibility function, the disparity map representation has an impact. We can consider a disparity map as a series of disconnected 3D points or as a continuous mesh. We define $O(i,d,d')$ as the visibility of pixel i at disparity d for the best solution with pixel $i-1$ at disparity d'. In the discontinuous case, O is defined as

$$O(i,d,d') = \begin{cases} 0 & \text{if } i+d = j + f_{i-1,d'}(j) \\ & \quad \text{for some } j < i \\ -1 & \text{otherwise.} \end{cases} \qquad (5)$$

It takes the value 0 when occlusion occurs and -1 when the left camera is visible. In the continuous case, we can lower the complexity by introducing the function O' defined as

$$O'(j,d') = \max_{0\le k\le j}(k + f_{j,d'}(k)).$$

This function can be computed inductively using the relations

$$\begin{aligned} &O'(0,d') = d' \\ &O'(j,d') = \max\{O'(j-1, f_{j,d'}(j-1)), j + d'\} \\ &\qquad\qquad\qquad\qquad\qquad\qquad \text{for } j > 0. \end{aligned}$$

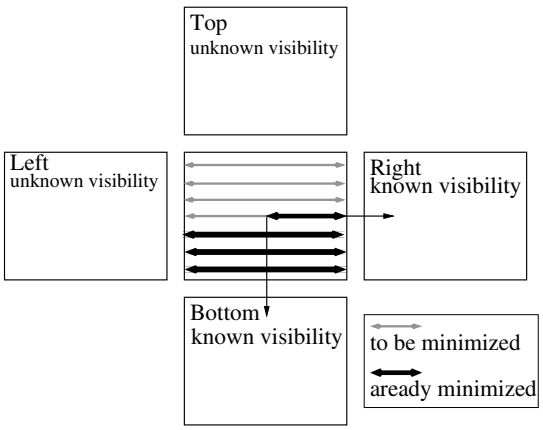

Figure 5. Optimization of a line: the visibility information is available for the right and the bottom cameras. For the current line, the right camera visibility is computed simultaneously with the disparity. The bottom camera visibility depends on the disparity of the previous lines which is fixed.

Now O can be simply defined as

$$O(i, d, d') = O'(i-1, d') - (i+d) \tag{6}$$

which is negative when the left camera is visible. Using a continuous mesh is equivalent to imposing the ordering constraint [6] in the selection of visibility masks, but not on the disparity map itself. Note that it is much faster to compute the continuous case. Moreover, as will be shown in section 4, always treating disparity maps as continuous does not affect the quality of the reconstruction in a scene that breaks the ordering constraint. The relations for right to left, top to bottom and bottom to top minimization are obtained similarly.

3.3. Visibility-aware iterative dynamic programming

In the previous section, we discussed visibility computation along an epipolar line. When working with a 5-camera cross-shape configuration, illustrated in Fig. 5, we can use the solution of the previous lines to compute the visibility of one of the cameras perpendicular to the line being processed. The visibility function O for such a camera is computed similarly as that of the camera with *correct* visibility along the current line. There is an important difference between the visibility information coming from the disparity maps of the previous lines and that from the line currently being processed: for latter, the disparity map is part of the minimization process, for the former it is fixed (Fig. 5).

In order to apply smoothing across epipolar lines, we use Iterative Dynamic Programming (IDP) proposed by Leung *et al.*[4]. For binocular stereo, they proceed in two steps: first they solve along horizontal lines and then along vertical lines. They repeat these two steps until a certain convergence criteria is met. The spatial smoothing term is not limited to a single line, but uses the last disparity information obtained from previous lines, step or iteration. We use the same smoothing strategy, but proceed in four steps when solving for lines and columns. In a first step, illustrated in Fig. 5, we start solving for horizontal lines from bottom to top, applying dynamic programming (DP) from right to left inside each line. The visibility computation relies on the disparity map obtained for the lower lines. In a second step, we solve for vertical lines from left to right, applying DP from bottom to top inside each line. Once again, solutions to previous lines are used for visibility. In the third step, we solve for horizontal lines from bottom to top, applying DP from right to left; for the fourth and last step, we solve for vertical lines from left to right, applying DP from top to bottom. Note that $\mathcal{C}_c$ and $\mathcal{M}_c$ vary from one step to the next. Table 6 shows the different masks for which the visibility is either *correct* or *heuristic* depending on the current step.

Optimization	Mask	Visibility
PIX right to left	$\mathcal{M}_c$ = { (0,**1**,0,**0**),(0,**0**,0,**1**),(0,**1**,0,**1**) }	correct
LINE bottom to top	$\mathcal{M}_h$ = { (1,**0**,0,**0**),(0,**0**,1,**0**) }	heuristic
PIX bottom to top	$\mathcal{M}_c$ = { (**1**,0,0,**0**),(**0**,0,0,**1**),(**1**,0,0,**1**) }	correct
LINE left to right	$\mathcal{M}_h$ = { (**0**,1,0,**0**),(**0**,0,1,**0**) }	heuristic
PIX left to right	$\mathcal{M}_c$ = { (**1**,0,0,**0**),(**0**,0,0,**1**),(**1**,0,0,**1**) }	correct
LINE bottom to top	$\mathcal{M}_h$ = { (**0**,1,0,**0**),(**0**,0,1,**0**) }	heuristic
PIX top to bottom	$\mathcal{M}_c$ = { (**1**,0,**0**,0),(**0**,0,**1**,0),(**1**,0,**1**,0) }	correct
LINE left to right	$\mathcal{M}_h$ = { (**0**,1,**0**,0),(**0**,0,**0**,1) }	heuristic

Figure 6. Visibility masks and their status depending on the current step. PIX refers to the order inside the line being solved,while LINE refers to the order in which lines are processed. In bold are the cameras belonging to $\mathcal{C}_c$. The camera order in a mask is left,right,top and bottom.

We also propose a different initialization; in [4], the disparity is initialized to a constant value. However, we do not use any prior disparity solution, the spatial smoothing is constrained to the current line during the first step of the algorithm. For subsequent steps, smoothing is performed along and across lines based on previously obtained solutions.

An iteration consists of the four steps described above. After the first iteration, every camera in $\mathcal{C}$ has been in $\mathcal{C}_c$ at least once. With a 5-camera cross configuration, in each step there is exactly one camera along the current epipolar line for which we have *correct* visibility at all time. For this reason, this configuration performs particularly well.

We can iterate to improve the disparity map. Our algorithm does not necessarily converge, as it is possible for the process to cycle. In practice, we stop after 1 to 8 iterations since changes are minimal after that. After one iteration,

the algorithm already provides high quality disparity maps.

When using the visibility function of Eq. 6 and hence representing the disparity map as a continuous mesh, the asymptotic complexity of the algorithm remains the same as for ordinary IDP, that is $\Theta(\#\mathcal{P}\,\#\mathcal{D}^2)$ where $\#\mathcal{P}$ is the number of reference pixels and $\#\mathcal{D}$ the number of disparity labels. When using the visibility function of Eq. 5, the asymptotic complexity increases to $\Theta(\#\mathcal{L}\,\#\mathcal{P}\,\#\mathcal{D}^2)$ where $\#\mathcal{L}$ is the highest number of pixels in any line. In our experiments, a continuous mesh representation was always used.

4. Experimental results

In all our experiments, the matching cost function was the same for all algorithms, that of [14] which is based on [2]. In our experiments we used color images; only the reference images in gray scale are shown here. For the smoothing term, we used the experimentally derived smoothing function that also comes from [14]:

$$s(\mathbf{p},\mathbf{r},f(\mathbf{p}),f(\mathbf{r})) = \lambda\, t(\mathbf{p},\mathbf{r})\,\delta(f(\mathbf{p})-f(\mathbf{r}))$$

where δ is 1 at 0 and 0 elsewhere and t is defined as

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

While we chose the Potts model for smoothing, dynamic programming can in fact use any model. As previously mentioned, we added to our method a visibility smoothing, taking the value 0 if the mask of the current pixel and that of its neighbor are both in $\mathcal{M}_c$ or both in $\mathcal{M}_h$, and the value γ if they are not. The details are similar *mutatis mutandis* as for the usual disparity smoothing (see [4]). The parameters λ and γ are user-defined smoothness levels. For each disparity map computation, we chose the λ and γ that achieved the best performance. A pixel disparity is considered erroneous if it differs by more than one disparity step from the ground truth. This error measurement is compatible with the one used in two comparative studies for 2-camera stereo [24, 22, 14].

4.1. Tsukuba Head and Lamp scene

This dataset is from the Multiview Image Database from the University of Tsukuba (Fig. 7). It is composed of a 5×5 image grid. Each image has a resolution of 384×288. The search interval is between 0 and 15 pixels and we used 16 disparity steps. The reference image is the center one and the four supporting images are the closest to it, forming a cross.

In order to show the validity of our algorithm, we compared our method with other state-of-the-art multi-baseline

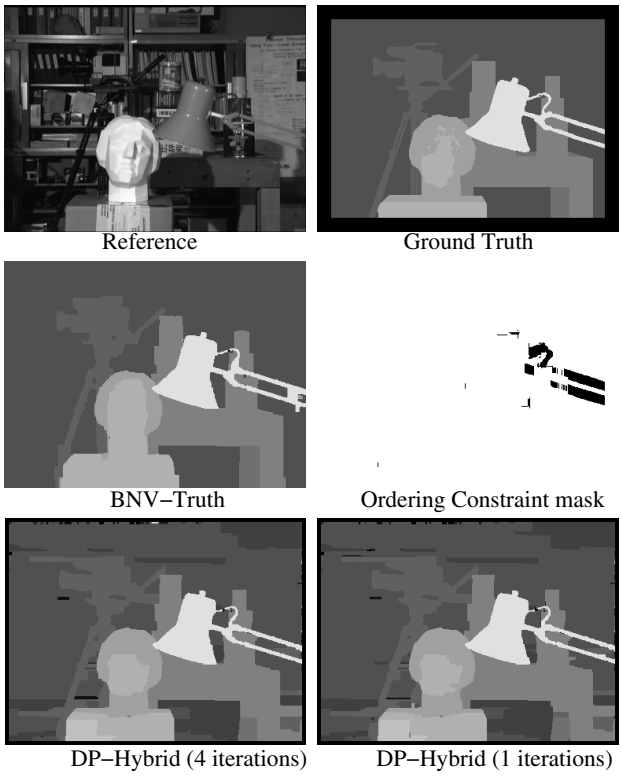

Figure 7. Disparity maps for various algorithms for the Head and Lamp scene (Multiview Images database of the University of Tsukuba). DP-Hybrid was run with a value of $\gamma = 19$. A linear mapping of disparities to gray levels was used. The mask of pixels breaking the ordering constraint in at least one supporting camera is also shown.

algorithms. Results from Graph Cut using Nakamura's visibility masks are labeled BNV-Heuristic. This is an adaptation of [16] where the maximum flow formulation of [21] was used with the precomputed visibility masks of [17]. We replaced the maximum flow by the Graph Cut algorithm of [3] (ranked the best stereo matcher in two comparative studies [24, 22]), having observed that, for this scene, it achieves a lower error rate. We also ran a version of the previous algorithm using IDP instead of Graph Cut as the optimization method (labeled DP-Heuristic), with 1 and 12 iterations. For the two versions, we tried different sets of masks $\mathcal{M}_h$ and picked the one achieving the best performance, namely the one that uses only 2 supporting cameras in each mask. This set has a total of 6 visibility masks.

Results of our method are shown under the label DP-Hybrid, with 1 and 4 iterations when using visibility smoothing, with 1 and 12 iterations when not. The only difference between DP-Heuristic and DP-Hybrid is the occlusion model. We also compared our algorithm with that of [14], a method with *correct* visibility handling (labeled

Algorithm	Error
BNV-Truth	1.01%
DP-Hybrid (4 iterations, $\gamma = 19$)	1.67%
BNV-heuristic	1.77%
DP-Hybrid (1 iteration, $\gamma = 19$)	1.82%
DP-Hybrid (12 iterations, $\gamma = 0$)	2.01%
KZ1	2.30%
DP-Heuristic (12 iterations)	2.35%
DP-Heuristic (1 iteration)	2.63%
DP-Hybrid (1 iteration,$\gamma = 0$)	2.77%

Figure 8. Percentages of error of the different algorithms for the Head and Lamp scene, using 5 images. All algorithms use the same matching cost and smoothing function.

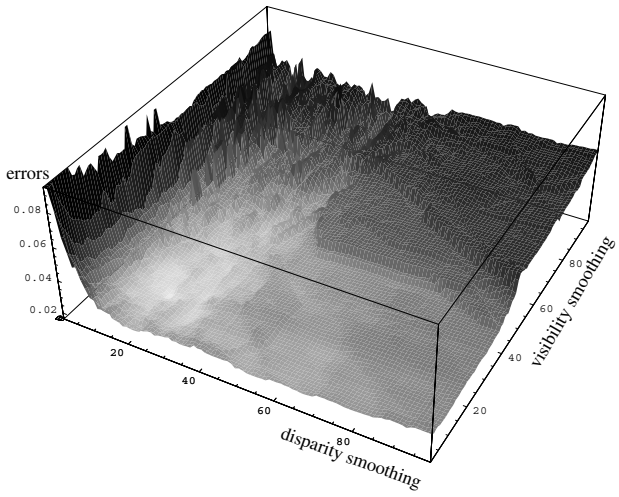

Figure 9. Resistance to change of the smoothing parameter for the Head and Lamp scene for one iteration. Both smoothing parameters increase by a factor of more than 100.

KZ1). Its error measurement was taken directly from the same article. Finally, we computed the disparity maps using Graph Cut with the *exact* visibility masks computed in advance from the ground truth. We labeled this method BNV-Truth.

Some disparity maps are shown in Fig. 7; all the error percentages are shown in Fig. 8. Our method with 4 iterations achieved the lowest error rate after BNV-Truth. Even after one iteration, the error rate is low and requires less than 4 seconds of running time on a 2.0 GHz Athlon 64 with a non optimized implementation. Figure 8 also shows the minimal impact of subsequent iterations after the first when using visibility smoothing. The error goes down with additional iterations, but only by a small amount. This figure also shows the impact of visibility smoothing for our algorithm.

Figure 9 shows a stability analysis of the smoothing parameter for our algorithm (DP-Hybrid), giving the error percentages over a broad range of values.

There are pixels for which the ordering constraint is broken, in particular in the arm of the lamp (Fig. 7). They were identified by re-projecting the ground truth in each supporting camera. They did not affect our algorithm even if our visibility computation makes the hypothesis that the ordering constraint is respected.

4.2. Others scenes

We used the Plant and the Santa scenes from the Multiview Image Database of the University of Tsukuba (Fig. 10). These datasets contain 81 images in a 9×9 grid taken with a camera having a 10 mm focal length. We only used 5 images in our usual cross configuration. Images were reduced by a factor of 2 to achieve a resolution of 320×240. Each disparity map was computed using 24 disparity steps. For these datasets, the value of γ did not have a significant impact. We display the results for $\gamma = 0$.

The Plant scene features a lot of occlusion coming from very thin objects and constitutes a very good test for the validity of any occlusion model. The red flower is located at 59cm of the reference camera, the back blue ones at 92cm and the background at 184cm. The baseline is 20 mm. The disparity map obtained after only one iteration is shown in Fig. 10. The running time was under 4 seconds. The results for 8 iterations are similar.

The second scene features a Santa doll. The hand is located at 59 cm of the reference camera and the background at 184 cm. The disparity map obtained after 8 iterations is shown in Fig. 10 (it is slightly better than the one obtained after one iteration). Note the details on the right side of the hat and on the candle.

5. Conclusion

We have presented a new stereo matching algorithm for multiple-baseline stereo. Our approach is a hybrid between the fast methods that use photo-consistency to approximate correct visibility and slower methods that use correct visibility. In this paper, we used a Potts model, but the proposed algorithm is flexible enough to be used with any type of smoothing term. It is fast and also succeeds in obtaining sharp and well-located depth discontinuities. The validity of our framework has been demonstrated on standard datasets with ground truth and compares favorably with other state-of-the-art occlusion models for multiple-view stereo.

As for future work, we would like to build a real-time implementation using dedicated hardware such as FPGA's. In

Figure 10. Disparity maps for the Plant and Santa scene using DP-Hybrid (Multiview Image Database of the University of Tsukuba). A linear mapping of disparities to gray levels was used.

addition, the extension of this occlusion model to arbitrary multi-camera configurations and volumetric reconstruction should be explored.

6. Acknowledgment

This work was made possible by NSERC (Canada) and NATEQ (Québec) grants.

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] 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.

[4] B. A. C. Leung 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.

[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] 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.

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

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

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

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

[11] 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.

[12] 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.

[13] 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.

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

[15] K. Kutulakos and S. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):133–144, 1987.

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

[17] 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.

[18] 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.

[19] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(4):353–363, April 1993.

[20] J. Park and S. Inoue. Hierarchical depth mapping from multiple cameras. In *Int. Conf. on Image Analysis and Processing*, volume 1, pages 685–692, Florence, Italy, 1997.

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

[22] 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.

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

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

[25] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.

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