Geo-consistency for Wide Multi-Camera Stereo

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

Abstract

This paper presents a new model to overcome the occlusion problems coming from wide baseline multiple camera stereo. Rather than explicitly modeling occlusions in the matching cost function, it detects occlusions in the depth map obtained from regular efficient stereo matching algorithms. Occlusions are detected as inconsistencies of the depth map by computing the visibility of the map as it is reprojected into each camera. Our approach has the particularity of not discriminating between occluders and occludees. The matching cost function is modified according to the detected occlusions by removing the offending cameras from the computation of the matching cost. The algorithm gradually modifies the matching cost function according to the history of inconsistencies in the depth map, until convergence. While two graph-theoretic stereo algorithms are used in our experiments, our framework is general enough to be applied to many others. The validity of our framework is demonstrated using real imagery with different baselines.

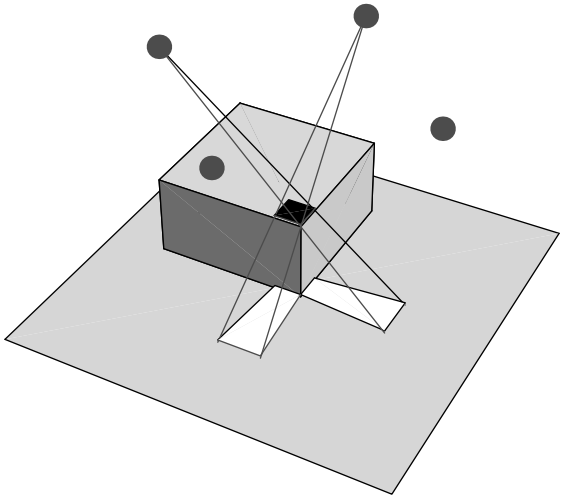

Figure 1. Example of occlusion. Occluded pixels appear in white, occluders in black.

1. Introduction

The goal of binocular stereo is to reconstruct the 3D structure of a scene from two views. As the baseline gets wider, the problem of occlusion, wich is often considered negligible with small baseline configurations, can become severe and limit the quality of the obtained depth map. Occlusion occurs when part of a scene is visible in one camera image but not in the other (see figure 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, as it is the final goal of the algorithm. We propose a novel multiple camera stereo algorithm that relies on photometric and geometric inconsistencies in the depth map to detect occlusions. As this algorithm is iterative, it does not explicitly model an occlusion state or add extra constraints to the cost function. This makes it possible to use a standard efficient algorithm during each iteration, instead of tackling a very difficult optimization problem. Futhermore, our approach guarantees to preserve the consistency between the recovered visibility and geometry, a property we call geo-consistency. In this paper, the maximum flow [19] and graph cut [2] formulations are used to solve each iteration. Our framework is general enough to be used with many other stereo algorithms. A survey paper by Scharstein and Szeliski [21] compares various standard algorithms.

The rest of this paper is divided as follows: in Section 2, previous work will be presented. Section 3 describes occlusion modeling and geometric inconsistency. Our proposed algorithm is described in Section 4. Experimental results are presented in Section 5.

2. Previous work

In a recent empirical comparison of strategies to overcome occlusion for 2 cameras, Egnal [4] 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 [9, 6, 26, 10]. These methods are binocular in nature and do not generalize well to the case of multiple arbitrary cameras. Other algorithms use dynamic programming [16, 7, 3] because of its ability to efficiently solve more complex matching costs and smoothing terms. Two methods using graph theoretical approaches [8, 11] have been proposed, but again they do not generalize to multiple camera configurations.

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, such as a regular grid of cameras [15]. Some researchers have proposed specially designed algorithms to cope with occlusion in multiple camera configurations. Amongst these, Kang et al. [10] proposed a visibility approach. While they did not improve over adaptive windows, their scheme was based on the hypothesis that a low matching cost function implies the absence of occlusion. This hypothesis is also made in [15, 20, 17, 18]. In contrast, we do not rely on such an assumption. In [27], a relief reconstruction approach based on belief propagation is presented where the correct visibility is approximated by using a low resolution base surface obtained from manually established correspondences. In [14, 23], *visibility-based* methods are introduced. The matching cost incorporates the visibility information into a photo-consistency matching criteria, thereby implicitly modeling occlusion in the reconstruction process. Our method differs completely in the way it handles smoothing and by its ability to recover from bad "carving". Similarly, a level-set method [5] uses the visibility information from the evolving reconstructed surface to explicitly model occlusion. In [12] a stereo algorithm based on graph cuts is presented. It strictly enforces visibility constraints to guide the matching process and ensures that it does not contain any geometric inconsistencies. The formulation imposes strict constraints on the form of the smothing term, constraints that will not apply to our method as we will see.

3. Modeling occlusion and Geo-consistency

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

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

where $\mathcal{N}_\mathbf{p}$ is a neighborhood of pixel $\mathbf{p}$. This can be solved efficiently because the likelihood term $e(\mathbf{p}, f(\mathbf{p}))$ is independent from $e(\mathbf{p}', f(\mathbf{p}'))$ for $\mathbf{p} \neq \mathbf{p}'$, and the smoothing term has a simple 2-site clique form.

To model occlusion, we must compute the volumetric visibility $V_i(\mathbf{q}, f)$ of a 3D reference point $\mathbf{q}$ from the point of view of a camera i, given a depth configuration f. It is set to 1 if the point is visible, and 0 otherwise. Visibility is a long range interaction and knowledge about immediate neighborhood configuration is insufficient most of the time for computing it. The visibility information is collected into a vector, the *visibility mask*

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

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, $(0, \ldots, 0)$ that it is invisible instead. 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. Using this, we transform Eq. 1 into an energy function with mask

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

Typically, we define

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

where the 3D point $\mathbf{p}|z$ is $\mathbf{p}$ augmented by z and $C(\mathbf{q}) = (C_1(\mathbf{q}), \ldots, C_N(\mathbf{q}))$ is the vector of matching costs of 3D point $\mathbf{q}$ for each camera. We use $|\mathbf{m}|$ to represent the l_1-norm which is just the number of cameras used from $\mathbf{q}$. The case where $|\mathbf{m}| = 0$ is discussed in section 4.2. A simple cost function is $C_i(\mathbf{q}) = (I_{ref}(\mathbf{M}_{ref}\mathbf{q}) - I_i(\mathbf{M}_i\mathbf{q}))^2$ where $\mathbf{M}_{ref}$ and $\mathbf{M}_i$ are projection matrices from the world to the images of camera *ref* and i respectively, and I_{ref} and I_i are these images. Now, in order to model occlusion properly, we simply need to examine the case $g(p) = V(\mathbf{p}|f(\mathbf{p}), f)$.

If the visibility masks were already known and fixed, the occlusion problem would be solved and only photogrametric ambiguity would remain to be dealt with; the energy function (2) would then be relatively easy to minimize.

Since this is not the case and f and $V(.,f)$ are dependent, we relax the problem by introducing the concept of *g*eo-consistency: we say that a $\mathcal{Z}$-configuration f is geo-consistent with an $\mathcal{M}$-configuration g if

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

for each component of these vectors and all $\mathbf{p} \in \mathcal{P}$. The inequality thus allows the mask to contain a subset of the visible cameras. The removal of extra cameras has been observed to have little impact on the quality of the solution [15]. Our problem becomes the minimization of Eq. 2 in f and g, with the constraint that f is geo-consistent with g.

3.1. Solving simultaneously for depth and visibility

Lets define $g^0(\mathbf{p}) = (1, \ldots, 1)$ for all $\mathbf{p} \in \mathcal{P}$; this corresponds to the case where all cameras are visible by all points. Minimizing $E(f, g^0)$ in f is equivalent to minimizing $E(f)$. In general, it is possible to minimize $E(f, g)$ by explicitly testing all combinations of depth labels and visibility masks in $\mathcal{Z} \times \mathcal{M}$. Since $\#\mathcal{M} = 2^N$, this effectively makes the problem too big to be solved except in the simplest cases. One way to reduce the number of visibility masks is to realize that for a given camera configuration, some masks may occur for no configuration f. This makes it possible to precompute a smaller subset of $\mathcal{M}$. Another way to reduce the number of masks is simply to decide on a *reasonable* subset to use [15]. Unfortunately, even with a small number of masks, it is still not practical to minimize in f and g simultaneously. We can however use photo-consistency alone to select the visibility mask of a pixel, if it is assumed equivalent to geo-consistency. In order to determine the mask for a pixel $\mathbf{p}$ at depth $f(\mathbf{p})$, we can try each mask and select the most photo-consistent one, i.e. we define g^*_f as

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

where $w(m)$ is a weight function favoring g^0 and eliminating improbable masks. The problem thus becomes the minimization of $E(f, g^*_f)$ in f. Since e is point-wise 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 [15, 20, 17]. However, the selected masks are not guaranteed to preserve geo-consistency.

In space carving[14], the depth $f(\mathbf{p})$ of a pixel is increased at a given step if it is not photo-consistent (which is determined using a threshold). When depth is changed at a point, the mask configuration g is updated accordingly, and so preserves geo-consistency. Space carving is a greedy algorithm that solves Eq. 2 subject to the constraint of Eq. 3 without smoothing. Kolmogorov and Zabih [12] tried to minimize an approximation of Eq. 2 subject to the constraint of Eq. 3 with spatial smoothing by moving iteratively from one geo-consistent solution to the other.

4. Stereo with a new implicit occlusion model

We propose to reduce the dependency between f and g by making it *temporal*: we let f^0 be the $\mathcal{Z}$-configuration minimizing $E(f, g^0)$ in f and for $t > 0$, we define iteratively f^t as the function minimizing

$$\sum_{\mathbf{p} \in \mathcal{P}} e(\mathbf{p}, f^t(\mathbf{p}), V(\mathbf{p}|f^t(\mathbf{p}), f^{t-1}) + \textit{smoothing} \tag{4}$$

and g^t as

$$g^t(\mathbf{p}) = V(\mathbf{p}|f^t(\mathbf{p}), f^{t-1}),$$

that is to say, f^t minimizes $E(f^t, g^t)$, where g^t depends on f^t according to the above equation. Now, this can be done using any standard algorithm. Unfortunately, this process does not always converge [10].

4.1. Using history for convergence

Because of the way g^t is defined, cameras that are removed at one iteration can be kept at the next, possibly introducing cycles. To guarantee convergence, we introduce a *visibility history mask* independent of the matching cost function value

$$H(\mathbf{q}, t) = (H_1(\mathbf{q}, t), \ldots, H_N(\mathbf{q}, t))$$

where N is again the number of cameras other than the reference and

$$H_i(\mathbf{q}, t) = \prod_{0 \leq k \leq t} V_i(\mathbf{q}, f^k) = \min_{0 \leq k \leq t} V_i(\mathbf{q}, f^k) \tag{5}$$

The new problem is obtained by substituting H for V in Eq. 4 to obtain

$$E^t_H(f^t) = \sum_{\mathbf{p} \in \mathcal{P}} e(\mathbf{p}, f^t(\mathbf{p}), H(\mathbf{p}|f^t(\mathbf{p}), t-1) + \textit{smoothing} \tag{6}$$

Mutatis mutandis, f^t now minimizes $E^t_H(f^t)$ and $g^t(\mathbf{p}) = H(\mathbf{p}|f^t(\mathbf{p}), t-1)$. This iterative process always converges (or stabilizes) in a polynomial number of steps. Indeed, $H(\mathbf{q}, t)$ is monotonically decreasing in t for all $\mathbf{q}$; moreover, if $H(\mathbf{q}, t-1) = H(\mathbf{q}, t)$ for all $\mathbf{q}$, then $f^t = f^{t+1}$ since both are solutions to the same minimization problem, and the process has stabilized. We see that the number of iterations is bounded by $N \cdot \#\mathcal{P} \cdot \#\mathcal{Z}$.

Furthermore, after convergence, the final configuration $f^{T+1} = f^T$ is geo-consistent with g^{T+1}; this comes from the fact that for all $\mathbf{p}$:

$$\begin{aligned} g^{T+1}(\mathbf{p}) &= H(\mathbf{p}|f^{T+1}(\mathbf{p}), T) = H(\mathbf{p}|f^T(\mathbf{p}), T) \\ &\leq V(\mathbf{p}|f^T(\mathbf{p}), f^T) = V(\mathbf{p}|f^{T+1}(\mathbf{p}), f^{T+1}). \end{aligned}$$

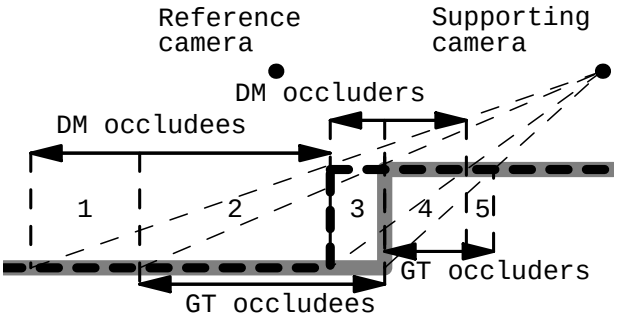

Figure 2. Effect of object enlargement on classification of occluders and occludees of a scene viewed by 2 cameras. The ground truth is in thick gray and the depth map in thick dashes. Occluders and occludees are shown for both ground truth (GT) and computed depth map (DM). Illustration of classification shift. Respectively, the 5 zones represent 1) regular pixels wrongly classified as occludees 2) occludees correctly classified 3) occludees wrongly classified as occluders 4) occluders correctly classified 5) occluders wrongly classified as regular.

We thus have an algorithm that converges to a geo-consistent solution, but that can transit through intermediate ones that are not. This type of behavior differentiates our approach from others that strictly enforce geo-consistency during the optimization process [14, 5, 12].

4.2. Pseudo-visibility

For a given f, an occluder $\mathbf{p}|f(\mathbf{p})$ is a 3D point blocking an occludee $\mathbf{p}'|f(\mathbf{p}')$ in some camera. Figure 1 illustrates the phenomenon. Each pixel of a depth map can be classified as an occluder, an occludee, or a regular pixel (neither occluder nor occludee). We have observed experimentally that many algorithms have a tendency to overestimate the disparity of occluded pixels. This has the effect of making close objects larger, creating a shift in the pixel classification of occludees and occluders. Occludees have a tendency to be classified as occluders, occluders as regular pixels and regular pixels as occludees (see figure 2). To validate this assertion, we used the results of two of the best stereo matchers evaluated with the Middlebury dataset. [21, 24, 2]. [2] was ranked the best stereo matcher in two comparative studies [25, 21]. [24] appeared later and achieved an even lower error rate. For each obtained depth map, we computed the percentage of pixels classified as occluder by the depth map that really are occludees and that of pixels classified as occludees that really are regular (figure 3). Both turned out to be quite high. Since most pixels are regular, the percentage of wrong classification for them is low. Nevertheless, there is a clear bias: more pixels classified as regular are occluders than occludees. The observation above discourages the direct use of visibility to update the visibility history mask. Instead, we introduce a pseudo-visibility

$$V'(\mathbf{q}, f) = (V'_1(\mathbf{q}, f), \ldots, V'_N(\mathbf{q}, f))$$

which compensates for the bias by labeling both occluders and occludees as invisible. An obvious consequence of this definition is the fact that

$$V'_i(\mathbf{p}|f(\mathbf{p}), f) \leq V_i(\mathbf{p}|f(\mathbf{p}), f) \quad \forall \mathbf{p} \in \mathcal{P}, \quad 1 \leq i \leq N.$$

The ordering constraint simply states that when scanning an epipolar line, the order in which we encounter two different objects visible in two images of a stereo pair must be the same in the two images (see Fig 4-left). This constraint holds for most scenes (see Fig 4-right) [13]. While this constraint is broken in some rare cases, it remains a powerful tool when dealing with occlusion and outliers. If we represent the depth map as an opaque mesh, we are guaranteed to preserve the ordering constraint between the reference and any supporting camera for any point visible from them. If a set of pixels $\mathcal{O}$ breaks the ordering constraint between the reference camera and some supporting image i at iteration t, then according to our definition of pseudo-visibility (and using an opaque mesh), the history mask is updated to $H_i(\mathbf{p}|f^{t+1}(\mathbf{p}), t) = 0$ for all $\mathbf{p}$ in $\mathcal{O}$. After convergence for the final configuration f^T we have for all $\mathbf{p}$ $H(\mathbf{p}|f^{T+1}(\mathbf{p}), T) = H(\mathbf{p}|f^T(\mathbf{p}), T-1)$. In particular $H_i(\mathbf{p}|f^{T+1}(\mathbf{p}), T) = 0$ for all $\mathbf{p} \in \mathcal{O}$. Since the offending camera i was not used to compute the final solution, the ordering constraint is respected between the reference camera and the supporting camera i.

The pseudo-visibility masks V'_i are computed by using rendering techniques. Two renderings of the current depth map f are done from the point of view of each supporting camera i: one with an ordinary Z-buffer and one with

Algo	**Scenes from Middlebury comparative study**					
	Tsukuba Head and Lamp			Sawtooth		
	Real status of pixels classified from depth map as occluders					
	occludee	occluder	regular	occludee	occluder	regular
bp [24]	**44.8**	16.3	38.9	**42.6**	3.8	53.6
bnv [2]	**50.4**	15.4	34.2	**42.6**	4.3	53.3
	Real status of pixels classified from depth map as occludees					
	occludee	occluder	regular	occludee	occluder	regular
bp [24]	15.5	5.9	**76.6**	5.5	1.1	**93.4**
bnv [2]	16.4	5.8	**77.8**	7.2	1.1	**91.7**
	Real status of pixels classified from depth map as regulars					
	occludee	occluder	regular	occludee	occluder	regular
bp [24]	1.0	**2.0**	97.0	0.5	**1.5**	98.0
bnv [2]	1.0	**2.0**	96.9	0.5	**1.5**	98.0

Figure 3. Real (ground truth) status in percentages of pixels according to their classification. Examples from the Middlebury comparative study [21]. In bold are the misclassifications favored by the overestimation of the disparity of occluded pixels.

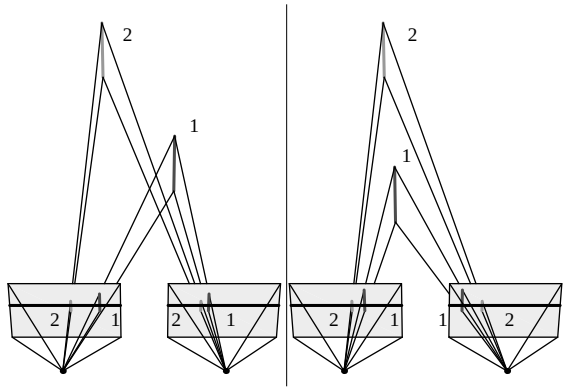

Figure 4. Left) Ordering constraint is satisfied. In this camera configuration, the epipolar lines are parallel to the X-axis. The line 2 is located to the left of the line 1 in both images. Right) Ordering constraint is broken, the line 2 appears to the left of the line 1 in one image and to the right in the other.

a reverse Z-buffer test. Two depth maps L_i^f and G_i^f are thus obtained and contains minimal and maximal depth observed by the camera. By comparing them, we can detect when two points of the mesh project to the same location for a given supporting camera. When using rectified images, this rendering process can be greatly sped up and simplified by replacing it by a line drawing using depth buffers. The pseudo-visibility function $V_i'(\mathbf{q}, f)$ can therefore be computed as

$$V_i'(\mathbf{q}, f) = \delta\left(L_i^f(\mathbf{T}_i\mathbf{q}) - G_i^f(\mathbf{T}_i\mathbf{q})\right)$$

where δ is 1 at 0 and 0 elsewhere.

It is possible for a voxel to have all its cameras removed, i.e. $H(\mathbf{p}|z, t-1) = \mathbf{0}$ even if $V(\mathbf{p}|z, t-1) \neq \mathbf{0}$ In practice, when this happens, we replace $e(\mathbf{p}, z, H(\mathbf{p}|z, t-1))$ by $e(\mathbf{p}, f^{t'+1}(p)), H(\mathbf{p}|z, t')$ in the minimization process that computes f^t (see Eq 6), where t' is the largest index such that $H(\mathbf{p}|z, t') \neq 0$. It this case, depth is assigned only from the neighborhood through smoothing.

5. Experimental results

In all our experiments, the matching cost function was the same for all algorithms, that of [12] which is based on [1]. We used color images but only the reference images in gray scale are shown here. As for the smoothing term, we used the experimentally defined smoothing function that also comes from [12]:

$$s(\mathbf{p}, \mathbf{r}, f(\mathbf{p}), f(\mathbf{r})) = \lambda\, g(\mathbf{p}, \mathbf{r})\, l(f(\mathbf{p}), f(\mathbf{r}))$$

Algorithm	Scenes from Middlebury					
	Barn1	Barn2	Bull	Poster	Venus	Sawtooth
FULL-BNV	3.5 %	3.1 %	0.7 %	3.7 %	3.4 %	3.3%
FULL-MF	4.0 %	5.4 %	0.7 %	3.4 %	4.4 %	3.8 %
GEO-BNV	**0.8 %**	**0.6 %**	0.4 %	1.1 %	**2.4 %**	**1.1 %**
GEO-MF	1.5 %	0.9 %	**0.3 %**	1.4 %	3.4 %	1.5 %
KAN-BNV	1.4 %	1.5 %	0.9 %	1.1 %	4.0 %	1.5%
KAN-MF	1.1 %	1.2 %	**0.3 %**	**0.9 %**	5.8 %	2.2 %

Figure 5. Error percentages for the different scenes of the Middlebury data set. The best performance for each image set is highlighted.

where g is defined as

$$g(\mathbf{p}, \mathbf{r}) = \begin{cases} 3 & \textit{if } |I_{ref}(M_{ref}\mathbf{p}) - I_{ref}(M_{ref}\mathbf{q})| < 5 \\ 1 & \textit{otherwise} \end{cases}$$

with $l(\mathbf{p}, \mathbf{r}) = |f(\mathbf{p}) - f(\mathbf{r})|$ for the maximum flow [19] formulation and $l(\mathbf{p}, \mathbf{r}) = \delta(f(\mathbf{p}) - f(\mathbf{r}))$ for graph cut formulation [2]. The parameter λ is user-defined. For each depth map computation, we chose the λ 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 [25, 21, 12].

When minimizing Eq. 6, a visibility mask must be keep for every voxel of the reconstruction volume, that is, for each $\mathbf{p} \in \mathcal{P}$ and $z \in \mathcal{Z}$. To reduce memory requirements and the number of iterations, we kept a single visibility history for each pixel p regardless of the disparity z, i.e. (5) becomes $H_i(\mathbf{p}, t) = \prod_{0 \leq k \leq t} V_i(\mathbf{p}|f^k(\mathbf{p}), f^k)$. This saves a lot of memory but the convergence is no longer guaranteed. We simply stop iterating when $H(\mathbf{p}, t) = H(\mathbf{p}, t-1)$ for all $p \in \mathcal{P}$. We observed that running the algorithm any longer only produce minor modifications to f^t. However, the number of pixels with final zero masks increases, usually in regions where the ordering constraint is broken. Pixels with zero masks are more prone to error, therefore we tried to improve results by adding a second step that reintroduces eliminated cameras. This step consisted in fixing to their final values the depth labels of the pixels with non-zero final camera masks. The history of the others was discarded and the volumetric visibility recomputed, considering only occlusion caused by the fixed pixels. Finally, an additional minimization was run to produce a better depth map.

5.1 Middlebury

This datasets from Middlebury [22] consists of 6 series of 9 images of size 434×383. We used images 0 to 7 in our experiments. The disparities between images 2 and 6 range from 0 to 19 pixels and 20 disparity steps were used. Since the ground truth was available for this dataset, we

Figure 6. Reference images for the Head and Lamp scene (left) and the Santa scene (right) from the Multiview Images database of the University of Tsukuba.

used it to compute error percentages when using the second image as the reference. We compared our method against Nakamura's [15] with a special choice of masks: either all the cameras to the left of the reference are visible or all the cameras to the right are. This specialized version of Nakamura is described in [10, 20]. The abbreviation used for this method is KAN. Our method is denoted by GEO. The results of GEO after one iteration are also shown under the label FULL, since this is a case where no occlusion modeling is made. We used 2 different stereo matchers: maximum flow [19] (MF) and graph cuts [2] (BNV). Results are shown in Figure 5. While KAN's modeling of occlusion achieves impressive results, our approach using the BNV stereo matcher perform better in 4 of the 6 sequences of images and were close to KAN in the other two. Oddly enough, in the Venus scene, KAN had a higher error rate than FULL, even though FULL is a simplified version of KAN (a single mask with all the cameras). Our algorithm takes an average of 8 iterations to converge, the improvement after just 4 is minimal.

5.2. Tsukuba Head and Lamp

This dataset is from the Multiview Image Database from the University of Tsukuba (see Figure 6). 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. We only used 5 images for each depth map computation. The reference image is the center one and the 4 supporting images are at an equal distance from it, arranged in a cross shape. In addition to those of GEO-BNV and GEO-MF, the results of GEO-BNV when using the recovery method described in section 5 are shown under the label "GEO-BNV pt". Some depth maps are shown in figure 7 and error precentages are shown in table 8. The entry KZ1 of the table comes directly from [12]. This method achieved a very low error rate. However, as the authors mentioned, the algorithm has trouble with low textured regions (the top right corner for instance), therefore the error is somewhat underestimated by the removal of an 18 pixel border in the ground truth. We also computed the error after removing the pixels breaking the ordering constraint, in particular part of the arm of the lamp. The mask was determined by re-projecting the ground truth in each supporting camera, hence it differs for the two baselines.

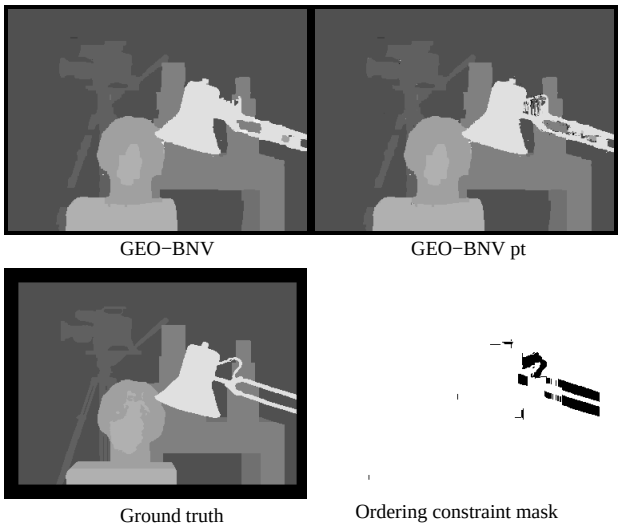

Figure 7. Depth maps for the Head and Lamp scene (Multiview Images database of the University of Tsukuba). Note for GEO-BNV how the errors are concentrated in regions breaking the ordering constraint. A mask of pixels breaking the ordering constraint for the smallest baseline is also shown.

Algorithm	Baseline	Error (whole image)	Error (mask)
GEO-BNV pt	1x	2.23%	1.53%
KZ1	1x	2.30%	2.01%
GEO-BNV	1x	2.46%	1.64%
GEO-MF	1x	3.42%	2.52%
GEO-BNV	2x	2.69%	2.11%
GEO-MF	2x	2.62%	1.28%

Figure 8. Percentages of error of the different algorithms for Head and Lamp scene, using 5 images. The right column contains the amount of error computed after the removal of the pixels breaking the ordering constraint, the left shows it for all the pixels.

GEO-BNV almost performed as well as KZ1; when removing pixels breaking the ordering constraint, it achieved a slightly lower error rate. For some algorithms, the error rate decreased for the larger baseline. This counter-intuitive behavior is explained by the fact that the matching cost function in the lamp region is less ambiguous when the baseline is larger. Figure 9 shows the stability to changes of the smoothing parameter of our algorithm using graph cuts, giving the error percentage for 6 values of this parameter.

5.3. Baseline test

As the baseline increases, the amount of occlusion in the scene increases as well. A stereo matcher not affected by occlusion would give identical depth maps for different

Algorithm	Smoothness parameter					
	1/30	1/10	1	2	3	4
GEO-BNV 1x	2.61	2.67	2.66	2.55	3.53	4.12

Figure 9. Resistance to change of the smoothing parameter for the Head and Lamp scene. The smoothing parameter increases by a factor of 120, while the error rate varies by less than 1.6% for the small baseline.

baselines. To measure the level of resistance to change of the baseline, for the different occlusion overcoming strategies, we introduce the notion of depth map incompatibility. A pixel $\mathbf{p}$ is incompatible in two depth maps i and j if

$$|f_i(\mathbf{p}) - f_j(\mathbf{p})| > 1$$

(a difference of 1 is meaningless as it could be the result of discretization errors). It is important to mention that a low incompatibility level is not necessarily a sign of low error level in the depth map. But the amount of occlusion increases with the baseline, and so should the error and incompatibility levels for stereo matchers that do not model occlusion. To test the stability of our algorithm, we used the Santa scenes from the Multiview Image Database of the University of Tsukuba (see figure 6). This dataset contains 81 images in a 9×9 grid and the focal distance of the camera was 10 mm with successive baselines of 20, 40, 60 and 80 mm. We only used 5 images in a cross shape configuration. Images were reduced by a factor of 2 to achieve a resolution of 320×240. Each depth map was computed using 23 disparity steps. Note the details on the right side of the hat and on the candle. Again, for each depth map, the smoothing parameter was adjusted to obtain the best possible performance. Since no ground truth was available, the choice was made by visual inspection of every depth maps.

The figure 11 contains bar charts of the percentages of pixels incompatible between the depth maps obtained for two baselines. In addition to GEO-MF and KZ1, results from the Nakamura approach [15, 20] using maximum flow (NAKA-MF) and graph cuts (NAKA-BNV) were also included. GEO-MF is twice as stable as NAKA-MF and yields less noisy depth maps. KZ1 and NAKA-BNV are less stable by a factor of 5 and more. The results for FULL-MF are again given. We can see in Figure 10 that GEO-MF achieves the best results for the third baselines. For the first baseline, KZ1, NAKA-MF and GEO-MF performed similarly. The running times for GEO-MF and GEO-BNV are respectively less than 5 and 9 minutes on a 2.0 GHz AMD Athlon(tm) XP 2600+.

6. Conclusion

We have presented a new framework to model occlusion in stereo by introducing geo-consistency. We also provided

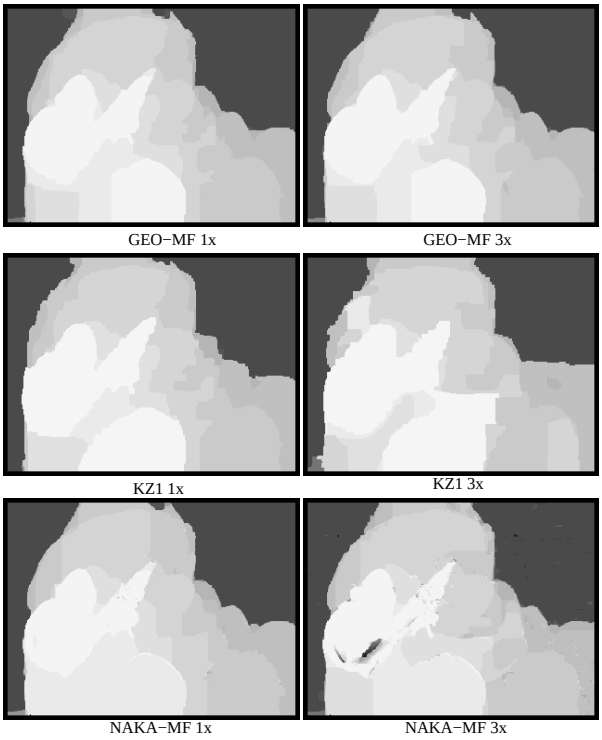

Figure 10. Depth maps obtained by 3 algorithms for 2 different baselines (1x and 3x) on the Santa scene (Multiview Image Database of the University of Tsukuba).

a way to apply this framework to add occlusion modeling to standard stereo algorithms. Rather than explicitly model occlusion, our iterative approach relies on geo-consistency of depth maps to determine visibility of cameras and to aggressively remove them to adjust the matching cost function to the scene structure and to the bias in the type of error committed by the stereo matcher. One of the main characteristic of our approach is that we do not discriminate between occluders and occludees. Our implicit occlusion model is successful in obtaining sharp and well-located depth discontinuities and allows the use of efficient standard stereo matching algorithms. Moreover, our framework does not add any parameter or constraint to the matching process. The validity of our framework has been demonstrated on standard datasets with ground truth and was compared to other state of the art occlusion models for multiple view stereo. Our approach was also tested on increasingly wider baselines in order to demonstrate its stability to increasing amount of occlusion in the scene. While the validity of our framework has been demonstrated using two stereo matching algorithms, it is general enough to be applied to others. It is not limited to regular grids of cameras and also works with other camera configurations.

As for future work, better approach to recover from error in scene breaking the ordering constraint should be in-

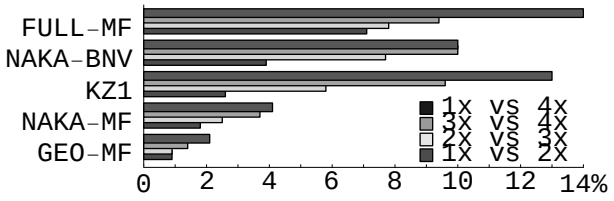

Figure 11. Resistance to baseline change for 5 algorithms for the Santa scene (Multiview Image Database of the University of Tsukuba); each bar represents a percentage of incompatible pixels between depth maps obtained for two different baselines.

vestigated. Also, the extension of this occlusion model to full volumetric reconstruction, where occlusion becomes the dominant problem, should be investigated.

7. Acknowledgment

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

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

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

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

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

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

[9] 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.

[10] 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.

[11] 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.

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

[13] J. Krol and W. van der Grind. The double-nail illusion. *Perception*, 11:615–619, 1982.

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

[15] 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.

[16] 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.

[17] 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.

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

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

[20] 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.

[21] 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.

[22] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

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

[24] 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.

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

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

[27] G. Vogiatzis, P. Torr, S. M. Seitz, and R. Cipolla. Reconstructing relief surfaces. In *Proc. of the IEEE Conf. on British Computer Vision*, September 2004.