Département d'Informatique et de recherche opérationnelle (DIRO)

Rapport technique 1273
Février 2006

Geo-consistency for Multi-Camera Stereo

by

Marc-Antoine Drouin
Martin Trudeau
Sébastien Roy
{drouim,trudeaum,roys}@iro.umontreal.ca
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, Canada H3C 3J7

Le Département d'Informatique et de recherche opérationnelle fait partie de la Faculté des Arts et Sciences de l'Université de Montréal.

ABSTRACT

This paper introduces the concept of geo-consistency to overcome the occlusion problems associated with wide-baseline multiple-camera stereo. By expressing more formally the geometric relationship between occlusion and visibility, it provides a new tool to understand and compare the behavior of stereo algorithms. In this context, we present a new visibility mask handling that can render most regular stereo algorithms ”occlusion aware”. Rather than explicitly modeling occlusions in the matching cost function, it detects occlusions in the depth map obtained from regular efficient stereo matching algorithms. The algorithm gradually modifies the matching cost function according to the history of inconsistencies in the depth map, until convergence. The final depth map is guaranteed to preserve the coherence between camera visibility and geometry. We also provide a fast specialized stereo algorithm, based on Iterative Dynamic Programming, that model visibility very efficiently. It is based on partial geo-consistency, in which some of the visibility information is always known exactly. Geo-consistency makes it easier to express the ordering constraint. This is helpful for the detection of the parts of a scene not fulfilling it. These zones are the most subject to error and should be handled separately. We observed that our fast IDP algorithm is especially well-suited for high discontinuity areas. For experiments, we applied our general occlusion algorithm to two common graph-theoretic stereo algorithms. The validity of our framework is demonstrated using real imagery taken with various baselines, as it is known that occlusion increases with it.

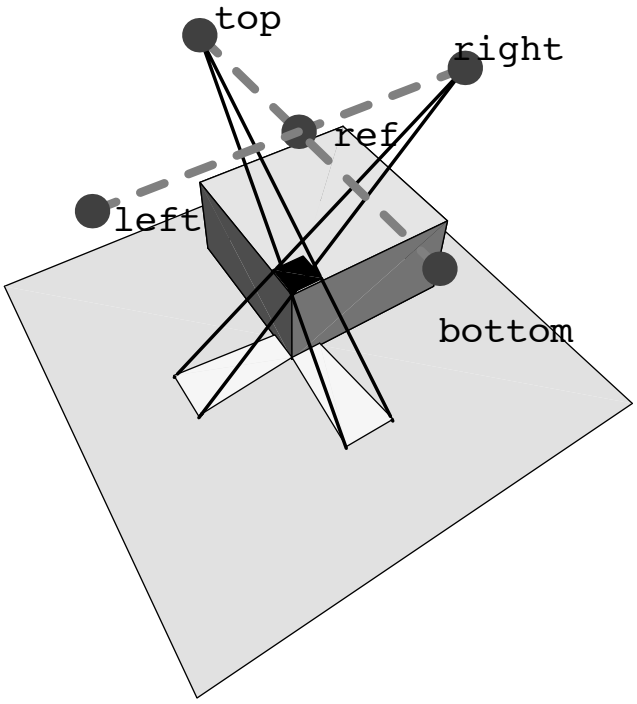

FIG. 1 – A cross-shaped camera configuration. The reference and supporting cameras are labeled *ref*, *left*, *right*, *top* and *bottom* respectively. Examples of occluder and occluded pixels are shown in black and white respectively.

1 Introduction

The goal of binocular stereo is to reconstruct the 3D structure of a scene from two views. When the baseline is wider, the problem of occlusion, that is often considered negligible with small baseline configurations, can become severe and limits the quality of the obtained depth map. The depth map is reconstructed from the point of view of a reference camera. Occlusion occurs when part of a scene is visible in the reference but not in some other supporting camera (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 when the correspondence is being established, as it is the final goal of the algorithm. We proposed two novel multiple-camera stereo algorithms. The first one 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 matching process. This makes possible the use of a standard efficient algorithm during each iteration, instead of tackling a very difficult optimization problem. Furthermore, the final solution is always guaranteed to preserve the consistency between the recovered visibility and geometry, a property we call geo-consistency. Our framework always recovers a geometry preserving the ordering constraint between the reference and the supporting cameras. The ordering constraint

simply states that during the scan of 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 2-left). This constraint holds for most scenes (see Fig 2-right) [15]. While this constraint is broken in some rare cases, it remains a powerful tool when dealing with occlusion and outliers. Futhermore, our framework can detect the regions of the depth map suspected to break the ordering constraint. Those regions are the most subject to errors and we provide a way to handle them separately. In this paper, our framework uses the maximum flow [24, 25] and graph cut [3] formulations to solve each iteration. It is general enough to be used with many other stereo algorithms. A survey paper by Scharstein and Szeliski [26] compared various standard algorithms.

We also propose a second algorithm that uses iterative dynamic programming (IDP) [17], a fast method for computing disparity maps. When applied to ordinary stereo, IDP 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 multiple-baseline stereo in a way that is impossible with Graph Cut. Interestingly, dynamic programming makes it possible to compute exactly part of the visibility information ,thereby preserving the geo-consistency between the recovered geometry and part of the camera visibility. The remaining visibility is obtained through heuristics. The proposed auxiliary algorithm is thus a hybrid between fast heuristics approaches and slower geo-consistent ones. The hybrid stereo matcher is fast and can also be used as an auxiliary stereo matcher in our implicit method to improve the quality of the disparity map in the regions suspected of breaking the ordering constraint.

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. Our hybrid algorithm is presented in Section 5. Experimental results are presented in Section 6.

2 Previous work

In a recent empirical comparison of occlusion overcoming strategies for 2 cameras, Egnal [5] enumerates 5 basic ones : left-right checking, bimodality test, goodness Jumps constraint, duality of depth discontinuity and occlusion and finally 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, 7, 32, 12]. These methods are binocular in nature and do not generalize well to the case of multiple arbitrary cameras. Other algorithms use dynamic programming [20, 9, 4, 8] 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 neither generalizes to multiple-camera configurations.

When extending binocular stereo to multiple cameras, the amount of occlusion increases since each pixel of the reference 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 regular grids of cameras [19]. Okutomi and Kanade have proposed a matching cost function designed to reduce ambiguity in stereo with multiple cameras having collinear optical centers[21]. However, their approach does

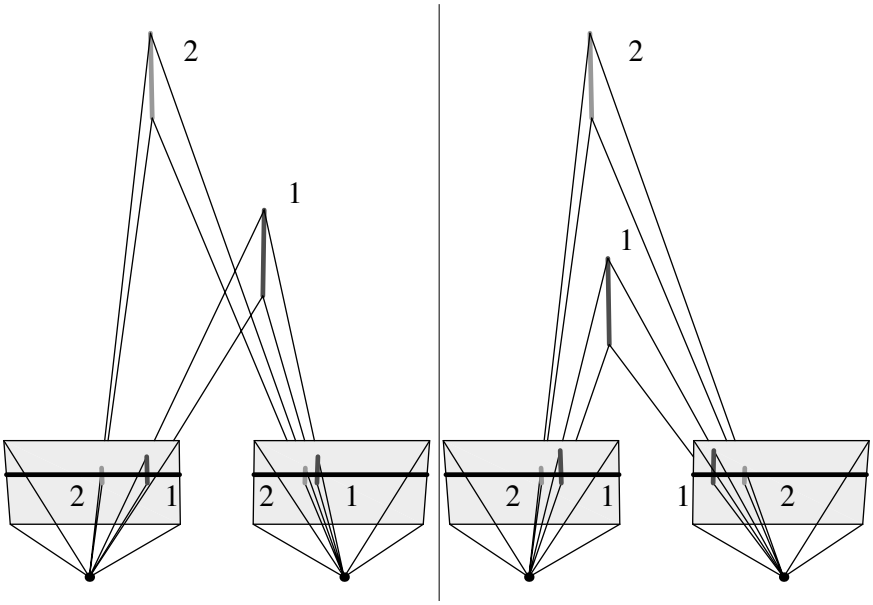

FIG. 2 – **Left)** The ordering constraint is satisfied. In this camera configuration, the epipolar lines are parallel to the X-axis. Line 2 is located to the left of line 1 in both images. **Right)** The ordering constraint is broken, line 2 appears to the left of line 1 in one image and to the right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

not model occlusion. Some researchers have proposed specially designed algorithms to cope with occlusion in multiple-camera configurations. Amongst these, Kang et al. [12] proposed spatial shiftable windows, half sequences (which is a particular case of [19]) , adaptative window sizes and explicit occlusion states.

They also proposed a visibility reasoning approach, but it does not perform better than adaptive windows. This scheme was based on the hypothesis that a low matching cost function implies the absence of occlusion. This hypothesis is also made in [19, 18, 22, 23]. In contrast, we do not rely on such an assumption. In [33], 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 [16, 28], *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 [6] uses the visibility information from its evolving reconstructed surface to explicitly model occlusion. In [14], 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.

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

$$e(\mathbf{p}, z, \mathbf{m}) = \frac{\mathbf{m} \cdot C(\mathbf{p}|z)}{|\mathbf{m}|} \quad \text{for } \mathbf{p} \in \mathcal{P},\ z \in \mathcal{Z},\ \mathbf{m} \in \mathcal{M} \tag{3}$$

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 $C_c(\mathbf{q}) = (I_{ref}(\mathbf{M}_{ref}\mathbf{q}) - I_c(\mathbf{M}_c\mathbf{q}))^2$ where $\mathbf{M}_{ref}$ and $\mathbf{M}_c$ are projection matrices from the world to the images of camera *ref* and c respectively, and I_{ref} and I_c 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

$$g(\mathbf{p}) \leq V(\mathbf{p}|f(\mathbf{p}), f) \tag{4}$$

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 [19]. Our problem becomes the minimization of Eq. 2 in f and g, with the constraint that f is geo-consistent with g.

3.1 Previous methods in a geo-consistency context

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 possible the precomputation of a smaller subset of $\mathcal{M}$. Unfortunately, even with a small number of masks, it is still not practical to minimize in f and g simultaneously.

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 the 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 [19]. 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 [19, 18, 22, 12]. Since the selected masks and the disparity map do not always respect Eq. 4, we call these methods *heuristic*. As already mentioned, these heuristic approaches rely on the hypothesis that photo-consistency implies visibility. The Figure 3 suggests that this is not always true. Using the matching cost function and images from the Middlebury comparative study [26], 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.

Other approaches try to minimize directly Eq. 2 in f and g, subject to the constraint of Eq. 4. Such *geo-consistent* methods have to solve a substantially more difficult problem than *heuristic* ones. In [16, 28], visibility-based methods are introduced. The matching cost incorporates the visibility information as a photo-consistency matching criteria, thereby implicitly modeling occlusion

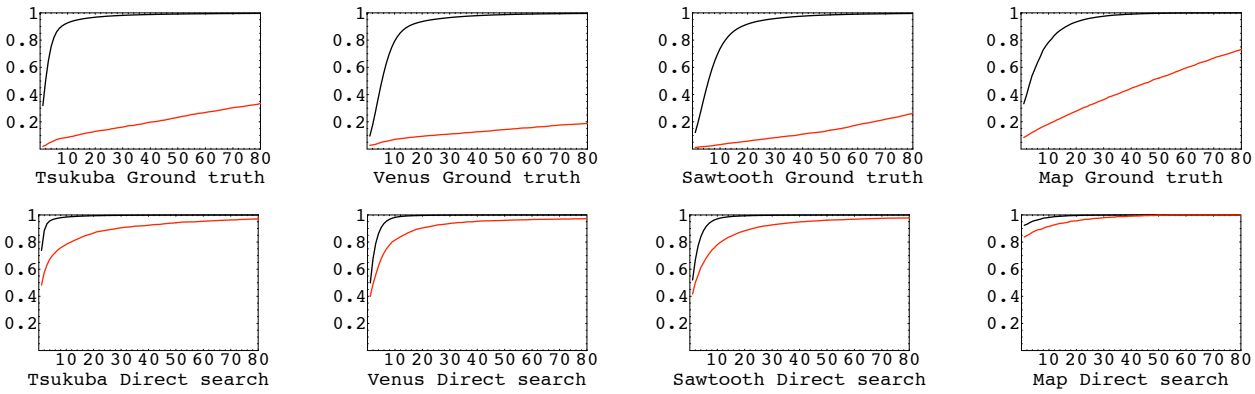

FIG. 3 – Cumulative histograms of likelihood values of occluded (red) and non occluded (black) pixels for the sequences of the Middlebury comparative study [26], for the **top)** real depths **bottom)** most likely depths (direct search).

in the reconstruction process. Space carving can be seen as a greedy algorithm that minimizes Eq. 2 subject to the constraint of Eq. 4 without smoothing. Similarly, a level-set method [6] 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 are consistent with the recovered surface. This algorithm jumps from one configuration respecting Eq. 4 to another. The formulation imposes strict constraints on the form of the smoothing term, constraints that will not apply to our method.

Ideally, we would like to benefit from the simplicity of heuristics without being affected by the similarity between the matching cost distribution of occluded and non occluded pixels and we would like to recover a geo-consistent solution without having to solve an intractable problem. We propose two new approaches. The first one, iteratively computes depth and visibility. At each iteration, a depth map is obtained from standard stereo algorithm. From this depth map the visibility is updated for the next iteration. While intermediate depth maps are not *geo-consistent*, the final one is guaranty to be *geo-consistent*. The second approach, rapidly finds a solution for which *geo-consistent* is always preserved for a sub-set of the supporting cameras, but not necessarily for the others.

4 An implicit geo-consistent algorithm

In the first approach, 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

$$\sum_{\mathbf{p}\in\mathcal{P}} e(\mathbf{p}, f^t(\mathbf{p}), V(\mathbf{p}|f^t(\mathbf{p}), f^{t-1}) + \textit{smoothing} \tag{5}$$

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 [12].

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. In order to guarantee convergence, we could fix the label f^t of the best matching pixels. g^t would be computed by considering only occlusion caused by these pixels. This process would be repeated until a certain convergence criteria is met. This strategy was tried out in [12], but the authors did not see any improvement when they added it to their algorithm. Again, Figure 3 suggests an explanation by illustrates the fact that photo-consistency does not imply visibility unless the ground truth is known. To guarantee convergence, we introduce instead a *visibility history mask* independent of the matching cost function value, namely

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

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

$$H_c(\mathbf{q}, t) = \prod_{0\leq s\leq t} V_c(\mathbf{q}, f^s) = \min_{0\leq s\leq t} V_c(\mathbf{q}, f^s). \tag{6}$$

Substituting H for V in Eq. 5, we obtain the new problem energy function

$$E^t_H(f^t) = \sum_{\mathbf{p}\in\mathcal{P}} e(\mathbf{p}, f^t(\mathbf{p}), H(\mathbf{p}|f^t(\mathbf{p}), t-1) + \textit{smoothing}. \tag{7}$$

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 solution to the same minimization problem, and the process has stabilized. We see that the number of iterations is bounded by $N \cdot \#\mathcal{P} \cdot \#\mathcal{Z}$ (# denotes the cardinality as usual).

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 $\mathbf{p}$:

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

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 enforce strict geo-consistency during the optimization process [16, 6, 14].

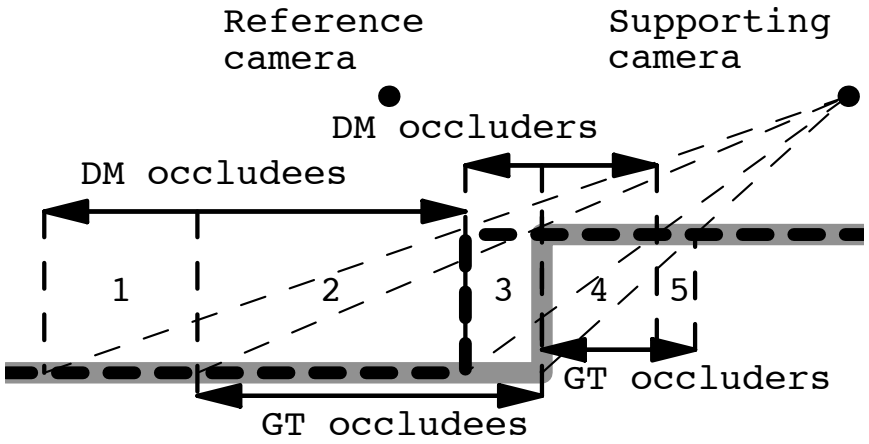

FIG. 4 – Effect of object enlargement on classification of occluders and occludees of a scene viewed by 2 cameras. The ground truth is in thick gray and the depth map in thick dashes. Occluders and occludees are shown for both ground truth (GT) and computed depth map (DM). Respectively, the 5 zones represent **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.

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 4). [fixe me] To validate this assertion, we used the results of two of the best stereo matchers evaluated with the Middlebury dataset. [26, 29, 3]. The Graph Cut algorithm was ranked the best stereo matcher in two comparative studies [3, 30, 26]. The Belief Propagation algorithm appeared later and achieved an even lower error rate [29]. 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 (Table 1). 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

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

algorithm	scene											
	tsukuba Head and Lamp			Venus			Sawtooth			Map		
	Real (ground truth) status of pixels classified from depth map as occluders											
	occludee	occluder	regular	occludee	occluder	regular	occludee	occluder	regular	occludee	occluder	regular
bp [29]	**44.8**	16.3	38.9	**25.9**	3.7	70.4	**42.6**	3.8	53.6	**60.9**	10.3	28.8
bnv [3]	**50.4**	15.4	34.2	**12.6**	61.0	26.4	**42.6**	4.3	53.3	**15.4**	64.6	20.0
	Real (ground truth) status of pixels classified from depth map as occludees											
	occludee	occluder	regular	occludee	occluder	regular	occludee	occluder	regular	occludee	occluder	regular
bp [29]	15.5	5.9	**76.6**	4.5	0.6	**94.9**	5.5	1.1	**93.4**	7.7	8.6	**83.7**
bnv [3]	16.4	5.8	**77.8**	65.9	1.9	**32.3**	7.2	1.1	**91.7**	68.4	2.5	**29.1**
	Real (ground truth) status of pixels classified from depth map as regulars											
	occludee	occluder	regular	occludee	occluder	regular	occludee	occluder	regular	occludee	occluder	regular
bp [29]	1.0	**2.0**	97.0	0.9	**1.4**	97.7	0.5	**1.5**	98.0	1.8	**4.9**	93.3
bnv [3]	1.0	**2.0**	96.9	0.5	**0.7**	98.8	0.5	**1.5**	98.0	0.8	**1.8**	97.4

TAB. 1 – Real status in percentages of pixels according to their classification. Examples from the Middlebury comparative study [26]. In bold are the misclassifications when overestimating the disparity of occluded pixels.

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

$$V'_c(\mathbf{p}|f(\mathbf{p}), f) \leq V_c(\mathbf{p}|f(\mathbf{p}), f) \quad \forall \mathbf{p} \in \mathcal{P}, \quad 1 \leq i \leq N. \tag{8}$$

The pseudo-visibility masks V'_c 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 c : one with an ordinary Z-buffer and one with a reverse Z-buffer test. Two depth maps L_c^f and G_c^f are thus obtained. 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 $V'_c(\mathbf{q}, f)$ can therefore be computed as

$$V'_c(\mathbf{q}, f) = \delta\left(L_c^f(\mathbf{T}_c\mathbf{q}) - G_c^f(\mathbf{T}_c\mathbf{q})\right) \tag{9}$$

where δ is 1 at 0 and 0 elsewhere.

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 c at iteration t, then according to this definition of pseudo-visibility (and using an opaque mesh), the history mask is updated to $H_c(\mathbf{p}|f^{t+1}(\mathbf{p}), t) = 0$ for all $\mathbf{p}$ in $\mathcal{O}$. After convergence for the final configuration f^T we have $H(\mathbf{p}|f^{T+1}(\mathbf{p}), T) = H(\mathbf{p}|f^T(\mathbf{p}), T-1)$ for all $\mathbf{p}$. In particular $H_c(\mathbf{p}|f^{T+1}(\mathbf{p}), T) = 0$ for all $\mathbf{p} \in \mathcal{O}$. Since the offending camera c was not used to compute the final solution, the ordering constraint is respected between the reference camera and the supporting camera c.

It is possible for a voxel to have all its cameras removed, i.e. $H(\mathbf{p}|z, t-1) = \mathbf{0}$ even if $V(\mathbf{p}|z, t-1) \neq \mathbf{0}$ (Figure 1-left). 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(Eq. 7), where t' is the largest index such that $H(\mathbf{p}|z, t') \neq \mathbf{0}$. It this case, depth is assigned only using the neighborhood through smoothing. This situation usually occurs in regions breaking the ordering constraint.

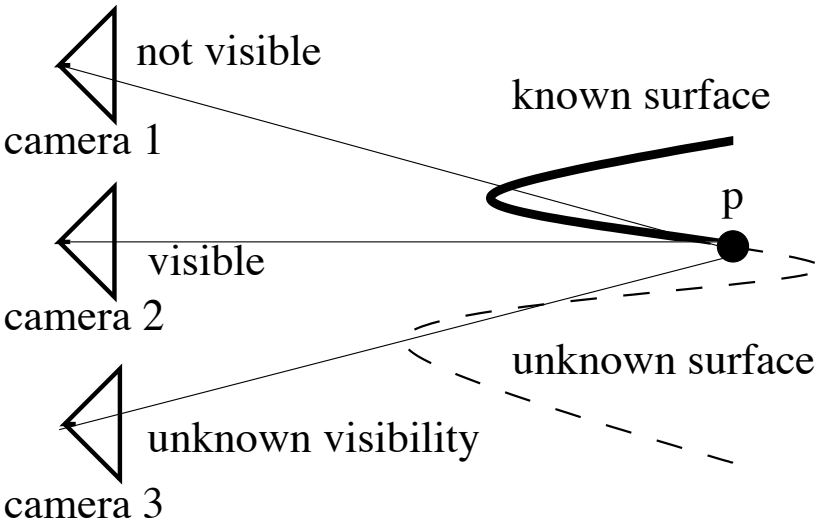

FIG. 5 – Example of the use of partial disparity map information. It is known that the point p is not visible from camera 1 but it is from camera 2. The partial disparity map does not allow us to tell for camera 3.

Our second approach does not impose the ordering constraint on the final solution but finds a depth map for which *geo-consistency* is guaranty to be preserved only for a sub-set of the supporting camera.

5 A fast partially geo-consistent algorithm

The second proposed algorithm goes through the reference image pixel by pixel, building potential depth 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 $\mathcal{C}_g$ of the set $\mathcal{C}$ of all supporting cameras. This visibility information comes from the partial knowledge of the depth map (Figure 5). We can build the set $\mathcal{M}_g$ of the masks using only cameras in $\mathcal{C}_g$. Given that each camera can be used or not, and discarding the empty mask, we are left with $2^{\#\mathcal{C}_g} - 1$ masks. When selected, a mask from $\mathcal{M}_g$ will always be *geo-consistent*, independently of the visibility status of the cameras not in $\mathcal{C}_g$. For the cases where no camera in $\mathcal{C}_g$ is visible, we must select a mask in another set $\mathcal{M}_h$ that only uses cameras in the subset $\mathcal{C}_h = \mathcal{C} - \mathcal{C}_g$. 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 make a selection. We expect the choice between $\mathcal{M}_g$ 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 depth map f constructed up to the previous pixel, if $\mathbf{p}$ is visible by at least one camera in $\mathcal{C}_g$, the mask of $\mathbf{p}$ at z is set to a partial visibility $V''(\mathbf{p}, z, f)$ with

each component defined as

$$V_c''(\mathbf{p}, z, f) = \begin{cases} V_c(\mathbf{p}, z, f) & \text{if } c \in \mathcal{C}_g \\ 0 & \text{otherwise.} \end{cases}$$

If $\mathbf{p}$ is not visible by any camera in $\mathcal{C}_g$, its mask is defined as $\arg\min_{m \in \mathcal{M}_h} e(\mathbf{p}, z, m)$. 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. 3 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

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

Since many visibility evaluations will be performed during the solving process, we would like to reduce its computational cost. In order to accomplish this, we work with rectified images and disparities instead depths. We assume that all cameras have collinear optical centers. This constraint will be progressively lifted in section 5.3 and 5.4. For more details about image rectification, see [31]. In the following discussion, we assume the cameras to be rectified. To compute a disparity map for a reference image, the set of reference pixels $\mathcal{P}$ remains the same, but the set of depth labels $\mathcal{Z}$ is replaced by a set of disparity labels $\mathcal{D}$. A $\mathcal{D}$-configuration $f : \mathcal{P} \mapsto \mathcal{D}$ associates a disparity label to every pixel. The relation between a disparity d and the depth z with respect to camera c is simply

$$d = \frac{B_c\, L}{W\, z}$$

where B_c is the baseline between the reference and supporting camera c, L is the focal length (assumed to be the same for all cameras) and W the width of a pixel on the CCD (again assumed to be constant). In a multiple-camera configuration, the disparities of a 3D point between the reference and the different supporting cameras vary. The disparity configuration f is expressed with respect to a fixed supporting camera c_0. The disparity for a reference pixel $\mathbf{p}$ between the reference and another supporting camera c is $s_c \cdot f(\mathbf{p})$ where the baseline scale factor s_c is equal to B_c/B_0. The pointwise likelihood term of Eq. 3 becomes

$$e'(\mathbf{p}, d, \mathbf{m}) = \frac{\mathbf{m} \cdot C'(\mathbf{p}, d)}{|\mathbf{m}|} \quad \text{for } \mathbf{p} \in \mathcal{P},\, d \in \mathcal{D},\, \mathbf{m} \in \mathcal{M} \tag{10}$$

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. Our choice of $\mathcal{M}_g$ and $\mathcal{M}_h$ ensures that $|\mathbf{m}| \geq 1$. In order to simplify the discussion, we will always consider disparities to be positive and normalized with respect to the camera c_0. Disparity and visibility will be solved simultaneously using dynamic programming, taking into account long range visibility interactions.

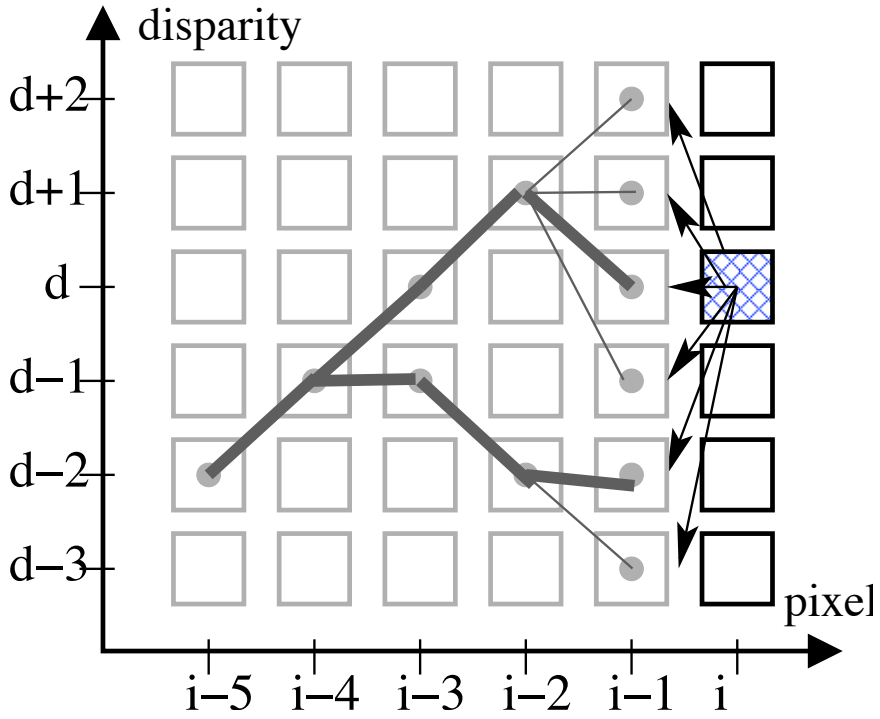

FIG. 6 – DP matching process. To determine the best depth map up to pixel i with depth d, for the different depth values of $i-1$, we look at the best solution up to $i-1$ available by construction.

5.1 Optimizing disparity and visibility

The stereo matching uses Dynamic Programming (DP) applied to epipolar lines, assumed here to be horizontal. We illustrate the process for a left-right epipolar line, with a center reference image and multiple left and right supporting images. When dynamic programming proceeds along, the computation of the disparity at pixel i can rely on the knowledge of the disparities of all preceding pixels (Figure 6). A similar strategy for binocular stereo was presented in [1]. In the following discussion, a left to right order is assumed, but the reverse order is of course possible. 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 (Figure 5). When going from left to right, $\mathcal{C}_g$ consists of all the cameras located to the left of the reference and $\mathcal{C}_h$ of all the ones to the right. Consequently, $\mathcal{M}_g$ contains only masks using one or more cameras to the left of the reference. The set $\mathcal{M}_h$ contains only masks with exactly one camera to the right. 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 $f_{i,d}$. Two sample disparity maps $f_{i-1,d}$ and $f_{i-1,d-2}$ are highlighted in Figure 6. The table t' is used to compute the different $f_{i,d}$'s.

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

$$\begin{aligned} t(0,d) &= e'(0,d,m_0) \\ t'(0,d) &= d \\ t(i,d) &= \min_{d' \in \mathcal{D}} \left(e_v(i,d,d') + s(i-1,i,d',d) + t(i-1,d')\right) \\ t'(i,d) & \quad \text{is the index of the minimum defining } t(i,d) \end{aligned}$$

where m_0 is a visibility mask with all cameras in $\mathcal{C}_g$ visible and all others invisible and

$$e_v(i,d,d') = \begin{cases} e'(i,d,(O_1(i,d,d'),\cdots,O_N(i,d,d'))) & \text{if } \exists c \in \mathcal{C}_g : O_c(i,d,d') = 1 \\ \min_{m \in \mathcal{M}_h} e'(i,d,m) & \text{otherwise.} \end{cases}$$

N is again the number of cameras other than the reference. $O_c(i,d,d')$ is a special visibility function that is 1 iff camera c in $\mathcal{C}_g$ is visible and 0 otherwise (see next section). 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

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

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 $O_c(i,d,d')$ for all d' and all camera c 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 depends on every pixel located to its left. As mentioned before, s may include visibility as well as disparity smoothing.

5.2 Computing visibility

When computing the visibility for each camera in $\mathcal{C}_g$, the depth 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 $O_c(i, d, d')$ as the visibility of pixel i at disparity d in camera c, supposing j has disparity $f_{i-1,d'}(j)$ for all pixels $j < i$. In the discontinuous case, O_c is defined as

$$O_c(i, d, d') = \begin{cases} 0 & \text{if camera } c \in \mathcal{C}_h \\ 0 & \text{if } c \in \mathcal{C}_g \text{ and } i + s_c\, d = j + s_c\, f_{i-1,d'}(j) \text{ for some } j < i \\ 1 & \text{otherwise.} \end{cases} \tag{11}$$

The visibility function takes the value 0 when occlusion occurs or when camera c is not in $\mathcal{C}_g$. It takes the value 1 when the left camera c is visible. In the continuous case, we can lower the complexity by introducing the function O'_c defined as

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

This function can be computed inductively using the relations

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

Now O_c can be simply defined as

$$O_c(i, d, d') = \begin{cases} 0 & \text{if camera } c \in \mathcal{C}_h \\ 0 & \text{if } c \in \mathcal{C}_g \text{ and } O'_c(i-1, d') - (i + s_c\, d) \ge 0 \\ 1 & \text{otherwise.} \end{cases} \tag{12}$$

Using a continuous mesh is equivalent to imposing the ordering constraint [5] 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 6, considering disparity maps continuous does not affect the quality of the reconstruction in scenes that break the ordering constraint. The relations for right to left, top to bottom and bottom to top minimization are obtained similarly. Care should be taken to ensure proper handling of the disparity sign.

5.3 Visibility-aware iterative dynamic programming

In the previous section, we discussed visibility computation along an epipolar line. When all cameras have optical centers located in a two dimensional grid configuration, such as a 5-camera cross illustrated in Figure 7, we can use the solution of the previous lines to compute the visibility of one of the cameras not belonging to the epipolar line being processed. The visibility function O_c for such a camera is computed in a fashion similar to that of the camera with exact visibility

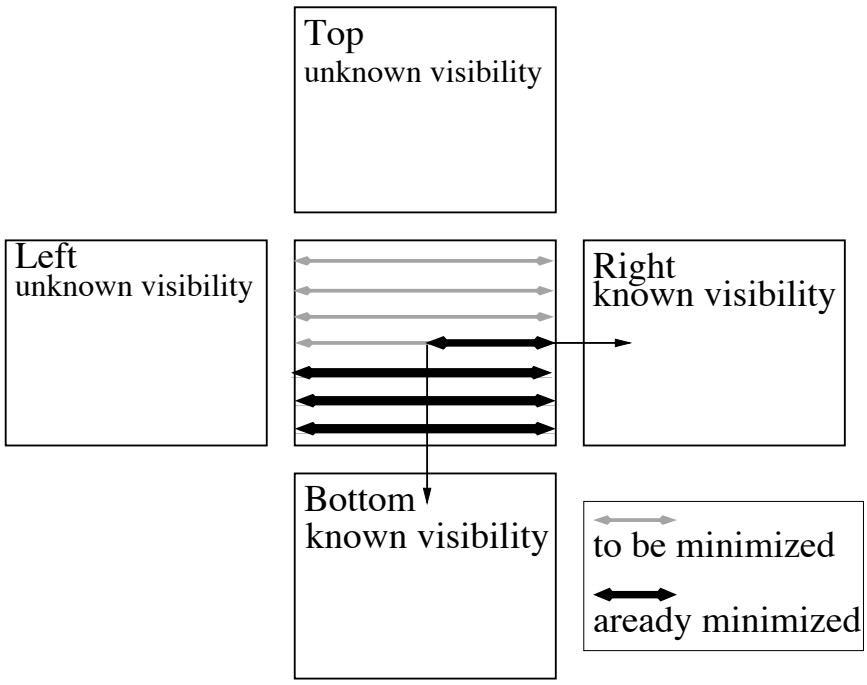

FIG. 7 – Optimization of a line : the visibility information is available for the right and the bottom cameras. For the current line, the right camera visibility is computed simultaneously with the disparity. The bottom camera visibility depends on the disparity of the previous lines which is fixed.

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

TAB. 2 – Visibility masks and their status depending on the current step for a 5-camera cross configuration. 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 $\mathcal{C}_g$. The camera order in a mask is left, right, top and bottom.

along the current epipolar line. General camera configurations are discussed in section 5.4. 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 the latter, the disparity map is part of the minimization process, for the former it is fixed (Figure 7).

In order to apply smoothing across epipolar lines, we use Iterative Dynamic Programming (IDP), proposed by Leung *et al.*[17]. 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 the first step, illustrated in Figure 7, 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 the 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 $\mathcal{C}_g$ and $\mathcal{M}_g$ vary from one step to the next. Table 2 shows the different masks for which the visibility is either *geo-consistent* or *heuristic* depending on the current step for a 5-camera cross-shape configuration.

We also propose a different initialization ; in [17], the disparity is initialized to a constant value. However, we do not use any prior disparity solution, the spatial smoothing is restricted 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 $\mathcal{C}_g$ 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 exact 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 8 or 12 iterations since changes are

minimal after that. Even after one iteration, the algorithm provides high quality disparity maps.

When using the visibility function of Eq. 12 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. 11, the asymptotic complexity increases to $\Theta(\#\mathcal{L}\,\#\mathcal{P}\,\#\mathcal{D}^2)$ where $\#\mathcal{L}$ is the highest number of pixels on any line. In our experiments, a continuous mesh representation was always used.

5.4 Arbitrary camera configuration

When the optical centers of the cameras are not located in a two dimensional grid, the approach described in the previous section is no longer suitable. At each step, in a general configuration, there can be at most one supporting camera for which the visibility is not fixed before processing a given line. The number of steps in an iteration increases and is the same as the number of supporting cameras. A mean direction between the reference and each supporting camera is computed, after which the images from all cameras are rectified with respect to it. For more details about image rectification, see [31]. With the camera configuration of the previous section only one rectification was required. In the current configuration at each step, a different rectified image of the reference camera is used. The visibility function for the supporting camera used to rectify the reference can be efficiently computed as described in section 5.2. Visibility computation for other cameras in $\mathcal{C}_g$ become more time consuming and may required rendering techniques. Applying smoothing across different epipolar lines is more complicated. When solving lines in an ordered fashion, disparity information is known for one of the neighbor lines (see Figure 7), but not for the others. To apply smoothing across epipolar line, knowledge of the disparity map of the previous step or iteration is thus required. With the camera configuration of the previous section, this was straightforward, since only one rectified image of the reference camera is used. This is no longer the case since multiple rectifications are made beforehand. Proper reprojection must be done between the image of the reference camera of the current step and that of the previous step in order to obtain the disparity information required to apply smoothing across epipolar lines.

6 Experimental results

In all our experiments, the matching cost function was the same for all algorithms, that of [14] which is based on [2]. 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 [14] :

$$s(\mathbf{p},\mathbf{r},f(\mathbf{p}),f(\mathbf{r})) = \lambda\, g(\mathbf{p},\mathbf{r})\, l(f(\mathbf{p}),f(\mathbf{r})) \tag{13}$$

where g is defined as

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

with $l(\mathbf{p},\mathbf{r}) = |f(\mathbf{p}) - f(\mathbf{r})|$ for the linear model used by the maximum flow [24] formulation and $l(\mathbf{p},\mathbf{r}) = \delta(f(\mathbf{p}) - f(\mathbf{r}))$ for the Potts model used by Graph Cut[3]. As previously mentioned, we added to our partially geo-consistent method a visibility smoothing, taking the value 0 if the mask of the current pixel and that of its neighbor are both in $\mathcal{M}_g$ 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 [17]). λ and γ are user-defined parameters. For each depth map computation, we chose the λ and γ that performed best. A pixel disparity is considered erroneous if it differs by more than 1 from the ground truth. This error measurement is compatible with the one used in two comparative studies for 2-camera stereo [30, 26, 14].

For our implicit algorithm, when minimizing Eq. 7, a visibility mask must be kept 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, we kept a single visibility history for each pixel $\mathbf{p}$ regardless of the disparity z, i.e. (6) becomes $H_i(\mathbf{p},t) = \prod_{0\leq s\leq t} V_i(\mathbf{p}|f^s(\mathbf{p}), f^s)$. This saves a lot of memory but the convergence is no longer guaranteed. We simply stopped iterating when $H(\mathbf{p},t) = H(\mathbf{p},t-1)$ for all $\mathbf{p} \in \mathcal{P}$. We have observed that running the algorithm any longer induced only minor modifications to f^t. Even with this simplification, the method produced high quality depth maps.

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. The first attempt consisted in fixing to their final value 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 using an auxiliary stereo matcher was run to produce a better depth map. Unfortunately, the error reduction was limited.

In the second attempt, we took into account the fact that using a continuous mesh enforces the ordering constraint, and that this constraint is often not respected by pixels with zero masks. Hence, we enlarged the set of pixels not fixed to include any pixel potentially blocked in one camera by another without camera. So to each pixel without camera and each supporting camera, we added a stripe of pixels along the epipolar line. The length of this stripe is the total number of disparities (supposing we are solving for disparities and not depths). Finally, we ran on this enlarged set of pixels an auxiliary stereo matcher, such as our fast hybrid algorithm, which does not enforce the constraint. This second attempt yielded an important error reduction.

6.1 Middlebury

This datasets from Middlebury [27] consists of 6 series of 9 images of size 434×383 with colinear optical centers. 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, we used it to compute error percentages when using image number 2 as the reference. We compared our method against Nakamura's [19] 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. This specialized version of Nakamura is described in [12, 18]. The abbreviation used for this method is KAN. Our

algorithm	smoothing model	Sequences from Middlebury						average
		barn1	barn2	bull	poster	venus	sawtooth	
FULL-BNV	Potts	3.5 %	3.1 %	0.7 %	3.7 %	3.4 %	3.3%	3.0%
FULL-MF	Linear	4.0 %	5.4 %	0.7 %	3.4 %	4.4 %	3.8 %	3.6%
FULL-IDP	Potts	3.0 %	4.9%	1.2%	6.0 %	5.8%	3.7%	4.1%
FULL-IDP	Linear	4.0 %	5.9%	1.3 %	4.7 %	4.7%	4.5%	4.2%
GEO-BNV	Potts	0.8 %	**0.6 %**	0.4 %	1.1 %	**2.4 %**	1.1 %	1.3%
GEO-MF	Linear	1.5 %	0.9 %	**0.3 %**	1.4 %	3.4 %	1.5 %	1.5%
KAN-BNV	Potts	1.4 %	1.5 %	0.9 %	1.1 %	4.0 %	1.5%	1.7%
KAN-MF	Linear	1.1 %	1.2 %	**0.3 %**	**0.9 %**	5.8 %	2.2 %	1.9%
HYBRID-IDP	Potts	**0.7 %**	3.9 %	0.8 %	4.0 %	5.3%	**1.0 %**	2.6%
HYBRID-IDP	Linear	2.0%	5.2%	0.6 %	3.0 %	4.5 %	2.7%	3.0%
KAN-IDP	Potts	1.6 %	6.0 %	1.9 %	4.5 %	7.4%	2.2 %	3.9%
KAN-IDP	Linear	3.5%	8.0 %	1.4 %	7.2 %	5.6 %	4.6 %	5.1%

TAB. 3 – Error percentages for the different scenes of the Middlebury dataset. The best performance for each image set is highlighted.

proposed method is denoted by GEO. The results of GEO after one iteration are also shown under the label FULL. This is a case where no occlusion modeling is made. We used 2 different stereo matchers : maximum flow [24] (MF) and Graph Cut [3] (BNV). Results are shown in Table 3. Iterative Dynamic Programming using a Potts model and a linear smoothing model are shown using our proposed hybrid occlusion scheme (Hybrid-IDP) and with KAN's heuristic (KAN-IDP).

While KAN's heuristic achieves impressive results when used with BNV and MF, our implicit approach using the BNV stereo matcher performs better in two of the six sequences and are close to KAN or Hybrid in the other four. GEO-BNV and GEO-MF achieve the lowest error rates on average. Oddly enough, in some 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). This is particuliary true when using the IDP stereo matcher. The camera configuration of the Middlebury dataset is not favorable to Hybrid-IDP. When solving along vertical lines, all cameras have their visibility fixed beforehand (i.e. the epipolar lines of all cameras pairs are horizontal). Nevertheless, when IDP uses our hybrid model, it achieves the lowest error rate in two of the six sequences and always performs better than IDP-KAN. When using visibility smoothing with Hybrid-IDP using both smoothing models, the error rate is slightly reduced.

The evolution of the error in function of the number of iterations for GEO is shown in Figure 8. While our algorithm takes an average of 8 iterations to converge, the improvement after 4 is already minimal. Note that according to Table 3, the choice of the occlusion model seems to be more significant then that of the smoothing model in order to obtain lower error rates for this dataset.

6.2 Tsukuba Head and Lamp

This dataset is from the Multiview Image Database from the University of Tsukuba. 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 is the center one and the 4 supporting images are at an equal distance from it, arranged in a cross shape. The results are shown in Table 4. The entry (KZ1) of the

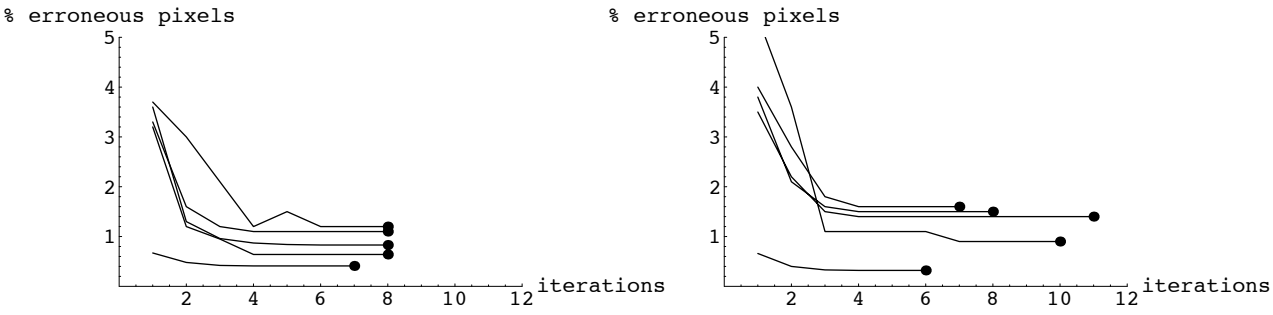

FIG. 8 – Percentage of error vs the number of iterations for the Middlebury dataset for GEO-BNV (left) and GEO-MF (right). Each curve is obtained for a different set of images. Note that the error rate stabilizes after about 4 iterations even if convergence is achieved after 8 iterations on average.

table comes directly from [14]. However, as the authors mentioned, the algorithm has trouble with low textured regions, therefore the error is somewhat underestimated by the removal of an 18 pixel border in the ground truth.

Results from Graph Cut using Nakamura's visibility masks are labeled NAKA-BNV. This is an adaptation of [18] where the maximum flow formulation of [24] was used with the precomputed visibility masks of [19]. We replaced the maximum flow by the Graph Cut algorithm of [3] (ranked the best stereo matcher in two comparative studies [30, 26]), having observed that it achieved a lower error rate for this scene. We also ran a version of the previous algorithm using IDP instead of Graph Cut as the optimization method (labeled Naka-IDP). For the two versions, we tried different sets of masks $\mathcal{M}_h$ and picked the one performing best, namely the one that uses only 2 supporting cameras in each mask. This set has a total of 6 visibility masks. Both NAKA-IDP and Hybrid-IDP were run using a linear and a Potts smoothing models. Even after one iteration of Hybrid-IDP using visibility smoothing, the error rate is low ; this required less than 4 seconds of running time on a 2.0 GHz Athlon 64. The only difference between Naka-IDP and Hybrid-IDP is the occlusion model. Table 4 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. Both algorithms based on IDP have trouble with the second baseline. The algorithm achieved good results near discontinuities, the errors are concentrated in flat regions. Hybrid-IDP performed very well on the first baseline independently of the smoothing model used. Finally, we computed the disparity maps using Graph Cut with the *exact* visibility mask computed in advance from the ground truth. We labeled this method BNV-Truth.

For KZ1 and the algorithms imposing the ordering constraint, we computed the error after removing the pixels breaking it, 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 base-

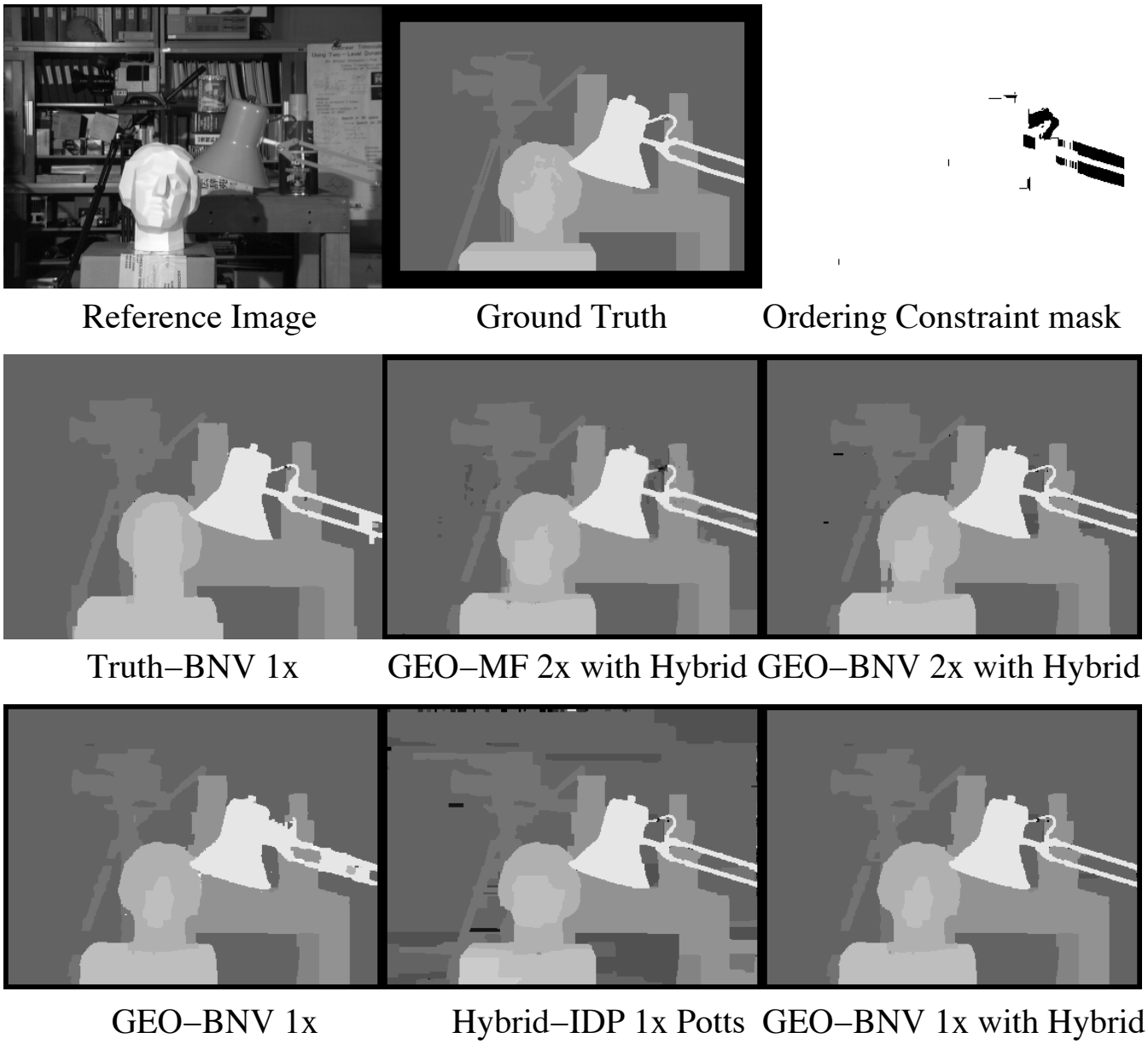

FIG. 9 – Depth maps for the Head and Lamp scene (Multiview Images database of the University of Tsukuba). DP-Hybrid was run with a value of $\gamma = 19$. Note for GEO-BNV how the errors are concentrated in regions breaking the ordering constraint. When combining GEO with Hybrid result are good for both baseline. Truth-BNV containes artifacts near image border, this comes from the fact that errors are not computed in a 18 pixels strip near image border. A mask of pixels breaking the ordering constraint for the smallest baseline is also shown.

Algorithm	Smoothing model	Baseline	Error (whole image)	Error (mask)
BNV-Truth	Potts	1x	1.01%	-
GEO-BNV with Hybrid-IDP	Potts	1x	1.17%	-
GEO-BNV with Naka-BNV	Potts	1x	1.53 %	-
GEO-MF with Hybrid-IDP	Linear	2x	1.55 %	-
Hybrid-IDP (4 iterations,$\gamma = 19$)	Potts	1x	1.67%	-
GEO-BNV with Hybrid-IDP	Potts	2x	1.68 %	-
Hybrid-IDP (4 iterations,$\gamma = 11$)	Linear	1x	1.72%	
NAKA-BNV	Potts	1x	1.77%	-
Hybrid-IDP (1 iterations,$\gamma = 19$)	Potts	1x	1.82%	-
Hybrid-IDP (12 iterations,$\gamma = 0$)	Potts	1x	2.01%	-
NAKA-BNV	Potts	2x	2.01%	-
Hybrid-IDP (12 iterations,$\gamma = 0$)	Linear	1x	2.04%	-
Hybrid-IDP (1 iterations,$\gamma = 23$)	Linear	1x	2.16%	
GEO-BNV with GEO-BNV	Potts	1x	2.23%	1.53%
GEO-BNV with Naka-BNV	Potts	2x	2.23 %	-
KZ1	Potts	1x	2.30%	2.01%
NAKA-IDP (12 iterations)	Potts	1x	2.35%	-
GEO-BNV	Potts	1x	2.46%	1.64%
GEO-MF	Linear	2x	2.62%	1.28%
GEO-BNV	Potts	2x	2.69%	2.11%
Hybrid-IDP (1 iteration,$\gamma = 0$)	Linear	1x	2.56%	-
Hybrid-IDP (1 iteration,$\gamma = 0$)	Potts	1x	2.77%	-
GEO-MF with Hybrid-IDP	Linear	1x	3.27%	-
GEO-MF	Linear	1x	3.42%	2.52%
NAKA-IDP (12 iterations)	Potts	2x	4.71%	-
Hybrid-IDP (12 iterations,$\gamma = 0$)	Potts	2x	4.71%	-
Hybrid-IDP (12 iterations,$\gamma = 0$)	Linear	2x	4.74%	
NAKA-IDP (12 iterations)	Linear	2x	4.76%	-

TAB. 4 – Percentages of error of the different algorithms for the 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.

lines. GEO-BNV almost performed as well as KZ1 ; when removing pixels breaking the ordering constraint, it achieved a slightly lower error rate.

To recover from regions breaking the ordering constraint, we added to GEO-BNV a second step using the first suggested approach described in section 6. This two-step approach is labeled "GEO-BNV with GEO-BNV". Improvement over plain GEO-BNV is minimal. When using the second suggested approach with Hybrid-IDP with the Potts smoothing model as an auxiliary to GEO-BNV ("GEO-BNV with Hybrid-IDP") or Hybrid-IDP with linear smoothing model as an auxiliary to GEO-MF ("MF-BNV with Hybrid-IDP"), the error rate is extremely low even for the second baseline. We also used NAKA-BNV and NAKA-MF as auxiliaries to GEO-BNV and GEO-MF, labeled respectively "GEO-BNV with NAKA-BNV" and "GEO-MF with NAKA-MF". The error rate is higher than when using our hybrid algorithm.

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. Table 5 shows the stability to changes in the smoothing parameter of GEO-BNV, giving the error percentage for 6 values of this parameter. Figure 10 investigates the stability of the smoothing parameters of our Hybrid algorithm, giving the amount of error over a

Algorithm	Baseline	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

TAB. 5 – Resistance to changes in 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.

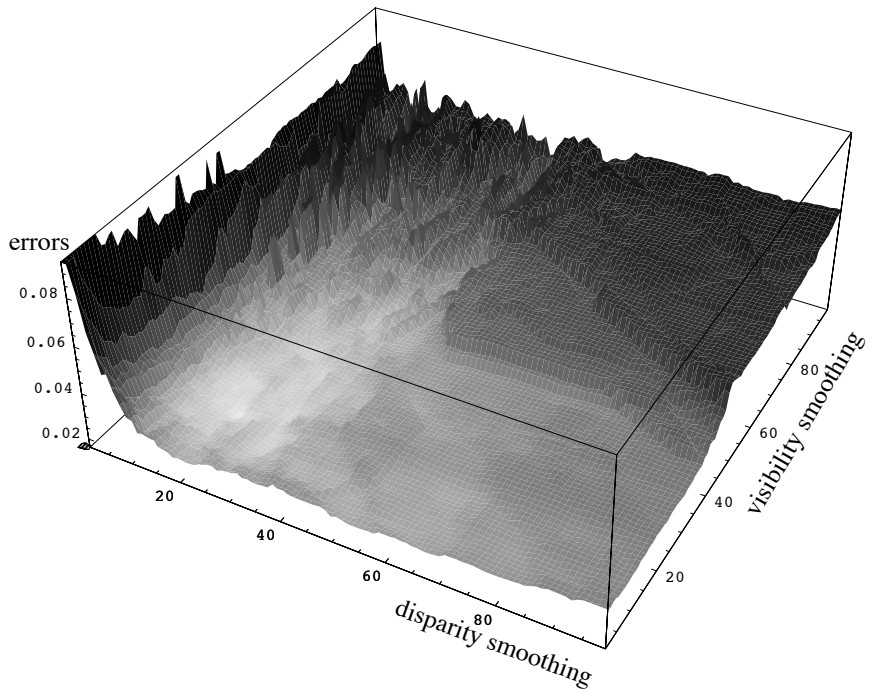

FIG. 10 – Resistance to changes in the smoothing parameters for Hybrid-IDP using the Head and Lamp scene for one iteration. Both smoothing parameters increase by a factor of more than 100.

broad range of values.

6.3 Pixel Classification test

In section 4.2, the shift in the classification of occluders and occludees for some standard stereo matchers is illustrated. We applied our framework together with the BNV stereo matcher to the multi-camera sequences of the Middlebury comparative study. Classification errors were recomputed using the same methodology as Table 1. Results are presented in Table 6. The border localization is more exact and thus the classification bias is now significantly reduced. The average width of occlusion zones in the Head and Lamp, Venus and Sawtooth scenes are respectively 4.5, 3.6 and 5.8 pixels. This has to be taken into account, for discontinuities occur at a sub-pixel level, therefore an error of one pixel in border localization could be the mere result of discretization. Since occlusion zones are relatively narrow, this has a non negligible impact on the statistics.

algorithm	scene								
	tsukuba Head and Lamp			Venus			Sawtooth		
	Real (ground truth) status of pixels classified from depth map as occluders								
	occludee	occluder	regular	occludee	occluder	regular	occludee	occluder	regular
BNV [3]	50.4	**15.4**	34.2	12.6	**61.0**	26.4	42.6	**4.3**	53.3
GEO-BNV + Hybrid-IDP	12.3	**69.2**	18.5	19.5	**63.3**	17.6	18.2	**69.8**	12.6
GEO-BNV + Hybrid-IDP 2x	7.7	**76.1**	16.2	-	**-**	-	-	**-**	-
	Real (ground truth) status of pixels classified from depth map as occludees								
	occludee	occluder	regular	occludee	occluder	regular	occludee	occluder	regular
BNV [3]	**16.4**	5.8	77.8	**65.9**	1.9	32.3	**7.2**	1.1	91.7
GEO-BNV + Hybrid-IDP	**70.6**	7.1	22.3	**61.8**	1.0	37.2	**69.8**	0.2	30.0
GEO-BNV + Hybrid-IDP 2x	**76.9**	5.7	17.3	**-**	-	-	**-**	-	-
	Real (ground truth) status of pixels classified from depth map as regulars								
	occludee	occluder	regular	occludee	occluder	regular	occludee	occluder	regular
BNV [3]	1.0	2.0	**96.9**	0.5	0.7	**98.8**	0.5	1.5	**98.0**
GEO-BNV + Hybrid-IDP	0.6	0.7	**98.7**	0.5	0.7	**98.8**	0.3	0.6	**99.1**
GEO-BNV + Hybrid-IDP 2x	0.8	1.0	**98.2**	-	-	**-**	-	-	**-**

TAB. 6 – Real status in percentages of pixels according to their classification. Examples from the Middlebury comparative study [26] for BNV and from our implicit model. Results for BNV are repeated from table 1. Correct classifications are in bold.

For instance, in the Head and Lamp scene, we can expect 11% (that is half a pixel over 4.5) of all occluders and occludees to be wrongly classified because of discretization. Results for the second baseline are also shown for the Head and Lamp scene ; since the width of occlusion zones doubles, the impact of sub-pixel border localization is halved for real occluders and occludees as can reasonably be expected.

6.4 Baseline test

As the baseline increases, the amount of occlusion in the scene increases as well. A stereo matcher not affected by occlusion should give identical depth maps for different baselines. To measure the level of resistance to changes in 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). 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 amount of error and incompatibility for stereo matchers that do not model occlusion. For instance, the incompatibility levels between the first and the second baselines in the Head and Lamp scene for GEO-BNV with Hybrid-IDP was 0.9%. Hybrid-IDP, which has trouble with the second baseline, has an incompatibility level of 4.8%. NAKA-BNV stands at 1.5%. To test the stability of our algorithm with wide baselines, we used the City and Santa scenes from the Multiview Image Database of the University of Tsukuba. Each dataset contains 81 images in a 9×9 grid. We always only used 5 images equally spaced and arranged in a cross. Images were reduced by a factor of 2 to achieve a resolution of 320×240. Again, for each depth map, the smoothing parameter was adjusted to obtain

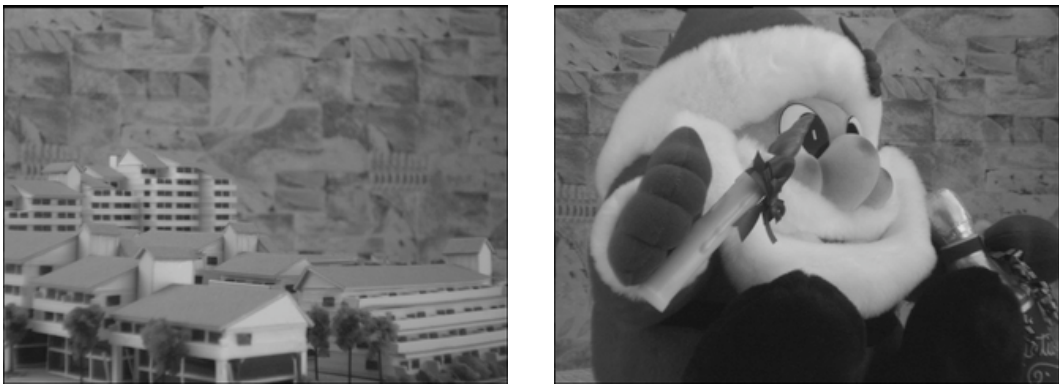

FIG. 11 – Reference image for the city scene (**left**) and santa scene (**right**).

the best possible performance. Since no ground truth was available, the choice was made by visual inspection.

6.4.1 City scene

This scene features a city with a lot of reflections in the front windows. The reference image is shown in Figure 11. The focal distance of the camera is 10 mm with successive baselines of 8, 16, 24 and 32 mm. The back building is located at 66.0 cm and the background at 184.0 cm. Every depth map was computed with 21 disparity steps. Figure 14 contains bar charts of the percentages of pixels incompatible between the depth maps obtained for two baselines. Nakamura's and our implicit method with the maximum flow formulation (NAKA-MF and GEO-MF) performed similarly, while KZ1 and Nakamura with BNV are slightly less stable. The results obtained after the first iteration of our algorithm (FULL-MF) are listed to illustrate what happens when no occlusion modeling is used. 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+.

GEO-BNV and NAKA-BNV do not perform well for large baselines. The occlusion model seems to be more important than the smoothing model for small baseline as illustrated in section 6.1, while the smoothing model seems to become predominant for large ones. Independently of the smoothing model, Hybrid-IDP and NAKA-IDP did not scale well with the increase of the baseline, performing worse than FULL-MF. Consequently, they were not included in the chart.

6.4.2 Santa scene

This scene features a Santa doll (see Figure 11). The focal distance of the camera is 10 mm with successive baselines of 20, 40, 60 and 80 mm. There is much more occlusion here than in the City scene since the baselines are larger. The hand is located at 59.0 cm of the reference camera and the background at 184.0 cm. Each depth map is computed using 23 disparity steps. Note the details on the right side of the hat and on the candle. Again, Figure 14 contains bar charts of the percentages of pixels incompatible between depth maps obtained for two different baselines. GEO-MF is twice as stable as NAKA-MF and yields less noisy depth maps. KZ1 and NAKA-BNV are less stable

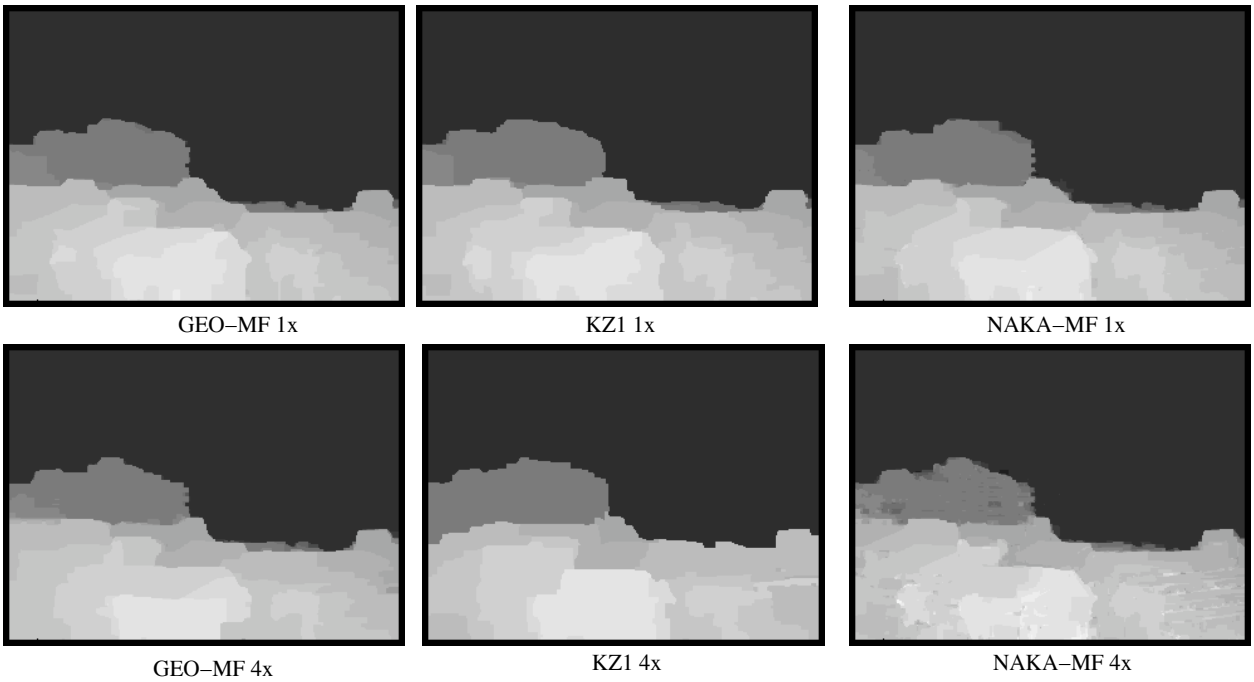

FIG. 12 – Depth maps obtained by 3 algorithms for 2 different baselines (1x and 4x) for the City scene (Multiview Image Database of the University of Tsukuba).

by a factor of 5 and more. The results for FULL-MF are again given. We can see in Figure 13 that GEO-MF achieves the best results for the third and fourth baselines. For the first two, KZ1, NAKA-MF and GEO-MF performe similarly.

7 Conclusion

We have presented a new framework to model occlusion in stereo by introducing the concept of *geo-consistency*. We also provided two complementary algorithms that model occlusion through the use of this concept. The first one adds occlusion modeling to standard stereo algorithms. Being implicit, our approach relies on *geo-consistency* of depth maps to determine the visibility of cameras and aggressively remove them to adjust the likelihood term to the scene structure and to the bias in the type of error made by the stereo matcher. One of the main characteristics of our implicit approach is that it does 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 implicit framework does not add any parameter or constraint to the matching process.

The second algorithm is a fast hybrid between the methods that use photo-consistency to approximate correct visibility and slower *geo-consistent* ones. While it is fast, it succeeds in obtaining sharp and well-located depth discontinuities. Our hybrid algorithm can be used with any smoothing model and performs well for small baselines. It can also be used as an auxiliary stereo

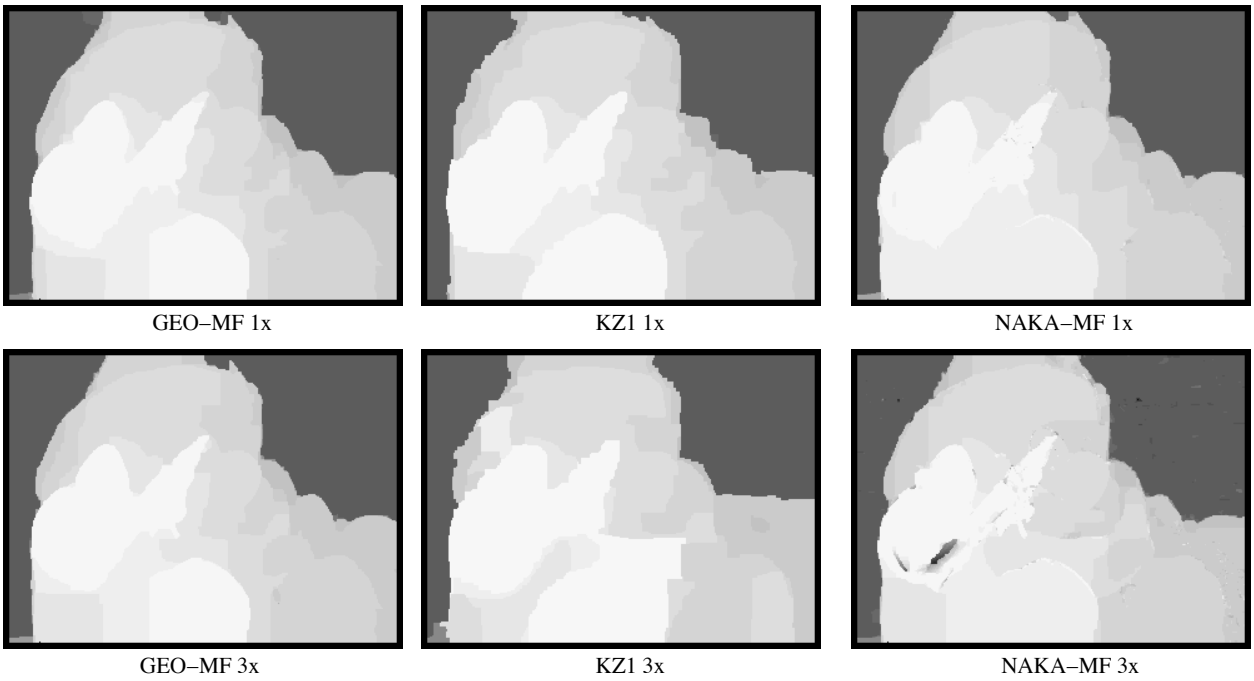

FIG. 13 – Depth maps obtained by 3 algorithms for 2 different baselines (1x and 3x) on the Santa scene (Multiview Image Database of the University of Tsukuba).

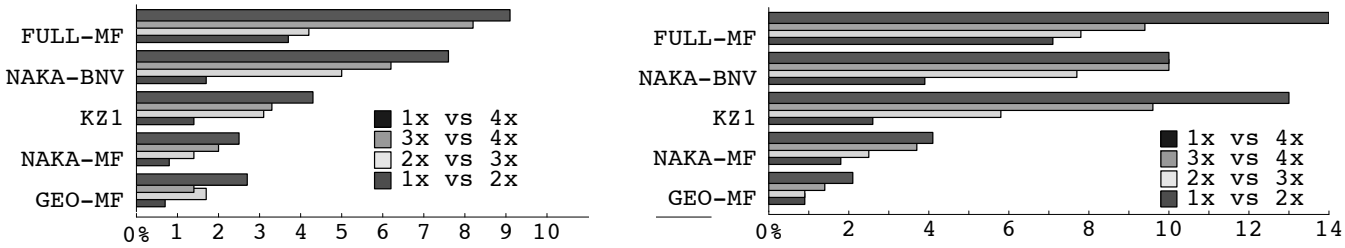

FIG. 14 – Resistance to baseline increase for 5 algorithms for the City (left) and Santa (right) scenes (Multiview Image Database of the University of Tsukuba) ; each column lists the percentages of incompatible pixels between the depth maps obtained for two different baselines.

matcher with our implicit framework when the recover depth map is suspected of containing regions not meeting the ordering constraint. The validity of both approaches 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 approaches were also tested on increasingly wider baselines ; the implicit one displayed a much higher stability to increasing amount of occlusion in the scene. While the validity of our implicit framework has been demonstrated using two stereo matching algorithms, it is general enough to be applied to others. Both approaches are not limited to regular grids of cameras and also work with other camera configurations.

As for future work, this occlusion model could be extended to full volumetric reconstruction, where occlusion becomes the dominant problem. In addition, we would like to build a real-time implementation of our hybrid algorithm, using dedicated hardware such as FPGA's.

8 Acknowledgments

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

Références

[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] 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.

[5] 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.

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

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

[8] G. Gimel'farb and U. Lipowezky. Accuracy of the regularised dynamic programming stereo. In *ICPR*, 2001.

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

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

[17] 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.

[18] G. Le Besnerais 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.

[19] 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.

[20] 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.

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

[22] 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.

[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] S. Roy and M-A Drouin. Non-uniform hierarchical pyramid stereo for large images. In *VMV*, pages 403–410, 2002.

[26] 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.

[27] 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.

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

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

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

[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] 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.