Symmetric Multi-View Stereo Reconstruction From Planar Camera Arrays

Matthieu Maitre
Microsoft
Redmond, WA
mmaitre@microsoft.com

Yoshihisa Shinagawa
Siemens Medical Solutions
Malvern, PA
sinagawa@uiuc.edu

Minh N. Do
University of Illinois
Urbana, IL
minhdo@uiuc.edu

Abstract

We present a novel stereo algorithm which performs surface reconstruction from planar camera arrays. It incorporates the merits of both generic camera arrays and rectified binocular setups, recovering large surfaces like the former and performing efficient computations like the latter. First, we introduce a rectification algorithm which gives freedom in the design of camera arrays and simplifies photometric and geometric computations. We then define a novel set of data-fusion functions over 4-neighborhoods of cameras, which treat all cameras symmetrically and enable standard binocular stereo algorithms to handle arrays with arbitrary number of cameras. In particular, we introduce a photometric fusion function which handles partial visibility and extracts depth information along both horizontal and vertical baselines. Finally, we show that layered depth images and sprites with depth can be efficiently extracted from the rectified 3D space. Experimental results on real images confirm the effectiveness of the proposed method, which reconstructs dense surfaces larger by 20% on Tsukuba.

1. Introduction

Online metaverses have emerged as a way to bring an immersive and interactive 3D experience to a worldwide audience. However, the fully automatic creation of realistic content for these metaverses is still an open problem. The challenge here is to achieve simultaneously four goals. First, the rendering quality must be high for the virtual world to look realistic. Second, the geometric quality must be sufficient to let physics-based simulation provide credible interactions between objects. Third, the computational complexity must be simple enough to enable real-time rendering. Finally, the data must admit a compact representation to allow data streaming across networks.

In this paper, we propose three contributions toward these goals. First, we introduce a special *rectified 3D space* and an associated rectification algorithm which handles planar arrays of cameras. It gives freedom in the design of camera arrays, so that their fields of view can be adapted to the scene being recorded. At the same time, rectification simplifies the reconstruction problem by making the coordinates of voxels and their pixel projection integers. This removes the need for further data resampling and simplifies changes of coordinate systems and visibility computations.

Second, we present a set of data-fusion functions which enable standard binocular stereo reconstruction [13] to handle arrays with arbitrary number of cameras. Using one depth map per camera, the algorithm reconstructs large surfaces, up to 20% larger on Tsukuba, and therefore reduces the holes in novel-view synthesis. We introduce two Markov Random Fields (MRF), a classical one over the arrays of pixels and a novel one over the array of cameras. The latter lets us treat all the cameras symmetrically by defining fusion functions over 4-neighborhoods of cameras.

Finally, we introduce a global fusion algorithm which merges the depth maps into a unique Layered Depth Image (LDI) [15], a rich but compact data representation made of a dense depth map with multiple values per pixel. We also show that the recovered LDI can be segmented fully automatically into sprites with depth [15]. Such sprites are related to geometry images, which can be efficiently rendered and compressed [7].

2. Relation to previous work

Surface reconstruction methods fall into two categories, those based on large generic camera arrays and those based on small rectified stereo setups, most often binocular, where the optical camera axes are normal to the baseline. The former [12, 14, 17, 21] handle a rich depth information and can reconstruct large surfaces. However, the genericity of the camera locations makes visibility computations difficult and voxel projections computationally expensive.

In rectified stereo setups [2, 13, 19], on the other hand, visibility and projections are simple. These setups also allow efficient reconstruction algorithms based on Maximum A-Posteriori (MAP) inference over MRFs. However, the depth information extracted from the images tend to be quite poor, especially for linear arrays which only take ad-

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

vantage of textures with significant gradients along their baseline. Moreover, the small number of cameras and the constrained viewing direction strongly limits the volume inside which depth triangulation is possible.

The constraint on the viewing direction can be removed using rectification, which trades view freedom for image distortion. So far, however, rectification has been limited to small stereo setups with two [4, 8] or three [1, 18] cameras.

In this paper, we introduce a special rectified 3D space and show that when the problem is defined in terms of transformations between 3D spaces, instead of alignment of epipolar lines, rectification can be generalized to planar arrays with arbitrary number of cameras.

Camera arrays have access to a much richer information than binocular setups. Quite surprisingly, however, the extra information can prove to be detrimental and actually reduce the quality of reconstructed surfaces [22]. The issue comes from partially visible voxels, whose number increases with the number of cameras. A number of methods tackle this issue [3, 10, 22]. However, most of them are asymmetric, choosing one camera as a reference. Cameras far apart tend to have less visible surfaces in common, which limits the number of cameras in the array and, as a consequence, the area of reconstructed surfaces. Moreover, many multi-view stereo methods disregard the relative locations of the cameras when extracting the depth information from images [6, 14], which reduces the discriminative power of the extracted information.

In the proposed method, we rely on multiple depth maps, one per camera, and treat all the cameras symmetrically. Furthermore, we define a novel MRF over the camera array and take into account the relative locations of the cameras. This way, the proposed method handles arrays with arbitrary number of cameras and extracts the depth information along both horizontal and vertical baselines.

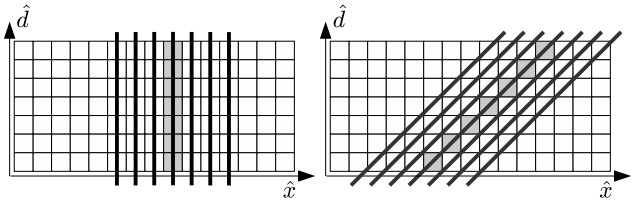
Surface reconstruction based on multiple depth maps has already been studied in [5, 6, 24] but these methods lacked the proposed rectified 3D space, which led to costly operations to compute visibility, enforce inter-camera geometric consistency, and merge depth maps.

The proposed extraction of sprites from LDIs is related to depth map segmentation [9], with the added complexity of multiple depth values per pixels. Moreover, unlike [16], the segmentation is performed automatically and is not limited to planar surfaces.

3. The rectified space

3.1. Overview

We first consider the problem of rectifying the 3D space and the 2D camera images to simplify the stereo reconstruction problem. In the following, points are represented in homogeneous vectors, with $\mathsf{x} \triangleq (x, y, 1)^\intercal$ denoting a point on the 2D image plane and $\mathsf{X} \triangleq (x, y, z, 1)^\intercal$ a point in 3D space. Points are defined up to scale: x and $\lambda\mathsf{x}$ are equivalent for any non-null scalar λ. This relation is denoted by the symbol '$\sim$'.



(a) Camera (0,0) (b) Camera (1,0)

Figure 1. A few rays of light in the rectified 3D space: rays passing through the optical centers of camera $(0, 0)$ (a) and camera $(1, 0)$ (b). The rays are aligned with the voxel grid, which simplifies visibility computations.

Under the pin-hole camera model [4], a 3D point X and its projection x onto an image plane are related by

$$\mathsf{x} \sim \mathbf{P}\mathsf{X} \tag{1}$$

where $\mathbf{P}$ is a 3×4 matrix which can be decomposed as

$$\mathbf{P} = \mathbf{K}\mathbf{R}\begin{pmatrix} \mathbf{I} & -\mathsf{c} \end{pmatrix} \tag{2}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{R}$ the camera rotation matrix, c the optical center and $\mathbf{K}$ the matrix of intrinsic parameters. All these parameters are assumed known.

The optical centers of the cameras are assumed to lie on a planar lattice, that is,

$$\mathsf{c} = \mathsf{o} + m\mathsf{v}_1 + n\mathsf{v}_2 \tag{3}$$

where o is the center of the grid, v_1 and v_2 are two non-collinear vectors, and m and n are two signed integers. The classical stereo pair is a special case of such an array.Since a pair (m, n) uniquely identifies a camera, we use it to index the cameras and denote by $\mathcal{C}$ the set of pairs (m, n).

The proposed rectification consists in rotating the cameras and transforming the Euclidean 3D space using homographies. The rectified 3D space is defined as a space where the projection matrices $\hat{\mathbf{P}}^{(m,n)}$ take the special form

$$\hat{\mathbf{P}}^{(m,n)} = \begin{pmatrix} 1 & 0 & -m & 0 \\ 0 & 1 & -n & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4}$$

It follows that, in the rectified space, a 3D point $\hat{\mathsf{X}} = (\hat{x}, \hat{y}, \hat{d}, 1)^\intercal$ is related to its 2D projection $\hat{\mathsf{x}}^{(m,n)} = \left(\hat{x}^{(m,n)}, \hat{y}^{(m,n)}, 1\right)^\intercal$ on the image plane of camera (m, n) by the equations

$$\begin{cases} \hat{x}^{(m,n)} = \hat{x} - m\hat{d}, \\ \hat{y}^{(m,n)} = \hat{y} - n\hat{d}. \end{cases} \tag{5}$$

The 2D motion vectors of image points from camera (m, n) to camera (m', n') are equal to $\hat{d}$ times the baseline

$(m-m', n-n')^\intercal$. Therefore, the third coordinate $\hat{d}$ of the rectified 3D space is a disparity, while the third coordinate z of the Euclidean space is a depth.

The projection of an integer-valued point $\hat{\mathsf{X}}$ is also an integer. Moreover, the rays of light passing through the optical centers are parallel to one another and fall on integer-valued 3D points, as shown in Figure 1, which simplifies visibility computations.

3.2. Rectification homographies

First, we need to recover the grid parameters o, v_1, and v_2 from the projection matrices $\mathbf{P}^{(m,n)}$. From (3), we obtain the system of equations

$$\begin{pmatrix} \mathbf{I} & m\mathbf{I} & n\mathbf{I} \end{pmatrix} \begin{pmatrix} \mathsf{o} \\ \mathsf{v}_1 \\ \mathsf{v}_2 \end{pmatrix} = \mathsf{c}^{(m,n)}, \quad \forall (m,n) \in \mathcal{C}. \quad (6)$$

In the general case, this system is over-constrained and the vectors are obtained by least mean-square. When the cameras are collinear, one of the vector is free to take any value. In that case, the constrained vector is computed by least mean-square and the free vector is chosen to limit the image distortion. To do so, the normal vector defined by the cross-product $\mathsf{v}_1 \wedge \mathsf{v}_2$ is set to the mean of the unit vectors on the optical axes. The free vector is then deduced by Gram-Schmidt orthogonalization.

We define an intrinsic-parameter matrix $\hat{\mathbf{K}}$ shared by all the rectified cameras as

$$\hat{\mathbf{K}} \triangleq \begin{pmatrix} \hat{f} & 0 & 0 \\ 0 & \hat{f} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (7)$$

where $\hat{f}$ is the rectified focal length. We also define a matrix $\mathbf{V}$ as $\mathbf{V} \triangleq (\mathsf{v}_1, \mathsf{v}_2, \mathsf{v}_1 \wedge \mathsf{v}_2)$ and two 4D homography matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ as

$$\mathbf{H}_1 \triangleq \begin{pmatrix} \hat{\mathbf{K}}\mathbf{V}^{-1} & -\hat{\mathbf{K}}\mathbf{V}^{-1}\mathsf{o} \\ 0 & \hat{f} \end{pmatrix}, \quad (8)$$

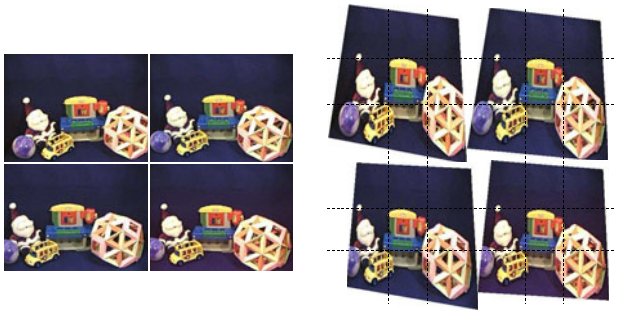
$$\mathbf{H}_2 \triangleq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (9)$$

The rectified focal length $\hat{f}$ is chosen as the mean focal length $\bar{f}$ of the actual cameras.

Multiplying (1) by $\hat{\mathbf{K}}\mathbf{V}^{-1}\mathbf{R}^{(m,n)-1}\mathbf{K}^{(m,n)-1}$, introducing $\mathbf{I} = \mathbf{H}_1^{-1}\mathbf{H}_2^{-1}\mathbf{H}_2\mathbf{H}_1$ between $\mathbf{P}$ and X, and using the relation $\hat{\mathbf{K}}\hat{\mathsf{c}}^{(m,n)} = \hat{f}\hat{\mathsf{c}}^{(m,n)}$, we obtain

$$\hat{\mathbf{K}}\mathbf{V}^{-1}\mathbf{R}^{(m,n)-1}\mathbf{K}^{(m,n)-1}\mathsf{x}^{(m,n)} \sim \hat{\mathbf{P}}^{(m,n)}\mathbf{H}_2\mathbf{H}_1\mathsf{X}. \quad (10)$$

By identification, we obtain the relations between Euclidean and rectified quantities



(a) Original images (b) Rectified images

Figure 2. Rectification of four images from the toy sequence [23]. After rectification, both the rows and the columns of the images are aligned.

$$\hat{\mathsf{x}}^{(m,n)} \sim \hat{\mathbf{K}}\mathbf{V}^{-1}\mathbf{R}^{(m,n)-1}\mathbf{K}^{(m,n)-1}\mathsf{x}^{(m,n)}, \quad (11)$$

$$\hat{\mathsf{X}} \sim \mathbf{H}_2\mathbf{H}_1\mathsf{X}, \quad (12)$$

which are two homographies.

The reconstruction of surfaces in the Euclidean space via depth estimation in the rectified space is then a three-step process. First, images are rectified by applying the homography (11). Then 3D points are estimated in the rectified space by matching the rectified images. Finally, these 3D points are transfered back to the Euclidean space by inverting the homography (12). Figure 2 shows an example of rectified images.

4. Stereo reconstruction

4.1. Overview

We now turn to the stereo reconstruction. In this section, we assume that the images have been rectified and we drop the hat over mathematical symbols in the rectified space.

In order to reduce the computational complexity, the dependencies between cameras in the array are modeled using a MRF where each camera (m,n) is associated with an image $\mathcal{I}^{(m,n)}$ and a disparity map $\mathcal{D}^{(m,n)}$, as shown in Figure 3. Specifically, each value $\mathcal{D}^{(m,n)}_{x,y}$ represents the disparity of a 3D point along the ray of light passing by pixel (x,y) in camera (m,n). At each camera, the dependencies between pixels are also modeled using a MRF. Stereo reconstruction then aims at inferring the hidden disparity maps from the observed images, relations between occupancy and visibility, unicity of the reconstructed scene, and the Markov priors.

An approximate solution is obtained by an iterative process, at the heart of which lie classical MAP-MRF inferences [2, 19, 24] applied *independently* on each camera. Each inference aims at solving an optimization of the form

$$\min_{\mathcal{D}} \sum_{(x,y)\in\mathcal{P}} \left(P_{x,y,\mathcal{D}_{x,y}} + \lambda_g G_{x,y,\mathcal{D}_{x,y}} + S_{x,y}(\mathcal{D}) \right) \quad (13)$$

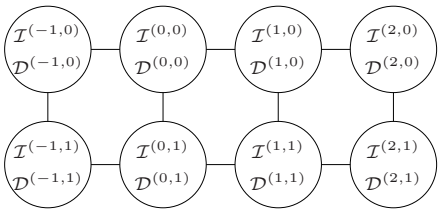


Figure 3. Camera MRF associated with a 2×4 camera array. Each node represents a camera with an observed image $\mathcal{I}$ and a hidden disparity map $\mathcal{D}$. Edges represent fusion functions.

where $\mathcal{P}$ denotes the set of 2D pixels, λ_g is a scalar weight, S is a clique potential favoring piecewise-smoothness [19], and $P_{x,y,d}$ and $G_{x,y,d}$ are respectively photometric and geometric cost volumes.

The proposed algorithm alternates between inferences and cost volume computations. Its novelty lies in the set of fusion functions computing the costs volumes. Due to the Markov assumption, the fusion functions are defined over 4-neighborhoods $\mathcal{N}_4$, i.e. cross-shaped groups of five cameras, which usually contain a rich depth information but only limited partial occlusions. The overall complexity of the proposed algorithm is linear in the size of the data.

Although limited, partial occlusions tend to create large photometric costs at voxels on the surfaces, which leads to erroneous disparities. These outlier costs can be removed by an explicit visibility modeling [3]. However, visibility depends on the surface geometry, which introduces a circular dependency. We solve this issue by introducing an implicit model of partial occlusions, which does not depend on the surface geometry.

Robust statistics over the four pairwise cliques of each camera 4-neighborhood can reduce the impact of outlier costs. However, classical robust statistics do not take into account the relative locations of the cameras and may fail to extract the depth information along both horizontal and vertical baselines, leading to photometric cost volumes with poor discriminative power.

Therefore, we propose a robust measure which strives to include the photometric costs from at least one vertical and one horizontal camera clique at each voxel. We do this by introducing an assumption we call "visibility by opposite neighbors": a voxel visible by a camera (m, n) is also visible by at least one of its horizontal camera neighbors $(m-1, n)$ and $(m+1, n)$, and by at least one of its vertical camera neighbors $(m, n-1)$ and $(m, n+1)$. This assumption usually holds, except for instance for surfaces like picket fences or cameras having less than four neighbors. In the following, we denote the quantities related to horizontal and vertical pairwise cliques by the superscripts h and v respectively.

4.2. Geometric cost volume $G^{(m,n)}$

The geometric cost volumes $G^{(m,n)}$ favor consistent disparity maps. In order to compute them, the disparity maps $\mathcal{D}^{(m,n)}_{x,y}$ are first transformed into binary occupancy volumes $\delta^{(m,n)}_{x,y,d}$, whose voxels take value one when they contain surfaces. An occupancy volume $\delta^{(m,n)}_{x,y,d}$ is obtained by initializing it to zero, except at the set of voxels $\{(x, y, \mathcal{D}^{(m,n)}_{x,y})\}$ where it is initialized to one.

Since all the occupancy volumes represent the same surfaces, they should be identical up-to visibility and a change of coordinate system. Thanks to the rectification leading to (5), changing the coordinate system of a volume δ from camera $(0, 0)$ to camera (m, n) is simply an integer 3D shear $\phi^{(m,n)}$ given by

$$\phi^{(m,n)}_{x,y,d}(\delta) = \delta_{x+md,y+nd,d}. \tag{14}$$

A change of coordinate system between two arbitrary cameras is obtained by concatenating two 3D shears.

Let us consider camera (m, n) and shear the occupancy volumes of its 4-neighbors to its coordinate system. Using the assumption of visibility by opposite neighbors, erroneous occupancy voxels are removed using

$$\begin{aligned}\delta^{(m,n)}_{x,y,d} \leftarrow \delta^{(m,n)}_{x,y,d} \wedge \left(\delta^{(m+1,n)}_{x,y,d} \vee \delta^{(m-1,n)}_{x,y,d}\right) \\ \wedge \left(\delta^{(m,n+1)}_{x,y,d} \vee \delta^{(m,n-1)}_{x,y,d}\right)\end{aligned} \tag{15}$$

where $\vee$ and $\wedge$ denotes respectively the "or" and "and" operators.

The geometric cost volume is then computed as

$$G^{(m,n)}_{x,y,d} \leftarrow \begin{cases} 0, \text{ if } \delta^{(m,n)}_{x,y,d'} = 0, \forall d' \\ \min\limits_{\delta^{(m,n)}_{x,y,d'} \neq 0} \min\left(\left|d - d'\right|, \tau_1\right), \text{ otherwise,} \end{cases} \tag{16}$$

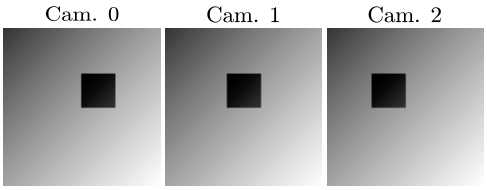
where τ_1 is a threshold.

4.3. Photometric cost volume $P^{(m,n)}$

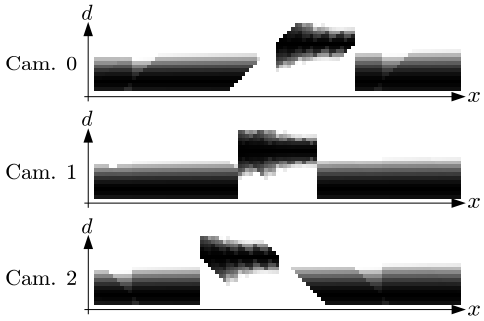
The photometric cost volumes favor voxels with similar intensities across images. They are based on a truncated quadratic error measure [13], in which we introduce an outlier removal process to discard errors from partially visible voxels. The outlier removal is based on a hybrid model with an implicit part, which does not need any occupancy information, and an explicit part, which takes advantage of the occupancy information when it becomes available. Figure 4 illustrates this occlusion model on a synthetic example and Figure 5 shows its impact on the disparity map estimation.

The explicit model relies on the dependency between occupancy and visibility. Due to the nature of the rectified 3D space, a binary visibility volume $\nu^{(m,n)}$ can be computed from its associated occupancy volume $\delta^{(m,n)}$ using a simple recursion along the disparity axis

$$\nu^{(m,n)}_{x,y,d} \leftarrow \nu^{(m,n)}_{x,y,d+1} \wedge \neg\delta^{(m,n)}_{x,y,d+1}, \tag{17}$$

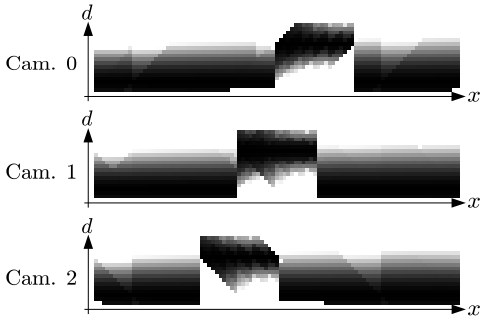


(a) Three images of two fronto-parallel planes: a dark square in front of a bright background.



(b) Photometric cost at iteration 1: the implicit model removes partial occlusions in camera 1 and limits their impact in cameras 0 and 2.

(c) Disparity maps at iteration 1: errors remain on cameras 0 and 2.



(d) Photometric cost at iteration 2: the explicit model removes partial occlusions in all the cameras.

(e) Disparity maps at iteration 2: no error remains.

Figure 4. A simple example demonstrating the behavior of the occlusion model. Perfect disparity maps are obtained in two iterations.

where $\neg$ denotes the "not" operator. The recursion is initialized by setting ν to one.

In the following, we only detail the computation of quantities related to horizontal cliques. The vertical ones are obtained by a similar reasoning. The computations are conducted independently at each voxel, so we drop the subscript (x, y, d). We define $I^{(m,n)}$ as the intensity volume obtained by replicating the image $\mathcal{I}^{(m,n)}$ along the disparity axis.

Let us consider the camera (m, n) and its 4-neighborhood. Using (14), the intensity and visibility volumes are sheared to the coordinate system of camera (m, n). From the truncated quadratic error model and the assumption of visibility by opposite neighbors, an horizontal error volume $E^{h(m,n)}$ is computed as

$$E^{h(m,n)} = \min\Big(\Big(I^{(m,n)} - I^{(m-1,n)}\Big)^2, \Big(I^{(m,n)} - I^{(m+1,n)}\Big)^2, \tau_2 \Big) \tag{18}$$

where τ_2 is a threshold.

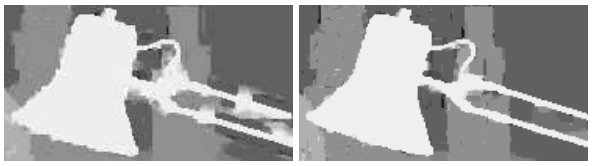
The photometric cost $E^{h(m,n)}$ may still contain large values when the assumption of visibility by opposite neighbors is violated. Therefore, we further discard outliers by explicitly computing visibility. Using De Morgan's laws, the validity of the costs is computed as

$$V^{h(m,n)} = \neg\nu^{(m,n)} \vee \nu^{(m-1,n)} \vee \nu^{(m+1,n)}. \tag{19}$$

We now have two pairs of error and validity volumes, $(E^{h(m,n)}, V^{h(m,n)})$ horizontally and $(E^{v(m,n)}, V^{v(m,n)})$ vertically. In order to create a photometric cost volume which includes the depth information from both vertical and horizontal texture gradients, we define this cost volume as the weighted average

$$P^{(m,n)} = \frac{V^{h(m,n)} E^{h(m,n)} + V^{v(m,n)} E^{v(m,n)}}{V^{h(m,n)} + V^{v(m,n)}} \tag{20}$$

which is only defined when at least one of the validity volumes takes a non-zero value. Values at voxels where this is not the case are obtained by interpolation.



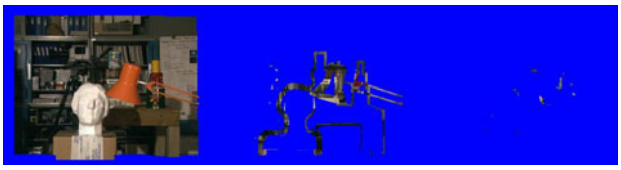
(a) Truncated quadratic cost (b) Proposed cost

Figure 5. Cropped disparity maps computed on Tsukuba with five cameras forming a cross. The proposed photometric cost reduces the disparity errors due to partial occlusions.

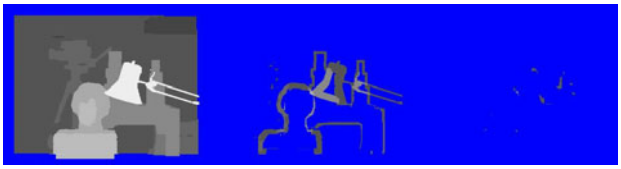
5. Global surface representation

5.1. Layered depth image

Using the special nature of the 3D rectified space, we present a simple and efficient procedure to merge the multiple disparity maps into a unique LDI [15]. The LDI offers a compact and global surface representation. Figure 6 shows an example of LDI.



(a) Texture



(b) Disparity

Figure 6. The 3-layer LDI obtained on Tsukuba with 25 cameras. By treating all the cameras symmetrically, the proposed algorithm recovers large areas, which may be occluded in the central camera.

To begin with, the disparity maps $\mathcal{D}^{(m,n)}$ are transformed into occupancy volumes $\delta^{(m,n)}$ as detailed in Section 4.2. These volumes are then sheared to a reference coordinate system, the one of camera $(0,0)$ for instance.

The disparity layers are extracted in a front to back order by voting. Visibility volumes $\nu^{(m,n)}$ are computed from their associated occupancy volumes using (17) and an aggregation volume A is obtained using

$$A_{x,y,d} = \sum_{(m,n)\in\mathcal{C}} \nu^{(m,n)}_{x,y,d}\,\delta^{(m,n)}_{x,y,d}. \tag{21}$$

A disparity layer $\mathcal{D}$ is extracted by selecting the voxels with the largest aggregation values along the disparity axis. These voxels are then removed from the occupancy volumes and the process is repeated until no occupied voxel remains.

5.2. Sprites with depth

Due to the smoothness term S in (13), the layers of the LDI are piecewise smooth. They can be converted to smooth sprites with depth by selecting regions of the LDI which do not contain discontinuities and which introduce as few new boundaries in continuous regions as possible. The extent of these regions may spread over multiple layers of the LDI. Figure 7 shows some examples of sprites.

Before the sprite extraction begins, the disparities are transformed into depth using (12), so that discontinuities be in the Euclidean space used for rendering.

A sprite is defined as a depth map $\mathcal{D}$ and a binary alpha map α, which takes value one inside the sprite. We focus here on the automatic extraction of sprite masks. Refinement techniques leading to high-quality textures have been addressed elsewhere [16] and are beyond the scope of this paper.

The sprites are extracted one at a time. First, an edge detection is performed on the depth map, followed by a distance transform and a watershed segmentation [20]. The sprite alpha map is then initialized to the largest watershed region and the sprite depth map is set to the LDI depth map inside this region.

Figure 7. Examples of sprites extracted from the LDI of Tsukuba with 25 cameras. Note the absence of occlusion on the cans.

The sprite is updated by looping through the layers of the LDI and solving a MAP-MRF inference each time, until convergence. The pixels inside the sprite are then removed from the LDI, the newly visible pixels moved to the first layer, and the process repeated.

The MAP-MRF inference proceeds as follows. Let $\mathcal{D}^{(LDI)}$ and $\alpha^{(LDI)}$ be respectively the depth map and the binary alpha map of the current LDI layer. The sprite and the LDI layer are first fused together to form $\bar{\mathcal{D}}$ and $\bar{\alpha}$ such that

$$\begin{aligned} \bar{\alpha}_{x,y} &= \alpha_{x,y} \vee \alpha^{(LDI)}_{x,y}, \\ \bar{\mathcal{D}}_{x,y} &= \alpha_{x,y}\mathcal{D}_{x,y} + (1-\alpha_{x,y})\mathcal{D}^{(LDI)}_{x,y}. \end{aligned} \tag{22}$$

At each pixel (x,y), we define a likelihood $p_{x,y}$ of belonging to the sprite and we model its dependencies by a MRF. The likelihoods inside the sprite mask are fixed to one and three transition functions are defined

$$p_{x',y'} = \begin{cases} (1-2\rho_0)p_{x,y} + \rho_0 \text{ where smooth,} \\ (1-2\rho_1)p_{x,y} + \rho_1 \text{ at small depth differences,} \\ \min\left(1-p_{x,y}, 1/2\right) \text{ at discontinuities,} \end{cases}$$

where ρ_0 and ρ_1 are two transition likelihoods with $0 \leq \rho_0 < \rho_1 \leq 1/2$. The third transition function states that at a discontinuity

1. if one side belongs to the sprite, the other one does not,
2. if one side does not belong to the sprite, there is no constraint on the other side.

Once the inference has been solved, the sprite alpha map is set to one where p is greater than $1/2$ and the sprite depth map is updated accordingly.

6. Experimental Results

First, the rectification and stereo reconstruction algorithms are validated on four images from the toy sequence [23]. The four cameras form a 2×2 array with non-parallel optical axes and non-square cells. Figure 2 shows the output of the rectification algorithm. Rectification aligns the rows and columns of the images and introduces a limited

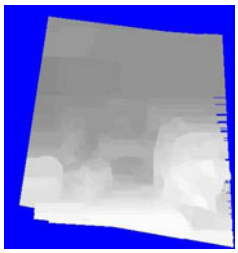


Figure 8. Disparity map obtained from the four rectified images of the toy sequence shown in Figure 2.

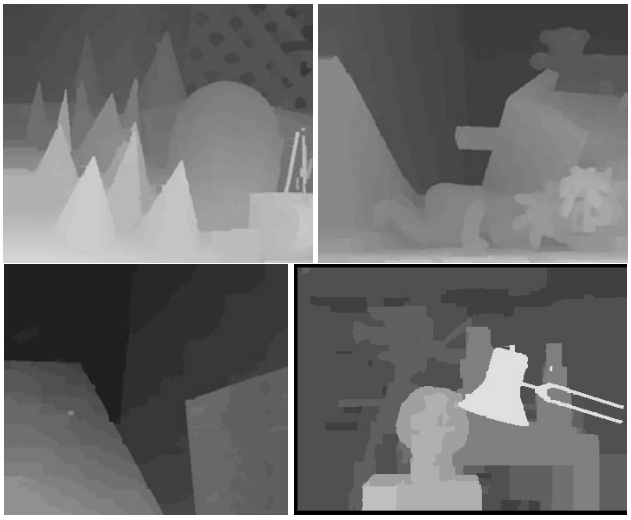


Figure 9. Disparity maps obtained on the Middlebury dataset with two cameras. The occlusion model leads to sharp and accurate depth discontinuities.

amount of distortions. Figure 8 shows the disparity map obtained by the proposed stereo reconstruction algorithm after five iterations. The geometry of the scene appears clearly.

The stereo reconstruction is then tested on the binocular sequences of the Middlebury dataset [13]. In this case, the configuration of the cameras is such that rectification does not introduce any image distortion. Figure 9 shows the disparity maps obtained by the proposed method using fixed parameters. The proposed method performs consistently well over the set of sequences. In particular, it does not suffer from foreground fattening [13]: occlusion modeling, geometric consistency and piecewise smoothness lead to disparity maps with discontinuities which are both sharp and accurately located. The disparity maps contain few errors, mostly located on the left and right image borders, where less depth information is available.

Since the groundtruth is known for this dataset, we also present numerical performance results in Table 1. The error rates of the proposed method are close to those of the best binocular methods.

Unlike binocular methods, however, the proposed method scales with the number of cameras. Table 2 presents

	Tsukuba	Venus	Teddy	Cones
Proposed method	1.53	1.04	10.9	8.65
Best method	1.29	0.21	6.54	7.86
Rank	3	13	6	6

Table 1. Performances on the Middlebury dataset with two cameras (from top to bottom: percentage of erroneous disparities over all areas for the proposed method, percentage for the best method on each image [13], and ranks of the proposed method).

2 cameras	**Proposed**	**1.5**
	New Kolmogorov, Zabih, 2005 [22]	2.2
	Wei, Quan, 2005 [22]	2.7
5 cameras	**Proposed**	**1.3**
	New Kolmogorov, Zabih, 2005 [22]	1.3
	Wei, Quan, 2005 [22]	1.3
	Drouin et al., 2005 [3]	2.2
	Kolmogorov, Zabih, 2002 [11]	2.3
25 cameras	**Proposed**	**1.3**

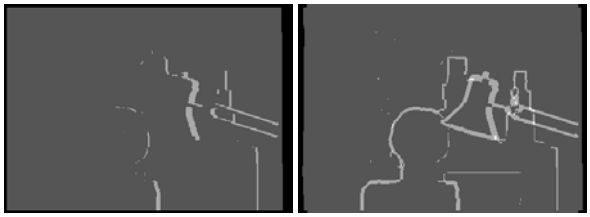
Table 2. Percentage of erroneous disparities over all areas on Tsukuba for several multi-camera methods. The proposed method achieves competitive error rates and scales with the number of cameras.

the error rates of the proposed algorithm and several multi-view algorithms on Tsukuba [13] under three camera configurations: two cameras forming a 1×2 binocular configuration, five cameras forming a 3×3 cross, and twenty five cameras forming a 5×5 square.

The proposed method achieves state-of-the art results in both the two and five camera case. Moreover, it scales to twenty five cameras and handles well the increased amount of partial occlusions. From these results, it seems that it is advantageous to switch from two to five cameras, but that little gain is achieved by further increasing the number of cameras to twenty five.

The real gain from the twenty-five camera array comes from the increased volume in which stereo reconstruction takes place. Figure 6 shows the LDI obtained from such an array. This LDI has three layers, which means that the rays of light originating from the optical center intersect the surfaces up to three times.

Figure 10 and Table 3 show the evolution of the LDI density as a function of the number of cameras. The number of disparity values increases by nearly 20% when switching from a unique disparity map to a 25-camera LDI. This behavior is confirmed by Figure 11, which shows the texture of the objects on the table recovered using two, five, and twenty five cameras. The texture area steadily increases with the number of cameras, which would reduce the size of holes in renderings from novel viewpoints. Since large parts of the textures are not visible in the central camera, they would not have been recovered by stereo algorithms relying on a reference image.



(a) 2 cameras (b) 5 cameras

(c) 25 cameras

Figure 10. Number of disparity values per pixel on Tsukuba (black: no value, white: 3 values). The area of the reconstructed surfaces increases with the number of cameras.

	Number of disparity values	Relative increase
Disparity map	106×10^3	0.0%
LDI, 2 cam.	108×10^3	+2.1%
LDI, 5 cam.	116×10^3	+9.7%
LDI, 25 cam.	127×10^3	+19.4%

Table 3. Number of disparity values in a standard disparity map and in an LDI, for various numbers of cameras on Tsukuba. Using an LDI and 25 cameras increases the area of reconstructed surfaces by almost 20%.

(a) 2 cameras (b) 5 cameras (c) 25 cameras

Figure 11. Cropped textures extracted from the LDIs of Tsukuba. Occlusions shrink when the number of cameras increases.

7. Conclusion

In this paper, we have first presented a novel rectification algorithm which handles planar camera arrays of any size and greatly simplifies the reconstruction of 3D surfaces. Second, we have introduced a stereo reconstruction method which treats all cameras symmetrically and scales with the number of cameras. Finally, we have presented novel algorithms to merge the estimated disparity maps into layered depth images and sprites with depth. We have validated the proposed methods by experimental results on arrays with various camera configurations and reconstructed dense surfaces larger by 20% on Tsukuba. Future work shall consider multiple planar arrays to obtain closed surfaces.

References

[1] N. Ayache and F. Lustman. Trinocular stereovision for robotics. *IEEE Trans. on PAMI*, 13, 1991.

[2] Y. Deng et al. Stereo correspondence with occlusion handling in a symmetric patch-based graph-cuts model. *IEEE Trans. on PAMI*, 29:1068–1079, 2007.

[3] M.-A. Drouin, M. Trudeau, and S. Roy. Geo-consistency for wide multi-camera stereo. In *Proc. CVPR*, 2005.

[4] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Appli.*, 12:16–22, 2000.

[5] P. Gargallo and P. Sturm. Bayesian 3D modeling from images using multiple depth maps. In *Proc. CVPR*, 2005.

[6] M. Goesele, B. Curless, and S. M. Seitz. Multi-view stereo revisited. In *Proc. CVPR*, 2006.

[7] X. Gu, S. J. Gortler, and H. Hoppe. Geometry images. In *Proc. SIGGRAPH*, 2002.

[8] R. I. Hartley. Theory and practice of projective rectification. *Int. J. of Comp. Vis*, 35:115–127, 1999.

[9] M. Jaesik, M. Powell, and K. W. Bowyer. Automated performance evaluation of range image segmentation algorithms. *IEEE Trans. on Sys., Man and Cyber.*, 34:263–271, 2004.

[10] S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *Proc. CVPR*, 2001.

[11] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *Proc. ECCV*, 2002.

[12] P. Merrell et al. Real-time visibility-based fusion of depth maps. In *Proc. ICCV*, 2007.

[13] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. of Comp. Vis*, 47(1–3):7–42, 2002.

[14] S. Seitz et al. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. CVPR*, 2006.

[15] J. W. Shade et al. Layered depth images. In *Proc. SIGGRAPH*, 1998.

[16] H.-Y. Shum et al. Pop-up light field: An interactive image-based modeling and rendering system. *ACM Trans. on Graphics*, 23:143–162, 2004.

[17] C. Strecha, T. Tuytelaars, and L. J. V. Gool. Dense matching of multiple wide-baseline views. In *Proc. ICCV*, 2003.

[18] C. Sun. Uncalibrated three-view image rectification. *Image and Vision comp.*, 21(3):259–269, 2003.

[19] J. Sun et al. Symmetric stereo matching for occlusion handling. In *Proc. CVPR*, 2005.

[20] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. on PAMI*, 13:583–598, 1991.

[21] G. Vogiatzis et al. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Trans. on PAMI*, 29:2241–2246, 2007.

[22] Y. Wei and L. Quan. Asymmetrical occlusion handling using graph cut for multi-view stereo. In *Proc. CVPR*, 2005.

[23] C. Zhang and T. Chen. View-dependent non-uniform sampling for image-based rendering. In *Proc. ICIP*, 2004.

[24] C. L. Zitnick and S. B. Kang. Stereo for image-based rendering using image over-segmentation. *Int. J. of Comp. Vis*, 2007.