SCHOOL OF ENGINEERING - STI
SIGNAL PROCESSING LABORATORY LTS4
Zafer Arıcan and Pascal Frossard

CH-1015 LAUSANNE

Phone: +4121 6934329
Fax: +4121 6937600
e-mail: zafer.arican@epfl.ch

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

DENSE DEPTH ESTIMATION FROM OMNIDIRECTIONAL IMAGES

Zafer Arıcan and Pascal Frossard

Ecole Polytechnique Fédérale de Lausanne (EPFL)

Signal Processing Laboratory LTS4 Technical Report

TR-LTS-2009-006

June 12th, 2009

Part of this work has been submitted to the International Journal of Computer Vision.

This work has been partly supported by a grant of the Indo-Swiss Joint Research Program.

Dense Depth Estimation from Omnidirectional Images

Zafer Arıcan and Pascal Frossard
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Laboratory (LTS4), Lausanne, 1015 - Switzerland.
{zafer.arican, pascal.frossard}@epfl.ch
Fax: +41 21 693 7600, Phone: +41 21 693 4329

Abstract

This paper addresses the problem of dense disparity estimation in networks of omnidirectional cameras. We propose to map omnidirectional images on the 2-sphere, and we perform disparity estimation directly on the sphere in order to preserve the geometry of images. We first perform rectification of the images in the spherical domain. Then we formulate a global energy minimization problem for the estimation of disparity on the sphere. We solve the optimization problem with a graph-cut algorithm, and we show that the proposed solution outperforms common methods based on block matching, for both synthetic scenes with varying complexity and complex natural scenes. Then, we propose a parallel implementation of the graph-cut algorithm that is able to perform dense depth estimation with an improved speed-up, which makes it suitable for realtime applications. Finally, we extend the spherical depth estimation framework to networks of multiple cameras, and we design two methods for dense depth estimation that are based respectively on disparity computation with pairs of images, or computation of inverse depth values. Both methods are shown to provide promising results towards depth estimation in networks of omnidirectional cameras.

Index Terms

disparity estimation, depth estimation, omnidirectional imaging, spherical images, camera networks

I. Introduction

Disparity or depth estimation from mutliple images has recently stimulated important research efforts due to numerous applications such as localization, 3D navigation or 3D scene rendering. When several cameras capture the same scene, the geometry of the scene can be estimated by comparing the images from the different vision sensors. The differences between the respective positions of a 3D point on multiple 2D images represent disparities that can typically be estimated by stereo matching methods. Disparities computed for each pixel form dense disparity maps, which are in general well studied for standard cameras where fast algorithms have been developed. These fast estimation methods are generally based on block matching and pixel correlation due to their simplicity and speed (e.g., [Takiguchi et al(2002)Takiguchi, Yoshida, Takeya, Eino, and Hashizume], [Meguro et al(2007)Meguro, Takiguchi, Amano, and Hashizume], [Gonzalez-Barbosa and Lacroix(2005)]). Alternatively, several works [Scharstein and Szeliski(2002)] have proposed the use of global energy minimization methods for dense disparity estimation on standard images, and such approaches have become quite popular with fast and accurate algorithms based on graph-cut [Boykov et al(1998)Boykov, Veksler, and Zabih] or belief propagation [Sun et al(2003)Sun, Shum, and Zheng]. These methods converge either to global minimum or to strong local minima in polynomial time, and become therefore suitable for realtime applications.

At the same time, omnidirectional imaging has shown recently interesting benefits in the representation and processing of the plenoptic function compared to classical projective camera models. It generally provides a better accuracy as it preserves the geometry information. It also provides a large field of view that simplify the deployment of camera networks. The strategies that have been developed for dense disparity estimation from standard camera images are also applicable to omnidirectional images. However, dense stereo matching on omnidirectional camera images has been mainly limited to planar omnidirectional images. The state-of-the-art algorithms are generally based on re-projection of omnidirectional images on simpler manifolds. For example, Takiguchi *et al* [Takiguchi et al(2002)Takiguchi, Yoshida, Takeya, Eino, and Hashizume] re-project omnidirectional images onto cylinders, while Gonzalez-Barbosa *et al* [Gonzalez-Barbosa and Lacroix(2005)] and Geyer *et al* [Geyer and Daniilidis(2003)] rectify omnidirectional images on the rectangular grid. Fleck *et al* [Fleck et al(2005)Fleck, Busch, Biber, Strasser, and Andreasson], [Fleck et al(2009)Fleck, Busch, Biber, and Straßer] also proposes a 3D modeling algorithm that is applied on planar omnidirectional images, along with post-processing for refinement of the disparity estimation. However, both cylindrical and rectangular projections are not sufficient to represent spatial neighborhood and correlations among the pixels.

In this paper, we propose to address the problems of stereo and dense disparity estimation from omnidirectional images into the spherical framework in order to preserve the geometry information. Most omnidirectional images generated by systems with a single point of projections or fish-eye lenses can be uniquely mapped on the 2D sphere by inverse stereographic projection, as shown in [Geyer and Daniilidis(2001)], [Ying and Hu(2004)]. We therefore exploit the advantages of the spherical framework for the dense estimation of disparity. We first rectify the omnidirectional images by rotation in the spherical domain where we

handle jointly the epipolar regions and central regions at the same time, contrarily to [Bartczak et al(2007)Bartczak, Koeser, Woelk, and Koch]. Then we formulate a global energy minimization method by considering the typical characteristics of images on the 2D sphere. We solve this optimization problem with a graph-cut algorithm on the sphere for an improved accuracy in the disparity estimation. Experimental results show that the proposed method provides better disparity estimation for synthetic and natural scenes compared to common block-based estimation methods. Next, we propose a parallel implementation of our graph-cut disparity estimation algorithm that lead to large speed-up without important accuracy loss. This generic solution permits to implement dense disparity estimation in realtime applications. Finally, we extend our framework to networks of more than two camera sensors where the accuracy of scene modeling can be improved by computation of disparity on multiple image pairs. In addition, we propose a alternative solution based on inverse depth and a different photo-consistency term to improve the performance and accuracy for multiple omnidirectional cameras. Both methods demonstrate promising experimental results towards the solution of depth estimation problems in networks of omnidirectional cameras.

The rest of the paper is structured as follows. Section II describes the problem of disparity estimation in the spherical framework. Section III describes the graph-cut algorithm that solves the disparity estimation problem and presents experimental results on natural and synthetic images. Section IV then presents a method to improve the speed of the depth estimation by parallelization of the graph-cut algorithm. The extension of the framework to networks of multiple cameras is finally presented in Section V.

II. Disparity estimation on the sphere

We formulate the problem of disparity estimation from omnidirectional images as an energy minimization problem on the sphere. We assume that the omnidirectional images captured by the vision sensors are uniquely mapped on the 2D sphere by inverse stereographic projection [Geyer and Daniilidis(2001)], [Ying and Hu(2004)]. We first perform a rectification of the spherical images and then we compute the disparities as differences between the spherical coordinates of the corresponding pixels. The optimal mapping function between pairs of images is obtained by minimizing a functional that inlcudes both a data fidelity and a smoothness term.

A. Rectification of Spherical Images

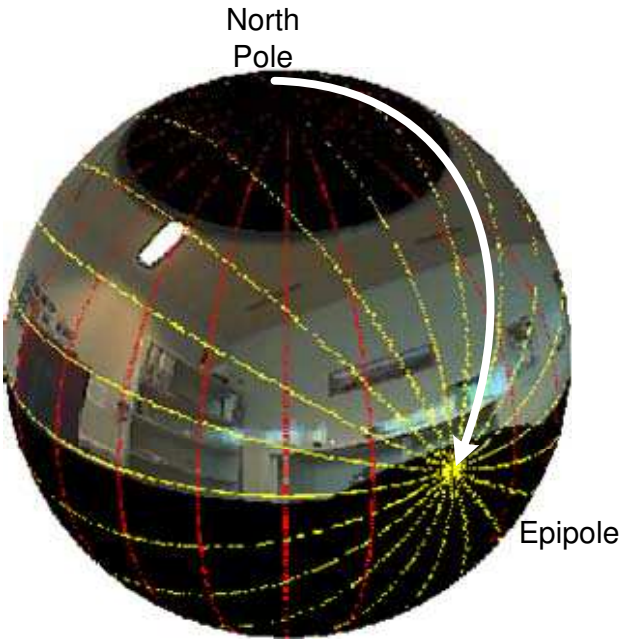

Fig. 1: Relation between longitudes epipolar great circles and rotation on the sphere.

Rectification is an important step in stereo vision where it permits to reduce the stereo correspondence estimation to a one-dimensional search problem. It basically consists in image warping, which is computed such that epipolar lines coincide with the scan lines. It does not only facilitate the implementation of disparity estimation algorithms, but it also makes computations faster.

Similarly to the case of perspective cameras, it is also possible to perform rectification on spherical images captured by omnidirectional cameras. We simply use the following observation about epipoles on spherical images: (i) epipoles resemble

the coordinate poles and (ii) epipolar great circles intersecting on epipoles are like longitude circles. Spherical image pairs can undergo rotation in the spherical domain such that epipoles coincide with the coordinate poles. In this way, epipolar great circles coincide with the longitudes and disparity estimation becomes a mono-dimensional problem.

Figure 1 illustrates the rectification strategy used in this paper. At the end of the rectification step, two rectified stereo spherical images are obtained in the form of rectangular images on a equiangular grid. It permits to extend the disparity estimation algorithms developed for standard images to spherical images, and thus perform fast computations. It is important to note that, even if the images are represented as rectangular images for the sake of clarity, they are actually sampled on an equiangular grid on the sphere, so that the geometrical information is preserved.

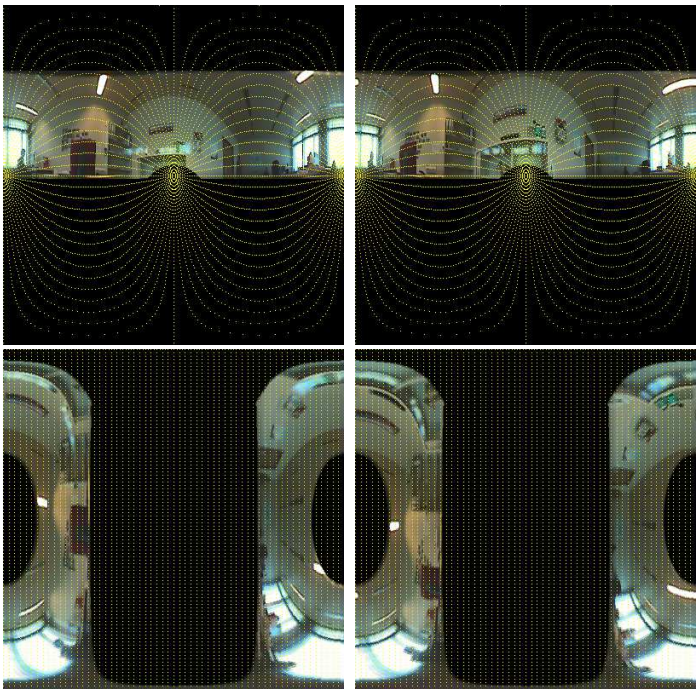

Fig. 2: Original images (top), and rectified ones (bottom). Epipolar great circles turn into straight vertical lines in rectified images.

B. Disparity between spherical images

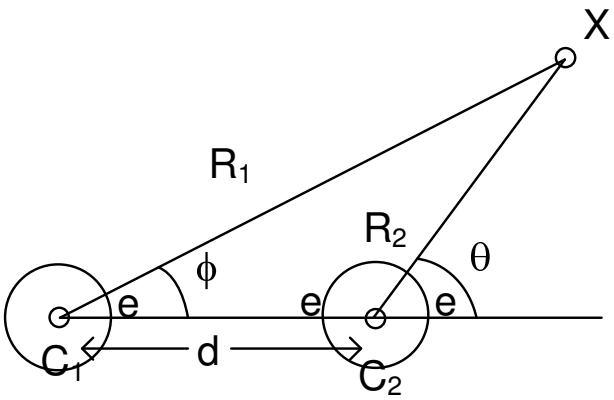

Fig. 3: 2D representation of the geometry between cameras and the 3D point

In the spherical framework, the disparity can be defined as the difference in angle values in the representation of the same 3D point in two different omnidirectional images. Since pixel coordinates are given with spherical coordinates, we define the disparity γ as the difference between the angles θ and ϕ corresponding to pixel coordinates on the two images, i.e., $\gamma = \theta - \phi$. Figure 3 shows the 2D representation of the geometry relationship between omnidirectional cameras and a point X in the

3D space. The depth $R1$ is defined as the distance of the point X to the reference camera center. The relation between the disparity γ, the depth $R1$ and baseline distance d, is given as

$$\gamma = arcsin\frac{dsin\theta}{R1} \tag{1}$$

This relation holds for all epipolar great circles on rectified stereo images. Based on equation (1), one could define equi-disparity fields as illustrated in Figure 4. We observe that the representation of depth is not very reliable around epipoles where depth is finely sampled so that high resolution is required for computing disparities. When the distance increases, or equivalently for far objects, the accuracy of the depth representation also suffers from the coarse sampling of the disparity. In these regions, a single disparity change leads to big changes in depth and might result into accumulated estimation errors.

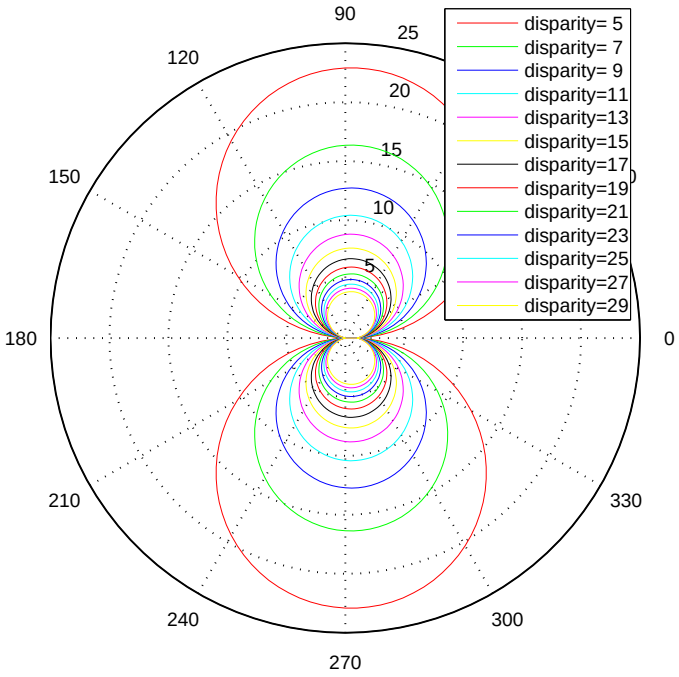

Fig. 4: Equidisparity fields for a system with baseline distance of $d = 1$ and pixel resolution of 360 for θ. Disparities are given in terms of number of pixels.

C. Dense disparity estimation problem

We formulate now the problem of disparity estimation between spherical images as an energy minimization problem. Let $\mathcal{P}_l$ and $\mathcal{P}_r$ denote the set of all pixels on left and right rectified spherical images. Note that left image is the reference image. Let further $\mathcal{L} = \{l_1, l_2, \ldots, l_{max}\}$ represent the finite set of discrete labels corresponding to disparity values. Recall that the disparity estimation problem rendered mono-dimensional due to the image rectification step. A single value per pixel is therefore sufficient to represent the disparity, and the disparity values for each pixel together form the disparity map. If $f : \mathcal{P}_l \rightarrow \mathcal{L}$ is a mapping function so that each pixel on the reference image is assigned a disparity label, our aim is to find the optimum mapping f^* such that the disparity map is as accurate and smooth as possible. The computation of the optimum mapping f^* can therefore be formulated as an energy minimization problem, where the energy function $E(f)$ is built on two components E_d and E_σ, which respectively represent the data and smoothness functions :

$$E(f) = E_d(f) + E_\sigma(f) \tag{2}$$

The data function first reports the photo-consistency between both omnidirectional images. It can be written as:

$$E_d(f) = \sum_{p \in \mathcal{P}_l, q \in \mathcal{P}_r} D(p, q) \tag{3}$$

where p and q are corresponding pixels in two images under a mapping function f. $D(.,.)$ is a function to check pixel dissimilarity. It can be truncated absolute difference in the form

$$D(p, q) = min\{\|I(p) - I(q)\|, K_d\} \tag{4}$$

or truncated square difference which can be written as

$$D(p,q) = \min\{(I(p) - I(q))^2, K_d\}, \tag{5}$$

where $I(i)$ represents the intensity or luminance of pixel i and K_d is a positive constant. In this work, we define the dissimilarity $D(p,q)$ in a way that is similar to the function proposed in Birchfield and Tomasi's work [Birchfield and Tomasi(1998)]:

$$D(p,q) = \min\left\{\bar{d}\left(p,q,I_L,I_R\right), \bar{d}\left(q,p,I_R,I_L\right)\right\}, \tag{6}$$

where I_L and I_R represent the intensity on the scanlines of the right and left images, and where $\bar{d}\left(q,p,I_1,I_2\right)$ describes the minimum difference in intensity between I_1 taken at pixel q and I_2 i that is linearly interpolated around pixel p. This metric presents the advantage to be insensitive to image sampling, which is quite useful since the equiangular grid on the sphere causes non-uniform sampling in terms of sampling density. For color images, the data cost is calculated for each band and then averaged by giving same priority.

The smoothness function then captures the variations of the disparity between neighboring labels. The goal of the smoothness function is to penalize the estimation of labels that are different from their neighborhood in order to obtain a smooth disparity field. The neighborhood $\mathcal{N}$ is generally represented by the four surrounding labels. The smoothness function under a mapping f can be expressed as:

$$E_\sigma(f) = \sum_{(p,q)\in\mathcal{N}} V_{p,q}(f(p), f(q)). \tag{7}$$

The term $V_{p,q}$ is a distance metric that reports the difference between labels attributed to neighboring pixels in $\mathcal{N}$. One of the possible functions for $V_{p,q}$ is defined by the Potts model [Boykov et al(2001)Boykov, Veksler, and Zabih] under the form

$$V_{p,q} = \lambda * \sigma_{l_p,l_q} \tag{8}$$

where σ is delta function which gives 1 when $l_p = l_q$ and 0 otherwise. The Potts model penalizes equally all disparity differences. Another function is truncated linear which is written as

$$V_{p,q} = \lambda * min(\|l_p - l_q\|, K_\sigma) \tag{9}$$

Truncated linear function takes into account the amount of difference between disparities but truncates at K_σ to prevent overpenalization of large values allowing discontinuities. The parameter λ is a smoothness weight factor to control smoothness level compared to data cost. Both Potts and truncated linear functions are metric functions which will be important for the graph-cut algorithm. The dense disparity estimation problem now consists in minimizing the energy function $E(f)$, in order to obtain an accurate and smooth disparity map.

III. Dense disparity estimation with graph-cut

Global energy minimization problems such as the estimation problem defined in the previous section typically find solutions inalgorithms based on graph-cut, or belief propagation methods. In this paper, we choose to minimize the function $E(f)$ with a graph-cut algorithm, which has been adapted to the specificities of the spherical framework. Basically, the graph-cut algorithm converts the energy minimization problem into several minimum cut problems. A graph is constructed on the pixels for each disparity label. Since the graph-cut algorithm is optimum in graphs for single label differences, multiple labels are processed iteratively by the construction of graphs for each label.

We design a graph-cut algorithm with α-expansion method in order to solve the energy minimization problem of Eq. (2). This method is based on graph construction illustrated in Figure 5. Two terminal nodes called source and sink represent the binary values for a label. The source node represents the label α and the sink one represents the label $\bar{\alpha}$. Intermediate nodes of the graph are given for each pixel, and edges are labeled with weights corresponding to costs. Edges between terminal nodes and pixel nodes represent the data cost function $D_{p,q}$ and edges between neighboring pixels represent the smoothness cost $V_{p,q}$. These edges are weighted by the value of the corresponding cost functions. The data cost $t_{p\alpha}$ for a pixel p with a disparity α is equal to

$$t_{p\alpha} = D(p, p-\alpha) \tag{10}$$

Similarly, $t_{p\bar{\alpha}}$ is the data cost for disparities other than α assigned to pixel p and can be written as

$$t_{p\bar{\alpha}} = D(p, p-\bar{\alpha}), \quad \bar{\alpha} \in \mathcal{L}, \quad \bar{\alpha} \neq \alpha \tag{11}$$

Smoothness term is denoted by e_{pq} in the graph and is equal to

$$e_{pq} = V(l_p, l_q) \tag{12}$$

With this representation, the optimization problem is equivalent to finding the cut in the graph with minimum total cost, which corresponds to the minimum of the energy $E(f)$. A cut is typically performed through the edges of the graphs and the total cost is the sum of the weights of the edges affected by the cut. In the experiments, we use the graph-cut algorithm based in part

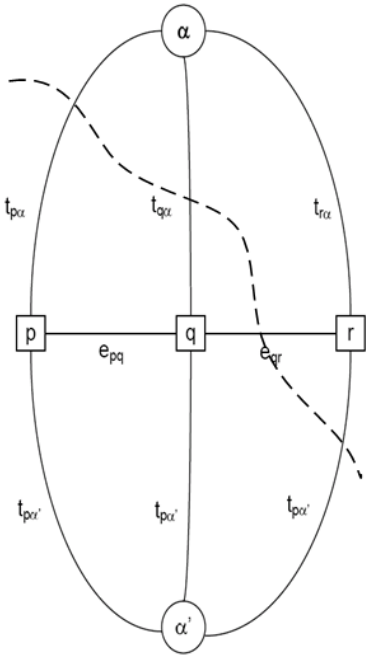

Fig. 5: The graph with source and sink nodes for the label and the intermediate nodes for the pixels, and edges with the weights corresponding to costs

on the implementations used in [Szeliski et al(2006)Szeliski, Zabih, Scharstein, Veksler, Kolmogorov, Agarwala, Tappen, and Rother] and [Boykov et al(2001)Boykov, Veksler, and Zabih], [Kolmogorov and Zabin(2004)], [Boykov and Kolmogorov(2001)]. The algorithm has been adapted to the spherical framework by taking into account the particular connectivity of the image boundaries. Interestingly, graph-cut algorithms are known to reach a global minimum or a strong local minimum independent of the initial disparity map.

It has to be noted that the alpha-expansion method in the graph-cut algorithm requires the smoothness term to be metric. The Potts and truncated linear functions comply with this requirement. In our implementation, we have applied the workaround used in [Szeliski et al(2006)Szeliski, Zabih, Scharstein, Veksler, Kolmogorov, Agarwala, Tappen, and Rother] to use semi-metric smoothness terms as well. This workaround truncates the values which fails to satisfy the regularity term explained in [Kolmogorov and Zabin(2004)]. In our tests, we have observed that the truncated linear function with truncation around a label difference of 5 gives the best results. On the other hand, the Potts model tends to oversmooth small differences due to an equal penalty value for each difference. This is a drawback for spherical dense disparity estimation where planar surfaces typically present smooth label differences.

A. Experimental results

1) Test images: We evaluate now the performance of the disparity estimation solution for both synthetic and real images. We have used 3 synthetic and one natural 3D scene. The synthetic scenes have been generated by the Blender [Blender(2007)] and Yafray [Yafray(2007)] programs. The first synthetic scene consists of 4 planes with distances 7,7,8 and 10 units from the reference camera which is positioned on the world origin. The second camera is one unit distant from the reference camera on the positive x-axis. For the sake of simplicity, there is no relative rotation between cameras. Reference camera is called left and the other camera is called right camera. Figure 6 shows the generated left and right images, and the corresponding synthetic scene with a top view. With these parameters, epipoles are located at 90° of latitude, 0° and 180° longitude angles for both cameras.

The second synthetic scene consists of different surface structures and a closed environment. There are 3 planes with different sizes, positions and orientations, a cylindirical surface patch and spherical surface patch. Similar to the first scene left and right cameras are placed at positions $(0, 0, 0)$ and $(1, 0, 0)$ respectively. The minimum distance to the objects has been set to 5. The left and right images and the scene model are shown in Figure 7.

The last synthetic scene is a realistic looking room scene with complex objects and illumination. Camera placement is same with the previous two synthetic scenes. The minimum distance to objects has been set to 3. Figure 8 shows the left, right images and the scene model.

Finally, the natural scene has been captured by a catadioptric system with a paraboloidal mirror. Two images of a room captured from two different positions are shown in Figure 9. We map the images onto the 2D sphere via inverse stereographic

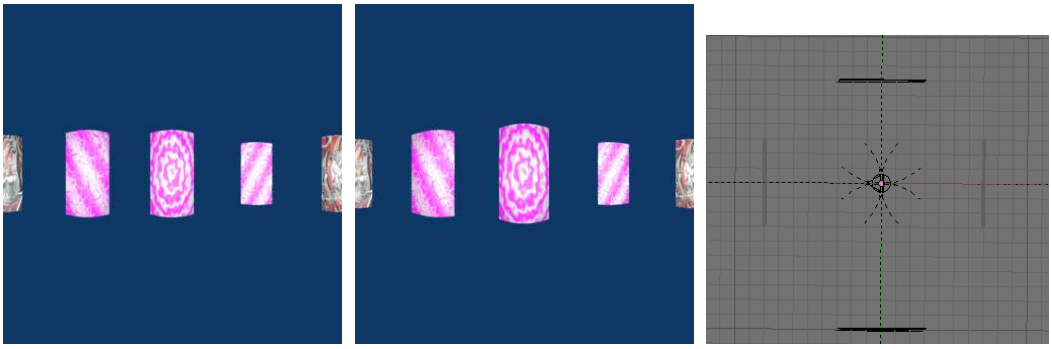

Fig. 6: Left and right images, and top view of the first synthetic scene.

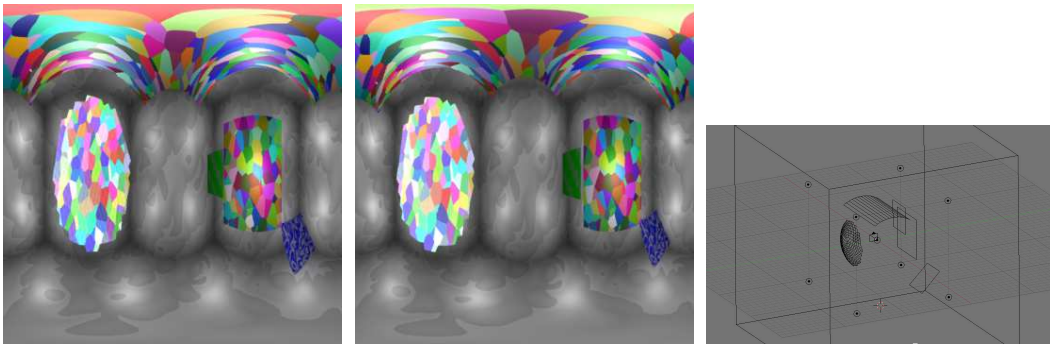

Fig. 7: Left and right images for the second scene, and scene model of the synthetic scene.

Fig. 8: Left and right images for the synthetic room scene, and scene model of the synthetic scene.

projection [Geyer and Daniilidis(2001)]. The images mapped on the sphere and the rectified images have been illustrated in Figure 2.

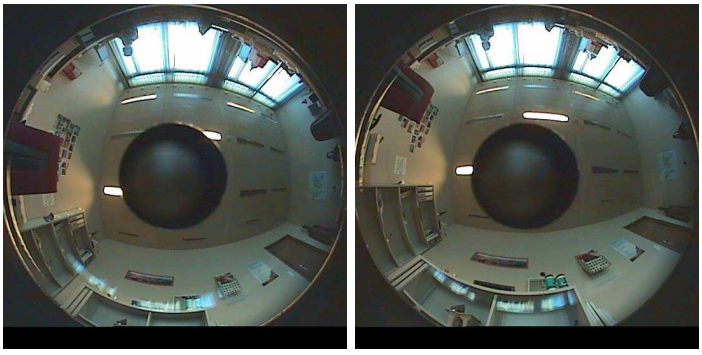

Fig. 9: Captured omnidirectional images.

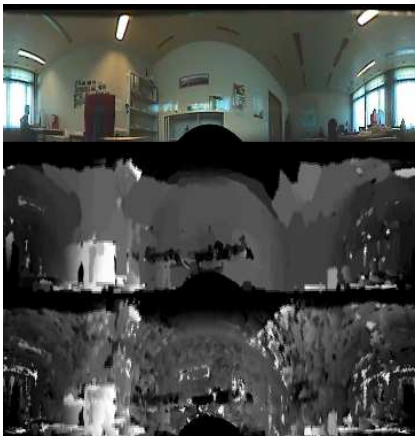

Fig. 10: Reference image and disparity images calculated with GC and WTA methods respectively for the real image set.

2) Performance analysis: We compare here the performance of the proposed graph-cut algorithm (GC) with a block matching algorithm (WTA) based on the winner-takes-all principle. It consists in a local optimization method that tries to find the best match between blocks of pixels on the sphere, based on the highest correlation score (i.e., $L2$-norm computed on the intensity values, averaged on the three color channels). Figure 10 shows the disparity maps computed by the graph-cut algorithm on the sphere and the disparity map obtained by the block matching algorithm for the natural scene. It can be seen that the disparity estimation is more precise in the GC method than in the WTA algorithm, which is not able to provide a very dense disparity map. Finally, disparity maps calculated with both GC and WTA for the 3 synthetic scenes are given in Figure 11 along with groundtruth disparity values. We observe that GC provides a better accuracy with increased smoothness of the disparity map. These observations are confirmed by the mean-square error (MSE) computation between the disparity maps produced by GC and WTA, and the groundtruth map. The MSE values given in Table I are clearly scene dependent, but they still show the superior performance of GC compared to WTA.

TABLE I: MSE values with respect to ground truth disparities

	GC	WTA
Synthetic scene 1	0.5456	15.9939
Synthetic scene 2	3.7526	13.2085
Synthetic scene 3	23.7460	25.0863

We also analyze the performance of the disparity estimation method in terms of depth computation. Recall that the depth and disparity values are linked by Eq. (1). We calculate the depth values from the estimated disparity maps, and we compute the MSE distortion with respect to the ground truth depth information. The depth estimation results are given in Table II for the 3 synthetic scenes for both the GC and WTA disparity estimation algorithms. We observe that, if the objects are close to cameras, the GC algorithm performs better and gives more accurate results that the WTA solution, as confirmed by the results on the first synthetic scene. However, if the objects are far from the cameras, the depth estimation suffer from the coarse estimation of disparities and the results become equivalent for both algorithms.

TABLE II: MSE distortion on estimated depth maps for GC and WTA (synthetic scenes)

MSE	GC	WTA
Synthetic scene 1	1.9676	40.1074
Synthetic scene 2	8.8849	29.3051
Synthetic scene 3	38.7207	52.8364

Finally, we analyze the performance of the disparity estimation algorithms from a scene reconstruction perspective. We wrap the left image using the computed disparity maps in order to estimate the right image. Figure 12 shows the ground truth right image, along with the results of warping based on the disparity maps obtained respectively with WTA and GC. We can see that GC clearly allows to obtain a better estimate of the right image. This is confirmed by the MSE distortion computed on the right image(see Table III). We can conclude that the GC algorithm gives better performance in terms of disparity estimation, both from a visual or MSE distortion viewpoints.

IV. Parallel graph-cut implementation

We have shown that the graph-cut algorithms provides promising results for dense disparity estimation on stereo image pairs. Even if it gives a solution in polynomial time, the computational cost is still high for real-time applications. We propose a

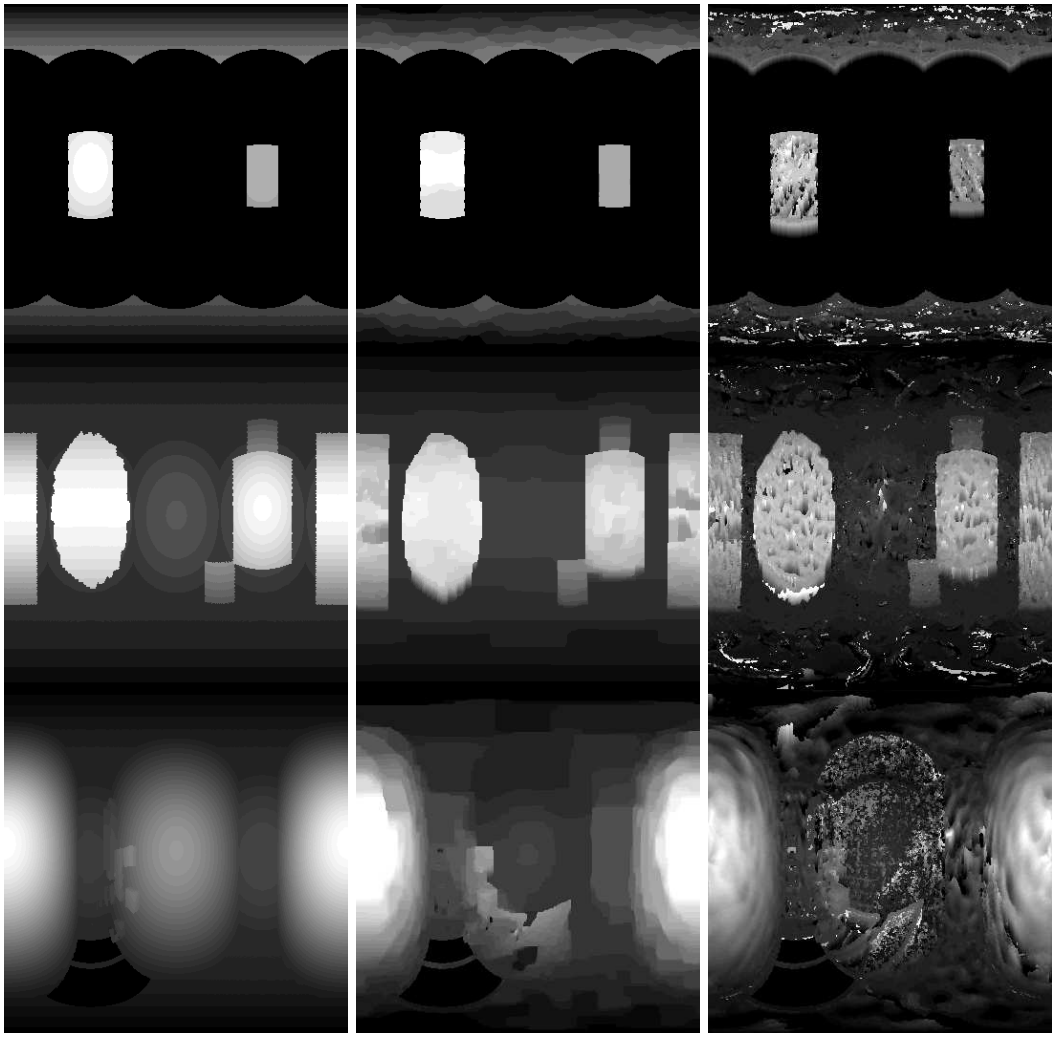

Fig. 11: Groundtruth disparities values (Leftmost column), GC disparities (Middle column) and WTA disparities (Rightmost column) for the synthetic scenes.

TABLE III: MSE distortion of the warped right images compared to the original images in the synthetic scenes.

	GC	WTA
1. Synth.	774.6006	3497.7
2. Synth.	576.6756	3894.1
3. Synth.	425.7070	3659.9

method to reduce the computational time by splitting the graph into subgraphs for rows and columns, which enables parallel processing.

The energy function in Section II-B can be described as a maximum a posteriori estimation of the probability density function $P(X|Y)$ where Y correspond to the observations or the pixel values, and X is the disparity function to be estimated. From Bayes' relation $P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}$, we see that both terms $P(Y|X)$ and $P(X)$ drives the estimation of the probability function $P(X|Y)$. They correspond to the two terms in the energy function. $P(Y|X)$ models the data cost and describes the distribution of the observations given the value to be estimated. It is modeled by intensity differences between two images given a disparity label and is a function of disparity label l. $P(X)$ represents the distribution of the disparity function X and models the relation among neighboring disparities for disparity estimation problem. It corresponds to smoothness term and modeled by label differences on pixel neighborhood. This is actually the term that induces the computational complexity of the depth estimation algorithm because it requires processing of the whole data at once and prevents separation of the processing on columns and rows.

We propose here a method that splits the graph into subgraphs corresponding to pixel rows and columns and reduces the computational time by a parallel implementation. It is based on the idea that the solution from an iteration of the disparity estimation algorithm can be considered as an observation for the next iterations. The smoothness terms corresponding to

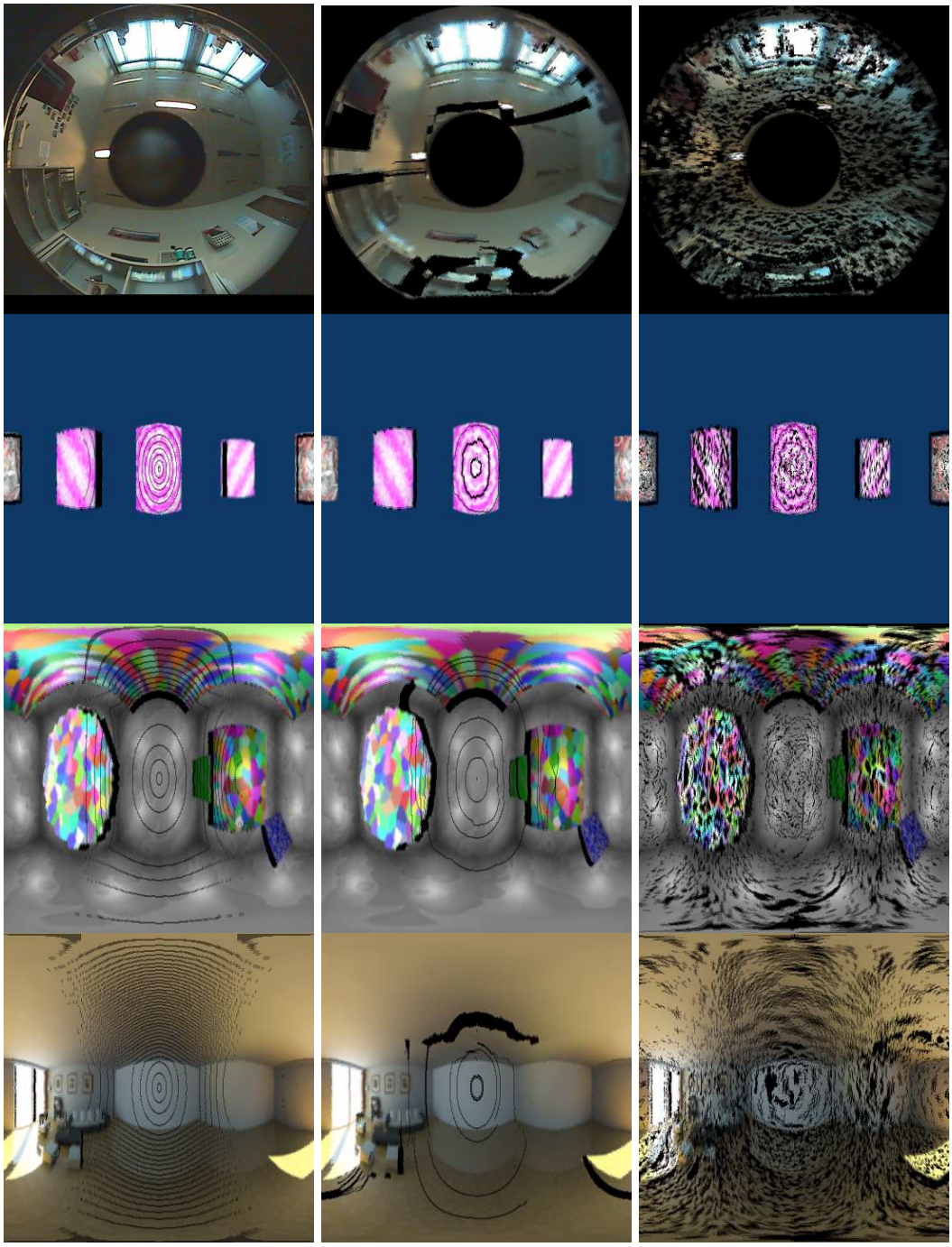

Fig. 12: Right image warped from the left image with groundtruth disparities, and with WTA and GC disparity maps, respectively, for the synthetic scenes.

columns or rows can be used in the data cost terms when the disparity is computed on rows, respectively columns.

In more details, multiple graphs are constructed for each of pixel rows. At each iteration of the disparity estimation algorithm, the weight of the edges connecting the pixels to the source and sink nodes become now

$$t_{p\alpha} = D(p, p-\alpha) + V(l_m, \alpha) + V(l_n, \alpha), m, n \in N_{col}, \tag{13}$$

and respectively

$$t_{p\bar{\alpha}} = D(p, p-\bar{\alpha}) + V(l_m, \bar{\alpha}) + V(l_n, \bar{\alpha}), m, n \in N_{col}, \bar{\alpha} \neq \alpha. \tag{14}$$

Note that l_m and l_n are the disparity labels obtained at the previous iteration of the algorithm, and N_{col} correspond to the

columns in the images. The new smoothness term is given as

$$e_{pq} = V(l_p, l_q), p, q \in N_{row}, \tag{15}$$

where N_{row} represents the pixel rows. The graph diagram and the new neighborhood is given in Figure 13. Since the connections among pixels in a subgraph can be represented in one dimension, each row can be processed independently by moving the smoothness term into the data cost term for processing in the other dimension. It can be noted that the above formulation describes the processing of pixel rows. Columns are processed similarly by interchanging rows by columns in the formulas given above.

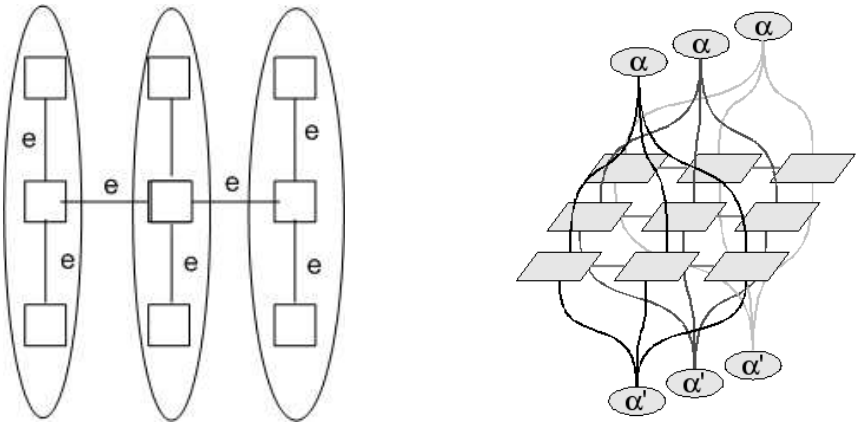

Fig. 13: The graph neighborhood and the connectivity provided by the edges weighted by the smoothness terms in the fast graph-cut algorithm.

The fast graph-cut algorithm is summarized below in Algorithm 1.

Algorithm 1 Parallel graph-cut algorithm for disparity estimation

Initialization: Set the initial disparity map to zero
repeat
  for each row **do**
    construct a graph
    find the minimum cut
  end for
  for each column **do**
    construct a graph using the disparity map from calculation on rows
    find the minimum cut
  end for
until minimum is reached

We analyze now the performance of the parallel graph-cut implementation, which might to different performance due to the separate computation of the disparity estimation on pixel rows and columns. We estimate the disparity map for the original graph-cut implementation described in the previous section, and for the fast implementation. We estimate the right image by warping the left image with help of the estimated disparity. Table IV shows MSE distortion when the wapred image is compared to the original right image for the 4 data sets and for both graph-cut implementations. The corresponding images are given in Figure 14.

TABLE IV: MSE distortion on the estimation of the right image by warping using disparity maps from full and parallel GC

	Full GC	Parallel GC
Natural Scene	764.79	744.62
Synthetic scene 1	854.50	853.86
Synthetic scene 2	581.78	570.80
Synthetic scene 3	425.71	449.59

Both the MSE distortion results and visual inspection show that the fast algorithm reaches similar performance as the original GC algorithm. Although the warping results are similar for both algorithms, it should be noted that the final energy values for both algorithms are different. The proposed algorithm is actually suboptimal and results into slightly higher values of the energy

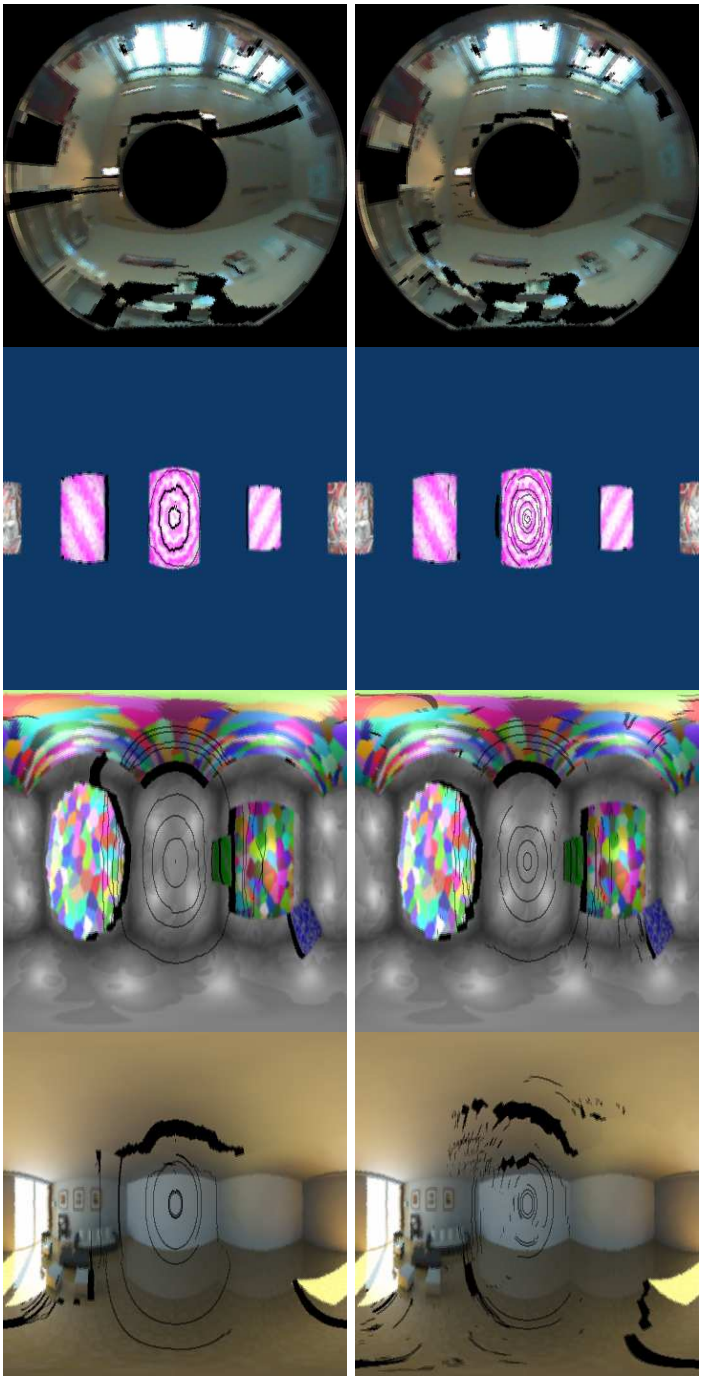

Fig. 14: Estimation of the right images by warping of the left images with the disparity maps computed with full graphcut (left column) and the parallel algorithm (right column), for the four test scenes.

cost function. This difference is due to the smoothness component which is distributed in the proposed parallel algorithm. At the same time it provides an important reduction of the computation time, as shown in Table V. The tests are performed on a machine with Intel Centrino 1.7GHz single core processor. The proposed algorithm is at least 3 times faster than the original graph-cut algorithm. These computation time values are given for a single processor. With the parallel structure of the proposed algorithm, the speed can yet be improved by a factor that depends on the number of available processors, which makes it particularly interesting for real-time processing.

V. Depth estimation multiple cameras

This section extends the depth estimation problem to scenarios with more than 2 cameras. Let I_k be k^{th} spherical image in a set with N such images captured from different viewpoints in a scene. We assume that the camera parameters are known

TABLE V: Computation time for both algorithms on a single CPU system.

	Full GC [sec]	Parallel GC [sec]
Natural Scene	13.4	3.6
Synthetic scene 1	4.1	1
Synthetic scene 2	6.6	2
Synthetic scene 3	9.8	1.6

for all cameras. The problem consists in estimating the depth values from an arbitrary point in the same scene using multiple spherical images. We want to determine depth values from a point in the 3D scene that either corresponds to one of the camera centers or to an arbitrary point. When the camera image from the arbitrary point is available, this image can be used as reference. If the camera is not available, it leads to a no-reference depth estimation. We propose below two algorithms for solving the depth estimation problem, which are respectively based pairwise disparity computation, and on computation of the inverse depth with global photo-consistency.

This problem is similar to the dense depth estimation problem with multiple perspective cameras. Differences however appear in the sampling structure of the images, as well as in the large field of view of the omnidirectional sensors. The latter actually makes the problem quite different from conventional dense disparity estimation since the omnidirectional cameras are outward looking and the objects are outside of the hull encapsulating the camera set. The sampling grid also requires a different processing because of the non-uniform density of the samples. Interestingly, it can be noted that the addition of extra cameras is expected to overcome the problem of inaccuracies around the epipolar regions that appears when only two images are used for disparity estimation, unless the cameras are colinear.

Without loss of generality, we study the depth estimation problem in the case of three cameras. We generate three images for each one of the synthetic scenes described above, by placing the cameras $C1$, $C2$, and $C3$ on vertices of an equilateral triangle with positions $(0, 0, 0)$, $(1, 0, 0)$, and $(0.5, \sqrt{3}/2, 0)$ respectively. The resulting images are illustrated in Figure 15.

A. Pairwise disparity estimation

The depth estimation with multiple cameras can be solved by extending the algorithm proposed in Section III by merging the results of successive disparity estimation on paris of images. For example, the proposed graph-cut algorithm can be applied on the two pairs of cameras $(C1, C2)$ and $(C2, C3)$, respectively. After rectification, the GC algorithm runs in parallel on both pairs of images in order to compute the disparities and distances to cameras $C1$ and $C2$. Later, the estimated disparity maps are merged together for global depth estimation. For example, the distances values obtained for the camera $C2$ are mapped onto the coordinate system of the first camera $C1$. Each disparity maps is mutliplied by a spherical weighting function that reports the importance of each pixel depending on its position. Typically, one can define the weighting function as a sine function on the sphere (see Eq. (1), with appropriate rotation accroding to the epipolar constraints. Finally, the depth values is selected from the disparity map that presents the highest weight value at a given pixel position.

Figure 16 presents the depth estimation results for the cases of two and three cameras, respectively. It shows that the artifacts on the epipolar regions are effectively removed by the introduction of a third camera. This improvement is more visible on objects around epipolar regions and on close objects. Due to lack of precision caused by coarse sampling of depth for far objects, results vary much and are not always consistent. Even if the GC algorithm provides promising results when the number of cameras increases, we have observed that low texture areas still prevent the algorithm to converge to better results, as it can be observed on the synthetic room scene.

B. Global photo-consistency

We present here an alternative algorithm for the depth estimation with multiple spherical images. This algorithm uses all the camera images together. It first discretizes the depth values Z such that inverse depth $1/Z$ is uniformly sampled in the range $[1/Z_{min} 1/Z_{max}]$, similarly to [Strecha et al(2006)Strecha, Fransens, and Van Gool]. A ray with discretized depth values thus emanates from each pixel on the virtual camera and depth discretization forms a set of concentric spheres that are quite interesting in our spherical framework.

Similarly to the stereo case, we formulate the depth estimation as an energy minimization problem in the form

$$E(f) = E_{data}(f) + E_{smooth}(f) \tag{16}$$

The function $E_{smooth}(f)$, represent the same smoothness constraint that penalizes the disparity differences between neighboring pixels. The data cost function $E_{data}(f)$ however considers the global photo-consistency for corresponding pixels in all images, at a given discretized depth value. If there is no reference image at the desired position, the data cost at pixel position p for depth level l is calculated as

$$D(p, l) = \sum_{i=1}^{N} I_k(M(p, l)) - \mu(l) \tag{17}$$

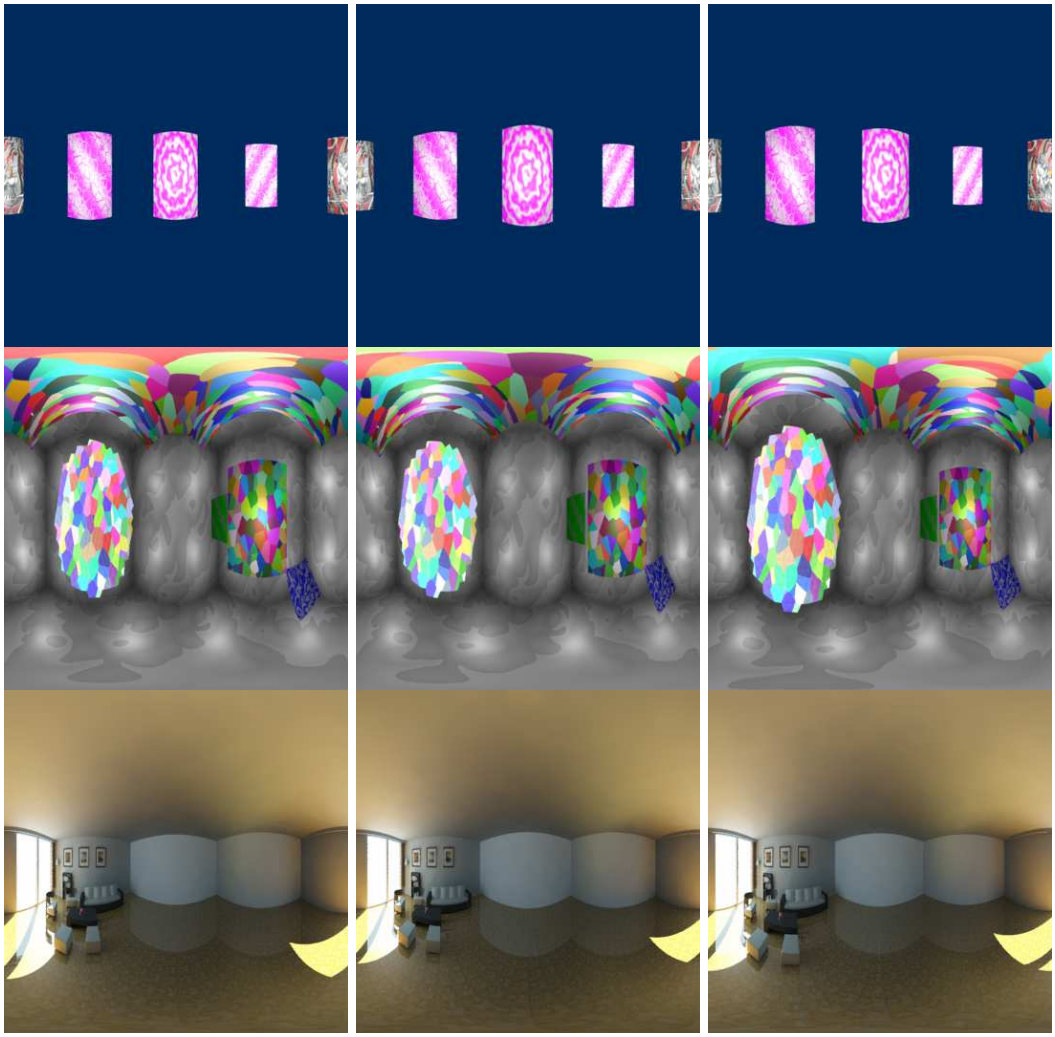

Fig. 15: Synthetic images for three cameras $C1$ (left), $C2$ (middle) and $C3$ (right).

where $M(.,.)$ computes the projection of 3D point defined by pixel point p and depth level l and $\mu(l)$ is the mean of all intensity values corresponding to this 3D point. This data cost will give high cost value when the 3D point does not correspond to the real 3D point of interest and will give low cost in the proximity of the real 3D point. In addition, the normalization with the mean function reduces the impact of outliers or occluded regions. If there is a reference image at the virtual camera position, the mean value $\mu(l)$ is replaced by the pixel value of the reference image at the pixel position p. The data cost function becomes

$$D(p,l) = \sum_{i=1}^{N} I_k(M(p,l)) - I_*(p) \tag{18}$$

where I_* is the reference image. The minimization of the energy function is finally performed using a graph-cut algorithm, as explained in the previous sections.

The proposed method is tested on spherical images from a realistic looking synthetic scene. Images are taken from different positions in the scene. Figure 17 shows the scene from the virtual view, the ground truth non-discretized depth map and the estimated depth map with and without a reference image. As in the stereo case, the smoothness factor affects the estimation results around the regions with depth discontinuities. Note that the results computed with and without reference image are similar. Although the knowledge of exact intensity values is known to improve the photoconsistency, the lack of such a reference does actually not penalize the depth estimation.

Finally, we compare the inverse depth based estimation with the above solution that uses pairwise disparity estimation. Figure 18 illustrates the depth estimation performance for both schemes. Since it considers the information of all images simultaneously, the approach based on inverse depth compuation performs better than the solution that merges pairwise disparity maps. It has to be noted however that this algorithm requires some prior information about the scene, such as minimum and maximum depth values that are used in the depth discretization process.

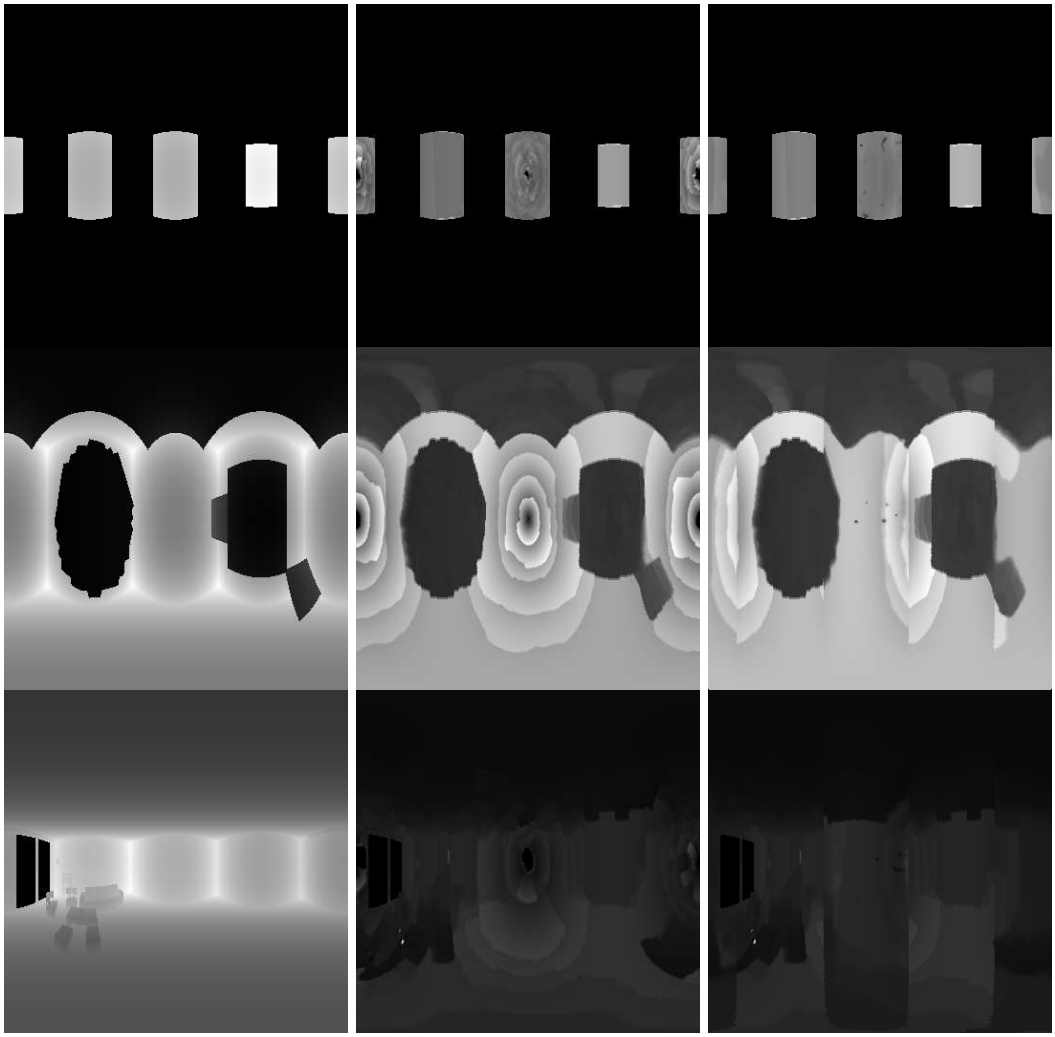

Fig. 16: Ground truth distances, and distances obtained by GC with two and three cameras.

VI. Conclusion & Discussion

We have addressed the problem of dense depth estimation from omnidirectional images. We have proposed a framework for processing the visual information on the 2D sphere in order to preserve the geometry information. We have adapted a strong energy minimization algorithm based on graph-cut for estimation of disparities on spherical images. A simple yet efficient rectification process based on simple rotation in the spherical framework, combined with an adapted graph-cut algorithm offers promising disparity estimation results. It outperforms a block matching algorithm based on a Winner-takes-all strategy. A fast implementation of the algorithm has been proposed by parallel processing of pixel rows and columns, which leads to important speed-up and permits implementation in realtime applications. The extension of the proposed algorithm with the introduction of additional cameras allow to overcome sampling problems in the regions around epipoles and offers promising performance for dense depth estimation in networks of omnidirectional cameras.

References

[Bartczak et al(2007)Bartczak, Koeser, Woelk, and Koch] Bartczak B, Koeser K, Woelk F, Koch R (2007) Extraction of 3D freeform surfaces as visual landmarks for real-time tracking. Journal of Real-Time Image Processing 2(2):81–101

[Birchfield and Tomasi(1998)] Birchfield S, Tomasi C (1998) A pixel dissimilarity measure that is insensitive to image sampling. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(4):401–406

[Blender(2007)] Blender (2007) Blender 2.43. http://www.blender.org

[Boykov and Kolmogorov(2001)] Boykov Y, Kolmogorov V (2001) An Experimental Comparison of Mm-cut/Max-flow Algorithms for Energy Minimization in Vision. Energy Minimization Methods in Computer Vision and Pattern Recognition: Third International Workshop, EMMCVPR 2001, Sophia Antipolis, France, September 3-5, 2001: Proceedings

[Boykov et al(1998)Boykov, Veksler, and Zabih] Boykov Y, Veksler O, Zabih R (1998) Markov random fields with efficient approximations. Computer Vision and Pattern Recognition, 1998 Proceedings 1998 IEEE Computer Society Conference on pp 648–655

[Boykov et al(2001)Boykov, Veksler, and Zabih] Boykov Y, Veksler O, Zabih R (2001) Fast approximate energy minimization via graph cuts. Pattern Analysis and Machine Intelligence, IEEE Transactions on 23(11):1222–1239

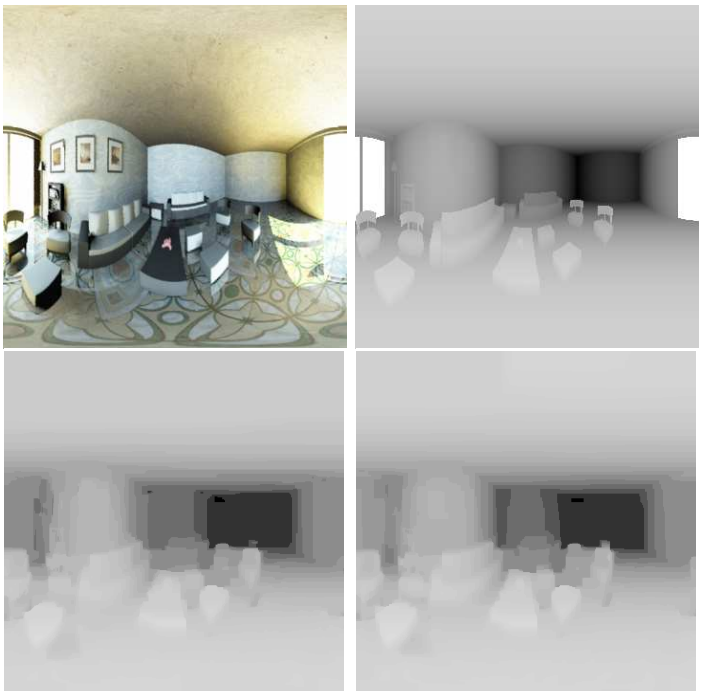

Fig. 17: First row: reference image and ground truth depth map. Second row: estimated depth map with reference (left), and without reference(right)

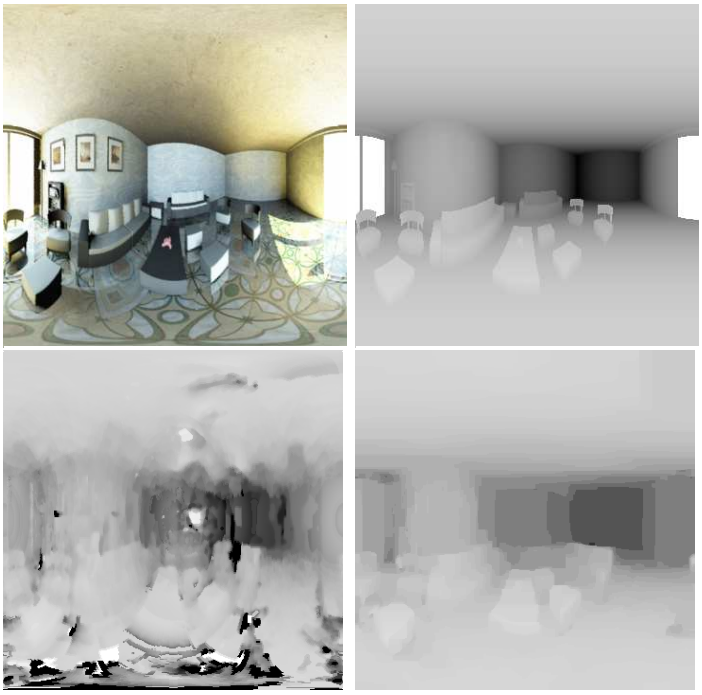

Fig. 18: First row: Reference image and ground truth depth map. Second row: depth estimation by merging disparity maps (left), and by the inverse depth method (right).

[Fleck et al(2005)Fleck, Busch, Biber, Strasser, and Andreasson] Fleck S, Busch F, Biber P, Strasser W, Andreasson H (2005) Omnidirectional 3D Modeling on a Mobile Robot using Graph Cuts. Robotics and Automation, 2005 Proceedings of the 2005 IEEE International Conference on pp 1748–1754

[Fleck et al(2009)Fleck, Busch, Biber, and Straßer] Fleck S, Busch F, Biber P, Straßer W (2009) Graph cut based panoramic 3D modeling and ground truth comparison with a mobile platform–The Wägele. Image and Vision Computing 27(1-2):141–152

[Geyer and Daniilidis(2001)] Geyer C, Daniilidis K (2001) Catadioptric Projective Geometry. International Journal of Computer Vision 45(3):223–243

[Geyer and Daniilidis(2003)] Geyer C, Daniilidis K (2003) Conformal Rectification of Omnidirectional Stereo Pairs. Omnivis 2003: Workshop on Omnidirectional Vision and Camera Networks

[Gonzalez-Barbosa and Lacroix(2005)] Gonzalez-Barbosa J, Lacroix S (2005) Fast Dense Panoramic Stereovision. Robotics and Automation, 2005 Proceedings of the 2005 IEEE International Conference on pp 1210–1215

[Kolmogorov and Zabin(2004)] Kolmogorov V, Zabin R (2004) What energy functions can be minimized via graph cuts? Pattern Analysis and Machine Intelligence, IEEE Transactions on 26(2):147–159

[Meguro et al(2007)Meguro, Takiguchi, Amano, and Hashizume] Meguro J, Takiguchi J, Amano Y, Hashizume T (2007) 3D Reconstruction Using Multibaseline Omnidirectional Motion Stereo Based on GPS/Dead-reckoning Compound Navigation System. The International Journal of Robotics Research 26(6):625

[Scharstein and Szeliski(2002)] Scharstein D, Szeliski R (2002) A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision 47(1):7–42

[Strecha et al(2006)Strecha, Fransens, and Van Gool] Strecha C, Fransens R, Van Gool L (2006) Combined depth and outlier estimation in multi-view stereo. In: Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, IEEE Computer Society, pp 2394–2401

[Sun et al(2003)Sun, Shum, and Zheng] Sun J, Shum H, Zheng N (2003) Stereo matching using belief propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(7):787–800

[Szeliski et al(2006)Szeliski, Zabih, Scharstein, Veksler, Kolmogorov, Agarwala, Tappen, and Rother] Szeliski R, Zabih R, Scharstein D, Veksler O, Kolmogorov V, Agarwala A, Tappen M, Rother C (2006) A comparative study of energy minimization methods for markov random fields. Proc Europ Conf Comp Vision

[Takiguchi et al(2002)Takiguchi, Yoshida, Takeya, Eino, and Hashizume] Takiguchi J, Yoshida M, Takeya A, Eino J, Hashizume T (2002) High precision range estimation from an omnidirectional stereo system. Intelligent Robots and System, 2002 IEEE/RSJ International Conference on 1:263–268

[Yafray(2007)] Yafray (2007) Yafray 0.09. http://www.yafray.org

[Ying and Hu(2004)] Ying X, Hu Z (2004) Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. Lecture Notes in Computer Science pp 442–455