A Surface Reconstruction Method Using Global Graph Cut Optimization

Sylvain Paris, François X. Sillion
ARTIS[†], GRAVIR/IMAG - INRIA

Long Quan
HKUST[‡]

Abstract

The surface reconstruction from multiple calibrated images has been mainly approached using local methods, either as a continuous optimization driven by level sets, or as a discrete volumetric method of space carving. We here propose a direct surface reconstruction approach. It starts from a continuous geometric functional that is then minimized up to a discretization by a global graph-cut algorithm operating on a 3D embedded graph. The method is related to the stereo disparity computation based on graph-cut formulation, but fundamentally different in two aspects. First, the existing stereo disparity methods are only interested in obtaining layers of constant disparity, while we focus on a surface geometry of high resolution. Second, only approximate solutions are reached by most of the existing graph-cut algorithms, while we reach a global minimum. The whole procedure is consistently incorporated into a voxel representation that handles both occlusions and discontinuities. It is demonstrated on real sequences, yielding remarkably detailed surface geometry up to $1/10$*th pixel.*

1 Introduction

In this paper, we consider the problem of 3D reconstruction of an object in volumetric or surface representations observed by several calibrated cameras. Many researchers have been interested in this problem and have proposed different methods, including a continuous geometric formulation solved by local optimization methods like *space carving* [18] or *level sets* [9] and a discrete labeling formulation for computing stereo disparities solved by global *graph-cut* methods [5, 6, 11, 13–16, 23, 31]. These methods have their own strengths and weaknesses. A continuous geometric formulation appears to be more suited to this problem as it is intrinsically related to the 3D space, while a global optimization ensures a better object solution seen as a whole rather than as a collection of small patches. A few of these methods are described here without being exhaustive.

Direct volumetric methods The method of *space carving* or *voxel coloring* [18, 19, 24, 25, 28] directly works on discretized 3D space, voxels, based on their image consistency and visibility. These techniques are characterized by their local treatment: voxels are considerated separately. For more details, one may read the following survey [27].

[†] {sylvain.paris|francois.sillion}@imag.fr
ARTIS is a research project in the GRAVIR/IMAG laboratory, a joint unit of CNRS, INPG, INRIA and UJF.
[‡] quan@cs.ust.fr

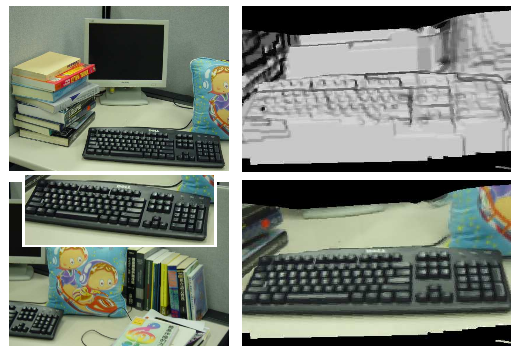


Figure 1: The reconstructed surface geometry (right) from 11 images at 640 × 480 (left). Notice that the keys of the keyboard are clearly distinguishable.

Volumetric method based on level sets A variational approach implemented by level sets has been proposed in [9]. This approach naturally handles the changes in topology and occlusion problems. But it is not clear under what conditions this method converges as the actual proposed functional seems highly non-convex.

Disparity methods based on graph cuts Originally, Roy and Cox [23] have formulated graph-cut methods as an extension of dynamic programming. An interpretation of the result as a labeling problem in the Markov Random Field framework is provided in [31]. Functionals with various concave smoothing terms have been introduced in [5, 14–17, 31]. The resulting problem is unfortunately NP-hard [5], and only an approximated solution is reached. Ishikawa and Geiger [11, 13] and Buehler et al. [6] restrict to a convex smoothing term which is proven to make a global minimum reachable in polynomial time [11, 12].

Geometric methods based on graph cuts A geometric formulation seems more suitable to handle the surface reconstruction problem. In such a geometric framework, Boykov and Kolmogorov [4] compute geodesic surfaces for data segmentation. Despite different goals, some few points are common with our method (*e.g.* the embedded graph and the discretization of a continuous formulation). One may even try to adapt their algorithm for 3D reconstruction from images but some non-trivial questions arise: location of the *seeds* that drive the topology, visibility, etc.

The method described here proposes its own formulation that is different from [4] while keeping a strong geometric interpretation that addresses all the specific issues of surface reconstruction.

Contributions The main contributions of this paper are summarized as follows:

- A new formulation of the surface reconstruction problem as a geometric optimization problem taking into account potential discontinuities of the object surface.
- A graph-cut technique reaching a global minimum to solve this problem up to an arbitrary discretization.
- The integration with a voxel-based method to characterize object boundaries and account for visibility.
- A formal analysis of the continuity of the result leading to a method avoiding spurious discontinuities.

2 Problem statement

Let $(u,v) \mapsto \mathbf{X}(u,v) \equiv (x(u,v), y(u,v), z(u,v))$ be a parameterized surface. A first approach is to find a minimum of the functional, $\iint c(\mathbf{X})du dv$, where $c(\mathbf{X})$ is a positive function measuring the consistency of $\mathbf{X}$ with different images like pair-wise sums of squared difference or of cross-correlation function. As reconstruction is ill-posed, this simple functional needs to be regularized.

Traditionally, the regularization terms are directly introduced for the parametric surface patch. It is formulated by Terzopoulos et al. [29] as a minimization problem with an additive smoothing term s containing up to the second derivatives: $\iint \left(c(\mathbf{X}) + \alpha\, s\left(\frac{\partial \mathbf{X}}{\partial u}, \frac{\partial \mathbf{X}}{\partial v}, \frac{\partial^2 \mathbf{X}}{\partial u^2}, \frac{\partial^2 \mathbf{X}}{\partial v^2}, \frac{\partial^2 \mathbf{X}}{\partial u \partial v}\right)\right) du dv$. The minimization is solved using a set of partial differential equations (PDE) provided by the Euler-Lagrange equation. Then a virtual time t is introduced with a steepest-descent method to make evolve a surface $\mathbf{X}(t)$ to a steady state.

In the approach developed in [9], the regularization is introduced intrinsically by considering the weighted minimal surface [7] for the functional $\iint c(\mathbf{X})ds = \iint c(\mathbf{X}) \left|\left|\frac{\partial \mathbf{X}}{\partial u} \times \frac{\partial \mathbf{X}}{\partial v}\right|\right|_2 du\, dv$. Since the formulation is intrinsic, *i.e.*, independent of any chosen parameterization, it makes the level-set formulation [20, 26] possible. Recently in [4], $\iint c(\mathbf{X})ds$ is shown to be also minimized by a graph cut when $c(\mathbf{X})ds$ is a Riemannian metrics. Compared to level sets, it reaches a global minimum but is less flexible with topology changes and visibility is accounted for.

The connection between these two different formulations, with a multiplicative regularization term and with an additive one, has been studied by Caselles et al. [7] in 2D: they have be shown to be equivalent. But it seems still to be an open question in 3D.

We propose the following functional

$$\iint \left(c(\mathbf{X}) + \left(\alpha_u(u,v) \left| \frac{\partial \mathbf{X}}{\partial u} \right| + \alpha_v(u,v) \left| \frac{\partial \mathbf{X}}{\partial v} \right| \right) \right) du\, dv.$$

Using only first derivatives for regularization is primarily due to the optimization method that we introduce. We also believe that introducing the second derivatives may lead to over-smoothed surfaces. Therefore, the first derivative smoothing terms are sufficient and even more desirable to capture fine geometric details as demonstrated in our experiments. This formulation is also not intrinsic, and is therefore dependent on parameterization. The $L1$ norm is used to fulfill the smoothing objective and allows an efficient computation. Note that α_u and α_v are not restricted to constants and make discontinuities possible.

3 Global discrete solution

From the previous framework, we set up a graph-cut method that yields a discrete global solution to the object surface. Details about graph cuts are in [1, 10].

Discretization Without loss of generality, let's assume that the 3D object space is coordinated by (x,y,z) and the surface is locally parameterized by a function f: $\mathbf{X}(u,v,f(u,v))$. Let the domain on which f is defined be $\mathcal{D}$. If multiple depth values are needed, multiple functions $f_1, f_2, \ldots$ are used.

Our proposed functional consists of a consistency term $\mathcal{C}$ and a smoothing term $\mathcal{S}$:

$$\mathcal{C}(f) = \iint_{\mathcal{D}} c(x,y,f(x,y))dx\,dy \tag{1}$$

$$\mathcal{S}(f) = \iint_{\mathcal{D}} \left(\alpha_x(x,y) \left| \frac{\partial f}{\partial x}(x,y) \right| + \alpha_y(x,y) \left| \frac{\partial f}{\partial y}(x,y) \right| \right) dx\,dy \tag{2}$$

Our solution strategy relies on an approximation of equations (1) and (2) by a discrete formulation. Let's consider that the surface domain $\mathcal{D}$ is rectangular and that the 3D space is discretized into voxels with size $\Delta x \times \Delta y \times \Delta z$. The extension to a general domain $\mathcal{D}$ with varying discretization steps is however straightforward.

The discrete consistency term $\mathcal{C}^{\mathrm{d}}$ is then:

$$\mathcal{C}^{\mathrm{d}}(f) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} c(x_i, y_j, f(x_i, y_j)) \Delta x \Delta y \tag{3}$$

and the discrete smoothing term $\mathcal{S}^{\mathrm{d}}$:

$$\begin{aligned} \mathcal{S}^{\mathrm{d}}(f) = & \sum_{i=1}^{n_x - 1} \sum_{j=1}^{n_y} \alpha_x(x_i, y_j) \left| f(x_{i+1}, y_j) - f(x_i, y_j) \right| \Delta y \\ & + \sum_{i=1}^{n_x} \sum_{j=1}^{n_y - 1} \alpha_y(x_i, y_j) \left| f(x_i, y_{j+1}) - f(x_i, y_j) \right| \Delta x. \end{aligned} \tag{4}$$

3.1 Building a first embedded graph

Like [4], our approach is based on a topologically embedded graph in the 3D geometric space. The graph is built such that the cut capacity is equal to the surface functional. Thus, a minimal cut is a solution to the discrete problem.

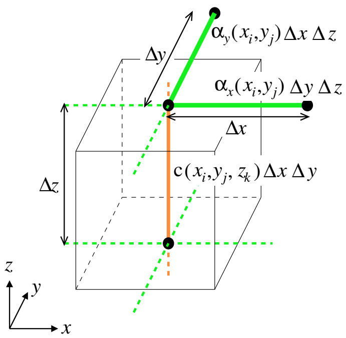


Figure 2: Correspondence between the voxel and the graph. Edge capacities are indicated.

The graph is a 3D grid superimposed on the voxels with correspondence shown in Figure 2.

Correspondence property: There is an one-to-one correspondence between a subset of cuts called the *potential minimal cuts* and the surfaces defined by a function f. Moreover, the cut capacity is equal to functional value of corresponding surface.

The proof is based on necessary criteria for a cut to be minimal and a careful count of cut edges (proof in [22]).

3.2 Improved smoothing with a second graph

Graph-cut techniques often yield flat and blocky results [6, 11, 14, 16]. This may not be important for disparity maps but it is crucial for shape reconstruction.

This artifact appears in regions with depth variation and rather uniform smoothing and consistency terms. In such a region, a continuous depth change and a discontinuous one have the same functional value because the smoothing term depends linearly on the derivatives. With a concave term [3, 16], the discontinuous change may even be lower. That is why spurious discontinuities appear with linear or concave terms and create blocky results (Fig. 7-a). This cannot be smoothed away because it would also remove the real discontinuities. The functional

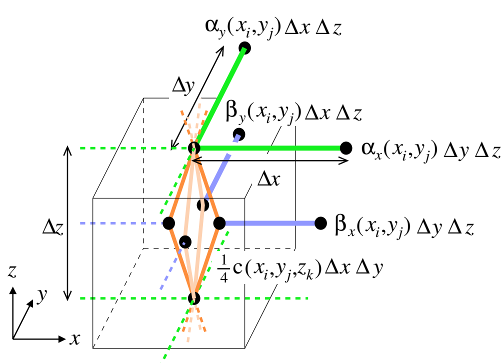


Figure 3: Correspondence between the voxel and the graph with a convex smoothing term. The 8 z-sub-edges have the same capacity.

must therefore differentiate a continuous change from a discontinuous one; this is done through a convex smoothing term. This is summarized in the following property:

Smoothing term property: Concave and linear smoothing terms introduce spurious discontinuities on the surface. To overcome this artifact, the smoothing term must be strictly convex.

For the implementation, we propose a new graph design (Fig. 3) which adds the following strictly convex term:

$$\mathcal{A}^{\mathrm{d}}(f) = \sum_{i=1}^{n_x-1}\sum_{j=1}^{n_y} \beta_x(x_i,y_j)\left[\left|f(x_{i+1},y_j)-f(x_i,y_j)\right|-\Delta z\right]^{+}\Delta y$$
$$+\sum_{i=1}^{n_x}\sum_{j=1}^{n_y-1} \beta_y(x_i,y_j)\left[\left|f(x_i,y_{j+1})-f(x_i,y_j)\right|-\Delta z\right]^{+}\Delta x$$

where $[\lambda]^{+} = \max(0,\lambda)$. Using $\beta \ll \alpha$, the continuous smoothing term (2) is approximated by $\mathcal{S}^{\mathrm{d}} + \mathcal{A}^{\mathrm{d}}$ as close as desired while being strictly convex. See [22] for details.

4 Algorithm description

This graph-cut method is then integrated into a consistent process corresponding to a practical reconstruction scenario. In the chosen configuration, we assume that there exists a separating plane: all the cameras are in the same half space and look toward the other half. This classical setup is similar to many others [4, 6, 15, 16, 25].

Input In addition to the calibrated input images, the user defines a bounding volume.

Initialization A discretization of this volume is then built with a scaling (see Fig 4). This discrete space is seen in turn as a voxel space and a graph. It fulfills the *constant footprint property* [28] (equal projection area) and ensures the *vertex coordinate property* [21]. Actually, this property extends to the case of non-aligned cameras as long as there is a separating plane:

Extended vertex coordinate property: If the scaling center used to build the voxel grid lies on a segment linking any two cameras C_i and C_j, in the epipolar planes of C_i and C_j, the consistent voxels form 2D regions which left-most and right-most points belong to the surface.

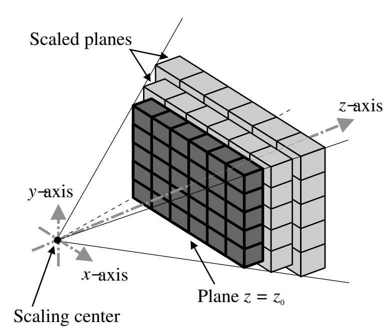


Figure 4: 3D configuration: planes are built with a scaling .

Sketch of proof: Get the result for only C_i and C_j and then show that any additional camera can only remove consistent voxels while keeping the points belonging to the surface.

This property characterizes the optimization domain by gathering the information from all the epipolar planes.

Main loop The algorithm has a multi-pass design to reconstruct objects one by one, from the closest one to the most distant. The separating plane defines unambiguously this "distance". It is a classical *front-to-back* approach [30] or it can also be seen as a *plane sweep* [18] in the z direction.

After each loop, the visibility is updated. The iteration stops when no unreconstructed object remains in the scene.

1: Consistency computation For each voxel, consistency is computed, thresholded and robustified with morphological operators [21]. In the rest of the pass, only the set of consistent voxels that is the closest to the cameras is considered.

2: Discontinuities Under the reasonable assumption that the aspect of a surface depends on its orientation (*e.g.* the light is not purely omnidirectional), surface discontinuities result in image discontinuities. Surface color is approximated using the voxel set and two maps (Fig. 5) of the potential discontinuities are computed by comparing adjacent voxels while accounting for local contrast (formulae in [22]). The maps control the continuity constraint of the functional through α and β and lead to sharper details (Fig. 7-b,e).

3: Graph cut with self-occlusions To account for self-occlusion, the graph-cut technique is adapted according to the geometric configuration. A *visibility term*[1] [16] is added in the functional: $+\infty$ if f corresponds to a self-occluding surface and 0 in all the other cases. The graph is adapted as shown in Fig. 6-a,b: if A and B are in adjacent columns and B occludes A, then an infinite edge is added from B to A. The orientation of the edge ensures that no cut containing B (or a voxel in front of B) and A (or a voxel behind A) is minimal. As the relationship "occluded by" is transitive (Fig. 6-c), these edges between adjacent columns are sufficient to avoid the self-occlusion of the whole surface. Therefore only 4 visibility edges are needed for a voxel.

In practice the results without these edges are never self-occluding, the constraints of the previous functional seems to be enough to achieve satisfying results. It seems these edges can be omitted to alleviate the computation without any loss of quality.

4: Visibility After each pass, the lines of sight blocked by the previously reconstructed objects are computed and ignored in the following passes.

Post-process: Mesh smoothing As the whole process is computed on a discrete grid, it suffers from aliasing artifacts (Fig. 7-c). Inspired by image denoising [2], we propose a PDE filter which is controlled by the principal curvatures of the surface and by the discontinuity maps (Fig. 5) previously computed. This makes a filter that preserves both the curvatures and the discontinuities of the surface. Formulae and implementation details are in [22].

[1]This term is not the same as [16] but has an equivalent effect when considering one disparity map: self-occlusion cannot appear.

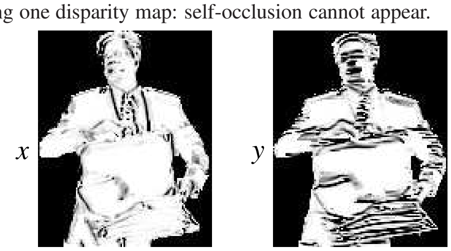


Figure 5: Example of discontinuity maps. Dark regions represent potential surface discontinuities.

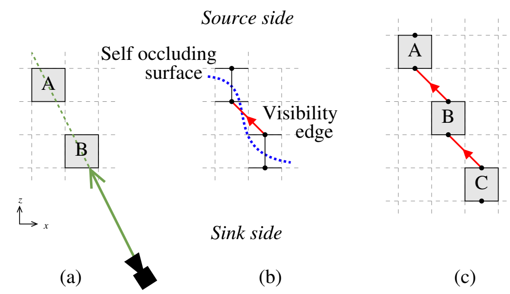


Figure 6: For illustration purpose, only a xz plane is presented and the graph is the non-convex one (Fig. 2). (a) A is occluded by B. (b) The corresponding visibility edge. (c) The BA and CB edges handle the occlusion of A by C.

5 Discussions

Global versus local minimum Our results are exact minima of the discrete functional, they are optimal for the given resolution. This partly explains why the reconstructed surfaces are so detailed compared to the methods that stops at a local minimum [5, 9, 14–16, 31].

Geometry versus labels We believe the geometric formulation gives a more suitable framework for shape reconstruction than the labels. For instance, the link between the functional and the 3D resolution is straightforward and may allow a multi-resolution approach.

Discontinuities In our algorithm, discontinuity location is controlled by images. It seems a more consistent approach than letting the optimization process decides [3, 5, 14–16, 31]. Moreover it allows the use of a convex smoothing term which makes the global minimum reachable without penalizing the discontinuities.

Occlusions Both self-occlusion and object-by-object occlusion are handled. This is comparable to level sets and goes farther than the disparity maps which only detect occlusions but do not reconstruct partially occluded objects. This comes from the mix between voxels and graph cut which is a major contribution of our technique.

Open and closed surfaces The problem that we solve deals with open surfaces. This is a major difference with closed-surface problems [4, 9, 18] because we have to cope with boundaries. It is all the more crucial that we do not compute disparity maps that use the image border as boundary. The beneficial counterpart is that we reconstruct partially occluded objects.

6 Experimental results

We show in Fig. 1 and Fig. 8 three sequences: a keyboard, a briefcase man, and a lantern. There are 40 frames at 692×461 for the man from Dayton Taylor's time-freezing setup with aligned cameras. The sequence is calibrated by a commercial system with about 70 points manually extracted from the sequence. There are 11 frames of resolution 640×480 for the keyboard sequence and 23 frames of 800×600 for the lantern sequence captured by a hand-held digital camera. The geometry of the cameras for these sequences is automatically computed. For consistency, *photo-consistency* [18, 25] is used in HSV.

The space resolution ranges typically from 1 to 10 million voxels with 5 nodes and 12 double edges per voxel. The precision is very high: the face of the man, the keys of the keyboard, the folds of the lantern. With the physical size of the keyboard and the keys, we have measured a 1/10 pixel accuracy. Note that the folded chess board is well reconstructed while being partially occluded by the lantern.

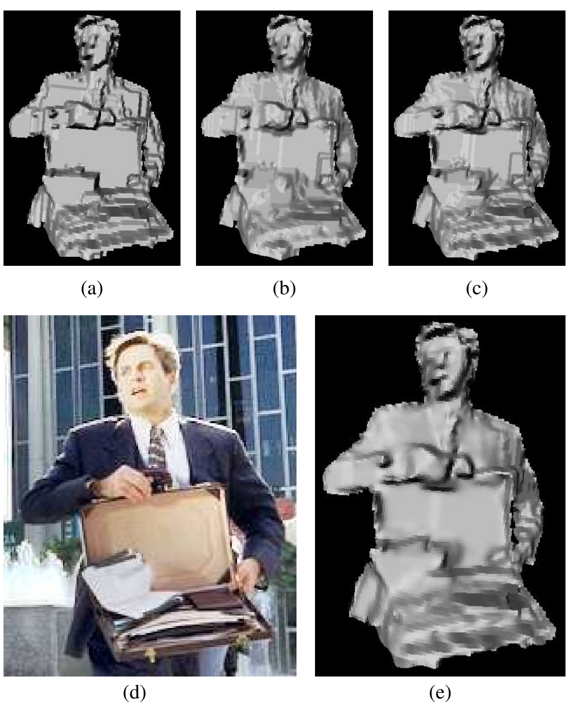


Figure 7: 3D reconstructions with various parameters (a) Without PDE filter and with a linear smoothing term: spurious discontinuities (on the shoulders and on the arms) (b) Without PDE filter and discontinuity maps: details are blurred (on the face and in the briefcase) (c) Without PDE filter: aliasing. (d) An input image. (e) Final result with all our contributions.

Graph flow implementation The graph-cut optimization process is time consuming. These examples took about 15 minutes on an Intel Xeon 2.4 MHz and they need between 300MB and 700MB of RAM. These values have to be evaluated considering the size of the graph ($\approx$ millions of vertices and edges) compared to the graphs used in [16] (at most 600 000 vertices and 4 millions directed edges). The huge graph-flow problem is made tractable through an implementation of [8] with our own improvements [22].

7 Conclusions and future work

We have described a new geometric formulation of the surface recovery problem. It is based on a functional that is simpler than those of the level sets or of some graph-cut approaches. However, we believe that the geometric formulation is more meaningful for 3D reconstruction. Moreover, it explicitly takes into account discontinuities. We have integrated this into a consistent voxel-based process to achieve our goal. It demonstrates that it handles self-occlusions and occlusions to reconstruct partially hidden objects. It achieves precise results even for complex configurations.

We believe that this new approach is promising and creates new solutions for interesting issues. The high robustness of the method coupled with a specific consistency evaluation may lead to results for a more general reflectance model. We also plan to study a generalization of the method to implicit surfaces.

Acknowledgements The visit of Sylvain Paris at HKUST has been supported by the Eurodoc program from Région Rhône-Alpes. We thank Maxime Lhuillier for calibrating the lantern and keyboard sequences, Victor Ostromoukhov for his fruitful discussions about the detection of the color discontinuities, Vladimir Kolmogorov for the statistics about his method and Guy Le Besnerais for his help on references.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

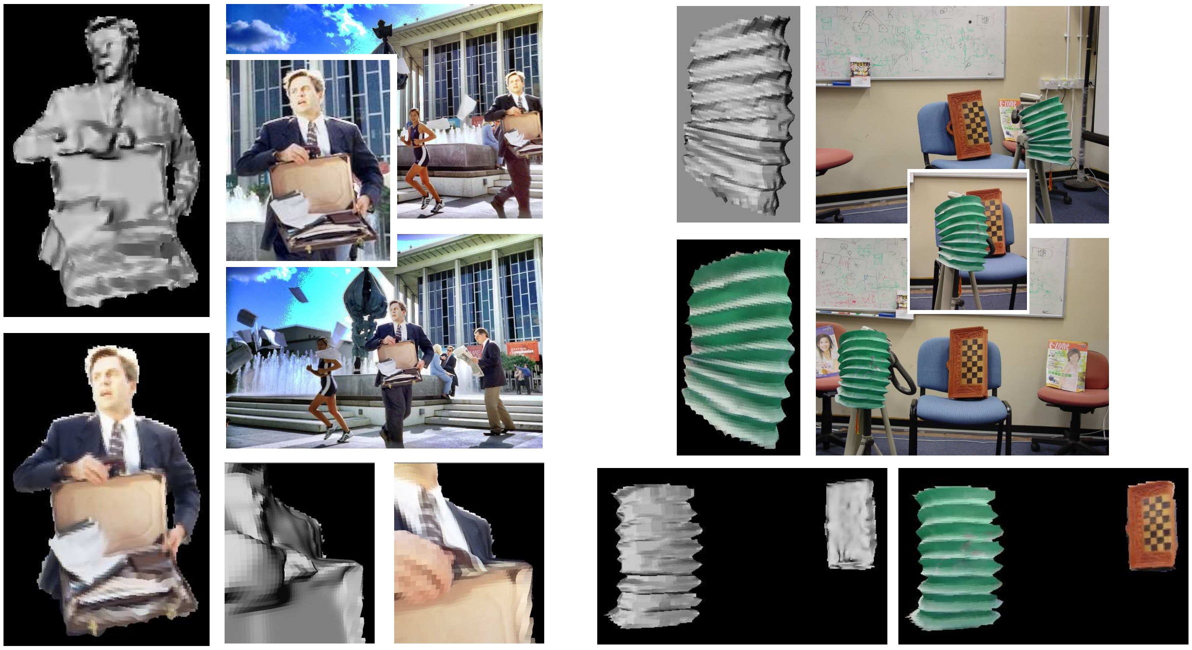
[2] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, volume 147 of *Applied Mathematical Sciences*. Springer-Verlag, 2002.

[3] A. Blake and A. Zisserman. *Visual reconstruction*. Artificial intelligence. MIT Press, Cambridge, 1987.

[4] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *International Conference on Computer Vision*, 2003.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001.

[6] C. Buehler, S. Gortler, M. Cohen, and L. McMillan. Minimal surfaces for stereo. In *European Conference on Computer Vision*, 2002.



(a) (b)

Figure 8: Results with the two extreme views of each sequence and a zoom on the interesting region. (a) Briefcase man (40 images at 692×461, face is about 40×40). Note how discontinuities are preserved. (b) Lantern and folded chess-board (23 images at 800×600). The folded chess-board is occluded from images 4 to 15.

[7] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 1997.

[8] B. V. Cherkassky and A. V. Goldberg. On implementing the push-relabel method for the maximum flow problem. *Algorithmica*, 19(4):390–410, 1997.

[9] O. Faugeras and R. Keriven. Variational principles, surface evolution, PDE's, level set methods and the stereo problem. *Transactions on Image Processing*, 1998.

[10] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[11] H. Ishikawa. *Global Optimization Using Embedded Graphs*. PhD thesis, New York University, 2000.

[12] H. Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[13] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *European Conference on Computer Vision*, 1998.

[14] J. Kim, V. Kolmogorov, and R. Zabih. Visual correspondence using energy minimization and mutual information. In *International Conference on Computer Vision*, 2003.

[15] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *International Conference on Computer Vision*, 2001.

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

[17] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *European Conference on Computer Vision*, 2002.

[18] K. Kutulakos and S. Seitz. A theory of shape by space carving. In *International Conference on Computer Vision*, 1999.

[19] K. N. Kutulakos. Approximate n-view stereo. In *European Conference on Computer Vision*, 2000.

[20] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 1988.

[21] S. Paris and F. Sillion. Robust acquisitions of 3D informations from short image sequences. *Graphical Models*, July 2003.

[22] S. Paris, F. Sillion, and L. Quan. A volumetric reconstruction method from multiple calibrated views using global graph cut optimization. Technical Report 4843, INRIA, June 2003.

[23] S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *International Conference on Computer Vision*, 1998.

[24] H. Saito and T. Kanade. Shape reconstruction in projective grid space from large number of images. In *Computer Vision and Pattern Recognition conference*, 1999.

[25] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 1999.

[26] J. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[27] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer. A survey of methods for volumetric scene reconstruction from photographs. In *VolumeGraphics*, 2001.

[28] G. Slabaugh, T. Malzbender, and W. B. Culbertson. Volumetric warping for voxel coloring on an infinite domain. In *3D Structure from Images - SMILE*, 2000.

[29] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motions. *Artificial Intelligence*, 1988.

[30] M. Ulvklo, H. Knutsson, and G. H. Granlund. Depth segmentation and occluded scene reconstruction using egomotion. *SPIE Visual Information Processing*, 1998.

[31] O. Veksler. *Efficient Graph-Based Energy Minimization Methods in Computer Vision*. PhD thesis, Cornell University, 1999.