INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A Volumetric Reconstruction Method from Multiple Calibrated Views using Global Graph Cut Optimization

Sylvain Paris — François Sillion — Long Quan

N° 4843

Juin 2003

THÈME 3

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--4843--FR+ENG

A Volumetric Reconstruction Method from Multiple Calibrated Views using Global Graph Cut Optimization

Sylvain Paris* , François Sillion† , Long Quan‡

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projets ARTIS§ et HKUST

Rapport de recherche n° 4843 — Juin 2003 — 18 pages

Abstract: In this paper, a new formulation and solution to volumetric reconstruction from multiple calibrated images is presented. This problem has been previously formulated either as a continuous geometric optimization process driven by local numerical methods, or as a discrete labelling problem solved by global techniques for computing only stereo disparities. Our new formulation builds a bridge between these two approaches and takes advantage of both: a continuous geometric functional is minimized up to a discretization by a global graph cut algorithm. The relation between the continuous and discrete formulations is straightforwardly established. The minimization operates on a 3D embedded graph whose minimal cut is a solution of the discrete problem, leading to a global minimum. This new approach handling both occlusions and discontinuities has been demonstrated on real sequences, giving remarkably detailed surface geometry up to $1/10$th pixel.

Key-words: Volumetric reconstruction, global optimization, graph flow, graph cut, discontinuity, occlusion, PDE

* ARTIS
† ARTIS
‡ Hong Kong University of Science and Technology (HKUST)
§ ARTIS est une équipe du laboratoire GRAVIR/IMAG, unité mixte de recherche CNRS, INPG, INRIA, UJF.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier (France)
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Méthode de reconstruction volumique à partir de caméras calibrées utilisant une optimisation globale basée sur une coupure de graphe

Résumé : Dans ce rapport, une nouvelle formulation et une solution au problème de la reconstruction volumétrique à partir de caméras calibrées sont présentées. Ce problème a été présenté jusqu'à présent soit comme un processus d'optimisation d'une géométrie continue à l'aide de techniques numériques locales, soit comme un problème de labélisation discrète résolu par des techniques globales calculant uniquement des disparités stéréo. Notre nouvelle formulation crée un lien entre ces deux approches et tire profit des deux: une fonctionnelle géométrique continue est minimisée à une discrétisation près par un algorithme global de coupure de graphe. La relation entre les formulations discète et continue est directe. La minimisation manipule un graphe qui est topologiquement plongé dans un espace 3D et dont la coupure minimale est solution du problème discret, atteignant un minimum global. Cette nouvelle approche manipule à la fois les occultations et les discontinuités. Elle a été testée sur des séquences réelles : elle donne une géométrie particulièrement détaillée jusqu'au dixième de pixel.

Mots-clés : Reconstruction volumétrique, optimisation globale, flot de graphe, coupure de graphe, discontinuité, occultation, EDP

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* [KS99] or *level sets* [FK98] and a discrete labelling formulation for computing stereo disparities solved by global *graph-cut* methods [RC98, Ish00, KZ02a, BGCM02]. 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 [SD99, KS99, SK99, Kut00, SMC00] directly works on discretized 3D space, voxels, based on their image consistency and visibility. All these techniques are characterized by their very local treatment of the surface: voxels are considerated separately. The different methods differ in the way they update the visibility. For efficiency purposes, some methods put constraints on the camera locations to simplify the visibility computation. A more detailed review can be found in [SCMS01]. Our method is different in that it is not purely local. Global constraints are added to handle the difficult situations of low textureness and small baseline sequences.

Volumetric method based on level sets A variational approach implemented by level sets has been proposed in [FK98]. This approach naturally handles the changes in topology and occlusion problems. It has great potential, and some results have been presented. However it is not clear under what conditions this method converges as the actual proposed functional seems highly non-convex. Our method is different from this approach both in formulation and solution methods. We propose a highly modular technique that clearly separates the main problems (image consistency, topology, occlusion, surface localization) whereas the level-set method defines an all-in-one framework. Moreover, as we restrict our functional to a simpler expression, we demonstrate that a global minimum can be reached.

Volumetric methods based on graph cut The link between discrete surface optimization and graph cut is first outlined in [RC98]. The meaning of the resulting surface is not very clear but they have emphasized the robustness of the technique through its smoothing effect. An interpretation of the result as a labelling problem in the Markov Random Field framework is provided in [Vek99] and [Ish00]. More recently, functionals with concave smoothing terms have been introduced in [Vek99, BVZ01, KZ01, KZ02b, KZ02a]. This concave smoothing term may take a wide variety of expressions. The resulting problem is unfortunately NP-hard [BVZ01], and only an approximation solution can therefore be reached. Methods for two or three cameras that restrict this term to convex functions are also proposed in [Ish00] and [BGCM02] to search for a global minimum. However, these methods cannot easily be generalized to multiple cameras. Moreover, all these methods lack precision in general as the results are disparity maps representing objects by flat regions.

Our approach is different in that it formulates the graph cut optimization as the discretization of a continuous geometric problem.

Our formulation and solution takes advantage of these approaches to look for a global discrete optimization method driven by a continuous geometric formulation. Its main contributions can be summarized as follows:

- A new formulation of the reconstruction problem as a geometric optimization process.
- The localization of the potential discontinuities of the object surface.
- The exact solution of a discretized version of this problem that takes the discontinuities into account.

Our method reconstructs accurate 3D geometry from real input images even with very small baselines.

2 Problem statement and formulation

Our goal is to create an object surface in 3D space. We first consider a general formulation that outlines the relations between our work and existing methods.

Let $(u,v) \mapsto \mathbf{X}(u,v) \equiv (x(u,v), y(u,v), z(u,v))$ be a regular parameterized surface representing the underlying object. Three-dimensional reconstruction can be cast as a variational problem of finding a suitable function or surface $\mathbf{X}$ that minimizes the functional:

$$\iint c(\mathbf{X}) du dv,$$

where $c(\mathbf{X})$ is a positive function measuring the consistency of pixels in different images, most of the time under the Lambertian assumption. Examples of such measurements are a function of pair-wise sums of squared difference (SSD, Zero-Mean SSD, or photo-consistency in color space) or of pair-wise cross-correlation function (CC or Zero-Mean Normalized CC). As reconstruction is ill-posed, this simple functional needs to be regularized to give smooth solutions.

Traditionally, the regularization terms are directly introduced for the parametric surface patch. Then, the regularized problem is formulated as a deformable surface models minimizing an energy [TWK88]. One possibility might be to consider the functional:

$$\iint (c(\mathbf{X}) + \alpha s(\mathbf{X}_u, \mathbf{X}_v, \mathbf{X}_{uu}, \mathbf{X}_{vv}, \mathbf{X}_{uv})) du dv.$$

The second smoothing terms $s(\cdot)$ are

$$a|\mathbf{X}_u|^2 + b|\mathbf{X}_v|^2 + c|\mathbf{X}_{uu}|^2 + d|\mathbf{X}_{uv}|^2 + e|\mathbf{X}_{vv}|^2.$$

The minimization is solved by local methods, a set of PDEs provided by the Euler-Lagrange equation. The problem with this approach is well-known [Set99] in that it does not handle the changes in the topology.

In the approach developed in [FK98], the regularization is introduced intrinsically by considering the weighted minimal surface [CRS97]. The weighted minimal surface is defined to be a minimizer of the functional

$$p(\mathbf{X}) = \iint c(\mathbf{X})ds = \iint c(\mathbf{X})||\mathbf{X}_u \times \mathbf{X}_v||_2 dudv.$$

Again the minimization is solved by an iterative steepest-descent method. However, further development shows that the formulation is intrinsic, *i.e.*, independent of any chosen parameterization. It makes the use of level-set formulation possible. The level-set method [OS88, Set99] regards the surface as the zero-level set of a higher dimensional function. The flow velocity is intrinsic, dependent only on the surface curvature. Topological changes, accuracy, and stability of the evolution are handled by using the proper numerical schemes developed by Osher and Sethian [OS88].

The connection between these two different formulations, one with a multiplicative regularization term and another with an additive one, has been studied by many researchers [CRS97]. In the case of 2D curves, these two formulations correspond respectively to geodesic active contours and classical snakes. These two formulations are equivalent [CRS97]. It seems that it is still an open question on the case of 3D surfaces.

We propose the following functional

$$\iint (c(\mathbf{X}) + (\alpha_u(u,v)|\mathbf{X}_u| + \alpha_v(u,v)|\mathbf{X}_v|))dudv.$$

Only first derivative terms are used for smoothing. Dropping the second derivative smoothing terms is primarily due to the optimization method we will introduce in the next section to solve the minimization problem. On the other hand, we believe that introducing the second derivative terms may lead to a complicated solution (for instance, the Euler-Lagrange solution of deformable models may contain fourth derivative terms) and 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 leads to an efficient computation. However, our work can be extended to the $L2$ norm using the work of Ishikawa [Ish00] with higher computation time. This formulation is also not intrinsic, and is therefore dependent on parameterization. And note that α_u and α_v are not restricted to constants and thus make discontinuities possible because they give local control of the smoothing.

Another major departure from these approaches [TWK88, FK98] is that our optimization method is not a local method based on a continuous formulation, but a global method based on a discretization of this functional. This is inspired by recent work that uses graph-cut algorithms for stereo-matching and reconstruction [RC98, Vek99, Ish00, BVZ01, KZ01, KZ02a, BGCM02]. However our discrete solution is different from those proposed in [Vek99, BVZ01, KZ01, KZ02a], and closer to that of [Ish00, BGCM02] in that we look for a global solution up to the discretization error rather than for an approximate solution. In addition, the relationship between continuous and discrete formulations is clearly identified.

3 Global discrete solution

From the previously studied general framework, we set up a method that yields a discrete global solution to the object surface.

We propose a graph cut technique to determine a solution to the discrete problem. The graph cut technique solves the following general problem: given a water source linked to a sink through a pipe network, what is the maximum water flow that can reach the sink or, equivalently, where is the network bottleneck that limits the flow? Formally speaking, we have a oriented graph (the pipe network) with a *source* node and a *sink* node and each graph edge has two associated values: the *capacity* (the maximum flow through a pipe) and the *flow* (the actual flow in a pipe). A *cut* is defined as set of edges which separate the nodes into two connected components: one including the source and the other including the sink. Polynomial algorithms exist to find the *minimal cut* (the bottleneck), which is the cut with the lowest capacity from the source component to the sink component. The references [FF62] give more details.

Discretization Without loss of generality, let's assume that the 3D object space is coordinated by (x,y,z) and the object surface is locally parametrized by a function, f: $\mathbf{X}(u,v,f(u,v))$, which can be seen as a depth function $z = f(x,y)$. Also let the domain on which f is defined be $\mathcal{D}$. This parameterization restricts the object being constructed to a single-valued depth. If multiple depth values are needed, multiple functions $f_1, f_2, \ldots$ could be used. As x and y often play equivalent roles, the $x|y$ notation is used when a statement is applied to both x and y.

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

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

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

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 discretized as a regular rectangular grid. Then, x, y, z have values $\{x_1,\ldots,x_{n_x}\}$, $\{y_1,\ldots,y_{n_y}\}$, $\{z_1,\ldots,z_{n_z}\}$ separated by Δx, Δy, Δz. The extension to a general domain $\mathcal{D}$ with varying discretization steps is however straightforward.

A discrete consistency term $\mathcal{C}^{\mathrm{d}}$ is obtained as:

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

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

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

3.1 Building a first embedded graph

Our approach is based on a topologically embedded graph in the 3D geometric space; that is to say, it can be seen as a 3D geometric entity. Nodes and edges correspond to 3D points and 3D segments. Therefore, a cut is a real surface that crosses these segments. Moreover, edge capacities are fully expressed with geometric measurements. The main idea is to build the graph such that the cut capacity is equal to the surface functional. Then, a minimal cut will be a solution to the discrete problem. The graph is a 3D grid superimposed on the voxels with correspondence shown in Figure 1. All edges are bidirectional (*i.e.* made of two directional edges): for x-edges, the capacity is $\alpha_x(x_i, y_j)\Delta y \Delta z$; for y-edges, $\alpha_y(x_i, y_j)\Delta x \Delta z$; and for z-edges, $c(x_i, y_j, z_k)\Delta x \Delta y$. We add source and sink nodes out of this grid and for each (x_i, y_j) voxel column, the voxel with the minimum depth is linked to the source and the one with the maximum depth is linked to the sink. All source and sink edges have an infinite capacity. From a complexity point of view, there are three edges and one 6-connected node per regular voxel (*i.e.* not on a border). Thus, the graph complexity is proportional to the number of voxels.

Note that Roy and Cox [RC98] and Ishikawa [Ish00] have also described embedded graphs. Roy and Cox associate each graph node to a disparity value although edges are the key entities in the graph cut methods. This is one of the main reasons why the results of their method are not well defined. Ishikawa embeds the graph in a generic Euclidean space but the links with the geometric space are unclear. In contrast to these approaches, our embedding is stronger: our graph is a 3D entity.

3.2 Establishing a correspondence between a minimal cut and an object surface

There are three necessary conditions for a cut to be minimal:

1. A minimal cut cannot cross an infinite edge;

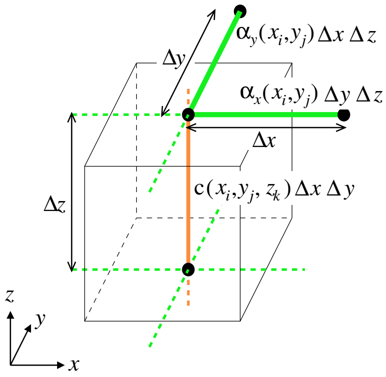

Figure 1: Correspondence between the voxel (x_i, y_j, z_k) and the graph.

2. It has to cross each (x_i, y_j) voxel column at least once to separate the source from the sink;

3. It cannot cross a column more than once for the capacity would be higher than if it crossed only once.

A cut satisfying these three conditions is called a *potential minimal cut*. A direct one-to-one correspondence between such a cut and a surface exists: the cut is limited to the consistent voxel space (condition 1) and the single crossing point on the (x_i, y_j) voxel column (conditions 2 and 3) unambiguously determines $f(x_i, y_j)$.

The capacity of a potential minimal cut represented by f_{cut} can be computed as follows. The capacity of the crossed z-edges is exactly the consistency term $\mathcal{C}^{\text{d}}(f_{\text{cut}})$. Then, if we consider the two adjacent columns (x_i, y_j) and (x_{i+1}, y_j), the cut depths are $f_{\text{cut}}(x_i, y_j)$ and $f_{\text{cut}}(x_{i+1}, y_j)$. This implies that it crosses $\frac{1}{\Delta z}\left|f_{\text{cut}}(x_{i+1}, y_j) - f_{cut}(x_i, y_j)\right|$ x-edges. Thus, the total capacity of crossed x-edges is $\alpha_x(x_i, y_j)\left|f(x_{i+1}, y_j) - f(x_i, y_j)\right|\Delta y$. There is a similar result for the y-edges. The total capacity of all the x-edges and y-edges crossed by the whole cut is exactly $\mathcal{S}^{\text{d}}(f_{\text{cut}})$.

By adding these results together, we draw the expected conclusion: there is an exact correspondence between a surface with the discrete functional $\mathcal{C}^{\text{d}} + \mathcal{S}^{\text{d}}$ and a potential minimal cut with its capacity. Therefore, any graph flow algorithm is able to reach a global minimum of this functional.

3.3 Improving the smoothing with a second graph

Graph cut techniques [Ish00, KZ02a, BGCM02] often yield flat and blocky results. This may not be fatal if we are only targeting building a disparity map with limited precision (*e.g.*, 16 or 32 disparity values), but it becomes serious if we target 3D shape reconstruction including small details and smooth slopes. We first elucidate the origin of this artifact, and then we propose a solution by introducing a new smoothing term.

As the smoothing term in (2) and (4) depends linearly on the derivatives, regions with uniform values suffer from strong ambiguities: low color variations in the images result in 3D regions with low variations both in the consistency term and the smoothing term. Under these conditions, a step variation and a smooth one have the same functional value as illustrated in Figure 2. Minimal cut algorithms will arbitrarily select a cut either with minimal or maximal depth, *i.e.*, a very discontinuous cut. As there is no objective difference between these cuts, the ambiguity cannot be solved. This

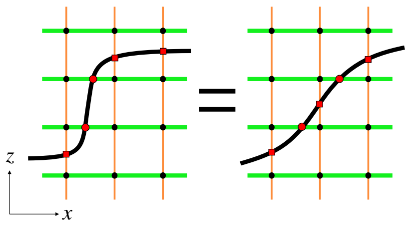

Figure 2: A step variation and a smooth one are equivalent in the uniform region: they cross the same number of x-edges (2 circles) and z-edges (3 squares). For clarity purpose, we present a 2D xz plane of the 3D grid.

is the problem inherent in most of the graph cut based methods, leading to flat and blocky aspects of their results.

Our solution is to add a convex smoothing term. We could have chosen the graph proposed in [Ish00] but it introduces some constant in the smoothing term that can be difficult to handle in a multi-resolution context. We therefore propose a new graph based on the correspondence shown in Figure 3. We conserve the global 3D grid layout of the graph but there are now two kinds of $x|y$-edges: the $\alpha_{x|y}$-edges and the $\beta_{x|y}$-edges and the z-edge is split into 8 sub-edges. Nonetheless, the graph complexity is still proportional to the number of voxels since there are twelve edges, four 3-connected nodes and one 12-connected node per voxel.

In this graph, the z-sub-edges are always cut by a group of four and therefore their total capacity is always equal to the consistency term $\mathcal{C}^{\mathrm{d}}(f_{\mathrm{cut}})$. Then, the $x|y$-axis, as the $\alpha_{x|y}$-edges correspond to the previous $x|y$-edges, always form $\mathcal{S}^{\mathrm{d}}(f_{\mathrm{cut}})$. Thus the only difference is introduced by the $\beta_{x|y}$-edges. Let's consider again two adjacent columns (x_i, y_j) and (x_{i+1}, y_j): there are $\frac{1}{\Delta z}\left|f_{\mathrm{cut}}(x_{i+1}, y_j) - f_{\mathrm{cut}}(x_i, y_j)\right|$ α_x-edges as previously shown and if this number is non-zero, because the cut can "go in the middle" of the z-sub-edges as seen in Figure 4, there is one less β_x-edge crossed. If we call $[\lambda]^+$ the value of $\max(0, \lambda)$, the capacity of the crossed β_x-edges between the columns is $\beta_x(x_i, y_j)\left[\left|f(x_{i+1}, y_j) - f(x_i, y_j)\right| - \Delta z\right]^+ \Delta y$, which gives a strictly convex function. Hence, $\beta_{x|y}$-edges add a convex term $\mathcal{A}_{\mathrm{d}}$ in the functional:

$$\mathcal{A}^{\mathrm{d}}(f) = \sum_{i=1}^{n_x-1}\sum_{j=1}^{n_y} \beta_x(x_i, y_j)\left[\left|f(x_{i+1}, y_j) - f(x_i, y_j)\right| - \Delta z\right]^+ \Delta y$$
$$+ \sum_{i=1}^{n_x}\sum_{j=1}^{n_y-1} \beta_y(x_i, y_j)\left[\left|f(x_i, y_{j+1}) - f(x_i, y_j)\right| - \Delta z\right]^+ \Delta x \tag{5}$$

As expected, with this convex term, the functional is lower for a smooth variation than for a step one (Fig. 4). This achieves our goal: $\mathcal{S}^{\mathrm{d}} + \mathcal{A}^{\mathrm{d}}$ is a smoothing term that "really smoothes" the surface. It is important to notice that the discrete $\mathcal{A}^{\mathrm{d}}$ term (5) has no continuous counterpart because of the Δz term, which is directly linked to the discretization step. However, $\beta_{x|y}$ can be made very small

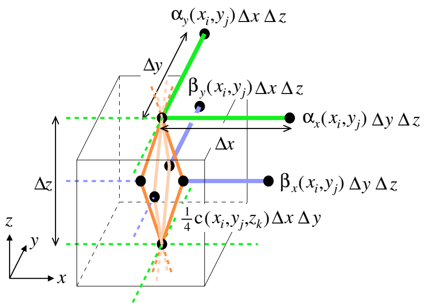

Figure 3: Correspondence between the voxel (x_i, y_j, z_k) and the graph with a convex smoothing term. The 8 z-sub-edges have the same capacity.

because we only need to penalize the step variations, no matter how big this penalty is. Therefore, $\mathcal{A}^{\mathrm{d}} \approx 0$. Then, we can compute a very close approximation of the discrete solution to the continuous functional while assuring smooth variations in the ambiguous regions.

The last point to be observed is that crossing two z-sub-edges may lead to a lower capacity than crossing one β-edge. To avoid this situation, we can either add a constant on the z-sub-edge capacity (but it can be incompatible with a multi-resolution approach) or use *constraint edges* [Ish00] (*i.e.* keep the same capacity for the oriented z-edges from the source to the sink and put an infinite capacity to their reverse oriented edges) and then have no additional constant.

3.4 Handling discontinuities

In our formulation (eq. 2), we have two functions (α_x and α_y) that control the smoothing term. More precisely, these functions control the discontinuities and can therefore be defined as $\alpha_{x|y}(x,y) = A\chi_{x|y}$ where A is a constant which corresponds to the desired smoothing strength when there is no discontinuity and the functions $\chi_{x|y}$ is a *discontinuity factor* varying between 0 (discontinuity) and 1 (no discontinuity). $\beta_{x|y}$ is defined similarly using $B\chi_{x|y}$ with constant B such that $B \ll A$.

Now we need to localize the discontinuities. The key idea is that depth discontinuities on the object can be detected in the images by assuming that any surface discontinuity results in color changes in the images. Of course, one can imagine some theoretical lighting conditions that make the previous assumption false. But such conditions are very hardly realized even in a studio so we can consider this assumption acceptable for all reasonable cases.

Since the object surface is unknown, we have to estimate the surface color according to the input images. This point depends on the implementation and a example will be described later. Let's assume that we have an estimate of the surface color for the moment. Then, the discontinuity factors can be constructed as follows.

Two *discontinuity maps* $\mathcal{D}_x$ and $\mathcal{D}_y$ are computed. $\mathcal{D}_x$ is related to the color distance between (x_i, y_j) and (x_{i+1}, y_j). As the estimation of the color discontinuities has to be independent of the local contrast (for instance, the estimation should not change between bright or dark lighting conditions), it is normalized by the standard deviation $\gamma_{\mathcal{N}_x^{ij}}$ of the color in a small neighborhood $\mathcal{N}_x^{ij}$. Small values of $\gamma_{\mathcal{N}_x^{ij}}$ cause numerical instabilities and make detect spurious discontinuities. We therefore

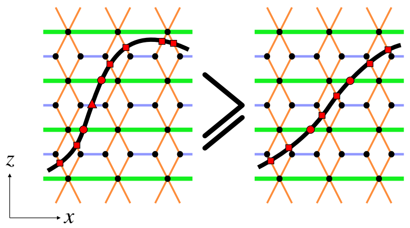

Figure 4: The step variation crosses one more β_x-edge (the triangle) than the smooth variation. Thus, it has a higher energy.

apply a threshold Γ according to the values of $\gamma_{\mathcal{N}_x^{ij}}$:

$$\mathcal{D}_x(x_i, y_j) = \begin{cases} 0 & \text{if } \gamma_{\mathcal{N}_x^{ij}} < \Gamma \\ \frac{\text{distance}(color(x_i,y_j),\text{color}(x_{i+1},y_j))}{\gamma_{\mathcal{N}_x^{ij}}} & \text{else} \end{cases} \tag{6}$$

We use the equivalent formula for $\mathcal{D}_y(x_i, y_j)$ to get:

$$\chi_{x|y}(x_i, y_j) = \left[1 - \mathcal{D}^2_{x|y}(x_i, y_j)\right]^+ \tag{7}$$

Remark Our approach is different from other methods which either consider discontinuities as normal depth variations [FK98, BGCM02]. This is obviously a drawback for important depth changes. Or they handle discontinuities in their optimization process but without controlling their localization [BZ87, BVZ01, Vek99, KZ01, KZ02a]. Our method adds this important issue of discontinuity localization according to the input images.

4 Implementation

To validate the approach we proposed here, we choose a practical scenario of a short image sequence of a still scene to demonstrate the potential of the method in extreme configurations. This corresponds, for instance, to a camera rail sequence that shoots simultaneous photographs or a short video extract. This kind of geometric configuration is characterized by its limited total view angle (a few degrees), which is known to introduce strong depth uncertainties. It makes the reconstruction by carving methods difficult due to the strong depth uncertainty and by level-set methods due to the open surfaces. However, it does ease the surface parameterization and visibility computation.

4.1 Process overview

The occlusions are handled by a multi-pass algorithm as presented in Section 3 according to a *front-to-back* design [UKG98]. Each pass reconstructs the nearest object to the cameras. Self-occluding objects are simply split into two objects that are reconstructed in different passes.

Each pass consists of the following steps.

1. The approximate localization of the object.
2. The estimation of the potential discontinuities.
3. The precise surface geometry computation using our global graph cut optimization.
4. The filtering of the final geometric surface.

After each pass, the visibility is updated by computing the region occluded by the previously reconstructed objects.

4.2 Geometric configuration

Our study case assumes that the cameras are all on one side of the scene, not surrounding the object. The process reconstructs only one side of the objects facing the cameras. The representation is not a closed surface, which implies that the surface domain $\mathcal{D}$ is not trivial. This geometric configuration as illustrated in Figure 5 introduces the following useful property that characterizes $\mathcal{D}$: the surface generating a set of consistent voxels spreads from the left-most boundary of the set to the right-most one. This can be first proved (the proof is omitted due to space limitation) for perfectly aligned cameras with variable viewing directions. But, in practice, this property still holds for a very rough linear trajectory of the camera.

Voxel geometry We need also to interactively delimitate a rough working volume containing the objects of interest and the potential occluders. Then, the volume is discretized into voxels and the c consistency function is evaluated for each of these voxels.

It seems reasonable to consider that a voxel represents a constant amount of information. It implies that the projection of a voxel onto an image space always has the same area. This is the *constant footprint* property defined in [SMC00]. There are different ways to achieve this property [SG98, SK99]. We have chosen a scaling based construction that is the second necessary condition to ensure the property.

4.3 Approximate localization

Since we are only interested in low values of c, the voxels with higher values than a given threshold are discarded. A 3D morphological filtering operator (closure and opening) is applied to the thresholded voxels as suggested in [MBWB02] to fill in the small holes and to remove the isolated voxels. The set of remaining voxels is comparable to that obtained from a *space carving* method in [KS99] in that we use only a threshold for consistency without any other considerations. Therefore, we make the same conclusion as Kutulakos and Seitz in [KS99] that this set is the largest set of voxels consistent with the input images. It guarantees that the object integrally lies inside these voxels and gives a first approximate localization of the object surface.

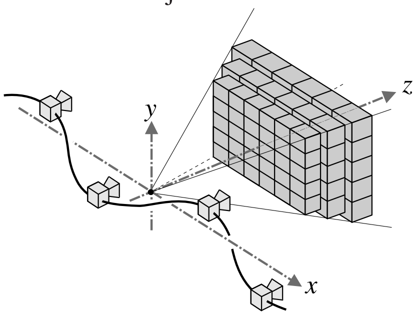

Figure 5: Geometric configuration of our study case: cameras roughly lie on the x-axis and the voxel space is in the z direction.

4.4 Discontinuities

To apply the method of handling discontinuities described in Section 3.4, we need an estimate of the color for every surface point. For each point (x_i, y_j), we average the colors of all the voxels in the corresponding column. This leads to an accurate estimation because all the voxels have almost the same color because they result from a uniform color as shown in [SD99, KS99].

Then, $\alpha_{x|y}$ and $\beta_{x|y}$ can be computed as previously described over a 6×5 neighborhood centered at $(x_{i+\frac{1}{2}}, y_j)$ for $\mathcal{N}_x^{ij}$. Figure 6 shows an example of χ_x and χ_y.

4.5 Surface localization

The current voxel set computed in 4.3 contains the object surface. It determines the domain $\mathcal{D}$ which supports the surface spreading from the left-most boundary to the right-most one.

Now, the main process described in Section 3 is used. We build a graph superimposed on the voxel set and use a graph flow algorithm to compute a minimal cut. This minimal cut is the surface we seek. The complexity of the graph flow algorithms [CG97] we implemented is $O(n^{2.5})$ for n voxels. The huge graph flow problem has been made tractable through a careful implementation of the algorithm and heuristics presented by Cherkassky and Goldberg [CG97]. The main aspect of this implementation is that it computes adjacency information instead of storing it and thus saves a lot of memory. Note that the other classical implementation proposed by Boykov and Kolomogorov [BK01] is not suitable for our graphs because it is specifically designed for small graphs and its higher complexity is a real drawback in our case.

4.6 PDE smoothing

As the whole process up to this step is purely discrete, it suffers from aliasing artifacts. A smoothing filter, inspired by PDE based image denoising [AK02] is well adapted to a geometric surface because it is driven by the principal curvatures κ_1 and κ_2. Also, because the potential discontinuity lines are already located, the PDE smoothing filter can be adapted to avoid crossing these lines.

First, since $\chi_{x|y}$ evaluates the discontinuity between two adjacent columns, we define $\hat{\chi}_{x|y}$ a column-centered function $\hat{\chi}_x(x_i, y_j) = \frac{1}{2}\left(\chi_x(x_i, y_j) + \chi_x(x_{i-1}, y_j)\right)$ and equivalently for $\hat{\chi}_y$. To adapt to the directions θ_1 and θ_2 of the principal curvatures, we define a discontinuity factor $\hat{\chi}_\theta$ for the direction θ with the classical interpolation: $\hat{\chi}_\theta = \cos^2(\theta)\,\hat{\chi}_x + \sin^2(\theta)\,\hat{\chi}_y$

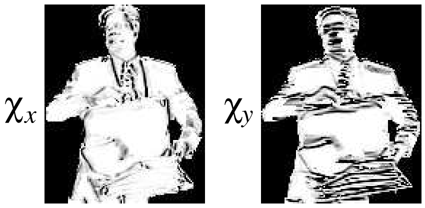

Figure 6: The χ_x and χ_y functions (black = 0 and white = 1) for the briefcase man (fig. 7-a).

Thus, we can formulate the filter as a surface evolution driven by the following PDE with virtual time t:

$$\frac{\partial f}{\partial t} = \hat{\chi}_{\tilde{\theta}_1} g(\tilde{\kappa}_1)\kappa_1 + \hat{\chi}_{\tilde{\theta}_2} g(\tilde{\kappa}_2)\kappa_2 \tag{8}$$

where g is a *stopping function* that controls diffusion to preserve the curvatures (its role is discussed in detail in [BSMH98]), $\tilde{\kappa}$ and $\tilde{\theta}$ are computed on a Gaussian filtered version of the surface, which leads to more robust estimations to control the filter [CLMC92]). Note that there are two controlling components: g driven by the surface geometry and $\hat{\chi}$ accounting for color discontinuity. These assure that both curvature and discontinuities are preserved.

5 Discussions

A discussion of comparisons of our method with existing methods is given in this section.

Global versus local minimum The main difference between our approach and other existing methods [FK98, Vek99, BVZ01, KZ01, KZ02a] is that we find an exact global minimum of the discrete functional. Other approaches use variational techniques: for level set methods, the variation is given by the Euler-Lagrange relation and for graph cut methods it comes from an α-*expansion* [BVZ01]. These methods suffer from the same limitations:

- They are dependent on the starting point.
- They only reach a local minimum. At best, Boykov et al. [BVZ01] have demonstrated a highest bound of the error committed by their method on the functional. Currently, the geometric error is unknown.

We can determine this global minimum because we use a convex smoothing term coupled to an *a priori* discontinuity detection which results in a polynomial problem opposite to [Vek99, BVZ01, KZ01, KZ02a] which rely on a concave term to handle discontinuities. Moreover our discontinuity handling seems more consistent as it is driven by input images as discussed in Section 3.4.

Geometry versus labels Another important feature of our approach compared to the existing graph cut techniques [Vek99, Ish00, BVZ01, KZ01, KZ02a] is its geometric formulation. Up to now, graph cut methods have always described the problem in terms of image disparity: different disparity values are selected and each image pixel has to be labeled with one of those values. Even if there is a link between the disparity and the depth, none of these methods has used it to characterize the functional. Therefore, the relation between the discretized resolution and the different terms of the functional is unclear, whereas our approach makes explicit the measure functions dx, dy and dz whose relation with Δx, Δy and Δz is straightforward.

From this point of view, our method is quite similar to the level-set approach that first defines a continuous geometric functional before looking for a solution with a specific method. But our solution technique is purely discrete whereas the level-set is a local numerical method.

Occlusions Our algorithm through the multi-pass design is able to handle strong occlusions for reconstructing partially hidden objects. From this point of view, these results are equivalent to those obtained with space-carving [KS99] or level-set methods [FK98]. They are more general than those obtained with disparity maps [Ish00, KZ02a, BGCM02] which detect occlusions but do not reconstruct hidden objects. Contrary to these methods, ours handles visibility to reconstruct objects which are only visible by a subset of the cameras.

6 Experimental results

The implemented system has been demonstrated on many real examples. We show three typical sequences: a briefcase man, keyboard, and lantern sequences illustrated in Figure 7. There are 40 frames for the street from Dayton Taylor's time-freezing setup with a perfectly aligned and regularly spaced 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 has been automatically computed. To compute effective solution for results, there is a broad range for the definition of consistency. For simplicity, we used the *photo-consistency* [SD99, KS99] in the *hue-saturation-value* color space for our current examples.

The space resolution ranges typically from 1 to 10 million voxels with 5 nodes and 12 edges per voxel (fig. 3). The precision of the reconstruction results is very high. We notice even the geometric details on the face of the man (Fig. 7-a), which comes from only a small patch of about $\approx 40 \times 40$ pixels. Almost every key on the keyboard is reconstructed and distinguishable (Fig. 7-b). We measured the physical size of the keyboard and the keys. This gives a 1/10 pixel accuracy.

The graph-cut optimization process is time consuming. These examples took between 30 minutes and 2 hours to compute on a 400MHz MIPS R12000 and they need between 300MB and 700MB of RAM.

Figure 7-c shows a case with strong occlusion. The folded chess-board is hidden by the lantern in half of the images. Nevertheless, our algorithm is able to exploit the remaining views to rebuild it.

7 Conclusions

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 set method or of some graph cut approaches. However, we believe that the geometric formulation is more meaningful than the labelling interpretation commonly proposed by the graph cut methods. Moreover, it explicitly takes into account discontinuities compared to the other methods which either do not handle discontinuity or consider them but have no control over their localization. We have built a system integrating our methods. It demonstrates that it can handle occlusions to reconstruct partially hidden objects. It also has the capacity to achieve very precise results even for complex configurations.

(a) (b) (c)

Figure 7: 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) Keyboard (11 images at 640×480). (c) Lantern and folded chess-board (23 images at 800×600). The folded chess-board is occluded from images 4 to 15.

References

[AK02] 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.

[BGCM02] C. Buehler, S. Gortler, M. Cohen, and L. McMillan. Minimal surfaces for stereo. In *European Conference on Computer Vision (ECCV 02)*, 2002.

[BK01] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. In *Third International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2001.

[BSMH98] M. J. Black, G. Sapiro, D. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 1998.

[BVZ01] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2001.

[BZ87] A. Blake and A. Zisserman. *Visual reconstruction*. Artificial intelligence. MIT Press, Cambridge, 1987.

[CG97] B. V. Cherkassky and A. V. Goldberg. On implementing the push-relabel method for the maximum flow problem. *Algorithmica*, 19(4):390–410, 1997.

[CLMC92] F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis*, 1992.

[CRS97] V. Caselles, R., and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 1997.

[FF62] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[FK98] O. Faugeras and R. Keriven. Variational principles, surface evolution, PDE's, level set methods and the stereo problem. *Transactions on Image Processing*, 1998.

[Ish00] H. Ishikawa. *Global Optimization Using Embedded Graphs*. PhD thesis, New York University, 2000.

[KS99] K. Kutulakos and S. Seitz. A theory of shape by space carving. In *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV-99)*, pages 307–314, 1999.

[Kut00] Kiriakos N. Kutulakos. Approximate n-view stereo. In *European Conference on Computer Vision (ECCV 00)*, 2000.

[KZ01] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *International Conference on Computer Vision*, 2001.

[KZ02a] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *European Conference on Computer Vision*, 2002.

[KZ02b] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *European Conference on Computer Vision*, 2002.

[MBWB02] K. Museth, D. Breen, R. Whitaker, and A. Barr. Level set surface editing operators. *ACM Transactions on Graphics (Siggraph 02)*, 21(3):330–338, July 2002.

[OS88] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 1988.

[RC98] S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *IEEE International Conference on Computer Vision*, 1998.

[SCMS01] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer. A survey of methods for volumetric scene reconstruction from photographs. In *VolumeGraphics 01*, 2001.

[SD99] Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. *IJCV*, 1999.

[Set99] J.A. Sethian. *Level Set Methods and Fast Marching Methods*. 1999.

[SG98] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. In *International Conference on Computer Vision (ICCV 98)*, 1998.

[SK99] H. Saito and T. Kanade. Shape reconstruction in projective grid space from large number of images. In *Computer Vision and Pattern Recognition (CVPR-99)*, pages 49–54, 1999.

[SMC00] G. Slabaugh, T. Malzbender, and W. B. Culbertson. Volumetric warping for voxel coloring on an infinite domain. In *3D Structure from Images - SMILE 2000*, 2000.

[TWK88] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motions. *Artificial Intelligence*, 1988.

[UKG98] M. Ulvklo, H. Knutsson, and G. H. Granlund. Depth segmentation and occluded scene reconstruction using ego-motion. *SPIE Visual Information Processing*, 1998.

[Vek99] O. Veksler. *Efficient Graph-Based Energy Minimization Methods in Computer Vision*. PhD thesis, Cornell University, 1999.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399