

Accurate and Scalable Surface Representation and Reconstruction from Images

Gang Zeng¹ Sylvain Paris² Long Quan¹ François Sillion³

¹ Dep. of Computer Science, HKUST, Clear Water Bay, Kowloon, Hong Kong

{zenggang,quan}@cse.ust.hk

² MIT CSAIL, 32 Vassar Street, Cambridge, MA 02139, USA

sparis@csail.mit.edu

³ ARTIS[†]/ GRAVIR-IMAG, INRIA Rhône-Alpes, 38334 Saint Ismier, France

Francois.Sillion@imag.fr

Abstract

We introduce a new surface representation method, called *patchwork*, to extend three-dimensional surface reconstruction capabilities from multiple images. A patchwork is the combination of several *patches* that are built one by one. This design potentially allows for the reconstruction of an object with arbitrarily large dimensions while preserving a fine level of detail. We formally demonstrate that this strategy leads to a spatial complexity independent of the dimensions of the reconstructed object, and to a time complexity that is linear with respect to the object area. The former property ensures that we never run out of storage and the latter means that reconstructing an object can be done in a reasonable amount of time. In addition, we show that the patchwork representation handles equivalently open and closed surfaces whereas most of the existing approaches are limited to a specific scenario, an open or closed surface but not both.

The patchwork concept is orthogonal to the method chosen for surface optimization. Most of the existing optimization techniques can be cast into this framework. To illustrate the possibilities offered by this approach, we propose two applications that demonstrate how our method dramatically extends a recent accurate graph technique based on minimal cuts. We first revisit the popular carving techniques. This results in a well-posed reconstruction problem that still enjoys the tractability of voxel space. We also show how we can advantageously combine several image-driven criteria to achieve a finely detailed geometry by surface propagation. These two examples demonstrate the versatility and flexibility of patchwork reconstruction. They underscore other properties inherited from patchwork representation: Although some min-cut methods have difficulty in handling complex shapes (e.g., with complex topologies), they can naturally manipulate any geometry through the patchwork representation while preserving their intrinsic qualities. The above properties of patchwork representation and reconstruction are demonstrated with real image sequences.

Index Terms

(I. Computing Methodologies).(4 Image Processing and Computer Vision).(5 Reconstruction) & (9 Applications): patchwork representation and reconstruction, space carving, graph-cuts, level-sets, patch-wise carving, patch-wise propagation.

I. INTRODUCTION

Three-dimensional automated reconstruction from multiple images is a natural extension of stereoscopic reconstruction. Combining information from several images makes the process more robust and precise. It is also possible to handle larger scenes since more viewpoints and view directions are available. A wealth of quality work has been produced to address the resulting challenges and to propose usable applications in the domains of virtual reality, movie making, entertainment, etc. In particular, great progress has been made in terms of camera calibration and surface optimization. The former work determines the parameters of the cameras such as their positions and focal lengths, while the latter work focuses on the actual geometry of the observed scene. In this paper, we focus on the geometry reconstruction part as we believe several points can be further improved.

Our paper focusses on two major issues that remain largely unaddressed in surface reconstruction: scalability and flexibility. First, even in a favorable situation, one cannot recover an arbitrarily large geometry due to resource limitations. Most of the existing techniques handle the entire scene at once. Therefore, for a given resolution, the size of the reconstructed scene is bound by the available memory of the machine that executes the program. In addition to this storage issue, since the time complexity of the optimization algorithms is higher than linear, increasing the scene size inherently leads to an explosion of the processing time (*i.e.* an increase of the object size yields an even greater increase of running time). Thus, large scenes are limited to coarse reconstructions that ignore the fine details. Second, existing methods represent the object's surface either with a single-value explicit depthfield, $z(x, y)$ (or $d(x, y)$ for disparity maps), or with a voxel space or an implicit function $\phi(x, y, z) = 0$ (*i.e.*, level set). These two representations address different configurations. Depthfields and disparity maps perform well with cameras that lie only on one side of the scene but it is difficult to extend these methods to arbitrary camera positions. Voxel spaces and level sets provide effective solutions when numerous cameras are available, but they break down with limited view directions. As a consequence, these techniques cannot cope with an arbitrary camera layout, and the user has to select the algorithm according to the scenario.

In order to overcome these limitations, in this paper we present our *patchwork* surface representation method. It consists of a collection of small surface pieces, the *patches*, that are progressively reconstructed and stitched together. Despite its apparent simplicity, it is built on the fundamental assumption that the reconstruction problem is a local issue. Let us consider the example of acquiring the geometry of a head. It seems reasonable and even desirable that, whatever process we use, the shape of one ear does not depend on the shape of the other. Any other behavior would mean, for instance, that adding an earring on one side changes the

geometry of the other ear. This would be incoherent. This assumption will be formally defined and assessed. We show that except for visibility (*i.e.*, whether or not a point is visible from a camera), the other components involved in the existing optimization techniques are local.

Independently of the selected optimization technique, our patchwork representation method induces several interesting gains. The first advantage is that dealing with patches makes the amount of handled data fixed and the processing time proportional to the number of patches. These properties are formally proven. Second, the patch parameterization can be adjusted for each patch. This allows for the representation of complex surfaces with methods that usually handle only depthfields or disparity maps. Third, the formulation is independent of the topology of the surface, and thus the same algorithm deals seamlessly with both open and closed surfaces depending on the setup. If the cameras provide enough information, the whole scene is built; if not, only a partial reconstruction is achieved.

We also address the practical issues that make this representation fully usable. All the patches are registered into a distance field to build a coherent structure. We define a proper shape for the patches in order to preserve the continuity at their boundaries. We expose an ordering strategy to maximize the quality of the produced surface. This complete framework is demonstrated with two practical reconstruction algorithms based on minimal cuts. The first one builds upon carving techniques to associate, in an effective way, voxels and graph optimization. The voxel space provides a robust estimation of the visibility and of the object topology whereas minimal cuts are used to produce a finely detailed geometry. The second one combines several geometric cues to recover the object shape. Reliable 3D points are used as starting points for a propagation process that uses images to build the final shape progressively.

Contributions: In summary, the patchwork representation and reconstruction described in this paper enables scalable and flexible algorithms by introducing the following contributions:

- 1) *Local Prior:* We introduce a new interpretation of the smoothness assumption. The scope of the corresponding prior is only local.
- 2) *Scalability:* The representation allows for the reconstruction of scenes of arbitrary size (or equivalently, a fine level of details).
- 3) *Versatility:* The reconstruction can be used with classical optimization techniques while preserving their intrinsic qualities.
- 4) *Flexibility:* The reconstruction makes it possible to overcome limitations such as topology handling inherent in some optimization techniques. The most significant advantage of this flexibility is the ability of our algorithm to retrieve both complete shapes (when the whole scene is visible) and open surfaces (when some regions are hidden).

II. PREVIOUS WORK

The three-dimensional reconstruction problem is inherently ill posed. There are several geometric solutions that are consistent with the input images. In order to alleviate the problem arising from the multiple solutions, the usual approach is to add an *a priori* hypothesis concerning the objects. Classically, this hypothesis states that the reconstructed surface must be regular, *i.e.*, the objects must be smooth. This assumption is interpreted in various frameworks, resulting in different mathematical formulations. Combining consistency with this *a priori* regularity leads to an optimization step that dominates the other steps in terms of spatial and temporal complexity. In the following sections, we review the existing reconstruction methods while focusing on their optimization techniques and their complexity management.

A. No Optimization

These techniques do not use optimization. Instead, the proposed surface is the largest one consistent with a given criterion.

1) *Visual Hull*: Laurentini *et al.* [1] introduced the *visual hull* as the largest volume consistent with the silhouettes observed from several viewpoints. This results in an approximate shape that captures the large features of the scene but mainly ignores the small details. Several efficient approaches have followed: fast computation [2], reconstruction from uncalibrated cameras [3], spline model [4], and so on. These approaches are mainly used for real-time applications [5] or as a first step to initiate a more accurate process [6], [7].

Relatively to our goal, the visual hull scales up nicely but cannot be considered as a final result because it lacks detail.

2) *Photo Hull*: Seitz and Dyer [8] popularized the use of a discrete volumetric representation (the voxels) in conjunction with a color criterion, the *photo-consistency*. Considering a point \mathbf{p} that is visible from the cameras $i \in \mathcal{V}_{\mathbf{p}}$ seeing colors $\{C_{\mathbf{p}}^i\}$, the photo-consistency $P_{\mathbf{p}}$ of \mathbf{p} is computed using the color distance d :

$$P_{\mathbf{p}} = \frac{1}{|\mathcal{V}_{\mathbf{p}}|} \sum_{i \in \mathcal{V}_{\mathbf{p}}} d(C_{\mathbf{p}}^i, \bar{C}) \quad \text{with} \quad \bar{C} = \frac{1}{|\mathcal{V}_{\mathbf{p}}|} \sum_{i \in \mathcal{V}_{\mathbf{p}}} C_{\mathbf{p}}^i. \quad (1)$$

The algorithm sweeps through the voxel space and carves out the voxels with a photo-consistency above a given threshold. The rationale is that a perfectly Lambertian point \mathbf{p} appears in the same color as from the viewpoint and thus, $P_{\mathbf{p}} = 0$. The threshold relaxes the hypothesis to process scenes that are not perfectly Lambertian. This approach has been developed in numerous directions such as a better sweep scheme [9], robustness against noise [10], transparency [11], a probabilistic framework [12], [13], and other voxel shapes [2]. More references can be found in the survey [14].

In practice, these methods are easy to set up but yields low-accuracy results on untextured regions because of the lack of color variation. The voxel approach is limited by the available resources because the necessary storage is proportional to the bounding volume of the scene.

B. Optimization by Local Operators

A number of methods formulate a global objective for their optimization stage and then solve the objective by means of local operators.

1) *Level Sets*: Level sets [15] are a flexible method to optimize functionals that can be expressed as a weighted minimal surface:

$$\iint w(\mathbf{x}) ds. \quad (2)$$

A time-evolving surface $\mathcal{S}(t)$ is represented at time t by the zero level set of an implicit function $\phi(\mathbf{x}, t)$, *i.e.* $\phi(\mathcal{S}(t), t) = 0$. To minimize Functional (2), the surface evolves according to a steepest-descent process. From the Euler-Lagrange formula, ϕ is driven by a partial differential equation (PDE):

$$\frac{\partial \phi}{\partial t} = \nabla w \cdot \nabla \phi + w \|\nabla \phi\| \operatorname{div} \frac{\nabla \phi}{\|\nabla \phi\|}. \quad (3)$$

It is important to note that the global integral (2) is minimized using local differential operators (3) that only consider the local neighborhood of each point. Despite the global formulation, the technique is driven on a local scale.

Faugeras and Keriven [16] cast the reconstruction problem into the level-set framework to allow for complex objects of arbitrary genus to be rebuilt. It also eases visibility management by estimating occlusions between each evolution step. The w function in Equation (2) is defined to account for the texture correlation by computing the zero-mean normalized cross-correlation (ZNCC) between pairs of cameras $\{C_i, C_j\}$. For a 3D point \mathbf{x} , the ZNCC value $Z_{ij}(\mathbf{x})$ is defined with the projections \mathbf{p}_i and \mathbf{p}_j of \mathbf{x} in cameras C_i and C_j . For an image point \mathbf{p} , $\bar{I}_{\mathbf{p}}$ and $\sigma_{\mathbf{p}}$ denote the mean and standard deviation of the intensity in the neighborhood $\mathcal{N}_{\mathbf{p}}$. Using a homography π to account for the perspective distortion between the two cameras (*i.e.* $\pi(\mathbf{p}_i) = \mathbf{p}_j$ and $\pi(\mathcal{N}_{\mathbf{p}_i}) = \mathcal{N}_{\mathbf{p}_j}$), we finally get:

$$Z_{ij}(\mathbf{x}) = \frac{1}{|\mathcal{N}_{\mathbf{p}_i}|^2 \sigma_{\mathbf{p}_i} \sigma_{\mathbf{p}_j}} \sum_{\mathbf{q} \in \mathcal{N}_{\mathbf{p}_i}} (I_{\mathbf{q}} - \bar{I}_{\mathbf{p}_i})(I_{\pi(\mathbf{q})} - \bar{I}_{\mathbf{p}_j}). \quad (4)$$

This results in convincing reconstructions, especially for high-genus objects. The counterpart is a lack of surface sharpness because of the high-order derivatives that control the process (Eq. 3). Several methods have extended the original technique: with contours [17], with contours and 3D points [18], for non-Lambertian objects [19], and so on.

These methods are limited by resources: A direct implementation handles the whole volume of the scene, while the *narrow band* technique [20] stores only the values close to the surface and requires an amount of memory proportional to the area of the scene.

2) *Generalized Cylinder*: Terzopoulos *et al.* [21] used general cylinders to retrieve the scene geometry from a set of silhouettes. They add symmetry constraints to their model to work from a single image. The optimization is expressed as an integral minimization, leading to local evolution rules based on partial derivatives. Relatively to our aim, the drawback is that it is unlikely to capture fine details because the solution space is limited to generalized cylinders.

3) *Snake*: Herni \acute{e} dez and Schmitt [7] determine the surface topology from the object's visual hull. They use a snake approach based on *gradient vector flow* to preserve this topological information. Akin to level sets, the evolution is driven by local differential operators.

The accuracy of the results is impressive but the cost is that both a surfacic and a volumetric data structure are maintained. Although a hierarchical structure is used, it still grows with the object size, impeding the scalability and inducing a long processing time (several hours).

4) *Free-Form Deformation*: Isidoro and Sclaroff [6] minimized the retro-projection error using free-form deformations. The applied transformations are also local although the goal is a global decrease in the errors. The surfacic representation is an obstacle to scalability.

C. Global Optimization

The previous methods adjust a deformable model step by step to fit the actual geometry and we have shown that the modification applied at each time step to a point is explicitly determined from its neighborhood. We now review another category of techniques that we call *global* in the sense that the treatment applied to a surface point depends on the whole surface, at least formally, and cannot be explicitly derived from its neighborhood.

1) *Minimal Cuts on Disparity Maps and Depthfields*: Roy and Cox [22] showed how to use the graph-flow theory [23] to generalize the purely one-dimensional Dynamic Programming technique to the two-dimensional problem raised by disparity maps. They designed a weighted graph such that computing its maximum flow and extracting a corresponding cut leads to an exact solution of a functional of the following form (with $c_{\mathbf{p}}$ being the consistency at a pixel \mathbf{p} , $d_{\mathbf{p}}$ the disparity, and \mathcal{A}_4 the set of the four-connected adjacent pixels):

$$\sum_{\mathbf{p}} c_{\mathbf{p}} + \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{A}_4} |d_{\mathbf{p}} - d_{\mathbf{q}}|. \quad (5)$$

This functional models the trade-off between the consistency (left term) and the regularity of the result (right term). The advantage compared to other techniques is that the functional (5) is

solved exactly, *i.e.*, a global minimum of the functional is found whereas most of the methods such as level sets and snakes reach only a local minimum.

Other approaches have been proposed to use minimal cuts. Equation (5) can be interpreted in the Markov Random Field framework [24]. Paris *et al.* [25] reinterpreted it in three-dimensional world to handle depthfields instead of disparity maps. They demonstrated how to solve the following continuous functional up to an arbitrary discretization (the surface is parameterized as a depthfield $z(x, y)$, and the α_x and α_y functions modulate the regularization term):

$$\iint \left(c(x, y, z(x, y)) + \alpha_x(x, y) \left| \frac{\partial z}{\partial x} \right| + \alpha_y(x, y) \left| \frac{\partial z}{\partial y} \right| \right) dx dy. \quad (6)$$

Kirsanov and Gortler [26] have described a generic optimization framework that leads to optimal solutions for such $z(x, y)$ or $d(x, y)$ parameterizations. This has been demonstrated on three-view reconstructions [27].

Boykov *et al.* [28] introduced the α -expansion technique to apply graph cuts to more general functionals. This opens the way for finer numerical models but the convergence to a global minimum is lost. Kolmogorov and Zabih [29] characterized a general theory on the set of functionals that can be handled by graph cuts. They also applied their method to disparity maps in the multi-view context [30]. In general, these disparity-map techniques yield accurate object boundaries but lack depth precision compared with the depthfield approach.

None of these methods scale up nicely because of their volumetric representation of space.

2) *Segmented Disparity Maps*: Several approaches [31], [32] have shown that satisfying disparity maps can be achieved by segmenting the input images into small regions of constant color. Although this clearly reduces the amount of data, it does not address the scalability issue. The challenge is to scale up while preserving details whereas the segmentation strategy takes advantage of the lack of precision to “smartly” downsample the disparity map.

3) *Minimal Cuts on General Surfaces*: Boykov and Kolmogorov [33] showed how weighted minimal surfaces (Eq. 2) can be minimized when $w ds$ is a Riemannian metric. The major novelty of this work is that general surfaces are handled compared to the disparity maps and depthfields of the previously discussed methods. Vogiatzis *et al.* [34] formulate the multi-view scene reconstruction problem using this framework. From the scalability point of view, the volumetric structure limits the scene size.

4) *Weak Membrane*: Blake and Zisserman [35] described a global optimization technique inspired by the mechanical properties of an elastic membrane. A remarkable point in their study is that they formally prove that two distant features do not interfere [35, p. 60] and behave as though each one were alone. Therefore, even if no local operator is determined, this result proves that the underlying process depends only on local neighborhoods.

D. Local Optimization

The following methods are the most similar to ours. They provide a local approach to the scene recovery problem.

1) *Particles*: Fua [36] introduced a particle technique to recover the scene geometry using particles. The particles obey a global optimization. Though it is a global scheme, it is defined by local interactions between the closest particles only. This representation can scale up because the particles can be handled separately. However, the accuracy of this representation is relatively low. The particles are flat disks whereas our patches have a more general shape.

2) *Image-Space Aggregation*: Szeliski and Golland [11] disambiguated regions with poor information by diffusing the adjacent data. This step can be seen as a local reconstruction of the geometry. It does not, however, address our goals because this local aggregation is only one step in an algorithm that is global most of the time.

3) *Depth Maps Fusion*: Narayanan *et al.* [37] reconstructed several depth maps that are aggregated into a single structure. This approach is similar to ours because it reconstructs the final scene by merging several partial results. However, the described process appears to be highly redundant (*i.e.*, large portions of the scene are reconstructed several times), inducing a large amount of unnecessary computation. Furthermore, the method does not focus on scalability and each depth map deals with the entire scene from a given viewpoint.

4) *Quadratic Patches*: In the context of stereo-vision, Hoff and Ahuja [38] constructed a disparity map by gathering the information stemming from several quadratic patches. The main difference in our approach is that our patches can be any depthfields, not only quadratic shapes. Our surface representation is also self sufficient and independent from the optimization technique whereas Hoff-Ahuja patches need to be interpolated to obtain the final result and rely exclusively on a least squares fit. Carceroni and Kutulakos [39] extended the approach to motion and reflectance recovery. However, the geometric accuracy is still limited by the patch shape. Ohtake *et al.* [40] showed impressive results in the context of surface reconstruction from points but the extension to multi-view stereo is unclear.

In comparison, our patchwork defines a complete surface representation, *i.e.*, it reconstructs a patchwork that is equivalent to reconstructing the surface itself; no additional treatment is needed, the patches spread across the whole surface and continuity is handled during the reconstruction process.

E. Summary

None of the existing methods reaches our scalability objective. In terms of running time, carving techniques are the most efficient because they do not perform any optimization. But

their lack of accuracy and their high memory requirement limit them. In terms of storage, the narrow band implementation of level sets is among the most efficient but this method still needs to store the whole surface in the memory to perform the optimization. Our approach strikes a new balance by building the surface piece by piece, thereby achieving linear time complexity and requiring only a subset of the data to be in memory at a time.

The existing methods are specialized for closed or open surfaces. One has to know *a priori* whether or not the camera configuration allows for a complete reconstruction. Our approach does not require such an information since both types of surfaces are seamlessly handled.

III. CONCEPT DEFINITION AND THEORETICAL STUDY

Here we formalize our problem to highlight the fundamental reasons that justify the use of patches. Let $\mathcal{F}(\cdot)$ be a functional that represents our goal, *i.e.*, \mathcal{F} assigns a value to any surface, \mathcal{S} , and \mathcal{F} is designed so that we consider a minimizer of \mathcal{F} as the result of the reconstruction problem. For now, we do not give more details about \mathcal{F} to keep it as general as possible. The design of such a functional is discussed later.

Patch definition: Intuitively, a *patch* is a small piece of a surface \mathcal{S} . Formally speaking, a patch, \mathcal{P} , is a connected subset of \mathcal{S} . A *patchwork* representation of \mathcal{S} is a set of patches $\{\mathcal{P}_i\}$ such that $\bigcup \mathcal{P}_i = \mathcal{S}$.

A. Patchwork Reconstruction

In the previous section, we showed that many reconstruction strategies are driven – either explicitly or implicitly – by local criteria. Here we state formally our base assumption: The result at a point depends only on its neighborhood, while distant points can be ignored.

1) *Locality Assumption:* We name \mathcal{S}_0 a minimizer of \mathcal{F} over the whole 3D space, *i.e.*, $\mathcal{S}_0 = \operatorname{argmin}_{\mathcal{S} \subset \mathbb{R}^3} \mathcal{F}(\mathcal{S})$. \dot{r} and \mathring{r} are such that $0 < \dot{r} < \mathring{r}$. $\dot{\mathcal{B}}_{\mathbf{p}}$ and $\mathring{\mathcal{B}}_{\mathbf{p}}$ denote the two balls centered on a point \mathbf{p} with radii \dot{r} and \mathring{r} . Minimizing \mathcal{F} in the ball $\mathring{\mathcal{B}}_{\mathbf{p}}$ returns a surface, $\mathring{\mathcal{S}} = \operatorname{argmin}_{\mathcal{S} \subset \mathring{\mathcal{B}}_{\mathbf{p}}} \mathcal{F}(\mathcal{S})$. See Figure 1 for a 3D illustration of these entities.

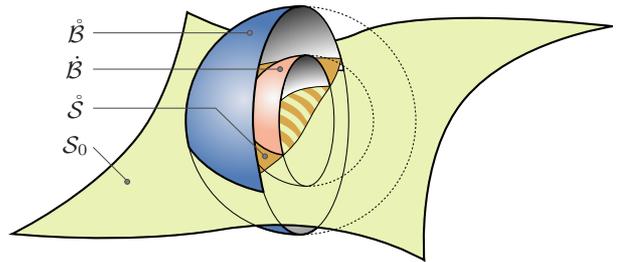


Fig. 1. We assume that there exist $\dot{\mathcal{B}}$ and $\mathring{\mathcal{B}}$ such that, inside $\mathring{\mathcal{B}}$, the result $\mathring{\mathcal{S}}$ of the optimization within $\mathring{\mathcal{B}}$ equals the global result \mathcal{S}_0 . This common portion corresponds to the striped area.

The *locality assumption* claims that, if the visibility information is known, there exist values for \dot{r} and \mathring{r} such that for any point $\mathbf{p} \in \mathcal{S}_0$:

$$\mathring{\mathcal{S}} \cap \dot{\mathcal{B}}_{\mathbf{p}} = \mathcal{S}_0 \cap \dot{\mathcal{B}}_{\mathbf{p}}. \quad (7)$$

- *Interpretation:* This assumption means that a local optimization yields a correct result except on the border of the considered volume (*i.e.*, between $\dot{\mathcal{B}}_p$ and $\mathring{\mathcal{B}}_p$). This restriction is reasonable because the border points have a truncated neighborhood (we cannot expect any optimization algorithm to give reliable results with partial data).

2) *Global Optimality:* In a number of cases, \dot{r} and \mathring{r} can be set so that a succession of local optimizations and a single global optimization produce identical or similar results. This demonstrates that a locally driven process is as stable as a global one.

- *Weak Membrane:* Blake and Zisserman [35] studied the weak-membrane model, which approximates the data d by a piecewise-smooth membrane u by optimizing the energy function controlled by the parameters α and λ , with dA and dl the area and length measures:

$$\mathcal{E} = \int \{(u - d)^2 + \lambda^2(\nabla u)^2\} dA + \alpha \int dl. \quad (8)$$

They showed that two discontinuities whose distance is significantly larger than λ do not interfere [35, page 60]. In our context, this means that setting $\mathring{r} \gg \dot{r} + \lambda$ ensures that any discontinuity interacting with points in $\dot{\mathcal{B}}_p$ are in $\mathring{\mathcal{B}}_p$, thereby considered by the local optimization. Therefore, potential differences between a local and global process can impact only the continuous regions, yielding at most limited differences since these areas are smooth.

- *Level Sets:* Level-set optimization is an iterative process whereby the current estimate is modified according to Equation (3). We name Υ the number of iterations (or a bound over it). Derivatives of order ω involve the adjacent values up to a distance $\lceil \omega/2 \rceil$. Thus, using the discretization step δ of the level-set grid and the maximum order of the involved derivatives Ω , we set $\mathring{r} = \dot{r} + \Upsilon \delta \lceil \frac{\Omega}{2} \rceil$. This guarantees that all the points in $\dot{\mathcal{B}}_p$ have been exactly computed as in the global strategy since all the involved data are in $\mathring{\mathcal{B}}_p$.

- *Graph Cuts:* For the graph cut approaches, Kolmogorov and Zabih [30] and Paris *et al.* [25] handle discontinuities, hence continuous regions are independent. Thus it is sufficient to set \dot{r} so that $\dot{\mathcal{B}}_p$ contains the largest continuous region.

- *Discussion:* In several cases, the locality assumption is either exact (for level sets and min cuts) or approximate (for weak membranes). However, determining the characteristic parameters of a given scene might be difficult. In particular, the graph-cut criterion requires an analysis of the whole scene. Therefore, in practice, the size of the local volume is set by the user and may not meet these criteria. Nonetheless, we have a strong result: For sufficiently large patches, the local optimization is equivalent or close to a global one. We further study this difference between global and local optimization in the following section.

3) *Study of the Functional:* \mathcal{F} always contains a term \mathcal{C} related to the consistency to ensure that the final surface \mathcal{S} matches the image content. With a consistency function c (*e.g.* photo-consistency or ZNCC) and a surface measure $d\mu$, this part can be written as:

$$\mathcal{C} = \iint_{\mathcal{S}} c \, d\mu. \quad (9)$$

Using $d\mu = ds$ to measure the surface area leads to the level set functional (2). The problem is then well posed but the sharp details of the scene are not captured.

Another option for the regularization is to add a smoothing term \mathcal{S} (*i.e.* $\mathcal{F} = \mathcal{C} + \mathcal{S}$). To do so, we parameterize \mathcal{S} as a depth field $z(x, y)$ (or $d(x, y)$ for a disparity map) and we introduce a function s that measures the variations of z . From Equation (6), this induces the plane measure $d\mu = dx \, dy$:

$$\mathcal{S} = \iint_{\mathcal{S}} s(z) \, dx \, dy. \quad (10)$$

This approach yields higher accuracy but it depends on the xyz coordinate system. Since the integrals in (9) and (10) consider the whole surface \mathcal{S} , this inherently limits the representable surfaces. Intuitively, splitting \mathcal{S} into small pieces makes it possible to define \mathcal{S} with several depth fields according to different coordinate systems.

Local Coordinate System: For each patch \mathcal{P}_i , a local coordinate system $x_i y_i z_i$ is defined to parameterize \mathcal{P}_i as $z_i(x_i, y_i)$. An appropriate choice for the z_i axis is the surface normal at the location of the patch. The orientation of x_i and y_i has no major influence. We propose two practical strategies to build these axes in the following sections.

Local Prior: The smoothness assumption is expressed locally. Instead of applying the smoothness term \mathcal{S} on the whole surface at once, we apply it to each patch separately:

$$\mathcal{S} = \sum_i \iint_{\mathcal{P}_i} s(z_i) \, dx_i \, dy_i. \quad (11)$$

The integration is now split in several domains \mathcal{P}_i , and a coordinate system $x_i y_i z_i$ is introduced for each of them. This overcomes the parameterization limitation of the global approach since \mathcal{S} is now represented as an assembly of depth fields instead of a single one. The same treatment can be applied to \mathcal{C} . Hence, with $f = c + s$, we can elegantly summarize the transformation from a global formulation to a local one:

$$\mathcal{F} = \iint_{\cup \mathcal{P}_i} f \, dx \, dy \quad \rightsquigarrow \quad \mathcal{F} = \sum_i \iint_{\mathcal{P}_i} f \, dx_i \, dy_i \quad . \quad (12)$$

Thus, our patchwork representation is relatively natural and simple from a formal point of view: A union in the geometric realm is transformed into a sum in the functional domain.

This local expression shows that the patches can be optimized independently. In practice, we minimize Equation (6) for each patch using the depth-field scheme [25].

4) *Surface Reconstruction*: The patchwork reconstruction consists of building a set of patches $\{\mathcal{P}_i\}$ that represents the whole surface \mathcal{S} . Several local optimization processes are run, *i.e.*, we use several local volumes \mathring{B}_i , each one producing a surface portion \mathring{S}_i . Because the border points of \mathring{S}_i are not reliable, we keep only the center part $\mathring{S}_i \cap \mathring{B}_i$. This is the actual patch \mathcal{P}_i produced by the local process. In order to smoothly connect these local patches together and form a global surface, we consider the following three aspects:

a) *Continuity*: We set the size of the local volumes so that the domains of adjacent patch reconstructions overlap with each other. As a consequence, adjacent patches share part of their data. This favors continuity. Moreover, we design the stitching process in order to guarantee the surface continuity (see Section III-D).

b) *Order*: Since a reliable patch is fixed after it has been built, it ignores the computation that occurs after its creation; and, as we have just described, it takes into account the already created patches. This temporal scheme can be seen as a data flow. A “new” patch receives information from the “old” patches. Thus, we can exploit the order in which the patches are built to reconstruct by priority the most reliable regions. We develop this ordering strategy in our practical implementations.

c) *Distance Field*: Once each patch is built, it is aggregated in a distance field as described by Curless and Levoy [41]. When all the patches are recovered, the final surface is extracted using the Marching Cube technique [42]. We give further details in Section III-D.

B. Study of the Complexity

We here compare the temporal and spatial complexities of a general global optimization and of our patchwork approach. Let us consider that \mathcal{S} has a 2D area $a_{\mathcal{S}}$ and a 3D volume $v_{\mathcal{S}}$, and that it is represented by a discrete structure with a discretization size δ . For instance, for level sets, this structure is the distance field embedding the surface; for min cuts, it is the quantized 3D (or disparity) space that supports the surface vertices.

Global optimization: An algorithm that minimizes \mathcal{F} over the whole surface \mathcal{S} deals with a data structure at least $\mathcal{O}(a_{\mathcal{S}} \delta^{-2})$ in size. This is the case for some graph-cut techniques [43] and for the narrow-band implementation of level sets [20]. Some algorithms (such as level sets, carving methods or some graph-cut techniques) use volumetric representations and hence have a space complexity in the order of $\mathcal{O}(v_{\mathcal{S}} \delta^{-3})$.

We consider a minimizing process with a complexity of degree $\alpha \geq 1$. Therefore, the time complexity is $\mathcal{O}(a_{\mathcal{S}}^{\alpha} \delta^{-2\alpha})$ or $\mathcal{O}(v_{\mathcal{S}}^{\alpha} \delta^{-3\alpha})$ depending on the surface representation. The complexity of level sets [16], [18] is unclear because it depends on the number of iterations, which in turn depends on the starting point and the target shape. Min-cut algorithms are typically

TABLE I

COMPARISON OF THE COMPLEXITY AMONG DIFFERENT METHODS

	SPACE			TIME		
	global	patches	gain	global	patches	gain
surfacic	$a_S \delta^{-2}$	$a_P \delta^{-2}$	η	$a_S^\alpha \delta^{-2\alpha}$	$\eta a_P^\alpha \delta^{-2\alpha}$	$\eta^{\alpha-1}$
volumetric	$v_S \delta^{-3}$	$v_P \delta^{-3}$	$\eta^{\frac{3}{2}}$	$v_S^\alpha \delta^{-3\alpha}$	$\eta v_P^\alpha \delta^{-3\alpha}$	$\eta^{\frac{3}{2}\alpha-1}$

cubic (or slightly better [44]). In practice, they behave almost linearly ($\alpha \approx 1.2$) [45]. Note that some min-cut techniques (e.g., Kolmogorov and Zabih [43]) are iterative and their complexity could be higher as mentioned for level sets.

Patch Optimization: Let us subdivide the surface \mathcal{S} into patches \mathcal{P} with area a_P . The number of patches η is on order of $\mathcal{O}(a_S/a_P)$. To compare with \mathcal{S} , we also define a pseudo-volume $v_P = \mathcal{O}(a_P^{-\frac{3}{2}})$ by considering that surfaces and volumes are related by a logarithmic ratio of $\frac{3}{2}$. Optimizing \mathcal{F} over a patch has a space complexity on the order of $\mathcal{O}(a_P \delta^{-2})$ or $\mathcal{O}(v_P \delta^{-3})$. Patches are processed one by one; therefore, the overall space complexity is the same. Only the storage of the final result requires more space but this can be done off-line (e.g., on the hard drive). Since we optimize η patches, the overall time complexity is $\mathcal{O}(\eta a_P^\alpha \delta^{-2\alpha})$ or $\mathcal{O}(\eta v_P^\alpha \delta^{-3\alpha})$.

Comparison: Table I summarizes all these results. It appears that the patches bring significant gains in terms of space and time complexity. The spatial complexity is the main gain since we can divide the memory needed by a factor of the number of patches used. However, we cannot decrease the size of the patches infinitely to increase their number because we would not be able to find a satisfactory result (see Section VI).

Scalability property: The patches allow for almost unlimited scalability because the space complexity depends only on the patch size and not on the object size.

Note that we need to store the position of each patch relative to the global surface. This requires storage on the order of $\mathcal{O}(\log(v_S \delta^{-3}))$, which is negligible because it always fits within three classical floating-point values xyz .

C. Study of the Parameterization

The patch also alleviates the limitation on the parametrization inherent in disparity map and depthfield methods. These methods handle a scalar field. In a nutshell, the depth is a function of the two other coordinates, i.e., $z = f(x, y)$ for some function f . This limits the usability of these techniques. First, special care is needed for proper handling of the cases that require several z values for a single (x, y) . Several functions, f_1, f_2, \dots , are then manipulated. Moreover, if the object surface is tangent to the z axis, these methods fail because of $\|\nabla f\| = \infty$.

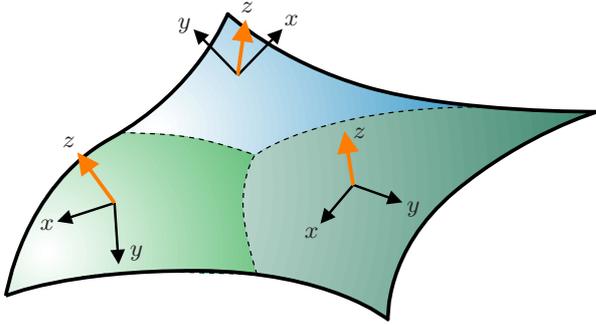


Fig. 2. Three patches with their local coordinate system

The patch approach eliminates these shortcomings. By definition, the patch reconstruction deals with several surfaces and intrinsically manipulates several f functions. Furthermore, the xyz coordinate system can be adapted to each patch. This means that the z axis can be chosen orthogonally to the surface to guarantee that the tangent case never occurs.

The topology is not a problem in the sense that patches can cope with any topology. However, the topology is not determined by the patches themselves. We rely on a side technique to determine the topology. We propose practical solutions to this side technique later.

Multi-resolution: This local parameterization opens avenues for a multi-resolution reconstruction. It would be possible to control the precision of the reconstruction patch by patch to focus on the most detailed parts. Though interesting, this is beyond the scope of this paper and is kept for future work.

D. Study of the Stitching Process

To collect all the patches and construct the final surface, we use a technique inspired by Curless and Levoy [41]. It has the advantage of allowing incremental updates with a fine control over the fusion. It is also shown to be optimal in a least square sense under some mild assumptions (see [41] for details). There are nonetheless two important caveats to consider: The patch borders should not be incorporated into the final surface since they are not reliable, and this step must not incur spurious discontinuities on the surface.

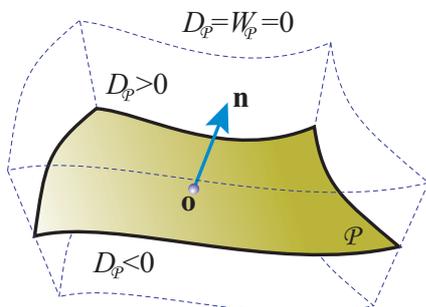


Fig. 3. The patch \mathcal{P} . The dashed lines delimit the neighborhood. \mathbf{o} is the center of \mathcal{P} , while \mathbf{n} is the local normal estimation.

Technically, the stitching process relies on two structures: a signed distance field D and a volumetric weight function $W \geq 0$, both sampled on a regular 3D grid. Each new patch locally modifies D and W . At the end of the process, the surface is extracted as the zero level set of D using the *Marching Cubes* technique [42]. W can be understood as the “history” of the construction of D ; each patch “records its influence” in W . Thus, we adapt the *Marching Cubes* algorithm to cope with a partially defined distance field. If a grid cell contains an uninitialized or null W value, no triangle is produced.

In practice, for each new patch \mathcal{P} , we compute a distance field $D_{\mathcal{P}}$ and a weight function $W_{\mathcal{P}}$

restricted to the neighborhood of \mathcal{P} (i.e., $D_{\mathcal{P}} = W_{\mathcal{P}} = 0$ outside the neighborhood, cf. Fig. 3). $D_{\mathcal{P}}$ is the signed distance to \mathcal{P} . $W_{\mathcal{P}}$ is related to the confidence we have in \mathcal{P} ; its design is discussed later. At each grid vertex \mathbf{x} , D and W are updated as follows:

$$D(\mathbf{x}) = \frac{W(\mathbf{x})D(\mathbf{x}) + W_{\mathcal{P}}(\mathbf{x})D_{\mathcal{P}}(\mathbf{x})}{W(\mathbf{x}) + W_{\mathcal{P}}(\mathbf{x})} \quad (13a)$$

$$W(\mathbf{x}) = W(\mathbf{x}) + W_{\mathcal{P}}(\mathbf{x}). \quad (13b)$$

These equations (13) show that $D(\mathbf{x})$ is the mean of patch distances $D_{\mathcal{P}_i}$ weighted by $W_{\mathcal{P}_i}$.

1) *Patch Weight:* The previous remark outlines the importance of $W_{\mathcal{P}_i}$ in determining the influence of \mathcal{P}_i on the final result. There are two major issues: discarding the unreliable points near the patch border and ensuring continuity across the patches. Both objectives are fulfilled by using a $W_{\mathcal{P}_i}$ function that smoothly decreases to 0 near the boundary. Thus, the border points have a negligible influence compared to the other patches (remember that the patches overlap). Continuity is guaranteed since the weights smoothly cross-fade.

Formal Study: To achieve surface continuity, from the Implicit Function Theorem, it suffices that:

- (1) D is C^1 continuous and,
- (2) ∇D is not null when $D = 0$.

From Equations (13), if $D_{\mathcal{P}}$ and $W_{\mathcal{P}}$ are C^1 , then Condition (1) is fulfilled. Condition (2) is not as direct. Theoretically, the gradient could vanish, but this is unlikely to occur in practice. First, $\nabla(W_{\mathcal{P}}D_{\mathcal{P}}) = D_{\mathcal{P}}\nabla W_{\mathcal{P}} + W_{\mathcal{P}}\nabla D_{\mathcal{P}}$ can vanish near the border because $W_{\mathcal{P}} = 0$ and $\nabla W_{\mathcal{P}} = 0$ but it does not affect ∇D since the patches overlap. Then, within the patch neighborhood, $\nabla D_{\mathcal{P}}$ cannot vanish because $D_{\mathcal{P}}$ is a signed distance function. However merging several patches at the same location may cancel the gradient ∇D . In practice, the zeros of D are near the zeros of $D_{\mathcal{P}}$, thus $D_{\mathcal{P}}\nabla W_{\mathcal{P}}$ is negligible compared to $W_{\mathcal{P}}\nabla D_{\mathcal{P}}$. The gradient cancellation would therefore imply that two patches have been reconstructed at the same place with their normals forming an angle greater than $\frac{\pi}{2}$. During our experiments, such an extremely large error never occurred.

Implementation: We use the patch center, \mathbf{o} , to define $W_{\mathcal{P}}$:

$$W_{\mathcal{P}}(\mathbf{x}) = \begin{cases} \left(1 - \frac{\|\mathbf{x} - \mathbf{o}\|^2}{\sigma^2}\right)^2 & \text{if } \|\mathbf{x} - \mathbf{o}\| < \sigma \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

We set σ such that, for any point \mathbf{p} on the border of \mathcal{P} , $\|\mathbf{p} - \mathbf{o}\| > \sigma$. With this condition, Condition (1) is fulfilled: $W_{\mathcal{P}}$ is C^1 , and the border discontinuities of $D_{\mathcal{P}}$ and $\nabla D_{\mathcal{P}}$ are cancelled by $W_{\mathcal{P}} = 0$ and $\nabla W_{\mathcal{P}} = 0$.

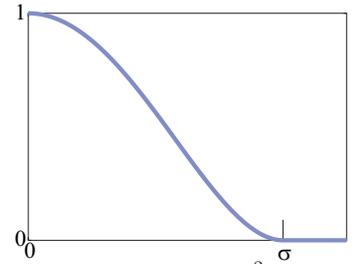


Fig. 4. $x \mapsto \left(1 - \frac{x^2}{\sigma^2}\right)^2$ if $|x| < \sigma$, 0 otherwise. This function is also known as the Tukey function.

2) *Weight Refinement*: The previous construction is independent of the input images: $W_{\mathcal{P}}$ depends only on the patch size. We refine this approach with $W_{\mathcal{P}}^*$ by accounting for the “quality” of the points: Consistent points are given more influence. In practice, this further reduces the influence of the border points if they are erroneous. A direct implementation could be: $W_{\mathcal{P}}^* = \max(0, Z) W_{\mathcal{P}}$ ($\max(\cdot)$ keeps it non-negative and cancels the gross errors). However, for real images, the ZNCC is unlikely to be C^1 , and Condition (1) would therefore be violated.

To address this point, we smooth the ZNCC while preserving its overall structure (we should not lower the influence of consistent regions close to inconsistent areas). We apply an edge-preserving filter inspired by Perona and Malik [46]. Using the $x_i y_i z_i$ coordinate system of \mathcal{P}_i , we consider $\varphi(x_i, y_i) = \max(0, Z(x_i, y_i, z_i(x_i, y_i)))$, the restriction of $\max(0, Z)$ to \mathcal{P}_i . Similarly to [47], we assume that surface areas of the same color are coherent regions. Thus, we preserve the edges where the color changes (we build a color map of \mathcal{P}_i by averaging the colors seen by the ZNCC cameras). The color intensity gradient ∇I then yields an effective and computationally efficient estimation of the edges. Putting this together with a stopping function g [48], we obtain:

$$\frac{\partial \varphi}{\partial t} = \operatorname{div}(g(\|\nabla I\|)\nabla \varphi). \quad (15)$$

Note that the g function is designed to slightly smooth the edges in order to avoid sharp discontinuities. Thus, Condition (1) is satisfied and the smoothing mainly occurs within regions of the same color. Finally, we extend φ to 3D: $\Phi(x_i, y_i, z_i) = \varphi(x_i, y_i)$ and define: $W_{\mathcal{P}}^* = \Phi W_{\mathcal{P}}$.

This refinement improves the accuracy because the inconsistent points have less influence. Moreover, it makes the boundaries of the open surfaces clean since the gross errors in the patch borders are discarded.

E. Discussion

1) *Problem Specificity*: The complexity study relies on the locality assumption stating that the patches can be optimized independently. This is different from the classical approach in parallel computing that subdivides a large problem (*e.g.*, equilibrium in Mechanics [49]) into small subproblems and boundary problems that assure the overall coherence between the subproblems. Classically, the subproblems are iteratively solved until convergence and lead to a complexity at least equal to the original. In our case, except for the visibility, which we handle separately, there is no phenomenon with an overall influence (unlike forces in Mechanics for instance). Thus we do not have to solve a boundary problem. This explains the gain in time.

2) *Normals and Topology*: The surface normal has to be determined to align the local z axis with it. To address this issue, we use a side technique that provides an initial guess. Numerous choices exist: photo hull [9], visual hull [1], level sets [16], etc. We do not require this side

technique to produce an accurate reconstruction; we only need an estimation of the normal. Typically, it can be run at a coarse resolution that fits within the available resources. In addition, we might also rely on this side technique to provide the topology. In the following sections, we detail a scenario for which a side technique is used for normals and topology, and one for which another side technique is used to bootstrap the reconstruction process.

IV. APPLICATION I: PATCH-WISE CARVING FROM MULTIPLE IMAGES

We introduce a practical algorithm¹ that is directly inspired by Space Carving [9]. Carving is flexible (any camera position, any object topology) but uses no prior and thus deals with an ill-posed problem. The outcome from untextured objects may significantly differ from the actual geometry. We revisit this carving strategy with our patchwork representation method. We approximately locate the object surface \mathcal{S} with voxels. The fine geometry is retrieved using a local graph-cut optimization on each patch \mathcal{P} .

A. Initialization

The algorithm starts with a set of calibrated images. If the background is known, we extract the object contours and we use the *visual hull* [1] as a bounding volume (this initialization is akin to [6], [7]). Otherwise, we require the user to provide a bounding box. This volume is then discretized into cubic voxels. It is important to emphasize that the voxels are used only to estimate the visibility and the topology, whereas the actual object surface is defined by the patches. The shape resolution is not directly linked to the voxel size. Thus, we can afford larger voxels than the ones used in the classical carving techniques. Although there might be some unusual cases for which this incurs topological inaccuracies, it has never occurred in our experiments (cf. the result section).

B. Local Optimization

We have chosen the depthfield optimization method [25] based on min-cuts because its geometric formulation is suitable for our goal and, in addition, it ensures the convergence to a global minimum of Equation (6). On the other hand, it is limited by a parameterization $z(x, y)$, but the patchwork representation addresses this point with its multiple local coordinate systems. We refer to the original paper [25] for the technical details.

¹The main idea of this algorithm has been proposed in our prior work [50]. In this section, we describe the algorithm based on the new theory that is proposed in the previous sections. It is a typical example derived from patchwork reconstruction.

C. Voxel Carving

We build upon a classical carving strategy. The voxels are considered one by one and the inconsistent ones are removed. Each time, the visibility is computed from the current voxel set (for this purpose, we use the effective technique described in [51]). The process is iterated until no more voxels can be carved. In this global framework, we define our own carving criterion and ordering scheme.

1) *Carving Criterion*: Instead of computing the photo-consistency of a voxel to decide whether it is carved, we reconstruct a patch within it². We run a graph-cut process; this results in a patch \mathcal{P} and a functional value $\mathcal{F}(\mathcal{P}) = \mathcal{C}(\mathcal{P}) + \mathcal{S}(\mathcal{P})$. The voxel is kept if the consistency value $\mathcal{C}(\mathcal{P})$ is less than a threshold, τ ; otherwise, it is carved. The rationale is that the consistency of \mathcal{P} is high (*i.e.*, $\mathcal{C}(\mathcal{P})$ is low) only if \mathcal{P} is part of the surface. Note that we do not use the smoothness value $\mathcal{S}(\mathcal{P})$ since the carving decision is not directly related to the creation of the fine surface. At the carving level, only the consistency is important.

This carving strategy might not carve enough voxels, akin to the original Space Carving method [9]. However, this would only happen with large, textureless regions since our voxels are one order larger than those of the classical method. In addition, our criterion is more robust than the original because it is based on a whole surface piece instead of a single point. Thus, we have not experienced any problem in our tests, even on faces that include large areas with little texture (cheeks, forehead – cf. Figures 5, 9 and 10).

Normal Estimation: To define the coordinate system, we need a normal estimation. We first start by fitting a plane to the current voxel and its adjacent surface voxels to get \mathbf{n}_0 (shown as short lines on Fig. 5-7.b). Then we build a patch $\mathcal{P}^{(0)}$ from which we estimate a new normal \mathbf{n}_1 . If $\mathbf{n}_1 \neq \mathbf{n}_0$, we build $\mathcal{P}^{(1)}$ using \mathbf{n}_1 . We iterate until $\mathbf{n}_{k+1} = \mathbf{n}_k$. In practice, this occurs in two or three steps. We define $\mathcal{P} = \mathcal{P}^{(k)}$ to compute the carving criterion $\mathcal{C}(\mathcal{P})$. In inconsistent regions, this may not converge. Therefore, if the process is not stabilized after k_{\max} iterations, the voxel is considered to be inconsistent and is carved.

Consistency Function: For the consistency function c (Eq. 6), we use the ZNCC value Z_{ij} (Eq. 4) computed from the two most front-facing visible cameras C_i and C_j according to the normal estimates. For a 3D point \mathbf{x} , we wish to choose a consistency function $c(\mathbf{x}) \geq 0$ that decreases when the match quality increases, which can be computed by $c(\mathbf{x}) = \arccos(Z_{ij}(\mathbf{x}))$. This corresponds to the interpretation of ZNCC as a dot product. In our experiments, it discriminates inconsistent points better than a linear inversion such as $1 - Z_{ij}$ does. This strategy yields satisfying results at a reasonable computational cost. As future work, it would be interesting to

²Note that the patch is not strictly within the voxel. It is large enough to overlap with its neighbors, cf. Section III-D.

test other consistency estimators [7], [16], [52].

Whenever the visual hull \mathcal{V} is available, we add a term v to constrain the patch within \mathcal{V} : $v(\mathbf{x}) = 0$ if $\mathbf{x} \in \mathcal{V}$, ∞ otherwise. In this case, $c(\mathbf{x}) = \arccos(Z_{ij}(\mathbf{x})) + v(\mathbf{x})$.

2) *Ordering Scheme*: The ZNCC is more reliable when computed with front-facing cameras because it limits the perspective distortion and the numerical inaccuracy inherent in it. Therefore, we use the following strategy to reduce the number of voxels processed with grazing view directions: For each voxel, we determine the angles with the normal of the two most front-facing unoccluded cameras. The voxels with small angles are considered first. The underlying idea is that processing the reliable voxels first is likely to carve away inconsistent voxels that were occluding front-facing cameras for other voxels. In other words, this ensures that we always consider the voxel with the “most reliable” ZNCC evaluation according to the current shape estimation. Once a voxel is found to be consistent, it is marked “definitely visible” and it is no longer examined by the carving process (except as a potential occluder). The corresponding patch is merged onto the surface.

D. Summary and Discussion

At a coarse level, our algorithm behaves like a carving technique except that we use the patch consistency \mathcal{C} instead of the photo-consistency and a visibility-driven order. At a fine level, we use a graph cut to build the patches by minimizing the functional (6) within each voxel. The optimization scheme [25] reaches a global minimum of the functional (6). In this respect, the patches are optimal. The consistent patches are then incorporated into a distance field as described in Section III-D. We have shown that, with a proper update scheme, this produces a continuous surface. Finally, when no more consistent voxels are found, the surface is extracted from the distance field.

It is important to highlight that the same algorithm handles complete and partial reconstructions. If the images cover the whole scene, the patches form a closed shape. Otherwise, if some regions remain hidden, an open surface is produced seamlessly. The Marching Cubes algorithm naturally creates a boundary when it reaches an uninitialized domain. Compared to a classical level-set approach, this may produce holes in small invisible regions whereas the level set would seamlessly fill the gap. However, we advocate that these holes are beneficial since they conversely ensure that the produced surface stems from actual image data and not from a “blind” interpolation. One can then apply effective hole-filling techniques to produce a high-quality interpolation [53].

V. APPLICATION II: PATCH-WISE PROPAGATION FROM 3D DATA IN MULTIPLE IMAGES

In this section, we apply the patchwork concept to combining several information sources, especially 3D points and images. This approach³ is motivated by the fact that most scanning devices, such as laser scanners, take a photograph of the scanned object. Purely image-based approaches, such as Lhuillier and Quan’s method [55], also provide reliable 3D points using only standard photographs. We propose a technique that addresses two major points. First, meshing a point cloud is difficult because of the noise and the sampling rate, which may be insufficient. Techniques, such as the ones by Amenta et al. [56] and by Hoppe et al. [57], exist but they do not exploit the images that are available in a number of cases, which would help. Associating images and points eases this reconstruction and yields accurate surfaces. Second, the point set may have holes, *e.g.*, image-based techniques do not extract reliable points in textureless regions. In that case, relying only on points allows for an interpolation surface that lacks details whereas using the available images makes it possible to recover these details. The patchwork provides an effective framework to handle these various situations coherently.

In our method, 3D points and images are considered as input. We do not assume that there is any special property, except that we can estimate the surface normal at the 3D points. This is possible as long as the point cloud is dense enough (see Appendix I for details). In practice, we use Lhuillier and Quan’s technique [55] to produce the 3D points. We have chosen this method because it gives irregularly distributed point sets that illustrate our work well. Nonetheless, the proposed technique can work with any range scanner that provides reliable 3D points.

Our strategy is to perform a propagation in 3D space starting from reliable feature 3D points, which help to avoid potential ambiguities and build a precise surface. To drive this propagation, we need to first define a set of control points, the *seeds*. We define a seed as a couple (\mathbf{s}, \mathbf{n}) , with \mathbf{s} being a 3D position, and \mathbf{n} being the surface normal estimation at this position. The seed list is initialized with the input 3D points and the normal computed from them (cf. Appendix I). We then proceed iteratively. Each iteration of the propagation loop picks a seed from the current list using a best-first strategy, estimates its visibility according to the current surface estimate, constructs an optimal patch around the seed and generates new seeds for further propagation. It is important to notice here that, in each step, the stereo points are regarded as hard constraints for building a new patch. The whole process ends with the last seed.

³This algorithm was presented in the ECCV conference [54]. Here, we describe it based on our proposed patchwork concept and reconstruction.

A. Patch Creation and New Seed Selection

Given a seed (the selection process is described later), we set a local coordinate based on the seed normal and run a min-cut optimization to build an optimal patch. This patch creation remains the same as the previous patch-wise carving as described in Section IV-B.

To continue the propagation, new seeds are created from this patch. These new seeds are selected in order to maximize their reliability because they are the anchor points of future patches. The location of the selected new seeds is determined from several criteria.

- 1) *Patch quality*: First of all, the value of the functional $F = \mathcal{F}(\mathcal{P})$ indicates the confidence of the optimal patch. If the confidence is too low (*i.e.*, \mathcal{F} is too high), the surface patch is discarded and no seed is created.
- 2) *Match quality*: A point with a high ZNCC value Z is more likely to provide a robust starting point for further propagation.
- 3) *Surface regularity*: A singular point does not represent accurate properties of the patch. With the principal curvatures κ_1 and κ_2 , points with high curvature $K = \kappa_1^2 + \kappa_2^2$ are therefore to be avoided.
- 4) *Propagation efficiency*: To ensure a faster propagation, distant points are preferred. This relies on the distance D between the patch center and the potential new seeds.

A value Λ is computed for each potential location of a new seed to represent its appropriateness relative to these objectives:

$$\Lambda = \frac{Z^{\omega(Z)} \cdot D^{\omega(D)}}{F^{\omega(F)} \cdot K^{\omega(K)}} \quad (16)$$

where $\omega(\cdot)$ are non-negative weights to balance the different criteria. From our experiments, $\omega(Z) = \omega(D) = \omega(F) = \omega(K) = 1$ yields satisfying results. Exploring the possibilities offered by these weights is kept as future work.

The number of new seeds created is inspired by the triangle mesh configuration. From the Euler property, the average number of neighbors of a vertex is 6 and the average angular distance between two neighbors is $\frac{\pi}{3}$. Thus, the directions of the new seeds in relation to the patch center are selected so that the angular distance between two neighboring seeds lies in $[\frac{2\pi}{5}, \frac{2\pi}{7}]$. In each direction, the location \mathbf{s}' with the highest Λ is selected and the normal \mathbf{n}' at \mathbf{s}' is computed and attached to form a new seed.

B. Selection of the Next Seed

To select the next seed (\mathbf{s}, \mathbf{n}) for propagation, we define a criterion Π to evaluate how “good for propagation” a seed is. With this criterion, we follow a classical best-first strategy to ensure

that the most reliable seed is picked each time. This choice drives the propagation directly because it indicates where the growing regions are.

First of all, the initial seeds (*i.e.*, the input 3D points) are regarded as reliable 3D points on the surface. Therefore, they are always selected before the seeds generated from the patches. The algorithm ends when there is no seed left in the list.

Selection Criterion for the Input 3D Points: Depending on how the input 3D points are obtained, an estimation of their accuracy may be available. In this case, the input points are ranked in order to pick the most accurate ones first. For instance, for the normal estimation we propose in Appendix I, we can estimate the normal precision from the local planarity of the point set. This corresponds to the ratio between the eigenvalue orthogonal to the plane (λ_3) and the smallest one within the plane (λ_2). Thus: $\Pi = \frac{\lambda_2}{\lambda_3}$.

Selection Criterion for Generated Seeds: For a generated seed, we use the ZNCC correlation score Z from its two most front-facing cameras, since a strong match gives a high confidence level. This strategy ensures that the surface grows from the part that is more likely to be precise and robust. Thus: $\Pi = Z$. If the criterion is computed from occluded cameras, the local textures in both images will not match and the ZNCC value is low. Therefore, a seed without occlusion is processed before a seed with occlusion. The occluded parts “wait” until other parts are reconstructed. The visibility of the processed seed is classically determined by the current propagated surface using ray tracing. The ordering scheme according to the matching score ensures that a seed is processed only when no better one is available. In all our experiments, this led to a correct visibility estimation, allowing for manipulating objects with strong occlusion (see Figure 11).

C. Summary and Discussion

This propagation algorithm reconstructs the surface of scene objects from a set of 3D points, which can be robustly computed. These points are the information sources from which the surface is grown along the tangent directions. Meanwhile, the images are used to guide the propagation, fill the holes and add high-resolution geometric details. Compared with patch-wise carving, which employs a low-resolution voxel space in the previous section, this propagation leads to a relatively faster reconstruction, since the 3D points provide accurate locations on several parts of the surface. The counterpart is that a side technique is required to obtain these 3D points.

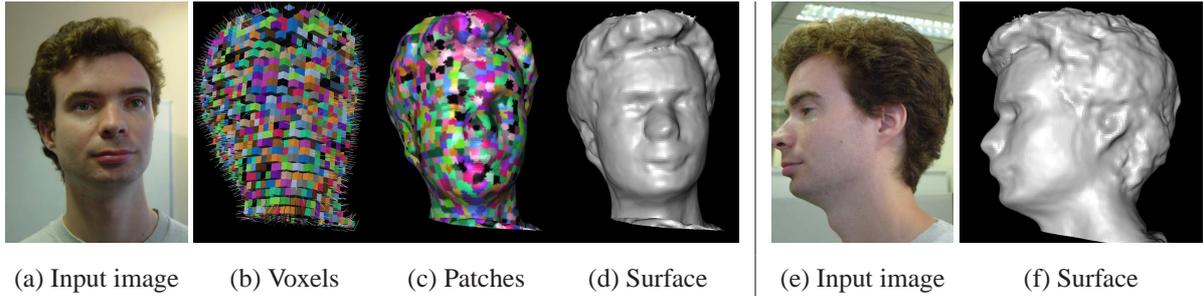


Fig. 5. Head reconstruction using our carving approach. This example demonstrates the ability of our approach to deal with non-Lambertian surfaces (skin and hair). The voxel resolution (b) is 32^3 ; this is one order coarser than traditional carving techniques. Although the process has been done patch by patch (c), no seam is visible on the final result (d,f).

VI. RESULTS AND DISCUSSION

A. Patch-wise Carving

Implementation Details: The presented results use real photographs shot with a hand-held consumer-grade camera. The camera geometry is computed by the quasi-dense approach in [55], [58]. The window size to compute the ZNCC is 11×11 . The patch size is set to twice the voxel size to ensure a sufficient overlap. To avoid grazing views, we ignore cameras whose angle to the normal is greater than $\frac{\pi}{3}$. The distance field D has a resolution 4^3 times finer than the voxel grid. The min-cut process is run on a grid of resolution of 15^3 . We stop the normal estimations after $k_{\max} = 4$ iterations. For example, for the owl sequence in Figure 7, we perform 3054 graph-cut optimizations and examine 1897 voxels. This corresponds to an average of 1.6 graph cuts to estimate the normal. In Equation (15), $g(\|\nabla I\|) = \max(0, 1 - \|\nabla I\|/16)$ with $I \in [0; 255]$. We use the min-cut code of the Boost library⁴ which leads to a computation time of between 20 min (the owl in Figure 7) and 45 min (the gargoyle in Figure 6).

As future work, we hope to try an implementation [59] that should run faster on our small graphs. We initialize all the sequences with the visual hull. Bounding boxes produce equivalent results, but in a longer time depending on the box size (more voxels have to be processed).

▷ The head sequence (Fig. 5) shows that non-Lambertian surfaces can be reconstructed by patch-wise carving. There are 21 views at 480×640 . The voxel space is 32^3 . It is important to notice that this kind of sequence is typically difficult for traditional space carving methods because the image appearance significantly changes from one view to another; skin and hair are well-known to be highly non-Lambertian.

⁴<http://www.boost.org>

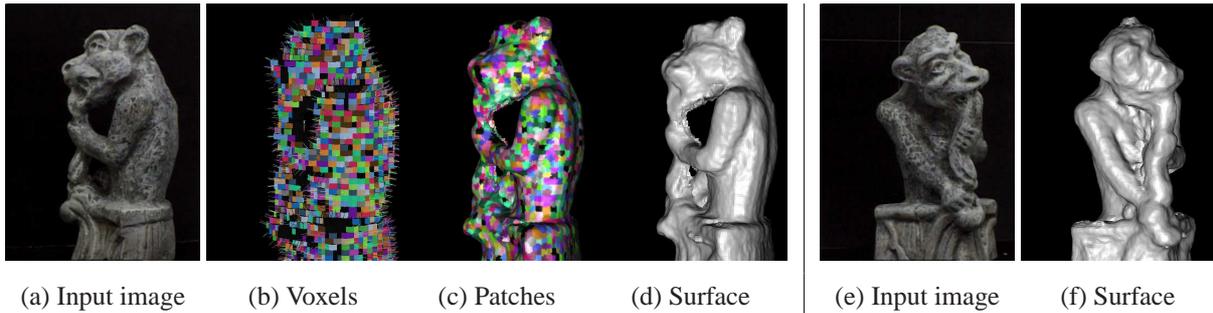


Fig. 6. Gargoyle reconstruction using our carving approach. This model has two holes (above and under its arm). The carving step correctly recovers this topology (b). Then, the patches (c) produce a fine surface (d,f). The back of the stick (d) is not as accurate as the rest of the model because the gargoyle’s body occludes most of the cameras. Only views with a grazing angle can be used for this part of the model.

The role of each step of the algorithm appears clearly. At a coarse level, our algorithm behaves as a carving technique (Figure 5b) except that we use the patch consistency as the carving criterion. At a fine level, minimal cuts build the patches that capture the fine geometry within the voxels (Figure 5c). These patches are stitched together to produce the final surface. As predicted, our stitching scheme achieves a seamless and continuous result (Figure 5d,f).

▷ The gargoyle sequence (Fig. 6) shows that a non-spherical topology can be reconstructed by patch-wise carving. There are 16 views at 720×486 although the gargoyle only covers an area of about 200×400 . This demonstrates the performance of our technique on low-resolution data. The voxel space is $25 \times 50 \times 25$. We encourage the reader to compare this result with the one obtained by existing techniques [9], [10] that work from the same images. The precision is dramatically improved.

▷ The owl sequence (Fig. 7) demonstrates the performance of the technique on concavities and thin sharp features. We correctly reconstruct the ears whereas many existing techniques would have difficulties because of the curvatures of the ears. There are 37 views at 600×800 . The voxel resolution is $25 \times 50 \times 25$.

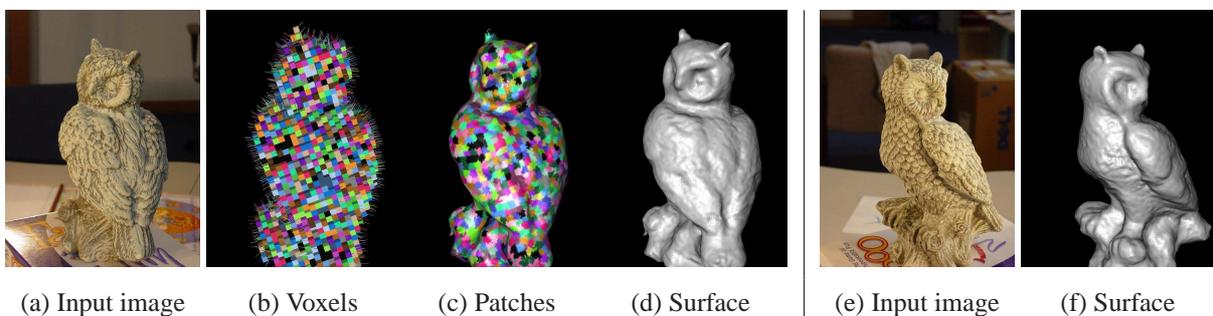


Fig. 7. Owl reconstruction using our carving approach. Our technique correctly recovers the geometry even within deep concavities. The thin and sharp ears are also accurately reconstructed. To our knowledge, few existing methods attain such precision on these kinds of features.

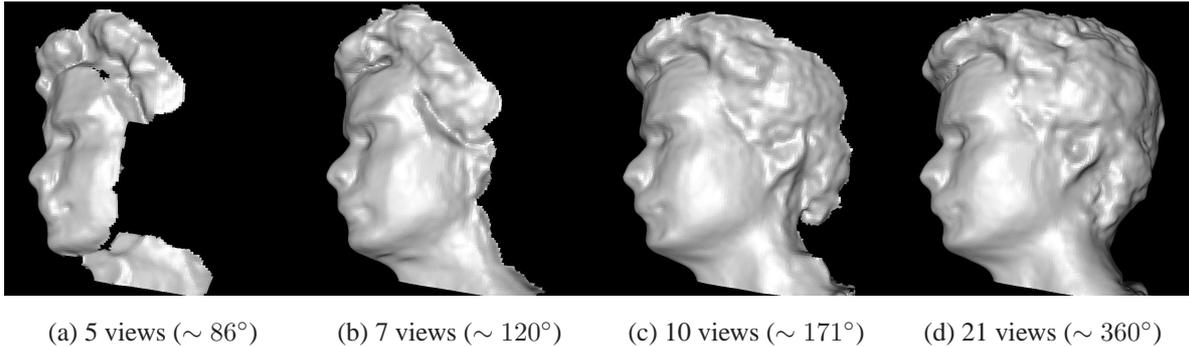


Fig. 8. Partial reconstruction. The 21 input images form a rough circle around the head. To demonstrate that the algorithm handles both partial and complete shapes, we have used only a subset of these images: 5 (a), 7 (b), 10 (c) and all views (d).

Partial versus Complete Reconstruction: To demonstrate the capabilities of our approach to handle both partial and complete reconstruction, we hid the back of the head by omitting some images. Without any change in the algorithm, the front part is reconstructed as an open surface (Figure 8a,b,c). When all the images are available, the technique naturally produces a closed surface (Figure 8d). Note that the geometry of the visible part is stable and independent of the setup. The Φ function makes the border clean (cf. Section III-D.2).

B. Patch-wise Propagation

▷ The two faces (Figures 9 and 10) illustrate the accuracy of our algorithm and its behavior with two different sampling densities. Figure 9 has a rather homogeneous point density (there is no large holes) whereas Figure 10 contains two large holes in the cheeks due to the lack of texture at this location. The point cloud is also denser in the first case than in the second one. Nonetheless, our technique achieves convincing results on both configurations, demonstrating its versatility. Our algorithm deals efficiently with different point densities, and the propagation strategy fills in holes with a consistent detailed surface. As future work, we want to quantify the influence of the point density and accuracy on the precision of the recovered surface.

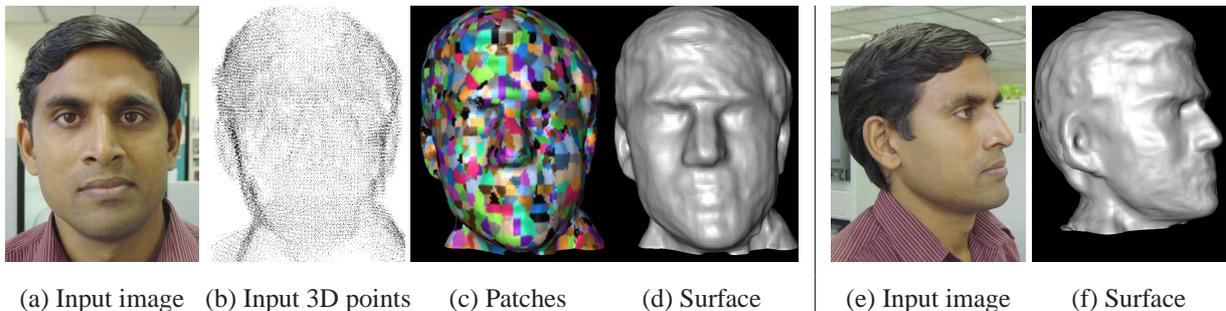
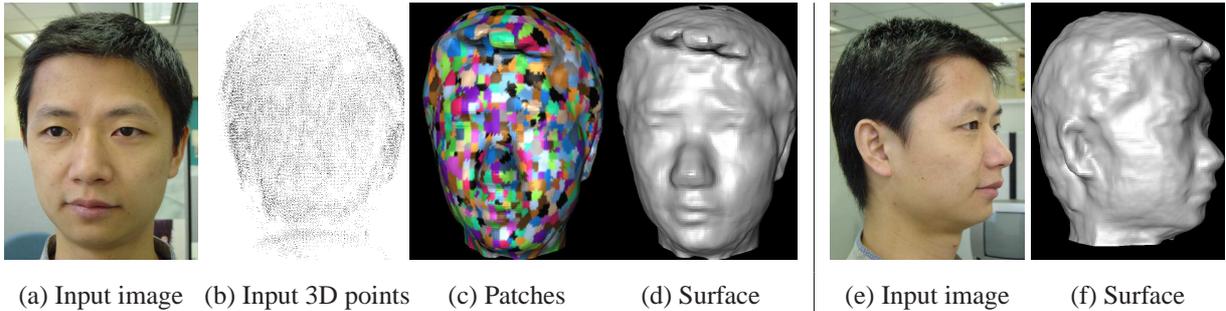


Fig. 9. Head reconstruction using our propagation approach. The input point cloud (b) is rather uniform on this model. Using the reliable input 3D points, small details (on the eyes, the nose and the ears) are obtained.



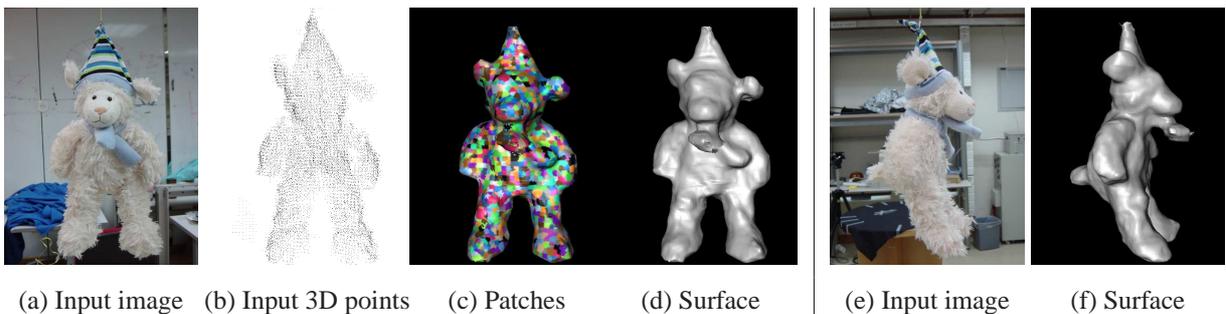
(a) Input image (b) Input 3D points (c) Patches (d) Surface (e) Input image (f) Surface
 Fig. 10. Head reconstruction using our propagation approach. The input point cloud (b) that we have extracted using an image-based approach [55] has two large holes in the areas of the cheeks, because these two regions have almost no texture in the input images (a,e). In addition, the point density is also coarser when compared to the first one in Figure 9. However, the proposed algorithm produces a surface with an equivalent quality.

▷ The toy example (Figure 11) illustrates the correctness and robustness of the patch-wise propagation. Fur is traditionally difficult in surface reconstruction because its appearance is strongly view dependent. This model also contains large occlusions (the legs and arms are hidden in several images). Despite these difficulties, our algorithm performs well: The geometry is accurately recovered and occlusions are correctly handled. There are 22 images with a resolution 480×640 .

▷ The bas-relief (Figure 12) is a typical scenario in which a technique dedicated to a closed surface would fail. This highlights the advantage of handling closed and open surfaces equivalently. This model is made of polished metal. Most of the geometry is correctly recovered, but there are two small artifacts. Such a borderline object is of high interest since it delineates the abilities of our technique. To handle more complex materials, one would have to implement more robust but also more computationally expensive consistency estimators such as [7], [19]. There are 23 images with the resolution 600×800 .

C. Comparison

In Figure 13, we use the same image sequence as in Figure 5 to compare our two algorithms with a level-set method [18] and Space Carving [9]. The first point is that Space Carving fails to capture a good geometry because of the non-Lambertian aspect of the head. To avoid over-carving, we had to sacrifice accuracy. Then, our two methods recover more details than level sets although the overall shape is smooth and thus should suit level sets. Note that our methods



(a) Input image (b) Input 3D points (c) Patches (d) Surface (e) Input image (f) Surface
 Fig. 11. Toy reconstruction using our propagation approach. This is a difficult example because of the fur and of the occlusions. Nonetheless, our algorithm yields a satisfactory result.

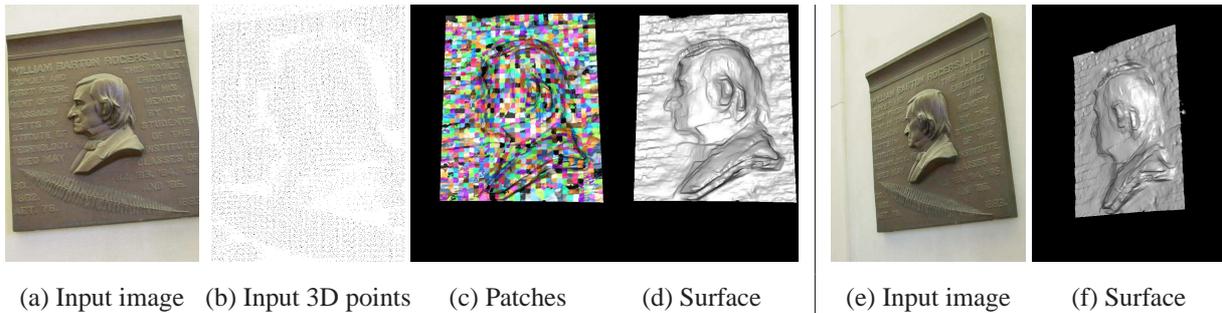


Fig. 12. Bas-relief reconstruction with patch-wise propagation. This situation underlines the advantage of being able to cope with open surfaces since obviously no information is available for the back part. The acquired geometry is mostly correct except in two regions: There are artifacts on the top of the head and the bottom of the bust. This means that this shiny metal is just at the borderline of the material that our algorithm can cope with. To better handle such highly non-Lambertian materials, one would have to use dedicated and more computationally costly consistency estimators [7], [19].

and the level-set technique work from the same image sequences and the same input 3D points. Then, between carving and propagation, the results look equivalent. The propagation is slightly more precise in most cases (see the nose and the mouth) with the help of the 3D points, except on regions where the visibility is difficult to estimate (*e.g.*, near the face-hair boundary). This advocates integrating both approaches, which is undoubtedly promising future work. From a performance point of view, the propagation is about 30% faster (about 20 min instead of 30 min) since the input 3D points directly indicate the areas to focus on. Nonetheless, the carving technique has the advantage of being usable even if 3D points are not available.

D. Role of the Resolution

We have compared several results from different settings of the distance field resolution and of the size of the graphs used for the optimizations (Figure 14). This confirms that the distance field resolution is directly linked to the details that can be recovered: A finer distance field makes it possible to represent finer details. These results also underline the importance of the spatial dimension of the patches. If the size of the graphs is kept constant while the resolution increases, the patches become smaller and smaller. First the precision increases but at some point, the results degrade. This behavior shows that there is a resolution beyond which the min-cut technique ceases to extract further information. Beyond this “limit resolution”, the patches rely comparatively on less information since they become smaller and no more information is gained from the finer resolution. Hence, the patches cannot be made infinitely small, and there is a bound to the complexity gain that can be achieved. On the other end, when the patches are too large, several advantages (*i.e.*, sharp normal or complex topologies) of our patchwork reconstruction are lost.

This experiment opens several promising research avenues. First, characterizing and comparing the “limit” resolution for different optimization techniques (*e.g.*, minimal cuts, level sets) would give valuable insights into their relative efficiency. A careful examination of these

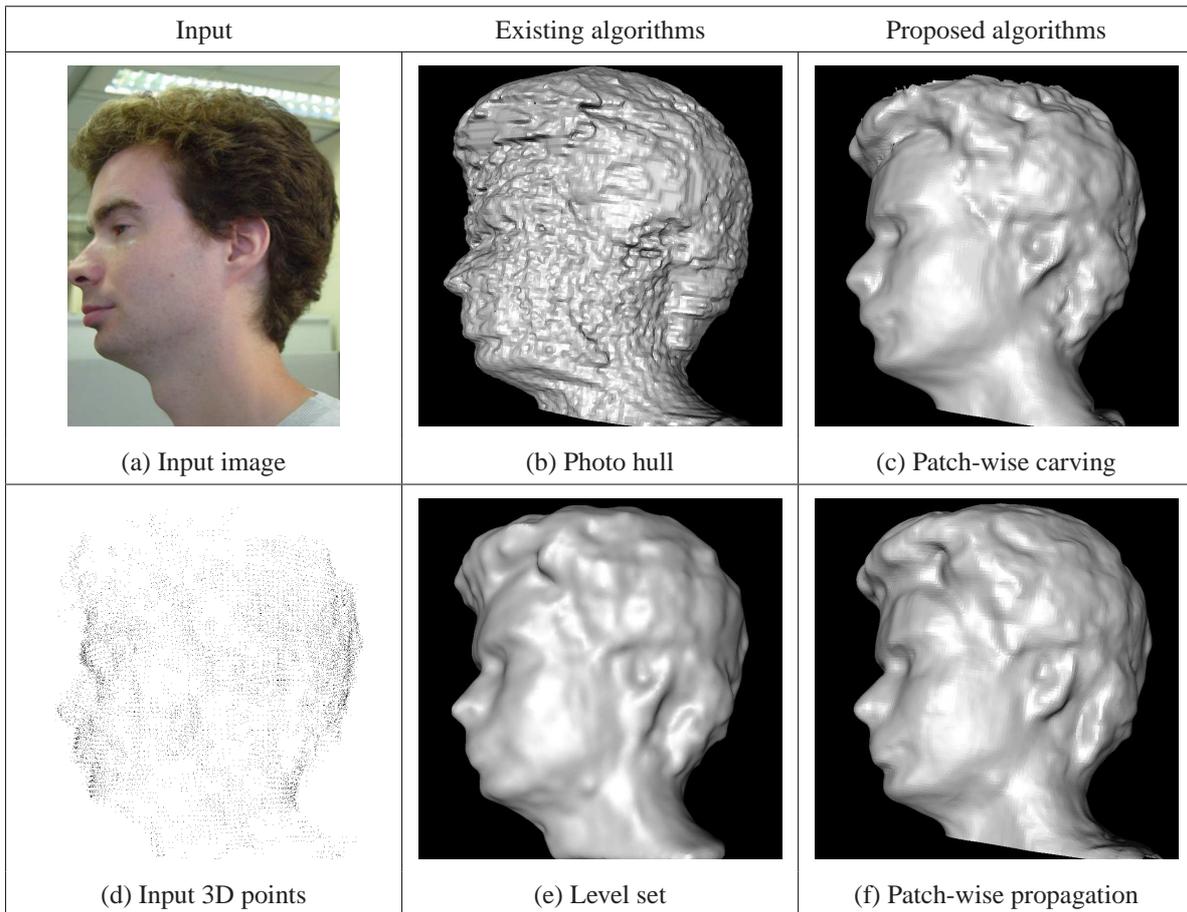


Fig. 13. Comparison. (a) One of the input images (b) Space Carving [9] fails to build a satisfying reconstruction due to the non-Lambertian surfaces. To achieve a fair comparison without aliasing, the voxel volume has been triangulated using Marching Cubes [42]. (c) Patch-wise carving and (f) propagation build reasonable results by patches that consider both image information and regularity. (e) The level-set technique [18] builds a satisfying geometry but is less detailed compared with our techniques (c,f); *e.g.*, observe the chin, the eyes and the forehead. (d) The input 3D points used in (e) and (f).

results also suggests that adjusting the patch size to the local characteristics of the surface would further enhance the accuracy of the final result (observe the lower lip on the bottom row of Figure 14 where smaller patches better match the high curvature). Such a multi-resolution approach could refine the most curved areas of a surface.

E. Quantitative Analysis

Table II shows typical values for memory usage and running times for our algorithm on an Intel PIII-1.9GHz. These numbers correspond to the experiment shown in Figure 14. This validates our space complexity analysis: The required storage for the optimization does not depend on the object size. Although the patch stitching and the surface extraction require more memory, we advocate that these steps are scalable since they involve only simple and local memory accesses that can be handled “out of the core” *i.e.*, by storing the data structure on the hard drive and performing only local updates in the memory. Such an approach would be non-trivial, if not impossible, for the optimization step because the involved algorithms need global access to the data.

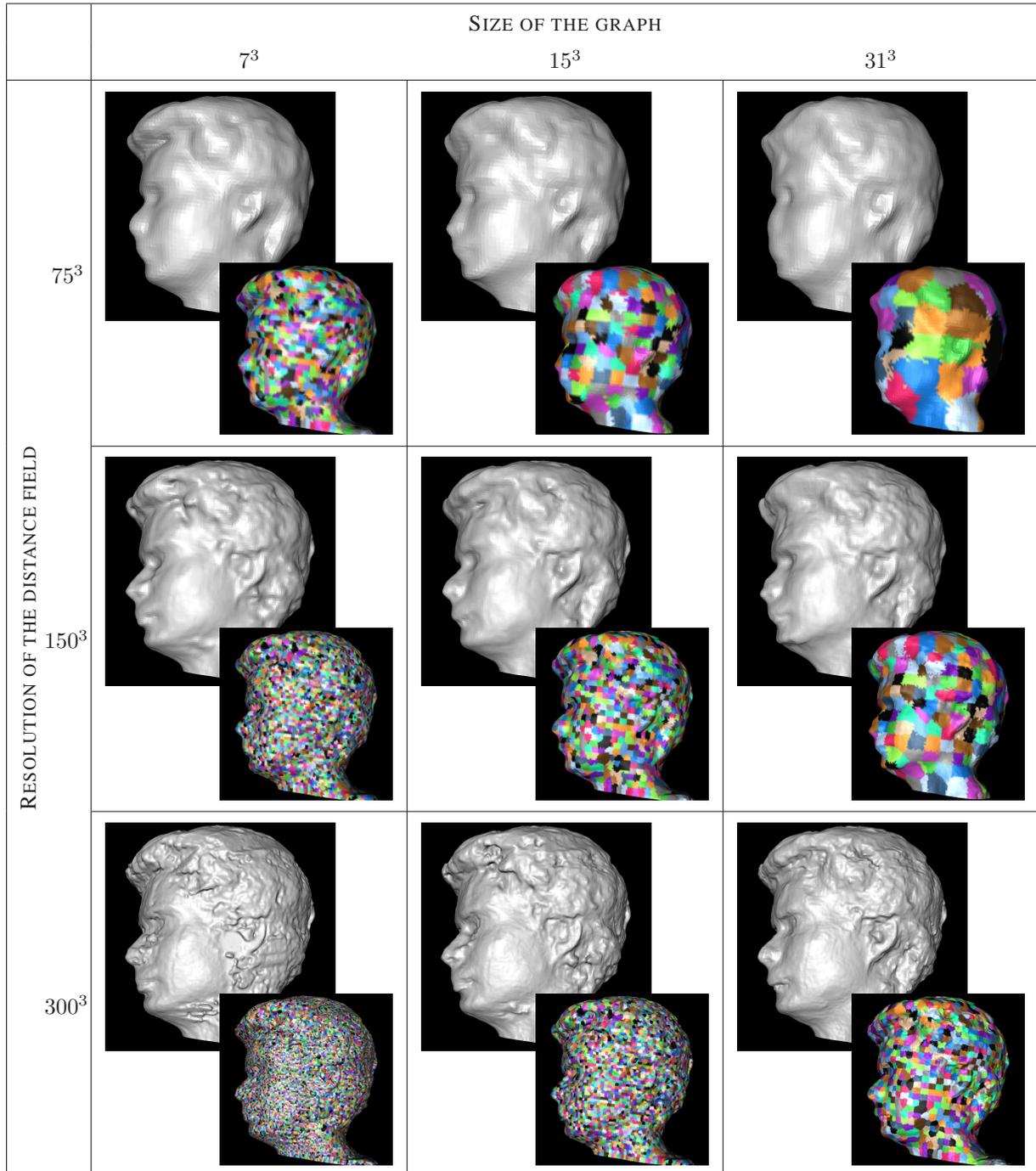


Fig. 14. The effect of the resolution of the distance field and of the graph size. We use the carving algorithm. Increasing the distance field resolution allows for capturing more details. When the graph size is kept constant, the corresponding patches become smaller. First the results improve (from the first row to the second one) and then they degrade (the first and second columns, from the second row to the third one). Note also that very large patches perform poorly (top right result).

QUANTITATIVE COMPARISON AMONG DIFFERENT RESOLUTIONS

DISTANCE FIELD RES.	GRAPH SIZE			DISTANCE FIELD RES.	GRAPH SIZE		
	7^3	15^3	31^3		7^3	15^3	31^3
75^3	229s (2785)	297s (559)	520s (104)	75^3	1M (105)	2M (106)	15M (119)
150^3	1010s (11876)	1455s (2772)	2406s (554)	150^3	1M (121)	2M (122)	15M (134)
300^3	3960s (45917)	6483s (11643)	12458s (2747)	300^3	1M (238)	2M (239)	15M (251)

(a) Running time (number of patches)

(b) Memory used by patch optimization (total space)

To validate the time complexity analysis of Section III-B, we first remark that the meaningful size of the problem in terms of complexity is the area of the surface to reconstruct relative to the targeted resolution. Formally speaking, the problem size is on the order of $\mathcal{O}(a_S/\Delta_{DF}^2)$, where a_S is the area of the surface to reconstruct and Δ_{DF} is the distance field discretization step. Thus to measure the influence of an increasing problem size, we can act upon a_S (*i.e.*, using a larger object) or upon Δ_{DF} (*i.e.*, using a finer distance field). Varying Δ_{DF} coherently uses the same object throughout the measure. We always use graphs of size 15^3 . The ratio a_P/Δ_{DF}^2 is therefore constant (with a_P is the patch area). Thus, the number of patches, $\eta = \mathcal{O}(a_S/a_P)$, is on the order of $\mathcal{O}(\Delta_{DF}^{-2})$. From our analysis, we expect a complexity proportional to η (cf. Table I) or equivalently quadratic in the distance field resolution $\frac{1}{\Delta_{DF}}$. This is the best possible complexity since it is linear to the problem size because $\eta = \mathcal{O}(a_S/\Delta_{DF}^2)$.

Figure 15 summarizes our measures. Fitting a polynomial curve gives a complexity of $\mathcal{O}(\Delta_{DF}^{-2.16})$. We obtain a nearly optimal result. The overhead stems from the fact that our carving algorithm needs to “dig through” the concavities to “reach” the actual surface. These steps introduce a volumetric component into the complexity. This is confirmed by the number of built patches (including the ones discarded by the carving process), which is also slightly higher than the quadratic, on the order of $\mathcal{O}(\Delta_{DF}^{-2.07})$. This result demonstrates the scalability of our approach. To our knowledge, our patchwork representation method is the first reconstruction technique that is proven to have a linear complexity that is practically confirmed on a real example.

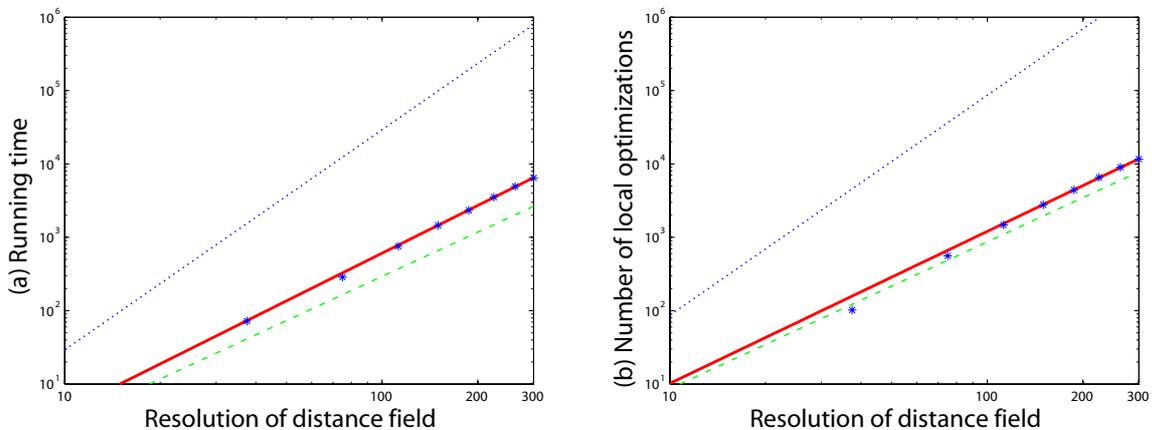


Fig. 15. We measure the running time and the number of local optimizations in terms of the resolution of distance field (from 38^3 to 300^3). Fitting a polynomial curve gives a running time in the order of $\mathcal{O}(\Delta_{DF}^{-2.16})$ and a number of built patches in the order of $\mathcal{O}(\Delta_{DF}^{-2.07})$. They are close to the optimal solution $\mathcal{O}(\Delta_{DF}^{-2})$ (the dashed green lines); the dotted blue lines show $\mathcal{O}(\Delta_{DF}^{-3})$ for comparison.

VII. CONCLUSION

We have presented a new patchwork representation method. It consists of a collection of small surface pieces that are progressively reconstructed and stitched together. It can represent both complete (closed) and partial (open) surfaces while being able to recover complex topologies. The achieved results are accurate, even on sharp features and concavities.

From a theoretical point of view, we have introduced a new mathematical formulation of the *a priori* smoothness of the objects. This formulation is purely local. *i.e.*, it involves only a patch whereas the existing technique relies on the whole surface. This local prior enables complex shapes by alleviating the parameterization problem inherent in some global formulations. The relationship with a global approach is rigorously characterized for a number of optimization techniques. We describe an efficient way to stitch the patches together that guarantees the continuity of the produced surface. Furthermore, our patch representation method is proven to induce an optimization process that requires a constant amount of memory that is independent of the object size. The temporal complexity is demonstrated to be optimal. These two theoretical results on the complexity are backed by actual measurements.

We have introduced two algorithms based on the patchwork concept. The first one combines a carving strategy with a min-cut optimization to retrieve the object's geometry. The second algorithm is specially designed to exploit reliable 3D points that are available from a number of configurations. Both are demonstrated on real examples. The reconstructed surfaces compare favorably with existing techniques such as Space Carving and level sets.

The patchwork approach strikes a balance between purely local techniques (*e.g.*, Space Carving) and global optimization methods, such as min-cuts and level sets. The patches aggregate a sufficient amount of data to be robust and precise while avoiding the manipulation of the whole surface that inherently makes the process less flexible. As a surface representation, the patchwork greatly broadens the range of objects recoverable by minimal cuts while preserving their key advantages: accuracy and convergence. We have demonstrated the patchwork concept with a min-cut optimization. Nonetheless, most of our results potentially extend to any optimization technique. As a consequence, we believe that the patchwork concept makes a significant contribution: Any optimization technique can enjoy enhanced scalability and flexibility simply by using patches to represent the object surface.

Future Work: Throughout this paper, we have mentioned several avenues for future research that we summarize here. Testing more robust consistency estimators would certainly further enhance the capacity of our algorithms. In some situations, it may be difficult to get reliable 3D points. Nonetheless, such “no-point” configurations are rare, thus combining our two algorithms into a single one is likely to improve the performance. A valuable extension is

to develop an out-of-core stitching process to work on very large and/or very detailed objects that would be impossible to acquire with other techniques. Finally, we have used the patches with min-cuts but other methods such as level sets would be interesting to investigate.

APPENDIX I: NORMAL ESTIMATION FROM 3D STEREO POINTS

As described in Section V, given a set of quasi-dense 3D points [55], we can estimate a normal for each of these points. The surface normals at these points are estimated to form the initial seeds. For each 3D point, \mathbf{p}_j , the surface normal, \mathbf{n}_j , is provided by the symmetric 3×3 positive semi-definite matrix, $\sum_{\mathbf{y} \in \mathcal{B}_r(\mathbf{p}_j) \cap \mathcal{P}} (\mathbf{y} - \mathbf{p}_j) \otimes (\mathbf{y} - \mathbf{p}_j)$, where $\mathcal{B}_r(\mathbf{p}_j)$ denotes a ball with radius r . Among the eigenvectors, $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, respectively associated to the eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$, we choose \mathbf{n}_j to be either \mathbf{v}_3 or $-\mathbf{v}_3$. The sign depends on the cameras used to reconstruct \mathbf{p}_j .

$\mathcal{B}_r(\mathbf{p}_j) \cap \mathcal{P}$ may contain very few points, baffling the orientation estimation. In dense regions, a large radius results in an over-smoothed estimation whereas a small radius makes the estimation sensitive to noise. Therefore, r is defined as a function of \mathbf{p}_j : in dense regions, r is fixed to a reference value, r_{dense} , representing the minimum scale. In the diluted regions, the radius is increased so that $\mathcal{B}_r(\mathbf{p}_j)$ contains at least k 3D stereo points. From our experiments, a good compromise is to define r_{dense} to be the radius of local patches and k to be 15-20.

ACKNOWLEDGMENT

This work was supported by the Hong Kong RGC Grant HKUST 6182/04E and HKUST 6190/05E. Sylvain Paris acknowledges a Eurodoc grant from the French ‘‘Région Rhône-Alpes’’ for his visit at the Hong Kong University of Science and Technology. His research at the Massachusetts Institute of Technology was partially supported by a Lavoisier Fellowship from the ‘‘French Ministère des Affaires Étrangères.’’ The authors would like to thank Kyros Kutulakos for the gargoyle images and Brian Curless for the interesting discussion about his work.

REFERENCES

- [1] A. Laurentini, ‘‘The visual hull concept for silhouette-based image understanding,’’ *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, pp. 150–162, February 1994.
- [2] E. Boyer and J.-S. Franco, ‘‘A hybrid approach for computing visual hulls of complex objects,’’ in *Proceedings of the Computer Vision and Pattern Recognition Conference*, vol. 1, June 2003, pp. 695–701.
- [3] R. Cipolla and K. K. Wong, ‘‘Reconstruction of sculpture from its profiles with unknown camera positions,’’ *IEEE Transactions on Image Processing*, vol. 13, no. 3, pp. 381–389, March 2004.
- [4] S. Sullivan and J. Ponce, ‘‘Automatic model construction and pose estimation from photographs using triangular splines,’’ *IEEE Transactions Pattern Analysis Machine Intelligence*, vol. 20, no. 10, pp. 1091–1097, 1998.
- [5] W. Matusik, C. Buehler, and L. McMillan, ‘‘Polyhedral visual hulls for real-time rendering,’’ in *Proceedings of the Eurographics Workshop on Rendering*, 2001.

- [6] J. Isidoro and S. Sclaroff, “Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints,” in *Proceedings of the International Conference on Computer Vision*. IEEE, 2003, pp. 1335–1342.
- [7] C. H. Esteban and F. Schmitt, “Silhouette and stereo fusion for 3d object modeling,” *Computer Vision and Image Understanding*, vol. 96, no. 3, pp. 367–392, December 2004.
- [8] S. M. Seitz and C. R. Dyer, “Photorealistic scene reconstruction by voxel coloring,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*. IEEE, 1997, pp. 1067–1073.
- [9] K. N. Kutulakos and S. M. Seitz, “A theory of shape by space carving,” *International Journal of Computer Vision*, vol. 38, no. 3, pp. 199–218, 2000.
- [10] K. N. Kutulakos, “Approximate N-view stereo,” in *Proceedings of the European Conference on Computer Vision*, 2000, pp. 67–83.
- [11] R. Szeliski and P. Golland, “Stereo matching with transparency and matting,” *International Journal of Computer Vision*, vol. 32, no. 1, pp. 45–61, 1999.
- [12] J. S. de Bonet and P. Viola, “Poxels: Probabilistic voxelized volume reconstruction,” in *Proceedings of the International Conference on Computer Vision*. IEEE, 1999.
- [13] A. Broadhurst, T. Drummond, and R. Cipolla, “A probabilistic framework for space carving,” in *Proceedings of the International Conference on Computer Vision*. IEEE, July 2001, pp. 388–393.
- [14] G. G. Slabaugh, W. B. Culbertson, T. Malzbender, and R. W. Schafer, “A survey of methods for volumetric scene reconstruction from photographs,” in *Proceedings of the International Workshop on Volume Graphics*, June 2001.
- [15] S. Osher and J. A. Sethian, “Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations,” *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.
- [16] O. Faugeras and R. Keriven, “Complete dense stereovision using level set methods,” *IEEE Transactions on Image Processing*, vol. 7, no. 3, 1998.
- [17] H. Jin, A. J. Yezzi, and S. Soatto, “Region-based segmentation on evolving surfaces with application to 3D reconstruction of shape and piecewise constant radiance,” in *Proceedings of the European Conference on Computer Vision*, 2004.
- [18] M. Lhuillier and L. Quan, “Surface reconstruction by integrating 3D and 2D data of multiple views,” in *Proceedings of the International Conference on Computer Vision*. IEEE, October 2003.
- [19] H. Jin, S. Soatto, and A. J. Yezzi, “Multi-view stereo beyond lambert,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2003, pp. 171–178 vol.1.
- [20] D. Adalsteinsson and J. A. Sethian, “A fast level set method for propagating interfaces,” *Journal of Computational Physics*, vol. 118, pp. 269–277, 1995.
- [21] D. Terzopoulos, A. Witkin, and M. Kass, “Constraints on deformable models: Recovering 3D shape and nonrigid motion,” *Artificial Intelligence*, vol. 36, no. 1, pp. 91–123, 1988.
- [22] S. Roy and I. J. Cox, “A maximum-flow formulation of the n-camera stereo correspondence problem,” in *Proceedings of the International Conference on Computer Vision*. IEEE, January 1998, pp. 492–499.
- [23] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993, ISBN 013617549X.
- [24] H. Ishikawa, “Exact optimization for markov random fields with convex priors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1333–1336, October 2003.
- [25] S. Paris, F. Sillion, and L. Quan, “A surface reconstruction method using global graph cut optimization,” *International Journal of Computer Vision*, vol. 66, no. 2, pp. 141–161, February 2006.
- [26] D. Kirsanov and S. J. Gortler, “A discrete global minimization algorithm for continuous variational problems,” Harvard Computer Science, Tech. Rep. TR-14-04, July 2004.
- [27] C. Buehler, S. Gortler, M. Cohen, and L. McMillan, “Minimal surfaces for stereo,” in *Proceedings of the European Conference on Computer Vision*, 2002.
- [28] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

- [29] V. Kolmogorov and R. Zabih, “What energy functions can be minimized via graph cuts?” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, February 2004.
- [30] V. Kolmogorov and R. Zabih, “Multi-camera scene reconstruction via graph cuts,” in *Proceedings of the European Conference on Computer Vision*, May 2002.
- [31] M. Lin and C. Tomasi, “Surfaces with occlusions from layered stereo,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 710–717, August 2004.
- [32] M. Bleyer and M. Gelautz, “Graph-based surface reconstruction from stereo pairs using image segmentation,” in *Proceedings of SPIE conference*, 2005.
- [33] Y. Boykov and V. Kolmogorov, “Computing geodesics and minimal surfaces via graph cuts,” in *Proceedings of the International Conference on Computer Vision*, vol. 1. IEEE, October 2003, pp. 26–33.
- [34] G. Vogiatzis, P. Torr, and R. Cipolla, “Multi-view stereo via volumetric graph-cuts,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2005.
- [35] A. Blake and A. Zisserman, *Visual reconstruction*. MIT Press, 1987, ISBN:0-262-02271-0.
- [36] P. Fua, “From multiple stereo views to multiple 3-D surfaces,” *International Journal of Computer Vision*, vol. 24, no. 1, pp. 19–35, 1997.
- [37] P. J. Narayanan, P. Rander, and T. Kanade, “Constructing virtual worlds using dense stereo,” in *Proceedings of the International Conference on Computer Vision*, 1998.
- [38] W. Hoff and N. Ahuja, “Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 121–136, 1989.
- [39] R. L. Carceroni and K. N. Kutulakos, “Multi-view scene capture by surfel sampling: From video streams to non-rigid 3d motion, shape & reflectance,” in *Proceedings of the International Conference on Computer Vision*, vol. 2. IEEE, 2001.
- [40] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel, “Multi-level partition of unity implicits,” *Transactions on Graphics*, vol. 22, no. 3, pp. 463–470, 2003, proceedings of the SIGGRAPH conference.
- [41] B. Curless and M. Levoy, “A volumetric method for building complex models from range images,” in *Proceedings of the SIGGRAPH conference*. ACM, 1996.
- [42] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3D surface construction algorithm,” in *Proceedings of the SIGGRAPH conference*. ACM, 1987, pp. 163–169.
- [43] V. Kolmogorov, R. Zabih, and S. Gortler, “Generalized multi-camera scene reconstruction using graph cuts,” in *Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, July 2003.
- [44] B. V. Cherkassky and A. V. Goldberg, “On implementing the push-relabel method for the maximum flow problem,” *Algorithmica*, vol. 19, no. 4, pp. 390–410, 1997.
- [45] S. Roy, “Stereo without epipolar lines: A maximum-flow formulation,” *International Journal of Computer Vision*, vol. 34, no. 2/3, pp. 147–162, August 1999.
- [46] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” *IEEE Transactions Pattern Analysis Machine Intelligence*, vol. 12, no. 7, pp. 629–639, July 1990.
- [47] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “High-quality video view interpolation using a layered representation,” *ACM Transactions on Graphics*, vol. 23, no. 3, July 2004, proceedings of the SIGGRAPH conference.
- [48] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, “Robust anisotropic diffusion,” *Transactions on Image Processing*, vol. 7, no. 3, pp. 421–432, March 1998.
- [49] P. Le Tallec, *Computational Mechanics Advances*. North Holland, 1994, vol. 1, no. 2, ch. Domain Decomposition Methods in Computational Mechanics, pp. 123–217.
- [50] G. Zeng, S. Paris, L. Quan, and F. Sillion, “Progressive surface reconstruction from images using a local prior,” in *Proceedings of the International Conference on Computer Vision*, 2005.
- [51] W. B. Culbertson, T. Malzbender, and G. G. Slabaugh, “Generalized voxel coloring,” in *Proceedings of the International*

- Workshop on Vision Algorithms*, ser. Lecture Notes on Computer Science. Springer Verlag, September 1999, pp. 100–115.
- [52] H. Jin, A. J. Yezzi, Y. Tsai, L. Chen, and S. Soatto, “Estimation of 3D surface shape and smooth radiance from 2D images: a level set approach,” *Journal of Scientific Computing*, vol. 19, no. 1-3, pp. 267–292, 2003.
- [53] A. Sharf, M. Alexa, and D. Cohen-Or, “Context-based surface completion,” *ACM Transactions on Graphics*, vol. 23, no. 3, July 2004, proceedings of the SIGGRAPH conference.
- [54] G. Zeng, S. Paris, L. Quan, and M. Lhuillier, “Surface reconstruction by propagating 3d stereo data in multiple 2d images,” in *Proceedings of the European Conference on Computer Vision*, 2004.
- [55] M. Lhuillier and L. Quan, “A quasi-dense approach to surface reconstruction from uncalibrated images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 418–433, 2005.
- [56] N. Amenta, S. Choi, and R. Kolluri, “The power crust, unions of balls, and the medial axis transform,” *Computational Geometry: Theory and Applications*, vol. 19, no. 2-3, pp. 127–153, 2001.
- [57] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle, “Piecewise smooth surface reconstruction,” in *Proceedings of the SIGGRAPH conference*. ACM, 1994, pp. 295–302.
- [58] M. Lhuillier and L. Quan, “Match propagation for image-based modeling and rendering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1140–1146, 2002.
- [59] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, September 2004.



Gang Zeng received his B.S. degree in mathematical science from Peking University (PKU), China, in 2001. He is currently pursuing his Ph.D. degree in computer science, the Hong Kong University of Science and Technology (HKUST). Under the supervision of Prof. Long Quan, he has carried out his research mainly on computer vision and graphics. He is expected to graduate in summer of 2006, and has been a visiting student in Microsoft Research Asia (MSRA) during the summer and winter of 2004. His research interests include 3D scene reconstruction, image-based modeling and rendering, interactive modeling and relighting, plant modeling and feature extraction.



Sylvain Paris graduated from École polytechnique in France and has a masters degree in Algorithmics. For his PhD, he was advised by François Sillion at INRIA in Grenoble, France. He also collaborated with Long Quan from Hong Kong University of Science and Technology where he spent six months. Since November 2004, he is a postdoctoral associate at the Massachusetts Institute of Technology in Boston where he is working with Frédo Durand. His research interest is at the interface between Computer Vision, Image Processing and Computer Graphics. More precisely, he has contributions in the fields of 3D reconstruction from images, three-dimensional capture of hair, and computational photography.



Long Quan received the Ph.D. degree in Computer Science from INPL-CNRS-INRIA, France, in 1989. Before joining the Department of Computer Science at the Hong Kong University of Science and Technology (HKUST) in 2001, he has been a CNRS senior research scientist at INRIA since 1990. His research interests include vision geometry, 3D reconstruction, structure from motion, and image-based modeling and rendering. He is on the editorial board of the International Journal of Computer Vision (IJCV) and a regional editor of Image and Vision Computing Journal (IVC). He has served as an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI); as an area chair of International Conference on Computer Vision (ICCV), European Conference on Computer Vision (ECCV), and IEEE Computer Vision and Pattern Recognition (CVPR); and as a programme co-chair of IAPR International Conference on Pattern Recognition 2006 (ICPR) Computer Vision and Image Analysis.



François Sillion is a senior researcher with INRIA (Institut National de Recherche en Informatique et Automatique), after working for CNRS (Centre National de la Recherche Scientifique). He is the head of the ARTIS research project of the GRAVIR laboratory. His research interests include lighting simulation, photorealistic and expressive rendering, shadowing techniques, and real-time rendering. He is the author of a comprehensive book on the radiosity method, and serves on the editorial boards of “ACM Transactions on graphics” and “Computer Graphics Forum”.