Progressive Surface Reconstruction from Images Using a Local Prior
Gang Zeng, Sylvain Paris, Long Quan, François X. Sillion

To cite this version:

HAL Id: inria-00510155
https://inria.hal.science/inria-00510155
Submitted on 13 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Progressive Surface Reconstruction from Images Using a Local Prior

Gang Zeng\textsuperscript{1}  Sylvain Paris\textsuperscript{2}  Long Quan\textsuperscript{1}  François Sillion\textsuperscript{3}

\textsuperscript{1} Dep. of Computer Science, HKUST, \{zenggang,quan\}@cs.ust.hk
\textsuperscript{2} MIT CSAIL, sparis@csail.mit.edu
\textsuperscript{3} ARTIS\textsuperscript{*} / GRAVIR-IMAG, INRIA Rhône-Alpes, Francois.Sillion@imag.fr

Abstract

This paper introduces a new method for surface reconstruction from multiple calibrated images. The primary contribution of this work is the notion of local prior to combine the flexibility of the carving approach with the accuracy of graph-cut optimization. A progressive refinement scheme is used to recover the topology and reason the visibility of the object. Within each voxel, a detailed surface patch is optimally reconstructed using a graph-cut method. The advantage of this technique is its ability to handle complex shape similarly to level sets while enjoying a higher precision. Compared to carving techniques, the addressed problem is well-posed, and the produced surface does not suffer from aliasing. In addition, our approach seamlessly handles complete and partial reconstructions: If the scene is only partially visible, the process naturally produces an open surface; otherwise, if the scene is fully visible, it creates a complete shape. These properties are demonstrated on real image sequences.

Keywords: Surface Reconstruction, Local Prior, Voxel Carving, Graph Cut, Complete/Partial Reconstruction.

1. Introduction

Three-dimensional reconstruction from multiple images has numerous applications in the domains of virtual reality, movie making, entertainment, and so on. Thus, a lot of efforts have been made leading to a wealth of quality work. Numerous techniques have been introduced in the domains of camera calibration and surface reconstruction. Our work focuses on the latter, as we believe that several points can be improved to acquire the scene geometry.

The most precise methods such as the technique of Hernández and Schmitt [11] yield an accuracy comparable to a 3D scanner. However, they require a cumbersome setup (high resolution images and a turn-table) and induce a long computation time (several hours). On the other hand, flexible algorithms such as the well-known Space Carving [21] lack precision because they do not regularize the underlying problem. Thus, one naturally assumes some additional hypothesis – or, formally speaking, one adds a prior to the optimization scheme. Classically, the assumption is that the object is “smooth”. There are several mathematical translations of this intuitive hypothesis. It leads for instance to the Level Set technique [12]. However, such a choice of prior

\textsuperscript{*}ARTIS is a team of the GRAVIR lab (UMR 5527), a joint unit of CNRS, INRIA and UJF.
of a discrete volumetric representation (the voxels). Carving [21] and Dyer [29] have popularized the use of carving, and the accuracy of graph cut.

Previous Work

This paper presents a new interpretation of the smoothness assumption leading to a local prior. We virtually cut the surface into pieces (the patches) and we explicitly limit the scope of the prior to a patch instead of the whole surface. This approach is more versatile and makes it possible to combine the flexibility of carving with the accuracy of the graph cut. Another consequence is the possibility to only recover a partial shape if some part of the scene remains hidden in the entire sequence. Switching between complete and partial geometry is seamless.

Overview

Our technique is directly inspired by Space Carving [21]. The process starts from a bounding volume of the object and a set of calibrated images. The volume is first discretized into voxels. Then the voxels are considered sequentially. For each voxel, we try to reconstruct a patch using a graph-cut optimization. If the “quality” of the resulting patch is low, it is discarded along with the corresponding voxel. However, if the reconstruction is successful, then the patch is aggregated in a distance field with the already built patches. The process is iterated until the whole surface has been recovered.

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 typically use voxels that are one order larger than the ones in the classical carving techniques.

Contributions

The contributions of this paper are as follows:

Local Prior: We introduce a new interpretation of the smoothness assumption. The scope of the corresponding prior is only local. Combination Carving/Graph Cut: This new prior leads to an effective implementation that enjoys both the flexibility of carving, and the accuracy of graph cut. Complete/Partial Reconstruction: Without any adaptation, our algorithm can retrieve both complete shapes (when the whole scene is visible) and open surfaces (when some regions are hidden).

2. Previous Work

Carving

Seitz and Dyer [29] have popularized the use of a discrete volumetric representation (the voxels) in conjunction with a color criterion, the photo-consistency. According to the Lambertian assumption, a surface point must have the same color from any view direction. The voxels are then examined one by one and the photo-inconsistent ones are cut out. Many improvements have been proposed, such as arbitrary camera positions [21], robustness against noise [20], transparency [31], probabilistic interpretations [5, 10], other voxel shapes [34], and so on.

Carving is flexible (any camera position, any object topology) but it has a drawback: The consistency issue is considered without any prior, leading to an ill-posed problem. It has been shown that a set of given images can correspond to several shapes with equivalent photo-consistency, and that the result is the largest one among these shapes [21]. For untextured objects, it may significantly differ from the actual geometry. In addition, the accuracy degrades when the scene is not Lambertian.

Level Sets

Level sets is a flexible method to optimize functionals expressed as a weighted minimal surface:

$$\int \int w(x) \, ds$$

A time-evolving surface \( S(t) \) is represented at time \( t \) by the zero level set of an implicit function \( \phi(x,t) \), i.e. \( \phi(S(t), t) = 0 \). To minimize Functional (1), the surface evolves according to a steepest-descent process. From the Euler-Lagrange formula, \( \phi \) is driven by a partial differential equation (PDE):

$$\frac{\partial \phi}{\partial t} = \nabla w \cdot \nabla \phi + \nu |\nabla \phi| \frac{\nabla \phi}{|\nabla \phi|}$$

Faugeras and Keriven [12] have cast the reconstruction problem into this framework. To regularize the problem, they assume the object to be smooth. This corresponds to the minimal surface formulation of the level sets (smooth surfaces have smaller area). The advantage is that arbitrary genus can be handled. The \( w \) function in Equation (1) accounts for the texture correlation by computing the zero-mean normalized cross-correlation (a.k.a. ZNCC) between pairs of cameras \( \{C_i, C_j\} \). For a 3D point \( x \), the ZNCC value \( Z_{ij}(x) \) is defined with the projections \( p_i \) and \( p_j \) of \( x \) in cameras \( C_i \) and \( C_j \). For an image point \( p_i \), \( p_j \) and \( \sigma_p \) denote the mean and standard deviation of the intensity in the neighborhood \( N_p \). Using \( \pi \) to account for the perspective distortion between the two cameras (i.e. \( \pi(p_i) = p_j \) and \( \pi(N_{p_i}) = N_{p_j} \)), we finally get:

$$Z_{ij}(x) = \frac{1}{||N_{p_i}||^2 \sigma_p \sigma_{p_j}} \sum_{q \in N_{p_i}} (I_{q} - \bar{I}_{p_i})(I_{\pi(q)} - \bar{I}_{p_j})$$

The method has been extended with contours [16], with non-Lambertian materials [17], with texture, contours and 3D points [23,24], with open surfaces [30], and so on. They can all recover high-genus objects but, because of the high-order derivatives (Eq. (2)), sharp features such as corners and creases are not captured. In addition, different algorithms are required to handle complete [12] and partial [30] reconstructions.
Graph Cut  Roy [28] uses the network-flow theory [1] to build disparity maps. They design a weighted graph such that computing its minimum cut leads to an exact solution of a functional of the following form \( c(p, d) \) is the consistency at a pixel \( p \) and disparity \( d, d_p \) the disparity of \( p \), and \( \mathcal{A}_4 \) the set of the 4-connected adjacent pixels:

\[
\sum_p c(p, d_p) + \alpha \sum_{(p,q) \in \mathcal{A}_4} |d_p - d_q|
\]

This functional models a trade-off controlled by \( \alpha \) between the consistency (left term) and the regularity (right term). Other approaches have been then proposed to use: more sophisticated functionals [19], minimal surfaces [6], and depth fields [26]. The strength of these methods is their global convergence [6, 26] or at least effective local convergence [4,32], whereas level sets reach only a local minimum whose characteristics are unclear. This convergence leads to higher accuracy. However, these methods are limited by their parameterization of the disparity \( d(x, y) \) [6,19] or the depth \( z(x, y) \) [26]. Intrinsic volumetric studies [2,18] exist but their use for 3D reconstruction seems nontrivial.

One of our contributions is to adapt the graph-cut technique to volumetric reconstruction. We had the choice between the disparity map approach [6,19] with precise boundaries, and the depth-field one [26] with accurate depth. Since depth is more important in our approach than boundaries, we employ the latter technique that minimizes the following functional for \( z(x, y) \) (\( \alpha_x \) and \( \alpha_y \) modulate the regularity term, see [26] for details):

\[
\int \int \left[ c(x, y, z) + \alpha_x(x, y) \frac{\partial z}{\partial x} + \alpha_y(x, y) \frac{\partial z}{\partial y} \right] \, dx \, dy
\]

This approach yields higher accuracy but it depends on the \( xyz \) coordinate system. Since the integrals (6) and (7) consider the whole surface \( S \), this inherently limits the representable surfaces. Intuitively, splitting \( S \) into small pieces makes it possible to define \( S \) with several depth fields according to different coordinate systems.

Patch Definition  A patch \( P \) is a connected subset of \( S \). Our goal is to retrieve a set \( \{P_i\} \) such that \( \bigcup P_i = S \) represents the object shape. For each patch, a local coordinate system \( x_iy_iz_i \) is defined to parameterize \( P_i \) as \( z_i(x_i, y_i) \).

Local Prior  Using this definition, we can express the smoothness assumption locally. Instead of applying the smoothness term \( S \) on the whole surface at once, we apply it on each patch separately:

\[
S = \sum_i \int \int_{P_i} s(z_i) \, dx_i \, dy_i
\]

This integration is now split on several domains \( P_i \), introducing a coordinate system \( x_iy_iz_i \) for each of them. This overcomes the parameterization limitation of the global approach since \( S \) is now represented as an assembly of depth fields instead of a single one. The same treatment can be applied to \( C \). Hence, with \( f = c + s \), we can summarize the transformation from a global formulation to a local one:

\[
\mathcal{F} = \int \int_{\bigcup P_i} f \, dx \, dy \leadsto \mathcal{F} = \sum_i \int \int_{P_i} f \, dx_i \, dy_i
\]

This local expression shows that the patches can be optimized independently. In practice, we minimize Equation (5) for each patch using the depth-field scheme [26].

Caveats  The points near the patch border have a truncated neighborhood and are likely to be erroneously reconstructed. Thus, we use patches that are slightly larger than the voxels so that they overlap, and we discard the border regions. We also have to ensure that the joints between...
patches are continuous since the optimizations are independent. Discarding the border points and ensuring the continuity will be handled during the stitching step. Finally, the coordinate systems $x_iy_iz_i$ have to be determined. Since each patch is a depth field $z_i(x_i,y_i)$, 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.

4. Reconstruction Algorithm

We now expose a practical algorithm to reconstruct the patches $P_j$ using the local prior. We use a carving approach to approximately locate the object surface $S$. The fine geometry is retrieved using a graph-cut optimization.

Initialization The algorithm starts with a set of calibrated images. If the background is known, we can extract the object contours and use the visual hull [22] as a bounding volume (this initialization is akin to [11, 15]). Otherwise, we require the user to provide a bounding box. This volume is then discretized into cubic voxels.

4.1. Voxel Carving

We use 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 that purpose, we use the effective technique described in [8]). The process is iterated until no more voxels can be removed. In this global framework, we define our own carving criterion and ordering scheme.

4.1.1 Carving Criterion

Instead of computing the photo-consistency of a voxel to decide whether it is carved, we reconstruct a patch within it. We run a graph-cut process; it results in a patch $P$ and a functional value $\mathcal{F} = \mathcal{C} + \mathcal{S}$. The voxel is kept if the patch consistency $\mathcal{C}$ is less than a threshold $\tau$, otherwise it is carved. The rationale is that the consistency of $P$ is high (i.e. $\mathcal{C}$ is low) only if $P$ is part of the surface. This criterion is more robust than photo-consistency because it is based on a whole surface piece instead of a single point. For the same reason, $\tau$ is relatively easy to set in practice.

Remark that the carving decision does not involve $\mathcal{S}$ since smoothness is not an issue at the voxel level.

Normal Estimation To define the coordinate system, we need a normal estimate. We first start by fitting a plane to the current voxel and its adjacent surface voxels to get $n_0$ (shown as short lines on Fig. 1b, 4b, 7b). Then we build a patch $P^{(0)}$ from which we estimate a new normal $n_1$. If $n_1 \neq n_0$, we build $P^{(1)}$ using $n_1$. We iterate until $n_{k+1} = n_k$. In practice, this occurs in 2 or 3 steps. We define $P = P^{(k)}$ to compute the carving criterion $\mathcal{F}(P)$. In inconsistent regions, this may not converge. Therefore, if the process is not stabilized after $k_{\text{max}}$ iterations, the voxel is considered inconsistent and carved.

Consistency Function For the $c$ function (Eq. 5), we use the ZNCC value (Eq. 3) computed from the two most front-facing visible cameras $C_i$ and $C_j$ according to the normal estimate. For a 3D point $x$, we wish to choose a consistency function $c(x) \geq 0$ that decreases when the match quality increases, which can be computed by $c(x) = \arccos(Z_{ij}(x))$. This corresponds to the interpretation of ZNCC as a dot product. In our experiments, it better discriminates inconsistent points than a linear inversion such as $1 - Z_{ij}$.

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

4.1.2 Ordering Scheme

ZNCC is more reliable when computed with front-facing cameras because it limits the perspective distortion. 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 estimate.

Once a voxel is found 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 into the surface (cf. the following section).

4.2. Surface Construction

To collect all the patches and construct the final surface, we use a technique inspired by Curless and Levoy [9]. It has the advantage of allowing for incremental updates with a fine control over the fusion. It 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 [25]. $W$ can be seen 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 partially defined distance field: If a grid cell contains an uninitialized or null $W$ value, then no triangle is output.

In practice, for each new patch $P$, we compute a distance field $D_P$ and a weight function $W_P$ restricted to the neighborhood of $P$ (i.e. $D_P = W_P = 0$ outside the neighborhood, cf. Fig. 2). $D_P$ is the signed distance to $P$, $W_P$ is related to the confidence we have in $P$, its design is discussed later. At each grid vertex $x$, $D$ and $W$ are updated as follows:
\[ D(\mathbf{x}) = \frac{W(\mathbf{x})D(\mathbf{x}) + W_P(\mathbf{x})D_P(\mathbf{x})}{W(\mathbf{x}) + W_P(\mathbf{x})} \quad (10a) \]
\[ W(\mathbf{x}) = W(\mathbf{x}) + W_P(\mathbf{x}) \quad (10b) \]

The equations (10) show that \( D(\mathbf{x}) \) is the mean of all the patch distances \( D_{P_i} \), weighted by \( W_{P_i} \).

### 4.2.1 Patch Weight

The previous remark outlines the importance of \( W_{P_i} \) in determining the influence of \( P_i \) on the final result. As discussed previously, there are two major issues: discarding the unreliable points near the patch border, and ensuring the continuity across the patches. Both objectives are fulfilled by using a \( W_{P_i} \) 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 the surface continuity, from the Implicit Function Theorem, it suffices that:

1. \( D \) is \( C^1 \) continuous and,
2. \( \nabla D \) is not null when \( D = 0 \).

From Equations (10), if \( W_P D_P \) and \( W_P \) are \( C^1 \), then Condition (1) is fulfilled. Condition (2) is not as direct. Theoretically, the gradient could vanish, but it is very unlikely to occur in practice. First, \( \nabla (W_P D_P) = D_P \nabla W_P + W_P \nabla D_P \) can vanish near the border because \( W_P = 0 \) and \( \nabla W_P = 0 \) but it does not affect \( \nabla D \) since the patches overlap. Then, within the patch neighborhood, \( \nabla D_P \) cannot vanish because \( D_P \) is a signed distance function. But merging several patches at the same location may cancel the gradient \( \nabla D \). In practice, the zeros of \( D \) are near the zeros of \( D_P \), thus \( D_P \nabla W_P \) is negligible compared to \( W_P \nabla D_P \). 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 happened.

**Implementation** For \( W_P \), we use the patch center \( \mathbf{o} \) to define (see plot on Figure 3):

\[
W_P(\mathbf{x}) = \begin{cases} 
(1 - \frac{||\mathbf{x} - \mathbf{o}||^2}{\sigma^2})^2 & \text{if } ||\mathbf{x} - \mathbf{o}|| < \sigma \\
0 & \text{otherwise} 
\end{cases} \quad (11)
\]

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

### 4.2.2 Weight Refinement

The previous construction is independent of the input images: \( W_P \) depends only on the patch size. We refine this approach with \( W_{P_i}^* \) 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:

\[ W_P^* = \max(0, Z(x_i, y_i, z_i(x_i, y_i))) \]

The color intensity gradient \( \nabla I \) then yields an effective and computationally efficient estimation of the edges. Putting this together with a stopping function \( g \), we obtain:

\[ \frac{\partial \varphi}{\partial t} = \text{div} \left( g(\|\nabla I\|) \nabla \varphi \right) \quad (12) \]

Note that the \( g \) function has to be designed to slightly smooth the edges in order not to create discontinuities. Thus Condition (1) is satisfied and the smoothing mainly occurs within regions of the same color. Finally we extend \( \varphi \) to 3D: \( \Phi(x_i, y_i, z_i) = \varphi(x_i, y_i) \) and define: \( W_P^* = \Phi W_P \).

This refinement improves the accuracy of the seams because it accounts only for the most consistent patch whereas a direct blending would loose details by averaging several different contributions. Moreover, it makes the boundaries of the open surfaces clean since the gross errors in the patch borders are discarded.

### 4.3. Summary and Discussion

At a coarse level, our algorithm behaves like a carving technique except that we use the patch consistency \( C \) in-

**Figure 3.** \( x \mapsto \left(1 - \frac{x^2}{\sigma^2}\right)^2 \) if \( |x| < \sigma \), 0 otherwise. This function is also known as the Tukey function.
stead 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 (5) within each voxel. The optimization scheme [26] reaches a global minimum of Equation (5). In this respect, the patches are optimal. The consistent patches are then incorporated in a distance field. We have shown that, with a proper update scheme, this produces a continuous surface. Finally when no more consistent voxel is 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.

5. Results

Implementation Details The presented results use real images. ZNCC is computed with a $11 \times 11$ window. The patch size is set to twice the voxel size to ensure a sufficient overlap. To avoid grazing views, we ignore cameras whose angle with the normal is greater than $\frac{\pi}{3}$. The distance field $D$ has a resolution $4^3$ times finer than the voxel grid. The graph-cut process is run on a grid of resolution $15^3$. We stop the normal estimations after $k_{\text{max}} = 4$ iterations. In Equation (12), $g(||\nabla f||) = \max(0, 1 - ||\nabla f|| / 16)$ with $I \in [0; 255]$. We use the graph-cut code of the Boost library\(^1\) which leads to computation time between 20 min (the owl) and 45 min (the gargoyle). As future work, we want to try the implementation [3] 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 running time of the other steps of the algorithm is negligible compared to the optimizations.

Complete Reconstruction We use a handheld camera to shoot images all around the object. The calibration is done as a pre-process.

$\triangleright$ The gargoyle sequence (Fig. 4) shows that non-spherical topology can be reconstructed. There are 16 views at $720 \times 486$ although the gargoyle only covers an area of about $200 \times 400$. This demonstrates the performance of our technique on low-resolution data. The voxel space is $25 \times 50 \times 25$.

$\triangleright$ 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 they are thin and sharp. There are 37 views at $600 \times 800$. The voxel resolution is $25 \times 50 \times 25$.

$\triangleright$ The head sequence (Fig. 1) involves significantly non-Lambertian materials (eyes, skin, hair). There are 21 views at $480 \times 640$. The voxel resolution is $32^3$. A comparison with other well-known techniques (Fig. 5) emphasizes the improvement brought around by our approach: The accuracy is obviously higher than a photo-hull [21]. We also compare our result with the level-set method [23] that works from images, 3D points (extracted from the images) and

\(^1\)http://www.boost.org
contours. Thus, we fairly use the same images and silhouettes. Note that the head is not a sharp shape and should suit the level sets. Nonetheless, we recover a finer geometry whereas level sets smooth out the details.

**Partial Reconstruction** To demonstrate the capabilities of our approach for partial 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 (see Figure 6).

(a) 7 views (~120°)  (b) 10 views (~171°)  (c) 21 views (360°)

**Figure 6.** Partial reconstruction. The 21 input images form a rough circle around the head. To demonstrate that the algorithm handles both partial and complete shape, we have used only a subset of these images: 7 (a) and 10 views (b). Note that the geometry of the visible part is stable independently of the setup. The \( \Phi \) function makes the border clean (cf. Section 4.2.2).

6. Conclusions

We have presented a reconstruction algorithm that combines carving and graph cut. This technique relies on a new formulation of the smoothness assumption: The prior is applied to small surface patches instead of the whole surface. The given technique is nonetheless shown to preserve the continuity of the surface. The created geometry has several valuable characteristics: Arbitrary topology is recovered, complete and partial shapes are handled by the same algorithm, detailed surfaces are captured and sharp features are reconstructed. This unique combination of properties makes a significant contribution to the state of the art.

**Future Work** This local approach to 3D reconstruction opens promising research avenues. First, it may be interesting to explore other combinations: We have demonstrated the value of a carving/graph cut couple. One can also imagine other interesting couples such as level set and graph cut. The patches introduce a new degree of flexibility. A challenging topic would be to reconstruct non-manifold surfaces to address very difficult cases such as tree leaves or complex thin objects.

**Acknowledgements** This work is supported by the Hong Kong RGC grant HKUST 6182/04E. Sylvain Paris’ research at MIT is supported by Shell, and Lavoisier program from French ‘ministère des Affaires étrangères’. We thank Eugene Hsu for his help with the paper and Kyros Kuttalakos for the gargoyle images.

**References**


