Accurate and Scalable Surface Representation and Reconstruction from Images

Gang ZENG[1] Sylvain PARIS[2] Long QUAN[1] François SILLION[3]

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

{zenggang,quan}@cs.ust.hk

[2] CSAIL - MIT, 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

Draft submitted to PAMI on Sep. 08, 2005.

Abstract

We introduce a new surface representation, the *patchwork*, to extend the problem of surface reconstruction from multiple images. A patchwork is the combination of several *patches* that are built one by one. This design potentially allows the reconstruction of an object of 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 linear with respect to the object area. The former property ensures that we never run out of storage (memory) 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 (open or closed surface but not both).

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 expose how it dramatically extends a recent accurate graph-cut technique. 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 the patchwork reconstruction. The above properties of the patchwork representation and reconstruction are extensively demonstrated on 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 reconstruction from multiple images is a natural extension to stereoscopic reconstruction. Combining the information from several images make 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 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 retrieves the parameters of the cameras such as their positions and focal lengths, while the latter produces the actual geometry of the scene. In this paper, we focus on the geometry reconstruction part.

Two major issues remain largely unaddressed: 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 bounded by the available memory of the machine that executes the program. In addition to this storage issue, since the temporal complexity of the optimization algorithms is high (*i.e.* more than linear), increasing the scene size inherently leads to an explosion in the processing time. Thus, large scenes are limited to large scale reconstructions that ignore the fine details. Second, existing methods represent the object 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 $\phi(x,y,z) = 0$ (*a.k.a.* level set). These two options address different configurations. Depthfields and disparity maps perform well with cameras that lie only on one side of the scene but they are hard to extend to arbitrary camera positions. 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 the *patchwork* surface representation. It consists of a collection of small surface pieces, the *patches*, that are progressively reconstructed and stitched together. Despite its apparent simplicity, it implies a 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. Another behavior would mean for instance that adding an earring on one side changes the geometry of the other ear. It would be incoherent. This assumption is formally defined and assessed. We show that except the visibility all the other components involved in the existing optimization techniques are local.

Independent of the selected optimization technique, the patchwork representation 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 stated and proven. Second, the patch parameterization can be adjusted for each patch. For instance, this allows the representation of complex surfaces with methods that usually handle only depthfields or disparity maps. Third, the formulation is independent of the surface topology, 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 also 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 progressively build the final shape.

***Contributions**:* In summary, the patchwork representation and reconstruction described in this paper focuses on the following contributions:

1) *Local Prior*: We introduce a new local 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 (such as graph cuts) while preserving their intrinsic qualities.
4) *Flexibility*: The reconstruction makes it possible to overcome limitations (such as topology handling) inherent to 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 3D reconstruction problem is inherently ill-posed: There exist several geometric solutions that are consistent with the input images. In order to alleviate this point, the usual approach add an a priori hypothesis concerning the objects. Classically, this hypothesis states that the reconstructed surface must be regular, *i.e.* the smoothness. This assumption is interpreted in various frameworks, resulting in different mathematical formulations. We here review the existing reconstruction methods while focusing on the optimization techniques and their complexity.

A. No Optimization

1) Visual Hull: Laurentini *et al.* [35] introduced the *visual hull* as the largest volume consistent with the silhouettes observed from several viewpoints, which is an over-estimation that captures the large scale features of the scene but ignores the small details. Several efficient approaches have followed: fast computation from Boyer and Franco [8], reconstruction from uncalibrated cameras by Cipolla and Wong [16], spline model by Sullivan and Ponce [47], etc. These approaches are mainly used for real-time applications that add in the details with a texture map (*e.g.* Matusik *et al.* [40]) or as a first step to initiate a more accurate process such as Isodoro and Sclaroff [26], and Hernández and Schmitt [20]. Several techniques [49], [51] exist to extract more information from contours but the process suffers from numerical instabilities.

2) Photo Hull: Seitz and Dyer [46] popularized the use of a discrete volumetric representation (the voxels) in conjunction with a color criterion, the *photo-consistency*. Considering a point $\mathbf{p}$ visible from the cameras $i \in \mathcal{V}_{\mathbf{p}}$ seeing colors $\left\{C_{\mathbf{p}}^{i}\right\}$, the photo-consistency $P_{\mathbf{p}}$ of $\mathbf{p}$ is computed using the color distance d:

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

The original method then sweeps through the voxel space and carves out the voxels with a photo-consistency criterion higher than a given threshold. The rationale of this technique is that for a perfectly Lambertian object, a consistent 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 better sweep scheme [34], robustness against noise [33], transparency [48], probabilistic framework [12], [19], other voxel shapes [8], [56].

Since no optimization is evolved, these two kinds of methods are efficient and have the advantage of being easy to set up. In practice, they yield satisfying results on convex or textured areas (where the color information is dense) whereas the concave or untextured regions are poorly reconstructed. The voxel approach is also drastically limited by the available resources because the necessary storage is proportional to the bounding volume of the scene.

B. Optimization by Local Operators

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

$$\iint w(\mathbf{x})\, ds \tag{2}$$

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):

$$\frac{\partial \phi}{\partial t} \;=\; \nabla w \cdot \nabla \phi \;+\; w \,||\nabla \phi||\, \mathrm{div} \frac{\nabla \phi}{||\nabla \phi||} \tag{3}$$

It is important to note that the global integral (2) is minimized by means of local differential operators (Eq. 3) that only consider a local neighborhood of each point. It shows that, despite a global formulation, the technique is driven at a local scale.

Faugeras and Keriven [21] have cast the reconstruction problem into this framework. The advantage is that complex objects of arbitrary genus can be rebuilt. It also eases visibility management because occlusions can be estimated between each evolution step. The w function in Equation (2) is defined to account for the texture correlation by computing the zero-mean cross-correlation (*a.k.a.* 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 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 have:

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

This results in convincing reconstructions, especially for the topology: High-genus objects are recovered automatically. The counterpart of this technique is a lack of surface sharpness. This comes from the high-order derivatives that control the process (Eq. 3).

Inspired by this work, several techniques have extended the original techniques. Jin *et al.* [28] use contours as a source of information to define w. They also extend the consistency criterion to handle non-Lambertian objects [29]. Lhuillier and Quan [37] combine texture correlation, silhouettes and 3D points to reach faithful models.

2) Generalized Cylinder: In a spirit akin to the level-set method of Jin *et al.* [28], Terzopoulos *et al.* [50] use a general cylinder representation to retrieve the scene geometry from a set of silhouettes. They add symmetry constraints to their model; thus they can work from a single image. Their optimization scheme is expressed as an integral minimization, leading to local evolution rules based on partial derivatives. Relatively to our aim, the main drawback of this method is its general cylinder representation that is unlikely to capture fine details.

3) Snake: Hernández and Schmitt [20] determine the surface topology from the object visual hull. Thus, they use a classical snake approach instead of the level sets to preserve this topological information. Then, they deform a 2D snake using the *gradient vector flow* technique to promote the surface data in 3D. The accuracy of the results is impressive but the cost is that both surface and volume data structure are maintained, impeding the scalability and inducing long processing time (several hours).

4) Free Form Deformation: Isidoro and Sclaroff [26] minimize the retro-projection error using free form deformations. In this framework, the applied transformations are local, and the goal is a global decrease of the error. The surface representation is an obstacle to scalability.

C. Global Optimization

1) Minimal Cuts on Disparity Maps and Depthfields: Roy and Cox [44], [45] have shown how to use the graph-flow theory [2] to generalize the purely one-dimensional Dynamic Programming technique to the two-dimensional problem raised by disparity maps. They design a valued graph such that computing its maximum flow and extracting a corresponding cut leads to an globally optimal solution of a functional of the following form ($c_{\mathbf{p}}$ being the consistency at a pixel $\mathbf{p}$, $d_{\mathbf{p}}$ the disparity, and $\mathcal{A}_4$ the set of the 4-connected adjacent pixels):

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

This functional models a trade-off between the consistency (left term) and the regularity of the result (right-hand term). The advantage compared to the 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 only reach a local minimum.

The original technique has been extended in several directions. Hishikawa and Geiger [25] demonstrate that Equation (5) can be interpreted in the Markov Random Field framework. They also extend the regularization term to any convex function. Paris *et al.* [42] reinterpret Roy and Cox's work in the three-dimensional world to handle depthfields instead of disparity maps. They show how to solve the following continuous functional up to an arbitrary discretization (the surface is parameterized by the depth z as a function of x and y, and the α_x and α_y functions modulate the regularization term):

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

Kirsanov and Gortler [30] 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 the three-view reconstruction by Buehler *et al.* [13] with a weighted minimal surface.

Boykov *et al.* [11] introduce the α-expansion technique to apply graph cuts to more general functionals. This opens the way to finer numerical models but the convergence to a global minimum is lost. Kolmogorov and Zabih [31] have characterized a general theory on the set of functionals that can be handled by graph cuts. They also apply their method to disparity maps in the multi-view context [32]. In general, none of these methods scale up nicely due to the complexity of the global optimization.

Segmented Disparity Maps: Wei and Quan [53] (in the stereoscopic case) and Bleyer and Gelautz [7] (in the multi-view case) have shown that satisfying disparity maps can be achieved by segmenting the input images into small regions of constant color. They expose modified algorithms to assign a disparity value per segment instead of per pixel, which clearly reduces the amount of data. The challenge is to preserve fine details whereas the segmentation strategy takes advantage of the lack of depth precision to "smartly" downsample the disparity map.

2) Minimal Cuts on General Surfaces: Boykov and Kolmogorov [9] have shown how a weighted minimal surface (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.* [52] formulate the multi-view scene reconstruction problem using volumetric representation. From the scalability point of view, the volumetric structure limits the scene size.

D. Local Optimization

1) Partition of Unity: Ohtake et al. [41] introduced a surface representation that shares some common properties with ours. To recover a surface from a dense set of points, they locally fit quadratic patches. The stitching weights sum up to 1, forming a *partition of unity*.

2) Particles: Fua [22] exposed 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 could scale up because the particles can be handled separately. However, the accuracy is relatively low, since the particles are regarded as flat disks.

3) Quadratic Patches: In the context of stereo-vision, Hoff and Ahuja [23] constructed a disparity map by gathering the information stemming from several quadratic patches. The differences with our approach are nevertheless important. First, we encompass a broader context by being independent from the number of cameras. Second, the shape of our patches is general and not limited to a quadratic parametrization. Moreover our patchwork representation can be combined with numerous optimization methods, while Hoff-Ahuja use a least-square technique. Carceroni and Kutulakos [14] have extended the approach to motion and reflectance recovery. However the geometric accuracy is still limited by the patch shape.

Summary

Almost all of the existing methods have difficulties in handling large objects with fine details. In comparison, the proposed patchwork reconstruction defines a complete surface representation as a set of patches: Reconstructing the patchwork is equivalent to reconstructing the surface itself; The patches spread the whole surface and the continuity is handled during the reconstruction process. Thus, a large surface is separated, and each part is reconstructed efficiently with a certain optimization technique. Furthermore, it helps some of the most accurate techniques based on minimal cuts to cope with complex shapes. In the rest of the paper, we present our patchwork representation that addresses these issues.

III. Concept Definition and Theoretical Study

Here we formalize our problem to outline 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 Work section, we have shown that many reconstruction strategies are driven – either explicitly or implicitly – by local criteria. Here we state formally our base assumption: Two distant points do not interfere. We then derive our reconstruction strategy.

1) Locality Assumption: We name $\mathcal{S}_0$ a minimizer of $\mathcal{F}$ over the whole 3D space *i.e.* $\mathcal{S}_0 = \operatorname{argmin}_{\mathcal{S}\subset\mathbb{R}^3} \mathcal{F}(\mathcal{S})$. We consider two real numbers $\dot{r}$ and $\mathring{r}$ 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 the figure on the right for a 3D illustration of these entities.

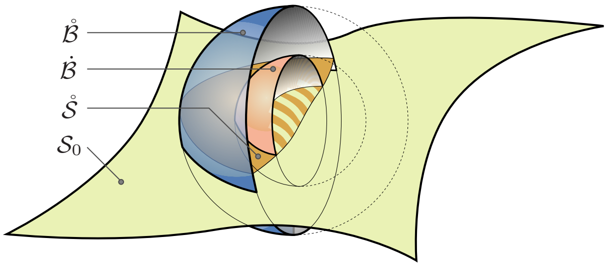


Fig. 1. We assume that there exist $\dot{\mathcal{B}}$ and $\mathring{\mathcal{B}}$ such that, inside $\dot{\mathcal{B}}$, the result $\mathring{\mathcal{S}}$ of the optimization within $\mathring{\mathcal{B}}$ equals the global result $\mathcal{S}_0$. This common portion corresponds to the stripped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

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

- *Interpretation:* This hypothesis means that a local optimization yields a correct result except on the border of the considered volume (*i.e.* between $\dot{\mathcal{B}}_{\mathbf{p}}$ and $\mathring{\mathcal{B}}_{\mathbf{p}}$). This restriction is reasonable since the border points have a truncated neighborhood (we cannot expect any optimization algorithm to give reliable results with partial data).
- *Discussion:* One can wonder how $\dot{r}$ and $\mathring{r}$ are determined in practice. This depends on the chosen functional and optimization technique. For instance, using the notation of Blake and Zisserman [6] on page 60, to guarantee a correct detection of the discontinuities, it is sufficient to set:

$$\dot{r} > 0 \quad \text{and} \quad \mathring{r} \gg \dot{r} + \lambda \tag{8}$$

In the case of level sets, if we know the number of iterations Υ (or a bound over it), we can derive $\dot{r}$ and $\mathring{r}$ since at each step, the derivatives of order ω involve the adjacent values up to a distance $\lceil \omega/2 \rceil$. Hence, using the discretization step δ of the level-set grid and Ω the maximum order of the involved derivatives, we obtain:

$$\dot{r} > 0 \quad \text{and} \quad \mathring{r} = \dot{r} + \Upsilon\,\delta \left\lceil \frac{\Omega}{2} \right\rceil \tag{9}$$

For the graph cut approaches, Kolmogorov and Zabih [32] and Paris *et al.* [42] handle discontinuities, hence continuous regions are independent. Thus it is sufficient to set $\dot{r}$ and $\mathring{r}$ to contain the largest continuous region. The previous examples show that, in several cases, the locality assumption is rigorously valid. 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 which is not compliant with our local approach. Therefore, in practice, the size of the local volume is set by the user. Nonetheless, we have this strong result that for sufficiently large patches, the local optimization is equivalent to a global one. We rigorously express this difference between global and local optimization in the following section.

2) Study of the Functional: $\mathcal{F}$ always contains a term $\mathcal{C}$ relating 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:

$$\mathcal{C} = \iint_{\mathcal{S}} c \; d\mu \tag{10}$$

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. Observing Equation (6), this induces the plane measure $d\mu = dx\; dy$:

$$\mathcal{S} = \iint_{\mathcal{S}} s(z) \; dx \; dy \tag{11}$$

This approach yields higher accuracy but it depends on the xyz coordinate system. Since the integrals (10) and (11) 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 will propose two practical strategies to build these axes.

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:

$$\mathcal{S} = \sum_i \iint_{\mathcal{P}_i} s(z_i) \; dx_i \; dy_i \tag{12}$$

The integration is now split in several domains $\mathcal{P}_i$, introducing a coordinate system $x_i y_i z_i$ 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:

$$\boxed{\mathcal{F} \;=\; \iint_{\bigcup \mathcal{P}_i} f \; dx \; dy \qquad \rightsquigarrow \qquad \mathcal{F} \;=\; \sum_i \iint_{\mathcal{P}_i} f \; dx_i \; dy_i} \tag{13}$$

Thus, the patchwork representation is relatively natural and simple from a formal point of view: A union in the geometric world 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 [42].

3) Surface Reconstruction: The patchwork reconstruction consists of building a set of patches $\{\mathcal{P}_i\}$ that represent the whole surface $\mathcal{S}$. Several local optimization processes are run *i.e.* we use several local volumes $\mathring{\mathcal{B}}_i$, each one producing a surface portion $\mathring{\mathcal{S}}_i$. Because the border points of $\mathring{\mathcal{S}}_i$ are not reliable, we keep only the center part $\mathring{\mathcal{S}}_i \cap \dot{\mathcal{B}}_i$: This is the actual patch $\mathcal{P}_i$ produced by the local process.

a) Continuity: We set the size of local volumes so that the domains of adjacent patch reconstructions overlap with each other. The overlapping region provides information for a seamless stitching among the patches. Moreover, when we build a new patch, we may further consider the neighboring reliable patches that have already been built. These data are used has a hard constraint for the new patch. Thus the optimization of $\mathcal{F}$ acts upon the new patch while considering the reliable ones. Formally, we name $\bar{\mathcal{S}}$ the surface built by the previously recovered reliable patches, and we compute:

$$\mathring{\mathcal{S}} = \operatorname{argmin}_{\mathcal{S} \subset \mathring{\mathcal{B}}} \mathcal{F}(\mathcal{S}) \quad \text{with the constraint} \quad \mathcal{S} \supset \left(\bar{\mathcal{S}} \cap \mathring{\mathcal{B}} \right) \tag{14}$$

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" ones. Thus, we can exploit the order in which the patches are built to reconstruct in priority the most reliable regions so that the weakest patches rely on them to be more accurate. 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 [18]. When all the patches are recovered, the final surface is extracted using the Marching Cube technique [39]. 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 and for graph-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 of size at least $\mathcal{O}(a_\mathcal{S}\ \delta^{-2})$. This is the case for some graph-cut techniques [32] and for the narrow-band implementation of level sets [1]. Some algorithms (such as level sets, carving methods or some graph-cut techniques) use volumetric representations, 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 [21], [37] 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 cubic (or slightly better [15]). In practice, they behave almost linearly ($\alpha \approx 1.2$) [44]. Note that some min-cut techniques (*e.g.* Kolmogorov and Boykov [32]) 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 $a_\mathcal{P}$. The number of patches η is in order of $\mathcal{O}(a_\mathcal{S}/a_\mathcal{P})$. To compare with $\mathcal{S}$, we also define a pseudo-volume $v_\mathcal{P} = \mathcal{O}\left(a_\mathcal{P}^{-\frac{3}{2}}\right)$ 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 in the order of $\mathcal{O}(a_\mathcal{P}\ \delta^{-2})$ (or $\mathcal{O}(v_\mathcal{P}\ \delta^{-3})$). 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 in $\mathcal{O}(\eta\ a_\mathcal{P}^\alpha\ \delta^{-2\alpha})$ or $\mathcal{O}(\eta\ v_\mathcal{P}^\alpha\ \delta^{-3\alpha})$.

Comparison: Table I summarizes all these results. It appears that the patches bring significant gain in terms of space and time complexity. The spatial complexity is the main gain since we divide the memory needed by a factor in order 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 (this issue is discussed later in the paper).

	SPACE global	patches	**gain**	TIME global	patches	**gain**
surfacic	$a_{\mathcal{S}}\ \delta^{-2}$	$a_{\mathcal{P}}\ \delta^{-2}$	η	$a_{\mathcal{S}}^{\alpha}\ \delta^{-2\alpha}$	$\eta\ a_{\mathcal{P}}^{\alpha}\ \delta^{-2\alpha}$	$\eta^{\alpha-1}$
volumetric	$v_{\mathcal{S}}\ \delta^{-3}$	$v_{\mathcal{P}}\ \delta^{-3}$	$\eta^{\frac{3}{2}}$	$v_{\mathcal{S}}^{\alpha}\ \delta^{-3\alpha}$	$\eta\ v_{\mathcal{P}}^{\alpha}\ \delta^{-3\alpha}$	$\eta^{\frac{3}{2}\alpha-1}$

TABLE I

COMPARISON OF THE COMPLEXITY

Scalability property: The patches allow for almost unlimited scalability because the space complexity depends only on the patch size and no more on the object size.

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

The gain in volumetric representations is more important because the patches ignore the inner volume of the object. In this way, they are comparable to a narrow band [1].

C. *Study of the Parameterization*

The patch also alleviates the limitation on the parametrization inherent in disparity map and heightfield 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 to properly handle the cases that require several z values for a single (x, y). Several functions f_1, f_2,... are then manipulated. Moreover, if the object surface is tangent to the z axis, these methods fail because of $||\nabla f|| = \infty$.

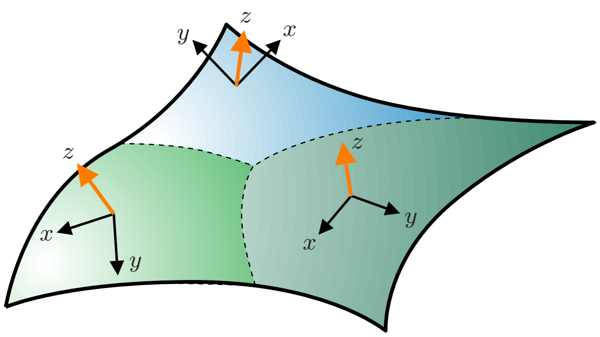


Fig. 2. Three patches with their local coordinate system

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: The z axis can be chosen orthogonally to the surface to guarantee that the tangent case never occurs.

Note that complex topology is not a problem in the sense that patches can cope with any topology. However, topology is not determined by the patches themselves: We rely on a side technique to determine it (this point is discussed 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, it is beyond the content of this paper and 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 [18]. It has the advantage of allowing incremental updates with a fine control over the fusion. There are nonetheless two important caveats to consider: First, the patch borders should not be incorporated into the final surface since they are not reliable. Also this step must not incur spurious discontinuities on the surface.

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 [39]. 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 a partially defined distance field: If a grid cell contains an uninitialized or null W value, no triangle is output.

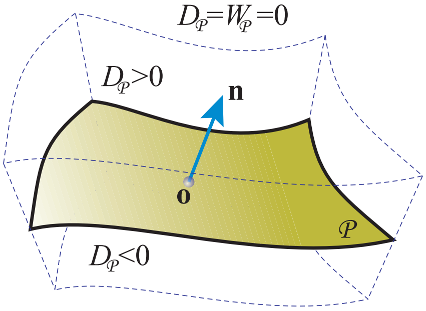


Fig. 3. The patch $\mathcal{P}$. The dashed lines delimit the neighborhood. $\mathbf{o}$ is the center of $\mathcal{P}$, and $\mathbf{n}$ the local estimation of the normal.

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:

$$D(\mathbf{x}) = \frac{W(\mathbf{x})D(\mathbf{x}) + W_\mathcal{P}(\mathbf{x})D_\mathcal{P}(\mathbf{x})}{W(\mathbf{x}) + W_\mathcal{P}(\mathbf{x})} \tag{15a}$$

$$W(\mathbf{x}) = W(\mathbf{x}) + W_\mathcal{P}(\mathbf{x}) \tag{15b}$$

The equations (15) show that $D(\mathbf{x})$ is the mean of all the 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. As previously mentioned, 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.

More formally, to achieve 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 (15), if $W_\mathcal{P} D_\mathcal{P}$ and $W_\mathcal{P}$ are C^1, then Condition (1) is fulfilled. Condition (2) is not as direct. Theoretically, the gradient could vanish, but it is very 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 happened. We use the patch center $\mathbf{o}$ to define $W_\mathcal{P}$ (see plot on Figure 4):

$$W_\mathcal{P}(\mathbf{x}) = \begin{cases} \left(1 - \frac{||\mathbf{x}-\mathbf{o}||^2}{\sigma^2}\right)^2 & \text{if } ||\mathbf{x}-\mathbf{o}|| < \sigma \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

We set σ 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: $W_\mathcal{P}$ is C^1, and the border discontinuities of $D_\mathcal{P}$ and $\nabla D_\mathcal{P}$ are cancelled by $W_\mathcal{P} = 0$ and $\nabla W_\mathcal{P} = 0$.

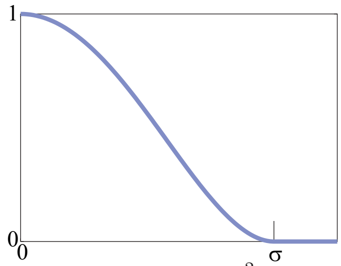


Fig. 4. $x \mapsto \left(1 - \frac{x^2}{\sigma^2}\right)^2$ if $|x| < \sigma$, 0 otherwise. This function is also known as the Tukey function.

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 $W^\star_\mathcal{P}$ 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: $W^\star_\mathcal{P} = \max(0, Z)\ W_\mathcal{P}$, ($\max(\cdot)$ keeps it non-negative and cancels the gross errors). However, for real images, ZNCC is unlikely to be C^1, thus Condition (1) would be violated.

To address this point, we smooth 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 [43]. 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 [57], 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 [5], we obtain:

$$\frac{\partial \varphi}{\partial t} = \mathrm{div}\big(g(||\nabla I||)\nabla \varphi\big) \tag{17}$$

Note that the g function is designed to slightly smooth the edges in order to preserve continuity. 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: $W^{\star}_{\mathcal{P}} = \Phi W_{\mathcal{P}}$.

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. In that it is different from the classical approach in parallel computing that subdivides a large problem (*e.g.* equilibrium in Mechanics [36]) 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: As previously discussed, 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 [34], visual hull [35], level sets [21], etc. Note that 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 describe in detail a scenario for which we use the side technique for normals and topology, and one for which it is only used to bootstrap the reconstruction process.

IV. Application I: Patch-wise carving from multiple images

Based on the new theory that are proposed in the previous sections, we now describe a practical algorithm [55] that is directly inspired by Space Carving [34]. 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. For untextured objects, it may significantly differ from the actual geometry. In addition, the accuracy degrades when the scene is not Lambertian. These have motivated us to adapt the carving criterion by considering the existence of a local patch $\mathcal{P}$. We use a carving approach to approximately locate the object surface $\mathcal{S}$. The fine geometry is retrieved using a local graph-cut optimization on each patch.

A. 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* [35] as a bounding volume (this initialization is akin to [20], [26]). 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 typically use voxels that are one order larger than the ones in the classical carving techniques.

B. Local Optimization

We have chosen the depth-field optimization method [42] 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 parametrization $z(x, y)$ but the patchwork representation addresses this point with its multiple local coordinate systems. We refer to the original article [42] for the technical details.

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 [17]). 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 it[1]. 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 [34]. However, this would only happen with large textureless regions since our voxels are one order bigger the one 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 low textures (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 2 or 3 steps. We define $\mathcal{P} = \mathcal{P}^{(k)}$ to compute the carving criterion $\mathcal{F}(\mathcal{P})$. 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 it is carved.

Consistency Function: For the consistency function c (Eq. 6), we use the ZNCC value (Eq. 4) computed from the two most front-facing visible cameras C_i and C_j according to the normal estimate. 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 better discriminates inconsistent points than a linear inversion such as $1 - Z_{ij}$. This strategy yields satisfying results at a reasonable computational cost. As future work, it would be interesting to test other consistency estimators [20], [21], [29].

If 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: $c(\mathbf{x}) = \arccos(Z_{ij}(\mathbf{x})) + v(\mathbf{x})$.

[1]Note that the patch is not strictly within the voxel, it is large enough to overlap with its neighbors, cf. Section III-D.

2) Ordering Scheme: 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 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 [42] reaches a global minimum of 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.

V. APPLICATION II: SURFACE RECONSTRUCTION BY PROPAGATING 3D STEREO DATA IN MULTIPLE 2D IMAGES

In this section, we apply the patchwork concept to combining several information sources, especially 3D points and images [54]. This approach is motivated by the fact that most of scanning devices such as laser scanners also take a photograph of the scanned object. Purely image-based approaches, such as the method of Lhuillier and Quan [38], also provide reliable

3D points using only standard photographs. We propose a technique which addresses two major points. First, meshing such a point cloud is difficult because of the noise, and of the sampling rate which may be insufficient, and so on. Techniques such as the ones by Amenta et al. [3], [4] and by Hoppe et al. [24] exist but they do not exploit the images that are available in a number of cases, which would help. Associating images and points ease 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 details. The patchwork representation provides an effective framework to coherently handle these various situations.

In our method, 3D points and images are considered as input. We do not assume 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 the technique of Lhuillier and Quan [38] to produce the 3D points. We have chosen this method because it gives irregularly distributed point sets that well illustrate our work. Nonetheless, the proposed technique can work with any range scanners that provide 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.

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.

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

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.* $\mathcal{F}$ 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. Using 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.

$$\Lambda = \frac{Z^{\omega(Z)} \cdot D^{\omega(D)}}{F^{\omega(F)} \cdot K^{\omega(K)}} \tag{18}$$

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 a new 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 first the most accurate ones. 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 second large eigenvalue λ_2 (the corresponding eigenvector lies in the tangent plane) and the smallest one λ_3 (the corresponding eigenvector is orthogonal to this plane). Thus, we have $\Pi = \frac{\lambda_2}{\lambda_3}$.

Selection Criterion for Generated Seeds: For a generated seed , we use the ZNCC correlation score Z by its two most front-facing cameras, since a strong match gives a high confidence. This strategy ensures that the surface grows from the part which 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 then low. Therefore a seed without occlusion is processed before a seed with occlusion. The occluded parts “wait” until other parts are reconstructed. The current visibility of the processed seed is classically determined by the current propagated surface using a ray-tracing technique. 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 stereo 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 the patch-wise carving, which employs a low-res. voxel space in the above section, the propagation leads to a relatively fast reconstruction, since the additional stereo points provide accurate locations on several surface regions. However, a side technique is required to obtain these stereo points.

VI. Results and discussion

A. Patch-wise Carving

Implementation Details: The presented results use real photographs shot with a handheld consumer-grade camera. The calibration is done as a pre-process. ZNCC is computed with a 11×11 window. The patch size is set to twice the voxel size to ensure a sufficient overlap.

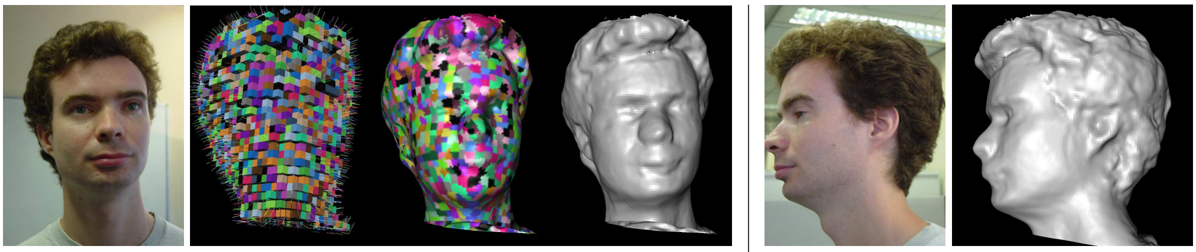


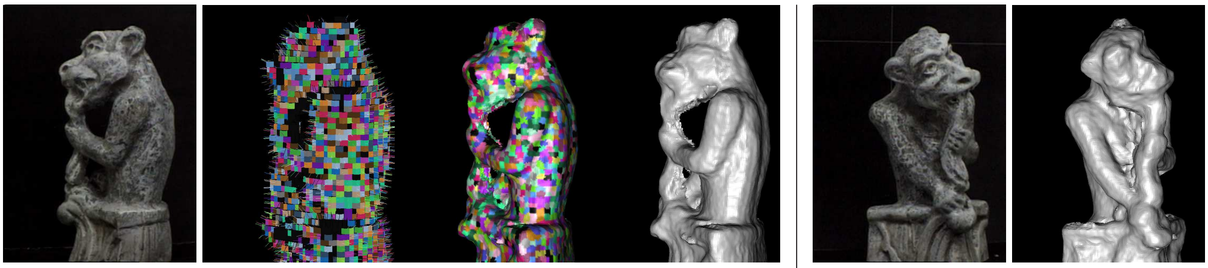
Fig. 5. Head reconstruction using our carving approach. This example demonstrates the ability of our approach to deal with non-Lambertian materials (skin and hair). The voxel resolution (b) is 32^3; this is one order coarser than traditional carving techniques. Although the process has been done patch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

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 15^3. We stop the normal estimations after $k_{\max} = 4$ iterations. For example, for the owl sequence, 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 (17), $g(||\nabla I||) = \max(0, 1 - ||\nabla I|| / 16)$ with $I \in [0; 255]$. We use the min-cut code of the Boost library[2] which leads to a computation time of between 20 min (the owl) and 45 min (the gargoyle). As future work, we want to try an implementation [10] 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).

$\triangleright$ The head sequence (Fig. 5) shows that non-Lambertian objects 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.

The role of each step of the algorithm is clearly put into evidence. At a coarse level, our algorithm behaves as a carving technique (Figure 5-b) 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 5-c). These patches are stitched together to produce the final surface. As predicted, our stitching scheme achieves a seamless and continuous result (Figure 5-d,f).

[2] `http://www.boost.org`



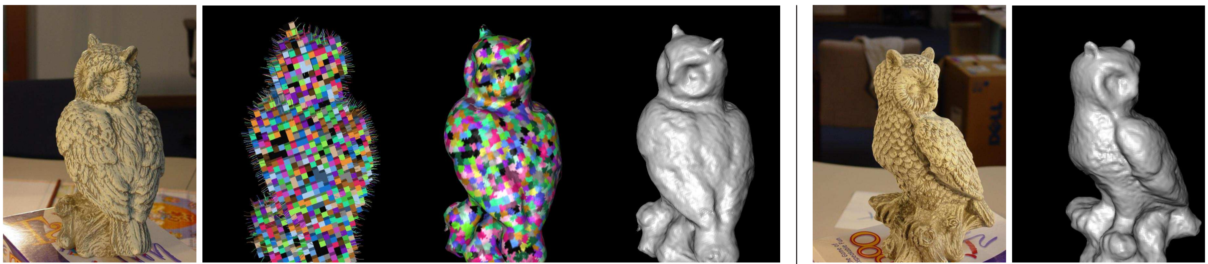
(a) Input image (b) Voxels (c) Patches (d) Surface (e) Input image (f) Surface

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 body occludes most of the cameras. Only views with a grazing angle can be used for this part of the model.

▷ The gargoyle sequence (Fig. 6) shows that 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 [33], [34]. The precision is improved.

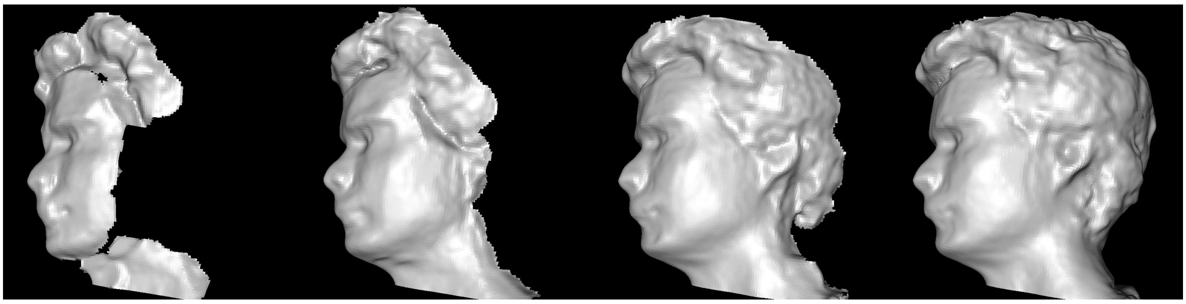
▷ The owl sequence (Fig. 7) demonstrates the performance of the technique on concavities and thin sharp features. We correctly reconstruct the ears whereas many existing techniques (such as level sets) would have some difficulties due to the high curvatures. There are 37 views at 600×800. The voxel resolution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

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 8-a,b,c). When all the images are available, the technique naturally produces a closed surface (Figure 8-d). Note that the geometry of the visible part is stable, independently of the setup. The Φ function makes the border clean (cf. Section III-D.2).



(a) Input image (b) Voxels (c) Patches (d) Surface (e) Input image (f) Surface

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.



(a) 5 views ($\sim 86°$) (b) 7 views ($\sim 120°$) (c) 10 views ($\sim 171°$) (d) 21 views ($\sim 360°$)

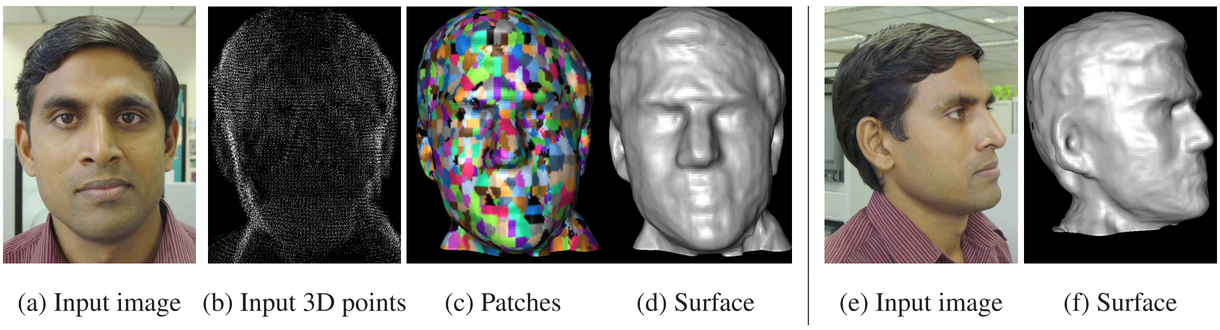
Fig. 8. Partial reconstruction. The 21 input images form a rough circle around the head. To demonstrate that the algorithm handles both partial and complete shape, we have used only a subset of these images: 5 (a), 7 (b), 10 (c) and all views (d).

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 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 density, 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.

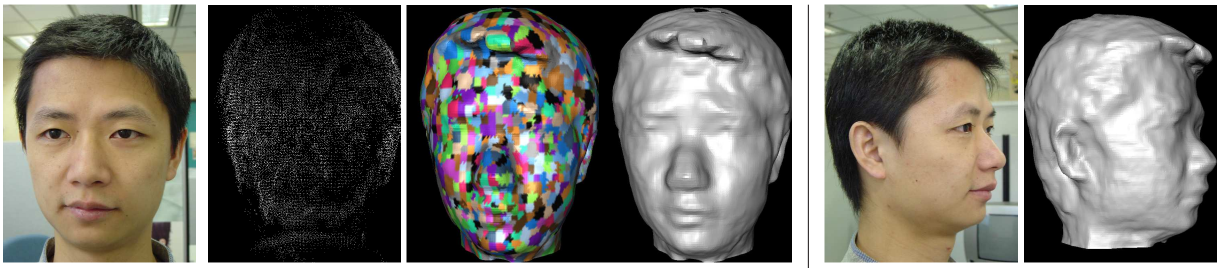
▷ The toy example (Figure 11) illustrates the correctness and robustness of the patch-wise propagation. Fur is traditionally hard for surface reconstruction because its appearance is strongly view-dependent. This model also contains large occlusions (the legs and arms are hidden in several images). Despite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



(a) Input image (b) Input 3D points (c) Patches (d) Surface (e) Input image (f) 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.



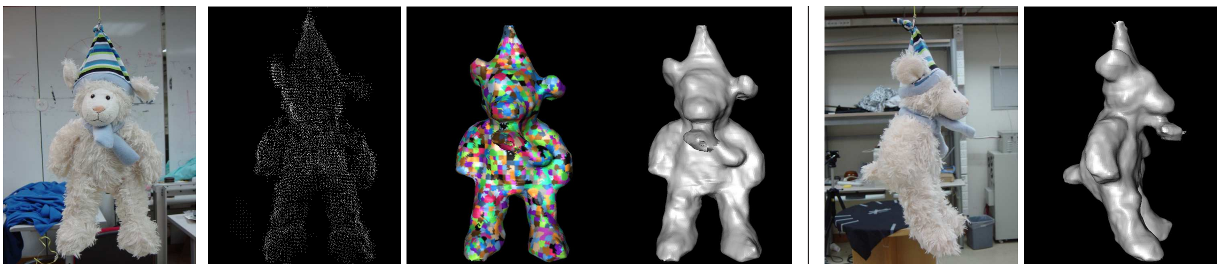
(a) Input image (b) Input 3D points (c) Patches (d) Surface (e) Input image (f) Surface

Fig. 10. Head reconstruction using our propagation approach. The input point cloud (b) that we have extracted using an image-based approach [38] has two large holes on the cheeks, because these two regions have almost no texture in the input images (a,e). In addition, the point density is also coarser compared to the first one. However, the proposed algorithm produces a surface with an equivalent quality.

alently. 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 computationally more expensive consistency estimators such as [20], [27]. There are 23 images with the resolution 600×800.

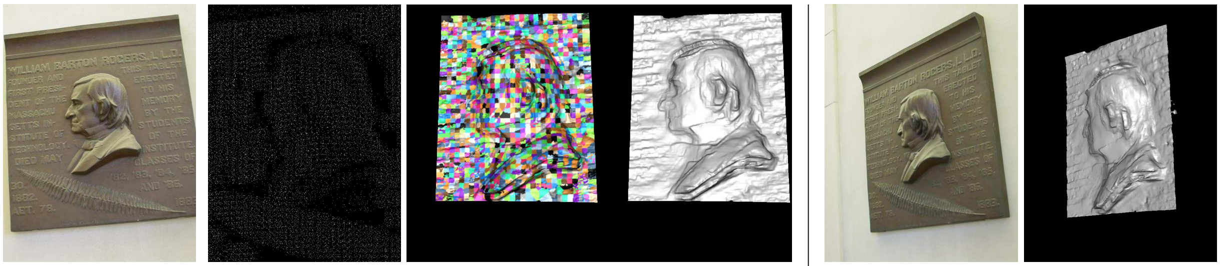
C. Comparison

In Figure 13, we use the same image sequence as Figure 5 to compare our two algorithms with a level-set method [37] and Space Carving [34]. The first point is that Space Carving fails to capture any 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 our methods and the level-set technique work fairly from the same image sequences and the 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 hard to estimate (*e.g.* near the face-hair boundary).



(a) Input image (b) Input 3D points (c) Patches (d) Surface (e) Input image (f) Surface

Fig. 11. Toy reconstruction using our propagation approach. It is a difficult example because of the fur and of the occlusions. Nonetheless, our algorithm yields a satisfying result.



(a) Input image (b) Input 3D points (c) Patches (d) Surface (e) Input image (f) Surface

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 on two regions: There are artifacts on the top of the head and the bottom of the bust. It shows 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 costly consistency estimators [20], [27].

This advocates for 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 is more suitable when 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 amount of 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 we use ceases to extract further information. Thus, 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, there is a bound to the complexity gain that can be achieved. On the other end, when the patches are too large, several advantages of patchwork reconstructions 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 on their relative efficiency. A careful examination of these 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, smaller patches better match its high curvature).

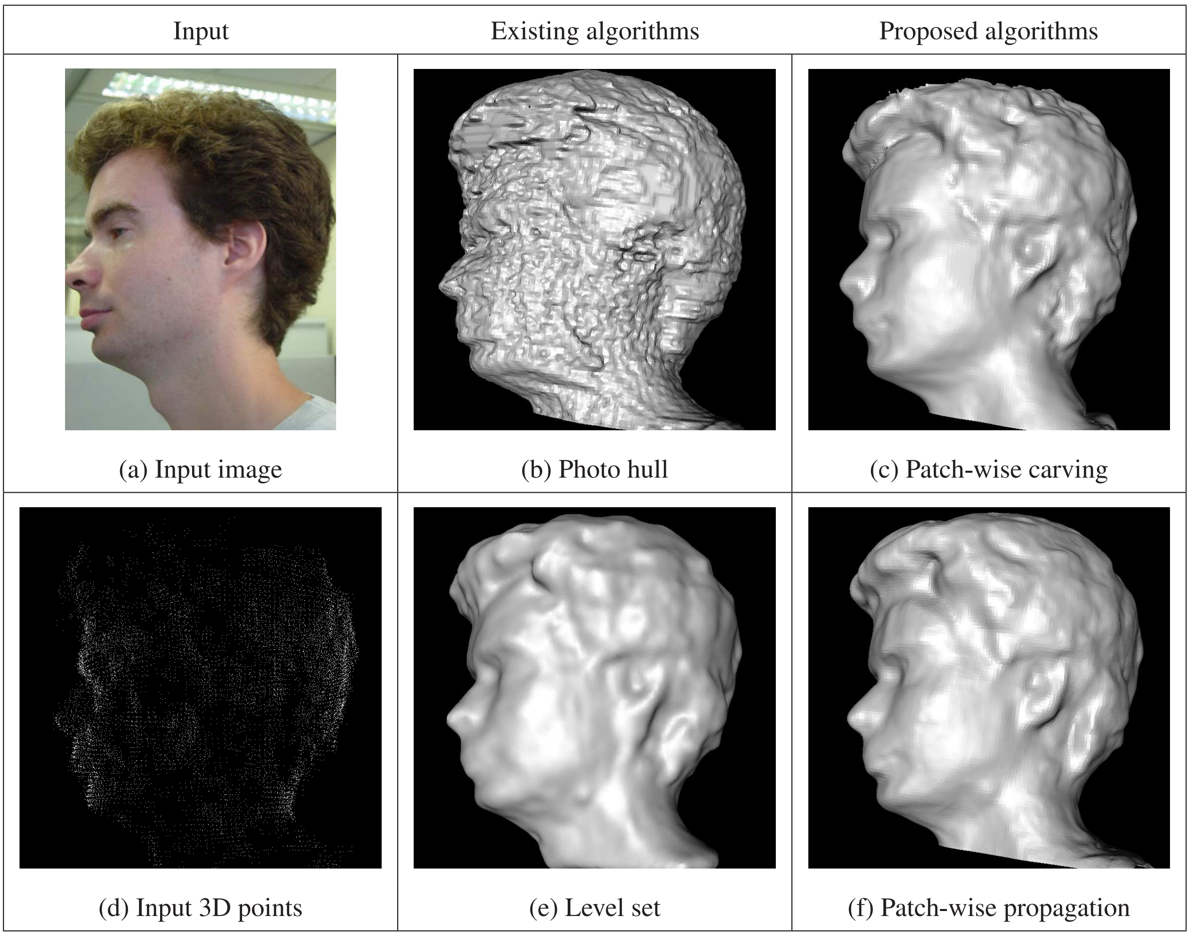


Fig. 13. Comparison. (a) One of the input images (b) Space Carving [34] fails to build a satisfying reconstruction due to the non-Lambertian materials involved. To achieve a fair comparison without aliasing, the voxel volume has been triangulated using the Marching Cubes [39]. (c) Patch-wise carving and (f) propagation build reasonable results by patches that consider both image information and regularity. (e) The level-set technique [37] builds a satisfying geometry but less detailed compared to our techniques (c,f) *e.g.* observe the chin, the eyes and the forehead. (d) The input 3D points used in (e) and (f).

E. Quantitative Analysis

Table II shows typical values for memory usage and running times on an Intel PIII-1.9GHz. These numbers correspond to the experiment of Figure 14. This validates our space complexity analysis: The required storage for the optimization does not dependent on the object size. Note that the global memory footprint increases because our implementation keeps the patches in memory after their creation. This strong result encourages us to implement an out-of-core method that stores the patches on the hard drive and thus enjoy an almost unlimited scalability.

To validate the time complexity analysis of Section III-B, we first demonstrate that the meaningful size of the problem in term of complexity is the area of the surface to reconstruct relative to the targeted resolution. Formally speaking, the problem size is in the order of $\mathcal{O}(a_{\mathcal{S}}/\Delta_{\mathrm{DF}}^2)$ where $a_{\mathcal{S}}$ 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 $a_{\mathcal{S}}$ (*i.e.*

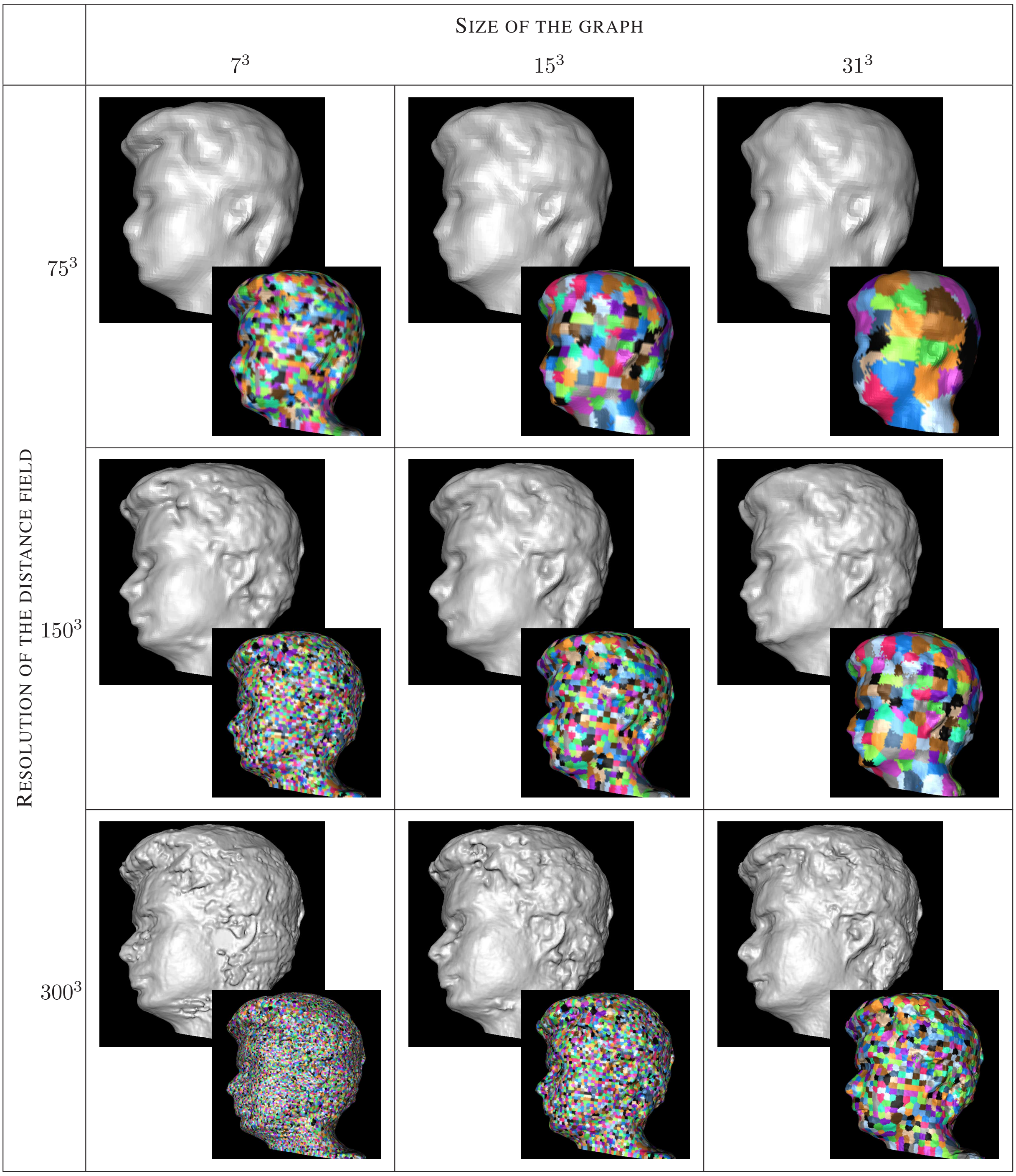


Fig. 14. Illustration of the effect of the resolution of distance field and of the graph size. We use the carving algorithm. Increasing the distance field resolution allows for capturing more details. When the graph size is kept constant, the corresponding patches become smaller. First the results improve (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, too large patches perform poorly (top right result). These issues are further discussed in the text.

DISTANCE FIELD RES.	GRAPH SIZE 7^3	15^3	31^3
75^3	229s (2785)	297s (559)	520s (104)
150^3	1010s (11876)	1455s (2772)	2406s (554)
300^3	3960s (45917)	6483s (11643)	12458s (2747)

(a) Running time (number of patches)

DISTANCE FIELD RES.	GRAPH SIZE 7^3	15^3	31^3
75^3	1M (105)	2M (106)	15M (119)
150^3	1M (121)	2M (122)	15M (134)
300^3	1M (238)	2M (239)	15M (251)

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

TABLE II

QUANTITATIVE COMPARISON AMONG DIFFERENT RESOLUTIONS

using a bigger 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, hence the ratio $a_{\mathcal{P}}/\Delta_{\text{DF}}^2$ is constant (with $a_{\mathcal{P}}$ is the patch area). Thus, the number of patches $\eta = \mathcal{O}(a_{\mathcal{S}}/a_{\mathcal{P}})$ is in the order of $\mathcal{O}(\Delta_{\text{DF}}^{-2})$. From our analysis, we expect a complexity relatively linear to η (cf. Table I) or equivalently quadratic in the distance field resolution $\frac{1}{\Delta_{\text{DF}}}$. This is the best possible complexity since it is relatively linear to the problem size $\eta = \mathcal{O}(a_{\mathcal{S}}/\Delta_{\text{DF}}^2)$.

Figure 15 summarizes our measures. Fitting a polynomial curve gives a complexity of $\mathcal{O}(\Delta_{\text{DF}}^{-2.16})$. 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 quadratic, in order of $\mathcal{O}(\Delta_{\text{DF}}^{-2.07})$. Nonetheless, we believe that this result is very strong in terms of scalability. To our knowledge, the patchwork representation is the first reconstruction technique that is proven to have a linear complexity which is practically confirmed on a real example.

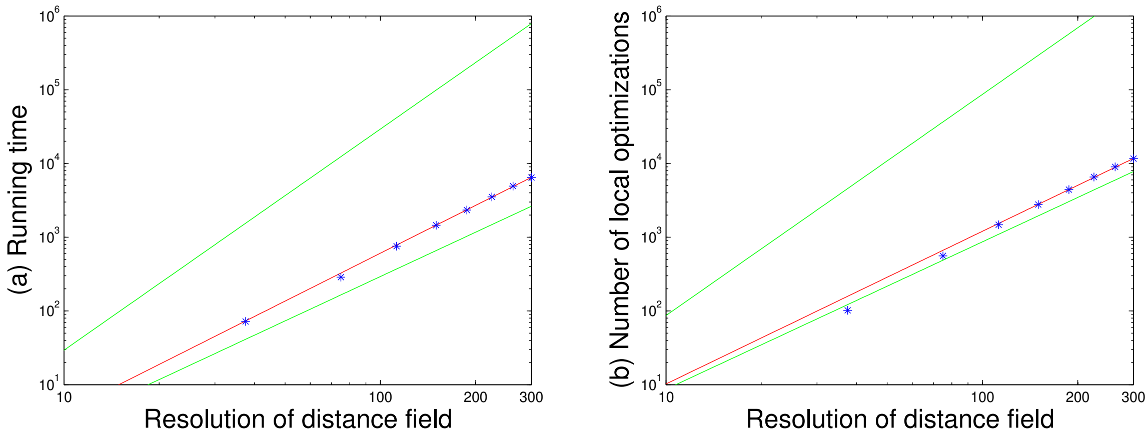


Fig. 15. We measure the running time and the number of local optimizations in terms of the resolution of distance field (from 38^3 to 300^3). Fitting a polynomial curve gives a running time in the order of $\mathcal{O}(\Delta_{\text{DF}}^{-2.16})$ and a number of built patches in the order of $\mathcal{O}(\Delta_{\text{DF}}^{-2.07})$. They are close to the optimal solution $\mathcal{O}(\Delta_{\text{DF}}^{-2})$ (*i.e.* the below green curve), and are much better than the global optimization, which is at least $\mathcal{O}(\Delta_{\text{DF}}^{-3})$ (*i.e.* the above green curve).

VII. Conclusion

We have presented a new patchwork representation. 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 a complex topology. 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, the patch representation is proven to induce an optimization process that requires a constant memory footprint, independently 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 described two algorithms based on the patchwork concept. The first one combines a carving strategy with min-cut optimization to retrieve the object geometry. The second algorithm is specially designed to exploit reliable 3D points that are available in a number of configurations. Both are demonstrated on real examples. The reconstructed surfaces compare favorably with existing techniques.

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. Representing the surface as a 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 extends to any optimization technique. As a consequence, we believe that the patchwork concept has this great contribution: Any optimization technique can enjoy enhanced scalability and flexibility simply by using patches to represent the object surface.

Future Work: Throughout the paper, we have mentioned several avenues for future research that we summarize here. Testing more robust consistency estimators would certainly further broaden the capacity of our algorithms. In some situations, it may be hard to get reliable 3D points. Nonetheless, the “no-point” configuration are rare, thus combining our two algorithms into a single one is likely to enhance their performances. An extension of developing an out-of-core stitching process for very large and/or very detailed objects (*e.g.* monuments) would be useful. Finally, we have obtained the patches with min-cuts but other methods such as level sets would be interesting to examine.

REFERENCES

[1] David Adalsteinsson and James A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118:269–277, 1995.

[2] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993. ISBN 013617549X.

[3] Nina Amenta, Marsahll Bern, and Manolis Kamvysselis. A new voronoi-based surface reconstruction algorithm. In *Proceedings of the ACM SIGGRAPH conference*, pages 415–421. ACM, 1998.

[4] Nina Amenta, Sunghee Choi, and Ravi Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19(2-3):127–153, 2001.

[5] Michael J. Black, Guillermo Sapiro, David H. Marimont, and David Heeger. Robust anisotropic diffusion. *Transactions on Image Processing*, 7(3):421–432, March 1998.

[6] Andrew Blake and Andrew Zisserman. *Visual reconstruction*. Mit Press, 1987. ISBN:0-262-02271-0.

[7] Michael Bleyer and Margrit Gelautz. Graph-based surface reconstruction from stereo pairs using image segmentation. In *Proceedings of SPIE conference*, 2005.

[8] Edmond Boyer and Jean-Sébastien Franco. A hybrid approach for computing visual hulls of complex objects. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, volume 1, pages 695–701, June 2003.

[9] Yuri Boykov and Vladimir Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 26–33. IEEE, October 2003.

[10] Yuri Boykov and Vladimir 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.

[11] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[12] Andrian Broadhurst, Tom Drummond, and Roberto Cipolla. A probabilistic framework for space carving. In *Proceedings of the International Conference on Computer Vision*, pages 388–393. IEEE, July 2001.

[13] Chris Buehler, Steven Gortler, Michael Cohen, and Leonard McMillan. Minimal surfaces for stereo. In *Proceedings of the European Conference on Computer Vision*, 2002.

[14] Rodrigo L. Carceroni and Kiriakos 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*, volume 2. IEEE, 2001.

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

[16] Roberto Cipolla and Kwan-Yee K. Wong. Reconstruction of sculpture from its profiles with unknown camera positions. *IEEE Transactions on Image Processing*, 13(3):381–389, March 2004.

[17] W. Bruce Culbertson, Thomas Malzbender, and Gregory G. Slabaugh. Generalized voxel coloring. In *Proceedings of the International Workshop on Vision Algorithms*, Lecture Notes on Computer Science, pages 100–115. Springer Verlag, September 1999.

[18] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the SIGGRAPH conference*. ACM, 1996.

[19] Jeremy S. de Bonet and Paul Viola. Poxels: Probabilistic voxelized volume reconstruction. In *Proceedings of the International Conference on Computer Vision*. IEEE, 1999.

[20] Carlos Hernández Esteban and Francis Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, December 2004.

[21] Olivier Faugeras and Renaud Keriven. Complete dense stereovision using level set methods. *IEEE Transactions on Image Processing*, 7(3), 1998.

[22] Pascal Fua. From multiple stereo views to multiple 3-D surfaces. *International Journal of Computer Vision*, 24(1):19–35, 1997.

[23] William Hoff and Narendra Ahuja. Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 121–136, 1989.

[24] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics journal*, 26(2), 1992. Proceedings of the ACM SIGGRAPH conference.

[25] Hiroshi Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, October 2003.

[26] John Isidoro and Stan Sclaroff. Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints. In *Proceedings of the International Conference on Computer Vision*, pages 1335–1342. IEEE, 2003.

[27] Hailin Jin, Stefano Soatto, and Anthony J. Yezzi. Multi-view stereo beyond lambert. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 171–178 vol.1, 2003.

[28] Hailin Jin, Anthony J. Yezzi, and Stefano 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.

[29] Hailin Jin, Anthony J. Yezzi, Yen-Hsi Tsai, Li-Tien Chen, and Stefano Soatto. Estimation of 3D surface shape and smooth radiance from 2D images: a level set approach. *Journal of Scientific Computing*, 19(1-3):267–292, 2003.

[30] Danil Kirsanov and Steven J. Gortler. A discrete global minimization algorithm for continuous variational problems. Technical Report TR-14-04, Harvard Computer Science, July 2004.

[31] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, February 2004.

[32] Vladimir Kolmogorov, Ramin Zabih, and Steven 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.

[33] Kiriakos N. Kutulakos. Approximate N-view stereo. In *Proceedings of the European Conference on Computer Vision*, pages 67–83, 2000.

[34] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[35] Aldo Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, February 1994.

[36] Patrick Le Tallec. *Computational Mechanics Advances*, volume 1, chapter Domain Decomposition Methods in Computational Mechanics, pages 123–217. North Holland, 1994.

[37] Maxime Lhuillier and Long Quan. Surface reconstruction by integrating 3D and 2D data of multiple views. In *Proceedings of the International Conference on Computer Vision*. IEEE, October 2003.

[38] Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):418–433, 2005.

[39] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the SIGGRAPH conference*, pages 163–169. ACM, 1987.

[40] Wojciech Matusik, Chris Buehler, and Leonard McMillan. Polyhedral visual hulls for real-time rendering. In *Proceedings of the Eurographics Workshop on Rendering*, 2001.

[41] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. *Transactions on Graphics*, 22(3):463–470, 2003. Proceedings of the SIGGRAPH conference.

[42] Sylvain Paris, François Sillion, and Long Quan. A surface reconstruction method using global graph cut optimization. In *Proceedings of the Asian Conference of Computer Vision*, January 2004.

[43] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions Pattern Analysis Machine Intelligence*, 12(7):629–639, July 1990.

[44] Sébastien Roy. Stereo without epipolar lines: A maximum-flow formulation. *International Journal of Computer Vision*, 34(2/3):147–162, August 1999.

[45] Sébastien Roy and Ingemar J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proceedings of the International Conference on Computer Vision*, pages 492–499. IEEE, January 1998.

[46] Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1067–1073. IEEE, 1997.

[47] Steve Sullivan and Jean Ponce. Automatic model construction and pose estimation from photographs using triangular splines. *IEEE Transactions Pattern Analysis Machine Intelligence*, 20(10):1091–1097, 1998.

[48] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. *International Journal of Computer Vision*, 32(1):45–61, 1999.

[49] Richard Szeliski and Richard Weiss. Robust shape recovery from occluding contours using a linear smoother. In *Proceedings of the conference on Computer Vision and Pattern Recognition*. IEEE, June 1993.

[50] Demetri Terzopoulos, Andrew Witkin, and Michael Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.

[51] Régis Vaillant and Olivier Faugeras. Using extremal boundaries for 3-D object modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):157–173, 1992.

[52] G. Vogiatzis, P.H.S. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2005.

[53] Yichen Wei and Long Quan. Region-based progressive stereo matching. In *Proceedings of the conference on Computer Vision and Pattern Recognition*, 2004.

[54] Gang Zeng, Sylvain Paris, Long Quan, and Maxime Lhuillier. Surface reconstruction by propagating 3d stereo data in multiple 2d images. In *Proceedings of the European Conference on Computer Vision*, 2004.

[55] Gang Zeng, Sylvain Paris, Long Quan, and François Sillion. Progressive surface reconstruction from images using a local prior. In *Proceedings of the International Conference on Computer Vision*, 2005.

[56] Remo Ziegler, Wojciech Matusik, Hanspeter Pfister, and Leonard McMillan. 3D reconstruction using labeled image regions. In *Proceedings of the Eurographics Symposium on Geometry Processing*, 2003.

[57] Charles Lawrence Zitnick, Sing Bing Kang, Matt Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics*, 23(3), July 2004. Proceedings of the SIGGRAPH conference.