Computer Vision and Image Understanding 114 (2010) 334–348

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Coarse-to-fine surface reconstruction from silhouettes and range data using mesh deformation

Y. Sahillioğlu, Y. Yemez *

Multimedia, Vision and Graphics Laboratory, College of Engineering, Koç University, Sarıyer, Istanbul 34450, Turkey

ARTICLE INFO

Article history:
Received 18 December 2007
Accepted 3 December 2009
Available online 8 January 2010

Keywords:
Coarse-to-fine surface reconstruction
Shape from silhouette
Shape from optical triangulation
Mesh-based deformation
Line-of-sight integration

ABSTRACT

We present a coarse-to-fine surface reconstruction method based on mesh deformation to build watertight surface models of complex objects from their silhouettes and range data. The deformable mesh, which initially represents the object visual hull, is iteratively displaced towards the triangulated range surface using the line-of-sight information. Each iteration of the deformation algorithm involves smoothing and restructuring operations to regularize the surface evolution process. We define a non-shrinking and easy-to-compute smoothing operator that fairs the surface separately along its tangential and normal directions. The mesh restructuring operator, which is based on edge split, collapse and flip operations, enables the deformable mesh to adapt its shape to the object geometry without suffering from any geometrical distortions. By imposing appropriate minimum and maximum edge length constraints, the deformable mesh, hence the object surface, can be represented at increasing levels of detail. This coarse-to-fine strategy, that allows high resolution reconstructions even with deficient and irregularly sampled range data, not only provides robustness, but also significantly improves the computational efficiency of the deformation process. We demonstrate the performance of the proposed method on several real objects.

1. Introduction

Shape from optical triangulation produces accurate range points but often fails to create complete and watertight reconstructions. The most prominent problem with active range scanning systems is that range points can only be collected from the observed portions of the object surface. The sensor may not be able to access the obstructed sections of the object (camera occlusion) or there may be parts of the object surface that cut off the projected light and prevent other parts from getting illuminated (laser occlusion), which eventually, cause the surface reconstruction to be incomplete. Shape from silhouette, on the other hand, as a passive reconstruction technique, yields robust, hole-free reconstruction of the visual hull of the object. In this paper, we present a surface reconstruction method that combines shape from silhouette with optical triangulation to build watertight, yet cavity-sensitive, object models.

In Fig. 1, we demonstrate our motivation on an example. Neither the shape from optical triangulation nor the shape from silhouette technique alone can satisfactorily recover the shape of the Hand object displayed in Fig. 1a. These two techniques actually provide complementary information about the surface geometry. The silhouette model displayed in Fig. 1b lacks concavities of the palm, but the fingers are successfully recovered. On the other hand, the optical triangulation model (Fig. 1c) exhibits holes due to severe occlusions. The fingers not only occlude each other, but also obstruct the palm on which the cavity information can only partially be recovered.

The range data actually contains more information about the object shape than the one represented as a triangulated surface. The triangle mesh obtained by shape from optical triangulation is the part of the true object surface that can be sensed by the scanner without any obstruction. Thus the range data also contains information about where the object surface cannot lie. Each range point sampled on the object, that appears as a vertex on the optical triangulation model, defines a line segment (or line-of-sight) that joins the range point and its projection on the camera screen. No part of the object surface or volume normally intervenes these line segments. This is depicted in Fig. 1d, where we display the lines of sight intersecting the silhouette model at the palm of the Hand. Even on some parts of the palm where no range point was sampled, there are lines of sight that intersect the surface. These line segments actually target at different locations on the surface, but the way they trace the space provides additional information about the cavities of the palm. Hence starting with the silhouette model and carving out the portions traced by these line segments, it is possible to recover some of the cavities that are missing on both optical triangulation and silhouette models.

* Corresponding author. Fax: +90 212 3381548.
E-mail addresses: ysahillioglu@ku.edu.tr (Y. Sahillioğlu), yyemez@ku.edu.tr (Y. Yemez).

1077-3142/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2009.12.003

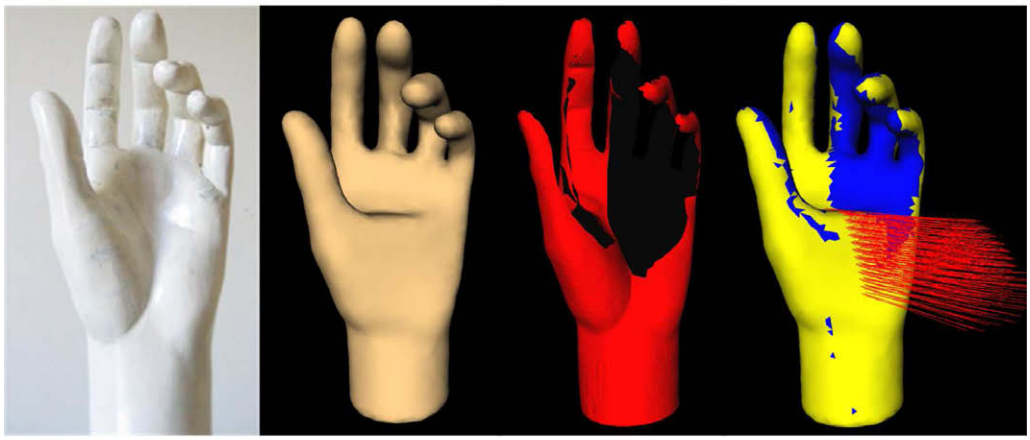

Fig. 1. (a) Original Hand image, (b) silhouette model, (c) triangulated range surface acquired with a single-camera single-stripe rotational scanner, (d) silhouette model painted with range data. Yellow color indicates sections of the silhouette model intersected by lines of sight whereas blue indicates sections with no intersection. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Our aim is to avoid holes by exploiting as much information as possible from range data and silhouette images and to build aesthetically pleasant 3D models of objects. To achieve this goal, we combine line-of-sight and silhouette information using a surface-based method, as opposed to the previously reported techniques which all integrate line-of-sight information in a volumetric fashion, such as in [1–5]. The major contribution of this work is a robust and computationally efficient mesh-based deformation scheme for surface reconstruction from sparse and deficient range data, that integrates silhouette and line-of-sight information in a coarse-to-fine framework. We use a deformable model, which is based on *dynamic triangle meshes* [6], that enables us to control topology, smoothness and resolution of the reconstruction.

The organization of the paper is as follows. In Section 2, we discuss the related work. In Section 3 we provide an overview of our reconstruction system. In Section 4, we describe our deformable model that we use later in Section 5 to combine silhouette and optical triangulation geometries. We present our experimental results in Section 6 and finally give some concluding remarks in Section 7.

2. Related work

In the last decade, a considerable amount of research has been devoted to enhancement of the active range scanning systems. Park et al. [7] explain how to integrate several laser projectors to reduce light occlusions, and Liska and Sablatnig [8] review next view planning techniques to optimize the surface coverage. Levoy et al. [9] have used large-scale enhanced acquisition systems to overcome the occlusion problem presenting very accurate and successful results. Rusinkiewicz et al. [10] describe a real-time acquisition system allowing to scan objects faster and with greater ease than conventional model acquisition pipelines. Despite all these enhancement efforts, and although some scanners perform better than others in generating more complete surfaces, final surface reconstructions for some objects always contain holes, no matter what kind of scanning configuration is implemented or how many scans are run. The use of multiple laser sources or cameras, and even mirrors [11], help improve the reconstructions but some portions of the object surface such as the inner walls of hollow parts are at best unreliably reconstructed.

There exist different ways of handling holes that appear in surface reconstructions. The common practice is to smoothly extrapolate the missing surface information either after reconstruction via hole filling [12] or during reconstruction [13–18]. These techniques work on point sets and perform very well when the holes on the range surface are small relative to the entire object. However they do not consider line-of-sight information which is very valuable especially in the presence of large holes due to severe occlusions. Curless and Levoy [1] propose a reconstruction technique that extracts a hole-free surface from range data in a more accurate way incorporating also the line-of-sight information which is discarded in previously reported reconstruction algorithms. The technique is based on volumetric space carving and creates a maximal surface which is consistent with the lines of sight derived from range images. However, at parts of the surface where the range data is severely occluded or very sparse, such as those sections that are visible by the sensor only at sharp angles, the algorithm fails to produce reliable reconstructions. They propose to shape such difficult parts of the object by further carving via background extraction without explicitly introducing the silhouette geometry. Another volumetric technique that uses line-of-sight information for surface reconstruction is Whitaker's work [2] which is based on level-set approach. Since the technique employs a deformation framework, the integration of range data is smoother as compared to the technique proposed in [1]. But the silhouette information is not considered at all and the problem of surface reconstruction from deficient and sparse range data remains unaddressed. As a result, Whitaker presents reconstruction results only on relatively simple shaped objects with almost complete range data that is free of large holes.

Early examples of the shape from silhouette technique were presented by Chien and Aggarwal [19] and later much improvement has been established concerning efficiency and space constraint issues [20–23]. In general, the strength of the technique lies in its simplicity, efficiency and robustness especially when applied to convex shapes. The drawback of this method is that it fails to capture hidden concavities. The robust output of the shape from silhouette technique however constitutes a solid initial foundation for further carving. This observation has led to several attempts to combine shape from silhouette with other techniques which alone do not produce complete or fully reliable model reconstructions. The common strategy for integrating silhouette information with others is to start off with an initial estimate of the object shape in the form of a visual hull obtained from its silhouettes and gradually recover the cavities with a cavity-sensitive method. Several researchers such as [24–27] fuse shape from silhouette and shape

from stereo in a volumetric fashion while others such as [28–31] adhere to surface deformation models for further enhancing the description of the silhouette model.

So far there have been very few attempts to explicitly combine the shape from silhouette technique with shape from optical triangulation. All prior techniques such as [3–5] are based on volumetric space carving, hence they are liable to topological problems and very sensitive to sensor noise, and thus do not permit high resolution reconstructions. They consider a fixed partitioning of the space and rely too much on the regularity of the range samples. However, the range sample density is hardly ever uniform throughout the object surface and the range data may indeed be deficient due to various factors such as scanner noise, occlusions, scanner configuration and surface orientation. Deficient range data, slight discrepancies between silhouette and optical triangulation geometries and missing range samples in the hidden surface cavities often result in protrusions and discontinuous surface transitions, and even topological problems, throughout the reconstructed model. Volumetric space carving techniques, especially in the presence of severe occlusions, cannot adequately address these challenges and seriously suffer from imperfections of the scanning system for which they have no cure other than post-smoothing.

Deformation-based techniques, either volumetric or surface-based, seem to be capable of overcoming the limitations of volumetric space carving. One alternative is the use of volumetric level-set approach which however suffers from certain drawbacks for surface modeling applications such as computational inefficiency, limited resolution and the lack of any direct surface representation during deformation [2]. Although it is possible to modify topology with level-set based deformation, it is usually difficult to control the changes that the initial topology may undergo, especially in the case of modeling complex objects from sparse and deficient range data. In this paper, we use a mesh-based deformation scheme that yields a computationally efficient surface reconstruction algorithm. We initially recover the correct topology of the object using the shape from silhouette method, hence the topology does not need to be modified during deformation. Moreover, the mesh-based deformation framework provides continuous levels of detail, hence allows the adjustment of the resolution in a flexible manner and thereby avails us to build a robust coarse-to-fine surface reconstruction algorithm.

3. System overview

The block diagram of our overall reconstruction system is given in Fig. 2. The basic tasks are briefly explained in the sequel.

3.1. Data acquisition

Our range scanner is a single-camera, single-stripe optical triangulation system, supporting only rotational movement of the object (see Fig. 3). Hence it does not require any alignment of patches as compared to systems that also incorporate translational movement, but in turn the surface coverage for a single scan is limited. In order to improve the surface coverage, we use several projectors and obtain separate scans from different angles. The components that make up the acquisition system, the laser projectors, the camera, and the turntable are calibrated with respect to a common reference frame. We should also note that, since the object is rotated with respect to the projector, the projected laser planes intersect in space, yielding irregular sampling of the object and may hence result in deficient range data at some parts of the surface.

3.2. Silhouette extraction

Our silhouette extraction method is based on the use of a sharp contrast maintained between a backdrop and the object [22]. The backdrop is saturated with light while the object is left in the dark creating a natural silhouette of the object. Simple thresholding then suffices to obtain the correct silhouette. For a successful extraction, the strength of the light sources and the camera settings have to be fine-tuned. The background saturation method, although problematic with very shiny surfaces, in general produces very clean and accurate results with radiosity and color confusion problems circumvented.

3.3. Shape from silhouette

The initial silhouette model can be obtained by any shape from silhouette technique that produces a topologically correct shape model which is eligible for further deformation. The shape from

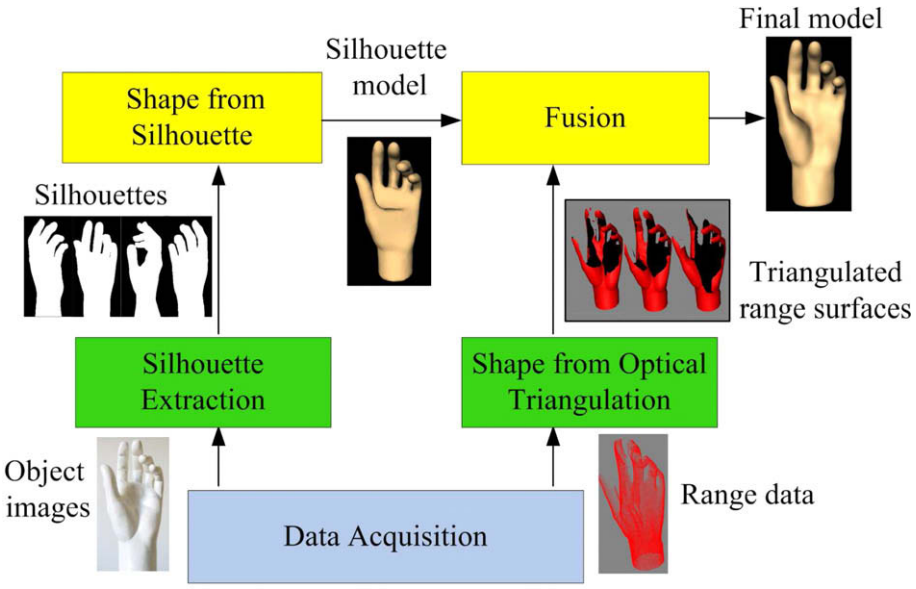

Fig. 2. Block diagram of the overall reconstruction system.

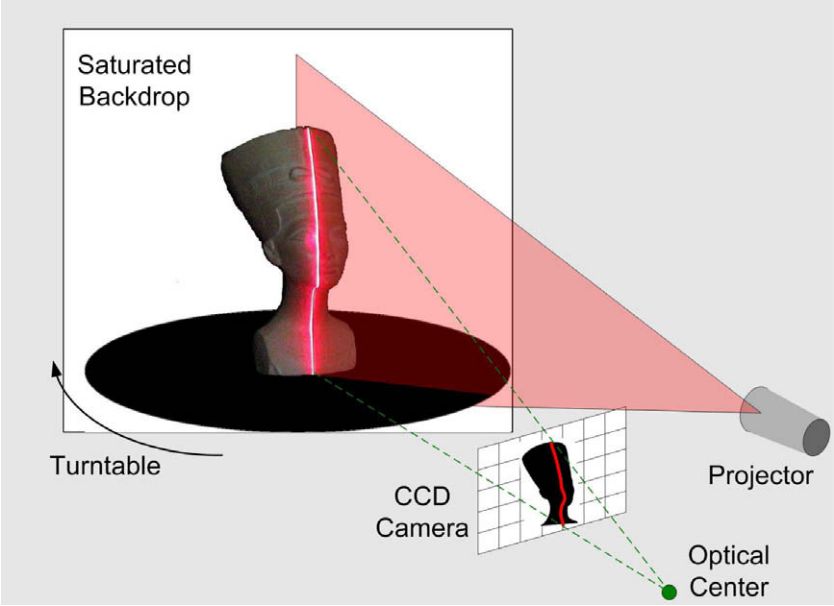

Fig. 3. Scanning setup.

silhouette technique that we employ is based on our two previous works. In [32], we construct the initial mesh by deforming the bounding sphere with a silhouette-driven snake-based deformable model. This method is however applicable only to objects of genus zero. Hence for building silhouette models of objects with higher genus, we employ the technique described in [22], which is based on iso-surface triangulation of an intermediate volumetric representation of the visual hull.

3.4. Shape from optical triangulation

The input to our shape from optical triangulation process is a series of laser images of the object in full rotation on the turntable. The routine starts with processing the laser images to draw 2D sample points from the laser stripes. A depth profile is computed from the sample points, creating a cloud of range points. Connecting nearest neighbors with triangles is a common strategy for systems that produce lattices of regular samples. The triangulation method that we use undertakes a similar path by weaving a web of faces across range points of adjacent stripes.

3.5. Fusion

The silhouette mesh, which represents the visual hull, is deformed towards the object surface under the guidance of the range points sampled on the object surface and their lines of sight. The fusion technique that will be described in detail in Section 5 does not explicitly use the connectivity information of the surface meshes generated from optical triangulation. The triangulation of range points rather serves us to estimate a surface normal for each range point. We will make use of these surface normals during the fusion process. The oriented range points resulting from different scans are integrated separately into the deformation process, thus they are not merged into a single surface representation.

4. Deformable model

Our deformation technique is based on iterative use of an appropriate transformation T that deforms an initial triangle mesh M_0 towards the object surface S through the following surface evolution equation:

$$M_{k+1} = T(M_k). \tag{1}$$

The deformable model M_k is required to remain as a smooth topologically correct mesh representation free of geometrical distortions during its evolution and to converge to an optimal mesh M_{k^*} that faithfully represents the object surface S at the equilibrium state,

$$M_{k^*} = T(M_{k^*}). \tag{2}$$

We define T as the composition of three transformations: $T = T_{\mathrm{d}} \circ T_{\mathrm{s}} \circ T_{\mathrm{r}}$, which we will refer to as displacement, smoothing and restructuring operators, respectively. The displacement operator pushes the deformable mesh towards the object surface while the smoothing operator regularizes the effect of this displacement and the restructuring operator modifies the mesh connectivity to eliminate any geometrical distortions that may appear during surface evolution. In this sense, the displacement operator corresponds to the external force whereas the other two correspond to the internal force of the classical snake formulation [33].

4.1. Displacement operator

The distance between the deformable mesh M_k and the object surface S can be approximated by the average distance from the vertex set of M_k to the surface S:

$$\epsilon(M_k, S) = \frac{1}{N_k} \sum_{i=1}^{N_k} d(\mathbf{v}_{i,k}, S), \tag{3}$$

where N_k is the number of vertices of the deformable mesh at iteration k, $\mathbf{v}_{i,k}$ is the position vector of the ith vertex, and $d(\mathbf{v}_{i,k}, S)$ is the Euclidean distance of the vertex $\mathbf{v}_{i,k}$ to the surface S. To reduce the distance $\epsilon(M_k, S)$, the operator $T_{\mathrm{d}}(M_k)$ maps the deformable mesh M_k to M'_k by moving each vertex $\mathbf{v}_{i,k}$ with a displacement $\mathbf{d}(\mathbf{v}_{i,k})$:

$$\mathbf{v}'_{i,k} = \mathbf{v}_{i,k} + \mathbf{d}(\mathbf{v}_{i,k}), \tag{4}$$

where $\{\mathbf{v}'_{i,k}\}$ is the vertex set of the transformed mesh M'_k which has the same connectivity as M_k. The displacement is set to be in the direction of the surface normal $\mathbf{N}(\mathbf{v}_{i,k})$, inwards or outwards,

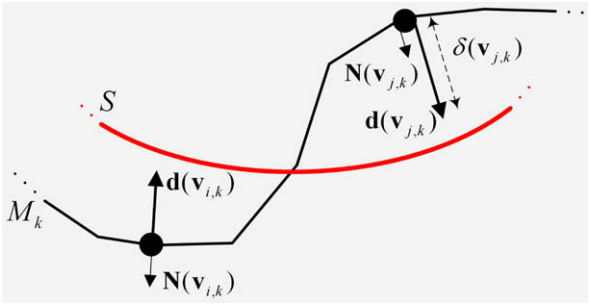

Fig. 4. Illustration of the displacement operator in 2D. The displacement $\mathbf{d}(\mathbf{v}_{i,k})$ on each vertex $\mathbf{v}_{i,k}$ of the deformable mesh M_k at iteration k is set to be in the direction of the surface normal $\mathbf{N}(\mathbf{v}_{i,k})$ and computed based on the signed distance from the vertex to the target surface S.

depending on positioning of the vertex with respect to the target surface S (see also Fig. 4):

$$\mathbf{d}(\mathbf{v}_{i,k}) = \delta(\mathbf{v}_{i,k}) \cdot \mathbf{N}(\mathbf{v}_{i,k}). \tag{5}$$

The displacement scalar $\delta(\mathbf{v}_{i,k})$ is computed based on the signed distance from the vertex $\mathbf{v}_{i,k}$ to the target surface S, as will later be explained in detail in Section 5.2. Also note that the magnitude of the displacement has to be bounded above for a stable surface evolution.

The distance function defined in Eq. (3) is only a discrete approximation of the true distance. Moreover it does not take into account the distance from the surface S to M_k. Hence the optimality of the surface obtained at convergence heavily depends on two factors, the resolution and the location of the initial deformable mesh M_0. If the initial mesh is of sufficiently high resolution and placed so as to bound the object surface S, the surface evolution is expected to converge to an optimal surface that accurately represents the target surface.

4.2. Smoothing operator

The smoothing operator, T_s, is necessary for a robust mesh evolution that is free of topological errors and to have eventually a visually pleasant fair surface representation. It should be easy to compute, yet must not yield any geometrical shrinkage and bias in the final surface estimate. To achieve this, we employ a combination of the classical Laplacian smoothing and Taubin's surface fairing technique [34].

The operator $T_s(M)$ maps the deformable mesh M to M' by moving each vertex $\mathbf{v}$ to $\mathbf{v}'$ (dropping the vertex index i and the iteration index k):

$$\mathbf{v}' = \mathbf{v} + \Delta\mathbf{v}_{\mathrm{T}} + \Delta\mathbf{v}_{\mathrm{N}}, \tag{6}$$

where the displacements $\Delta\mathbf{v}_{\mathrm{T}}$ and $\Delta\mathbf{v}_{\mathrm{N}}$ correspond to smoothing along tangential and normal directions of the surface, respectively. We obtain the tangential component, $\Delta\mathbf{v}_{\mathrm{T}}$, by tangential Laplacian smoothing [35]:

$$\Delta\mathbf{v}_{\mathrm{T}} = \mathbf{L}(\mathbf{v}) - (\mathbf{L}(\mathbf{v}) \cdot \mathbf{N})\mathbf{N}, \tag{7}$$

where $\mathbf{L}(\mathbf{v})$ denotes the Laplacian displacement that moves the vertex $\mathbf{v}$ to the centroid of the vertices in its one-ring neighborhood:

$$\mathbf{L}(\mathbf{v}) = \frac{1}{N_v}\sum_{i=1}^{N_v}\mathbf{v}_i - \mathbf{v}, \tag{8}$$

where N_v is the number of the vertices $\mathbf{v}_i$ which are adjacent to $\mathbf{v}$.The component $\Delta\mathbf{v}_{\mathrm{N}}$, on the other hand, is obtained by fairing the surface along its normal direction:

$$\Delta\mathbf{v}_{\mathrm{N}} = (\mathbf{F}(\mathbf{v}) \cdot \mathbf{N})\mathbf{N}, \tag{9}$$

where $\mathbf{F}(\mathbf{v})$ denotes the displacement created by the non-shrinking surface fairing algorithm described in [34]. The displacement $\mathbf{F}(\mathbf{v})$ is obtained at two steps. At the first step, each vertex $\mathbf{v}$ is moved to $\mathbf{v}^{(1)}$ by

$$\mathbf{v}^{(1)} = \mathbf{v} + \lambda\sum_{i=1}^{N_v} w_i(\mathbf{v}_i - \mathbf{v}), \tag{10}$$

where $\{\mathbf{v}_i\}$ are the vertices adjacent to $\mathbf{v}$. Each weight w_i is chosen as proportional to the inverse of the corresponding edge length $\|\mathbf{v}_i - \mathbf{v}\|$ and then normalized so as to meet $\sum_i w_i = 1$. The factor $\lambda, 0 \leqslant \lambda \leqslant 1$, controls the amount of shrinkage. At the second step, to counter the shrinking effect of the first step, the same operation is repeated using this time a negative factor μ such that $\mu < -\lambda$. Hence each vertex $\mathbf{v}^{(1)}$ is moved to its final position $\mathbf{v}^{(2)}$ by

$$\mathbf{v}^{(2)} = \mathbf{v}^{(1)} + \mu\sum_{i=1}^{N_v} w_i(\mathbf{v}_i^{(1)} - \mathbf{v}^{(1)}). \tag{11}$$

The overall fairing displacement $\mathbf{F}(\mathbf{v})$ for each vertex is then computed by $\mathbf{v}^{(2)} - \mathbf{v}$.

The fairing procedure described above corresponds to smoothing the surface in the direction of its normal using a low-pass filter with the transfer function $f(k) = (1 - \lambda k)(1 - \mu k)$, where $k \in [0, 2]$. The parameters λ and μ determine the pass-band frequency, k_{PB}, of the low-pass filter by $k_{\mathrm{PB}} = \frac{1}{\lambda} + \frac{1}{\mu} > 0$ [34]. Once the pass-band frequency is fixed, the parameter λ is chosen as large as possible in order to make the transfer function decay fast in the high-frequency band. One way to achieve this is to set $f(1) = -f(2)$ which ensures a fast and stable filter for $k_{\mathrm{PB}} < 1$, as proposed in [34]. The filter can then be designed with varying values of $k_{\mathrm{PB}} \in (0, 1)$. In all our reconstructions, we set $k_{\mathrm{PB}} \approx 0.1$ with $\lambda = 0.6307$ and $\mu = -0.6732$, as suggested in [34]. We have also conducted experiments to verify the optimality of this choice in our case. These experiments will be presented in Section 6.

4.3. Restructuring operator

The restructuring operator, T_{r}, is the composition of three operators:

$$T_{\mathrm{r}} = T_{\mathrm{flip}} \circ T_{\mathrm{col}} \circ T_{\mathrm{split}}, \tag{12}$$

where $T_{\mathrm{split}}, T_{\mathrm{col}}$ and T_{flip} are defined in terms of edge split, edge collapse and edge flip transformations which were introduced by Hoppe et al. [36] for mesh optimization. We use these elementary transformations in the way Kobbelt et al. [6] use them for mesh editing. At the end of each iteration of the surface evolution, the operator T_{split} first splits all edges longer than $\varepsilon_{\max}$ at their midpoints. Then, the operator T_{col} successively eliminates all edges shorter than $\varepsilon_{\min}$ by edge collapses. Finally, the operator T_{flip} is applied to reduce the number of irregular vertices possibly created by the previous collapse and split operations: The common edge of any two neighboring triangles is swapped with the one joining the unshared vertices of the triangles, if this operation increases the number of vertices with valence close to 6. For the split operation to be compatible with the collapse operation, the threshold $\varepsilon_{\max}$ has to be chosen such that $\varepsilon_{\max} \geqslant 2\varepsilon_{\min}$ since otherwise split operations would create edges with length smaller than $\varepsilon_{\min}$. We set $\varepsilon_{\max} = 2\varepsilon_{\min}$ to have uniformly sized triangles with small aspect ratios. Since the edge length ratio is then bounded by $\varepsilon_{\max}/\varepsilon_{\min} = 2$ and the valence distribution preserves its uniformity by flip operations, the deformable mesh maintains a high quality in terms of the aspect ratio of the triangles during surface evolution.

The minimum and maximum edge length requirements as well as the uniform valence distribution are not actually hard constraints for a stable and robust mesh evolution. For instance, flip operations that follow edge splits and collapses at each iteration

may occasionally create invalid edges that are smaller than $\varepsilon_{\min}$ or larger than $\varepsilon_{\max}$. Fortunately such invalid edges do not persist long and usually handled at the next iteration of the algorithm. Other instances of violation of these requirements are, though rarely encountered, illegal collapse and flip operations that are to be avoided and that would otherwise create fold-overs and non-manifold triangulations, as explained in [36].

Thanks to the restructuring operation applied at each iteration of the surface evolution, the deformable mesh can adapt its shape to the object surface, avoiding geometrical distortions such as degenerate triangles and irregular vertices. Note that, with the restructuring operator as formulated above, the surface evolution results in an optimal surface M_{k^*} that has the same topology as the initial mesh M_0 unless explicit topology modifying operators for merging and/or splitting, such as those in [37], are incorporated.

4.4. Coarse-to-fine deformation

The restructuring operator, T_r, can also serve as an effective tool to control the resolution of the deformable mesh. By adjusting the parameter $\varepsilon_{\min}$, the deformable mesh can be represented at any resolution, hence the surface of an object can be recovered at any desired level of detail. The computational complexity of the surface evolution algorithm however is drastically effected by the choice of $\varepsilon_{\min}$ due to two reasons. First, the number of edges (or triangles) of the deformable mesh increases quadratically as $\varepsilon_{\min}$ decreases. Second, the parameter $\varepsilon_{\min}$ puts an upper bound to the magnitude of the vertex displacement $\mathbf{d}(\mathbf{v})$, which we constrain with half of the minimum edge length parameter, i.e., $|\delta(\mathbf{v})| \leqslant \varepsilon_{\min}/2$. In this way, a more stable surface evolution is obtained, but the number of iterations needed for convergence becomes directly proportional to the chosen value of $\varepsilon_{\min}$.

To achieve high resolution reconstructions, yet in a computationally efficient manner, we use a coarse-to-fine deformation scheme. We start with a relatively large value of $\varepsilon_{\min}$ and apply the restructuring operator to put the deformable mesh initially into a low resolution form. We iterate the surface evolution equation (see Eq. (1)) until we obtain a coarse mesh representation of the surface, M_{k^*}, which is optimal at this resolution (see Eq. (2)). This optimal surface M_{k^*} then becomes the initial deformable mesh for the next run of the surface evolution algorithm with a smaller value of $\varepsilon_{\min}$, that is, at some higher resolution. The surface evolution algorithm is successively rerun in this manner with a predefined sequence of decreasing values of $\varepsilon_{\min}$ until the desired precision is obtained.

5. Fusion

The goal of the fusion process is to combine the 3D shape information provided by the silhouette images and the range data in the smoothest possible manner. This goal can be achieved by using the multiresolution deformable model described in Section 4. The most critical part of the fusion algorithm is to determine the appropriate displacements that will deform the initial silhouette model towards its final shape as faithfully as possible to the true object surface at a given resolution. This requires, prior to the deformation process, identifying the lines of sight that intersect the initial silhouette model. We will refer to these line segments as *carvers* of the fusion process.

5.1. Carver assignment

A carver, that is, a line segment that joins a range point and its projection on the camera screen ideally intersects the true object surface only at the range point itself. However since the true object surface lies inside the visual hull, some of these carvers may (and usually do) intersect the initial silhouette model at multiple locations. These carvers and their intersections with the model give indication of where, how much and in which direction the surface is to be deformed. Some triangles may have multiple intersections with carvers while others may have none. Triangles with no carver intersection correspond to those parts of the shape where we can rely only on silhouette information.

We associate each triangle of the deformable mesh, whenever available, with a carver that satisfies the following three conditions:

- The angle between the triangle normal and the carver direction (originating from the camera optical center and pointing to the range point) takes a value in the interval $[0, \pi/2]$.
- The carver intersects the triangle; or the line of the carver intersects the triangle and the associated range point lies in the vicinity of the triangle.
- The intersection of the triangle plane with the line of the carver is the closest to the triangle centroid as compared to all other intersections with eligible carvers.

Thus a carver, even if it does not intersect a given triangle, can be assigned to the triangle provided that its range point is close enough to that triangle (recall that carvers are line segments, not lines). In this way, the range points that systematically fall into the outside vicinity of the silhouette model (possibly due to imperfections of the acquisition system) are also taken into account. "The vicinity of a triangle" can be defined based on the current resolution of the deformable mesh. We define it as the sphere neighborhood of the triangle centroid with radius equal to the current value of $\varepsilon_{\min}$. Also note that, by definition, while at most one carver can be assigned to a given triangle, a carver may be assigned to multiple triangles. In Fig. 5, we illustrate the carver assignment process for three distinct cases.

The carver assignment process involves identifying ray-plane intersections which can efficiently be computed [38]. Moreover carver assignment is initially performed on a coarse silhouette model that is restructured (or already reconstructed) using a relatively large value of $\varepsilon_{\min}$ as compared to the resolution of the range data. The computational efficiency of the initial carver assignment algorithm can further be improved by exploiting the fact that nearby triangles in space have also nearby carvers with respect to the scanning order of the range stripes.

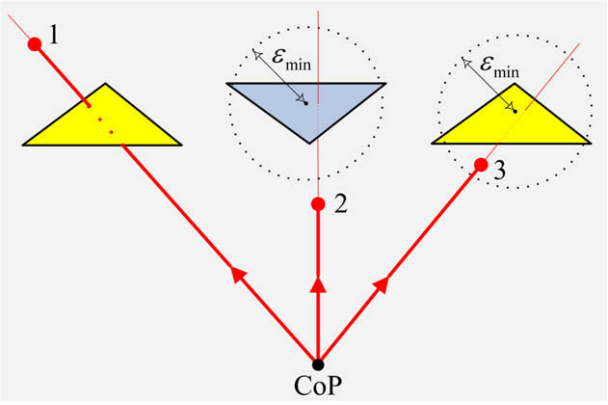

Fig. 5. Illustration of the carver assignment process for three distinct cases. The first carver intersects the leftmost triangle, hence it is an eligible carver for that triangle. The lines of the second and the third carvers intersect the middle and the rightmost triangles, respectively, but the carvers themselves do not. The second carver is not an eligible carver since its range point is not in the vicinity of the associated triangle whereas the third carver is an eligible one since its range point lies in the vicinity of the triangle.

5.2. Computation of displacement

Let $\{C_l\}, l = 1, 2, \ldots, L$, denote the carvers assigned to the triangles that share the vertex $\mathbf{v}$, where L is the number of these carvers. Each carver C_l, associated with a range point $\mathbf{r}_l$, exerts a displacement $\mathbf{d}_l(\mathbf{v})$ on the vertex $\mathbf{v}$:

$$\mathbf{d}_l(\mathbf{v}) = \delta_l(\mathbf{v})\mathbf{N}(\mathbf{r}_l), \tag{13}$$

where $\mathbf{N}(\mathbf{r}_l)$ represents the unit vector that is normal to the optical triangulation surface at the range point $\mathbf{r}_l$. Each displacement component, $\delta_l(\mathbf{v})$, is determined based on the Euclidean signed distance of the vertex $\mathbf{v}$ to the plane P_l which is tangent to the optical triangulation surface at the range point. Hence, the target surface S is locally approximated with the tangent plane of the range point as illustrated in Fig. 6. The average of these displacements, when projected to the surface normal of the deformable mesh, determines the overall displacement $\mathbf{d}(\mathbf{v})$ at the vertex $\mathbf{v}$ (see also Eq. (5)):

$$\mathbf{d}(\mathbf{v}) = \left(\frac{1}{L}\sum_{l=1}^{L}\mathbf{d}_l(\mathbf{v})\cdot\mathbf{N}(\mathbf{v})\right)\cdot\mathbf{N}(\mathbf{v}) \tag{14}$$

The magnitude of each displacement $\delta_l(\mathbf{v})$, hence the magnitude of the overall displacement $\mathbf{d}(\mathbf{v})$, is bounded above by $\varepsilon_{\min}/2$. We note that if a triangle has no carver, its contribution to the displacements of its vertices is assumed to be zero.

The above formulation for computing the displacement $\mathbf{d}(\mathbf{v})$ requires the normal information on the range surface, which we estimate based on the optical triangulation model. Alternatives also exist in case a triangulated range surface is not available but only a set of range points is provided or in case normal vectors are too much influenced by sensor noises and hence are not reliable to be used for surface reconstruction. One such alternative would be, for each displacement component $\delta_l(\mathbf{v})$, to use the signed distance of the range point $\mathbf{r}_l$ to the plane of the triangle to which the corresponding carver $\{C_l\}$ is assigned. The displacement $\mathbf{d}(\mathbf{v})$ could then be computed, though less accurately, in the same way as given in Eq. (14).

We also note that, in our formulation, the displacement $\mathbf{d}(\mathbf{v})$ does not include any component that enforces the final reconstruction to be consistent with the silhouettes. Incorporating silhouette information as a hard constraint [30,31] or as an additional force component [28] may indeed improve reconstructions when fusing it with noisy and unreliable information such as photo consistency (or stereo). However, the range data inferred from optical triangulation, though sparse, is usually very accurate. Hence, in our case, enforcement of silhouette consistency during deformation does not bring in any significant improvement, and might even deteriorate the quality of reconstruction, for instance in case there are slight discrepancies between silhouette and optical triangulation geometries due to imperfections of the acquisition system.

5.3. Update of carvers

As the deformable mesh evolves, its geometry changes. Thus the carver information has to be updated at each iteration. If the carver of a triangle no longer satisfies the properties specified in Section 5.1, a new carver is sought for that triangle. The search for a new carver is fast and affordable, since it suffices to search only in a small scan-order neighborhood of the invalid carver (see Fig. 7). We define the neighborhood of a carver by the lines of sight in its vicinity. The carver neighborhood can easily be deduced from the range-stripe adjacency information. Recall that each range image gives a laser stripe, and each pixel on a stripe gives a range point and a line-of-sight. To support the carver update process, we initially index the lines of sight into a two dimensional array L. Each line-of-sight, $L_{i,j}$, can thus be accessed by specifying two indices, i for the source range image and j for the row of the corresponding stripe pixel (recall that stripes are vertical in our case). Then the carver neighborhood is simply given by the set $\{L_{i+p,j+r}\}$ where p and r are integers ranging from $-W_1$ to W_1 and from $-W_2$ to W_2, respectively. The choice of the size of the neighborhood window, $(2W_1+1)\times(2W_2+1)$, mainly depends on the density of the range samples and the resolution of the deformable model.

As the deformable mesh evolves, its connectivity also changes due to restructuring operations. An instance of carver update is thus due to a split or flip operation that creates new triangles on the restructured mesh. The carver of a new triangle created by a split operation is inherited from its parent and then updated searching within its neighborhood. Likewise, the carver of a triangle created by an edge flip is inherited from one of the triangles that formerly share the flipped edge, whichever is more appropriate, and then locally updated.

The carver update process should ideally be invoked for all the triangles of the deformable mesh. However, that would be too costly to repeat at each iteration since it is not possible to define a carver neighborhood for the triangles currently having no carver. Fortunately, a triangle without carver, which is not neighboring to some triangle that already has one, is very unlikely change its state in one single iteration. Thus, we invoke the update process only for the triangles that already have carvers or that are neighboring to some triangles with carvers. The carver neighborhood for each of these non-carver triangles is obtained from the carver of the neighboring triangle.

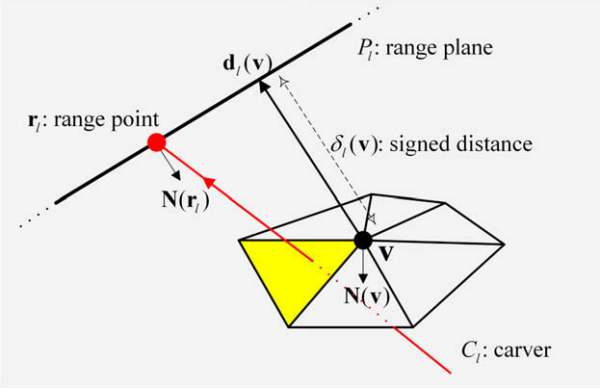

Fig. 6. Signed distance of a vertex to the range surface associated with the range point of a carver.

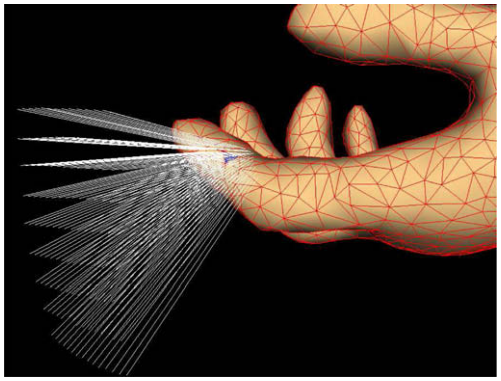

Fig. 7. A typical neighborhood of lines-of-sight around the carver of a triangle.

5.4. Algorithm

The input to the fusion algorithm is the initial silhouette model associated with carvers and one or more triangulated range surfaces. The initial model is iteratively deformed by applying the surface evolution Eq. (1). The algorithm is repeated for a set of decreasing values $\varepsilon_{\min}$. A carver is said to be neighboring to a vertex if it belongs to one of the triangles incident to the vertex. The vertices of the initial deformable mesh, M_0, are all set to be *active* except for those having no neighboring carver. The coarse-to-fine fusion algorithm is then as follows:

For a set of decreasing values of $\varepsilon_{\min}$
  Restructure the initial mesh by $T_r(M_0)$;
  Activate vertices with neighboring carvers;
  Iterate on k
    Update carvers of active triangles;
    Move active vertices by $T_d(M_k)$;
    Smooth active vertices by $T_s(M_k)$;
    Restructure active edges by $T_r(M_k)$;
    Deactivate vertices that no longer move;
  Till convergence
  Set $M_0 = M_{k^*}$;

Note that the displacement and smoothing operators are applied only to active vertices of the deformable mesh whereas the restructuring operator is invoked only for active edges, that is, for edges with at least one active vertex. Likewise the carver update process is applied only to active triangles having at least one active vertex. Thus as iterations proceed and as more and more vertices become inactive, the time spent at each iteration significantly reduces, yielding on overall a computationally efficient algorithm.

There are various alternatives to set the convergence criterion. Ideally, the algorithm converges when the vertices of the deformable mesh no longer move, that is, when the equilibrium condition in Eq. (2) is satisfied. In practice, the algorithm can be terminated if the number of iterations exceeds a given threshold or when the error between the object surface and the deformable mesh falls below a certain value. We terminate the algorithm when the average of the displacements of the active vertices falls below a predefined threshold that is proportional to $\varepsilon_{\min}$; more specifically, the threshold is set to be $\varepsilon_{\min}/15$ in our experiments.

The choice of the initial $\varepsilon_{\min}$ is critical and usually an initial value which is sufficient to represent the visual hull of the object works well in practice. As the deformation algorithm is reiterated by some decreasing values of $\varepsilon_{\min}$ each time at some higher resolution, the details on the surface cavities are gradually recovered. This coarse-to-fine fusion scheme reduces the risk of leaving some parts of the deformable mesh erroneously uncarved and yields a robust surface reconstruction against possible deficiencies in the range data.

6. Results

We have tested our fusion technique on four real objects. The original images of these objects, which are the Elephant (made of wood), the Head (made of stone), the Hand (made of plastic) and the Cup (made of stone) are displayed in Fig. 8. The resolution of the images acquired for both silhouettes and range data is 2000 by 1310 pixels. For reconstruction of the silhouette model of each object, we use a sequence of 72 images that corresponds to a complete turn of 360°. The number of optical scans performed for range data acquisition and the angle step size of the turntable depend on the shape complexity and size of the reconstructed object. For the Hand object that poses severe occlusions, we have conducted three separate scans, each resulting in a sequence of 180 range images for a complete rotation of the object with an angle step size of two degrees. The angle step size is set to be one degree for the Elephant and Head objects, and two degrees for the Cup. We have performed a single scan for the Head and Cup objects, and two separate scans for the Elephant. In all our experiments, we have used a Nikon-D1H camera for image acquisition and a Lasiris focusable non-Gaussian single-line projector (670 nm, 5 mW diode, 45° fan angle) for laser scanning.

6.1. Reconstruction experiments

In Fig. 9 we demonstrate the fusion algorithm on the Head object. In the first row, we display two views from the optical triangulation model obtained from a single scan and the silhouette model associated with carvers of the range data. The second row displays the surface models reconstructed at increasing levels of detail by setting the minimum edge length parameter, $\varepsilon_{\min}$, successively to 0.029, 0.015 and 0.007. The values of $\varepsilon_{\min}$ are normalized with respect to the object size, that is, by the radius of the smallest sphere that bounds the object volume. The last row provides various views from the highest level of detail. We observe that the fusion algorithm satisfactorily recovers the parts that are missing in the optical triangulation model such as the holes on the hair of the Head. This is thanks to the reconstruction algorithm that makes use of not only the range points but also their carvers and the visual hull. The recovered surface information on the cavities that are missing in the optical triangulation model is not necessarily exact, but faithful to the real surface. We also observe that the fine details of the optical triangulation model, even the periodical pattern of vertical ridges, are accurately represented at the highest level of the reconstruction. Note that the artifacts which are in the form of vertical ridges are also visible in the optical triangulation model. These artifacts, which are due to the imperfections of the range data acquisition system, are smoothed out at lower levels of detail. In Fig. 10, we zoom on the reconstructed Head

Fig. 8. Original images of Head, Elephant, Hand and Cup objects.

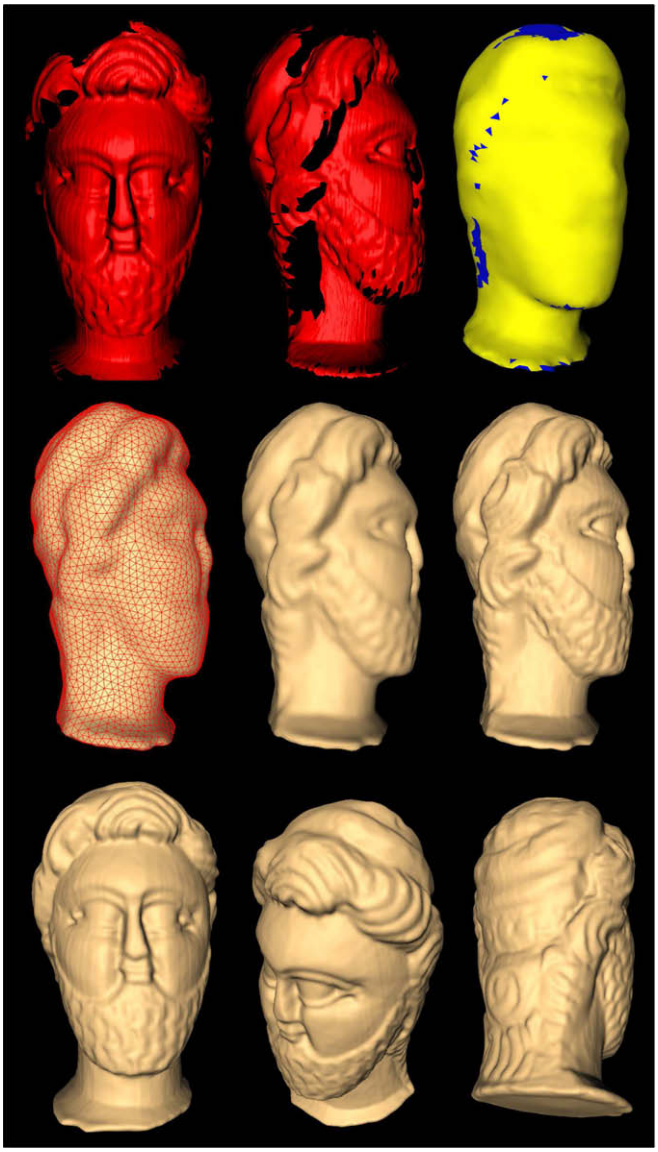

Fig. 9. Head reconstruction. (First row) Two views from the optical triangulation model and the silhouette model associated with range data (yellow and blue colors indicate triangles with and without carvers, respectively). (Second row) Fused models at increasing levels of detail. (Third row) Various views from the highest level of detail. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

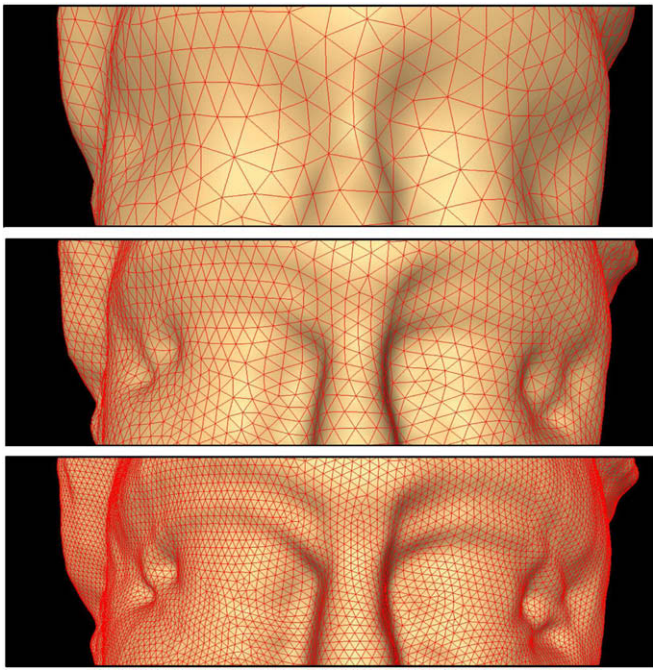

Fig. 10. Zoom on the reconstructed Head model at increasing levels of detail.

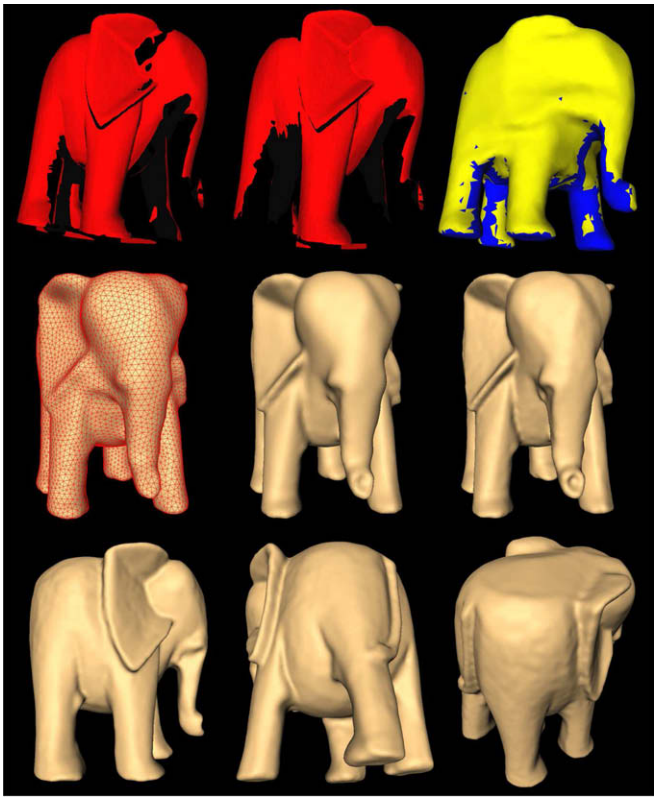

Fig. 11. Elephant reconstruction. (First row) Two separate optical triangulation models and the silhouette model associated with range data. (Second row) Fused models at increasing levels of detail. (Third row) Various views from the highest level of detail.

models to better observe the improvement obtained at each resolution.

The results of the reconstruction process for the Elephant are provided in Fig. 11. In the first row, we display the two optical triangulation models obtained from two separate scans and visualize the initial carver assignment on the silhouette model. The surface models displayed in the second row are reconstructed by setting $\varepsilon_{\min}$ successively to 0.024, 0.017 and 0.010. The last row provides various views from the highest level of detail. We observe that the obstructed inner faces of the legs, the trunk and the top sections of the model, which are missing in the optical triangulation models, are recovered partly with the line-of-sight information and partly with the silhouettes.

The Hand object poses severe occlusions due to the fingers, hence we perform three separate scans to improve the surface coverage. The first row of Fig. 12 displays the three optical triangulation models obtained from these scans along with the silhouette model. We observe that the range scanner still misses some parts of the visible surface that are actually not occluded. This is mainly because our rotational range scanner produces laser planes that

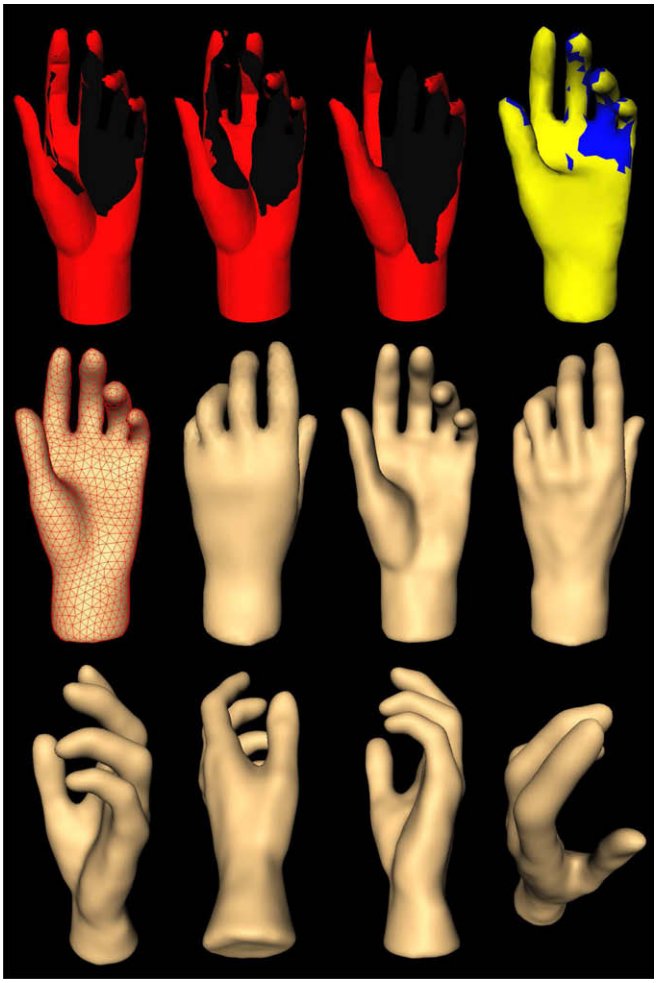

Fig. 12. Hand reconstruction. (First row) Three separate optical triangulation models and the silhouette model associated with range data. (Second row) Fused models at increasing levels of detail. The first two are at level 1 and the other two are at level 2. (Third row) Various views from the highest level of detail.

converge towards certain locations of the object surface. We reconstruct the surface model at two resolutions by setting $\varepsilon_{\min}$ to 0.038 and 0.019. The second row of Fig. 12 displays the models reconstructed at these two levels of detail side by side for comparison. The relatively low resolution of the acquired range data in this case prevents us to reconstruct the surface at higher resolutions. In the last row we provide various views from the highest level of detail. We observe that the fingers of the Hand are mostly recovered by the silhouettes whereas the cavity information of the palm mainly comes from the range data.

In Fig. 13, we demonstrate the reconstruction process for the Cup object. In the first row, we display two views from the optical triangulation model obtained from a single scan, and the silhouette model associated with carvers of the range data. The Cup object is of genus two, having two deep holes on its handles. As we observe from the figure, these holes are missing from our optical triangulation model. Inner parts of such deep hollow surfaces are indeed very difficult to capture even with scanners which are more sophisticated than ours. On the other hand, the initial silhouette model, though it lacks most of the cavity information, satisfactorily recovers these holes, and when combined with range data, yields cavity-sensitive complete surface reconstructions. In the second row of Fig. 13, we display the surface models reconstructed at increasing levels of detail by setting $\varepsilon_{\min}$ successively to 0.036, 0.013 and 0.007. In the last row we provide various views from the highest level of detail.

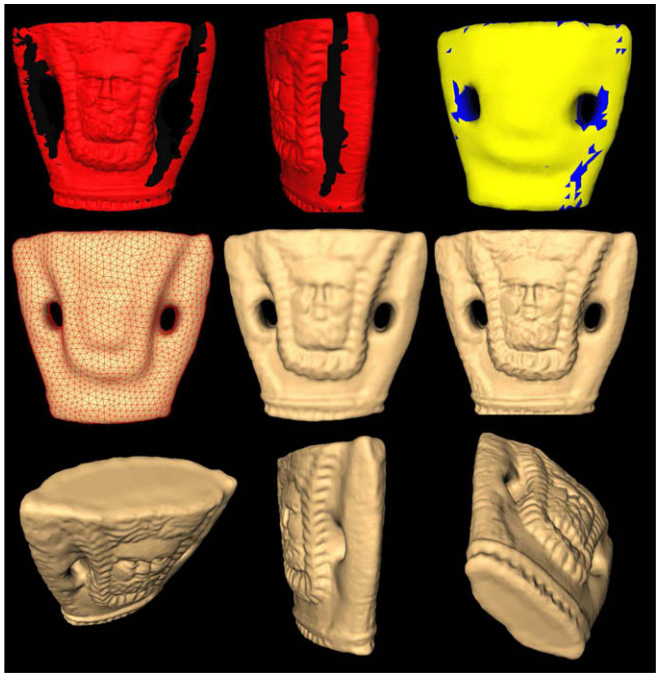

Fig. 13. Cup reconstruction. (First row) Two separate optical triangulation models and the silhouette model associated with range data. (Second row) Fused models at increasing levels of detail. (Third row) Various views from the highest level of detail.

In Fig. 14, for each object, we zoom on both the initial silhouette and the final reconstructed models to better observe the surfaces before and after deformation. This figure also demonstrates how the line-of-sight and silhouette information are integrated in a smooth and robust manner so as to obtain watertight reconstructions. When compared with the optical triangulation models shown in the previous figures, we observe that the holes on the hair and on the nose of the Head object, the ripples on the Cup, the palm of the Hand and some part of the trunk of the Elephant are recovered by the line-of-sight information to the extent that is possible with the available range data. On the other hand, the silhouette information compensates some missing surface information on the thumb and on the palm of the Hand, on the leg and on the trunk of the Elephant, and almost fully recovers the handle of the Cup.

We measure the improvement obtained by the fusion algorithm with respect to the silhouette model, using the average Euclidean distance of the range points to the reconstructed surface, $\epsilon(R,M)$, as defined in Eq. (3), where R denotes the set of range points and M is the reconstructed mesh. Note that ϵ is a directed distance metric that measures the faithfulness of the reconstructed surface to the accurate but incomplete range data. In Table 1 we provide, for each of the four objects, the average distances of the range dataset to the silhouette model and to the models reconstructed by fusion at increasing levels of detail. The distance measure is calculated by assuming that each object is circumscribed by a bounding sphere of radius 100. For example, the Head statue is actually bounded by a sphere of radius approximately 15 cm and hence the average distance of the range data to the fused model at the highest level of detail is about 0.04 mm.

In Table 2, for our method, we provide the number of triangles in the reconstructed model, the number of surface evolution iterations and the execution time needed for convergence at each level of detail. The execution times are measured on a 2.2GHz AMD Athlon 4200+ dual core processor. The given execution times do not

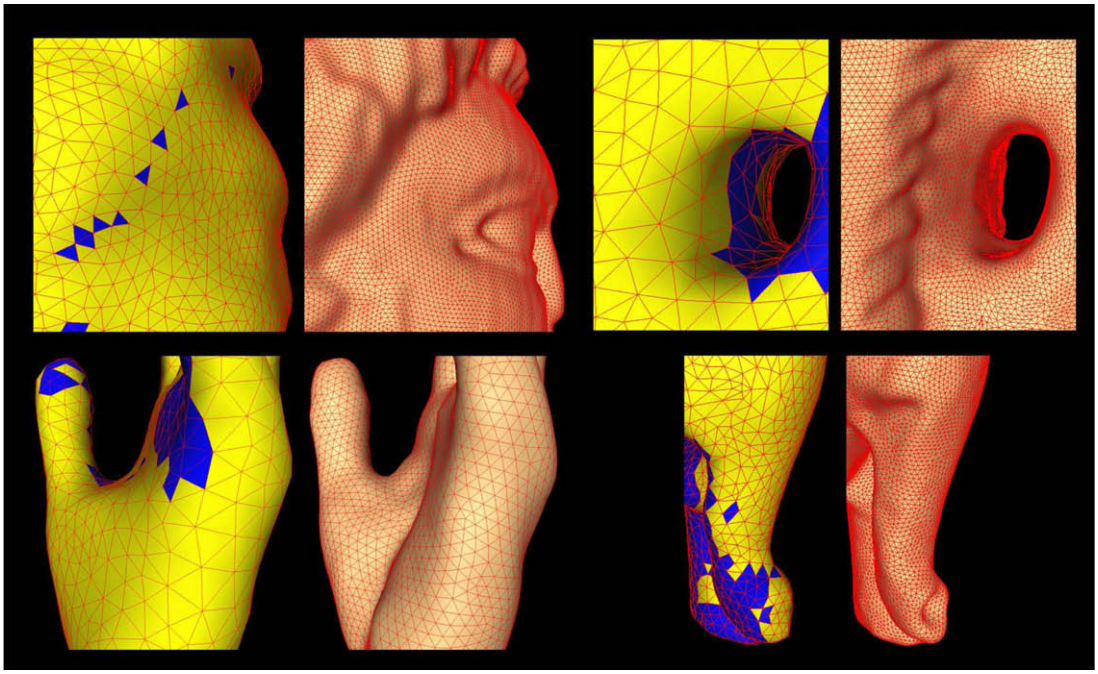

Fig. 14. Zoom on the reconstructed surfaces before and after deformation. For each object, the model on the left is the initial silhouette model and the one on the right is the final reconstruction at the highest resolution (yellow and blue colors on the initial silhouette models indicate triangles with and without carvers, respectively). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Average distances of the range datasets to the models reconstructed by shape from silhouette (SFS) and by fusion at increasing levels of detail.

Model	ϵ – SFS	ϵ – Level 1	ϵ – Level 2	ϵ – Level 3
Head	1.21	0.21	0.08	0.03
Elephant	1.89	0.27	0.15	0.12
Hand	1.03	0.43	0.30	–
Cup	1.72	0.37	0.12	0.08

Table 2
Number of triangles, number of iterations and execution time at each level of detail of the reconstructions.

Model	Triangles (#)			Iterations (#)			Time (s)		
Head	7.8K	30.2K	119.7K	6	4	3	19	54	171
Elephant	13.7K	29.1K	90.4K	9	7	5	60	72	142
Hand	2.9K	11.7K	–	9	3	–	18	18	–
Cup	5.2K	38.8K	152.2K	7	7	3	16	124	229

include the times spent for the initial carver assignment process. Our current implementation requires 17, 19, 33 and 91 s for initial assignment of carvers on the Cup, Hand, Head and Elephant objects, respectively.

We compare our reconstruction technique with the volumetric method proposed in [5]. The method is based on volumetric space carving. An initial volumetric model representing the visual hull is carved voxel by voxel at a fixed resolution through the lines of sight defined by the range data and then triangulated using the iso-surface information so as to obtain a cavity-sensitive watertight surface representation. The method produces accurate surface reconstructions which however suffer from severe topological problems and disturbing visual artifacts, especially at high resolutions. The Head, Cup, Elephant and Hand models, when each reconstructed with this method at the highest possible resolution using the same dataset, result in ϵ values, 0.06, 0.11, 0.09 and 0.27, respectively. For the Elephant and Hand objects, the maximum achievable resolutions match the resolutions of the reconstructions obtained with our technique. Hence the resulting ϵ values are comparable with, even slightly better than the values that we have obtained, though the reconstructions still contain small surface artifacts. For the Head and Cup objects, the volumetric method does not permit high resolution reconstructions with the available data and hence cannot obtain the accuracy that we can achieve with our technique. In Fig. 15, we display three Head models, one reconstructed with the volumetric method, and the other two with ours at two different resolutions using $\varepsilon_{\min} = 0.012$ and $\varepsilon_{\min} = 0.007$. The resolution obtained with $\varepsilon_{\min} = 0.012$ matches the resolution of volumetric method, which is limited with $128 \times 128 \times 128$ partitioning of the bounding box. At higher resolutions of the volumetric reconstruction, topological problems and visually very disturbing intolerable artifacts such as large protrusions begin to appear on the reconstructed surface due to irregular and incomplete range data. Some of these artifacts can even be observed on the current resolution as displayed in Fig. 15, e.g., the protrusion on the hair. Our deformation-based technique, on the other hand, can produce a higher resolution mesh representation of the same object surface (with $\varepsilon_{\min} = 0.007$), which is more accurate, smooth, and free of topological problems and artifacts. At the resolution obtained with $\varepsilon_{\min} = 0.012$, which is comparable to the resolution of the volumetric reconstruction, our method still obtains a visually much more pleasant result with almost the same accuracy ($\epsilon = 0.067$ against $\epsilon = 0.064$). As for the computational complexity at this relatively low resolution, the fusion algorithm takes about 143 s in our case whereas it takes about 95 s for volumetric reconstruction. Note also that the volumetric method requires the triangulated surface information during the fusion process while our method uses only the range points and their surface normals.

Fig. 15. The Head models reconstructed (left) with volumetric carving method, (middle) with our deformation-based technique at low resolution, and (right) with our deformation-based technique at high resolution. Note the surface artifacts on the model reconstructed with the volumetric carving method, such as the protrusion on the hair.

6.2. Parameter setting

There are several parameters used in our algorithm, that need to be set prior to reconstruction. These are the parameters of the smoothing operator, λ and μ, the initial value of the minimum edge length parameter $\varepsilon_{\min}$, and finally the size of the neighborhood window, $(2W_1+1)\times(2W_2+1)$, used for carver update.

Our coarse-to-fine reconstruction algorithm starts with an initial $\varepsilon_{\min}$ which can then gradually be decreased as desired until the final highest resolution is achieved. The lower bound of the initial $\varepsilon_{\min}$ is determined by the resolution and the quality (i.e., the regularity and the sparseness) of the available range data whereas the upper bound depends mainly on the shape complexity. The initial value of $\varepsilon_{\min}$ should be small enough to adequately represent the convex hull of the object, but also sufficiently large for the sake of computational efficiency. To set the initial value of $\varepsilon_{\min}$ for each reconstruction, we have conducted experiments with its varying values. In Figs. 16 and 17, we present the results of these experi-

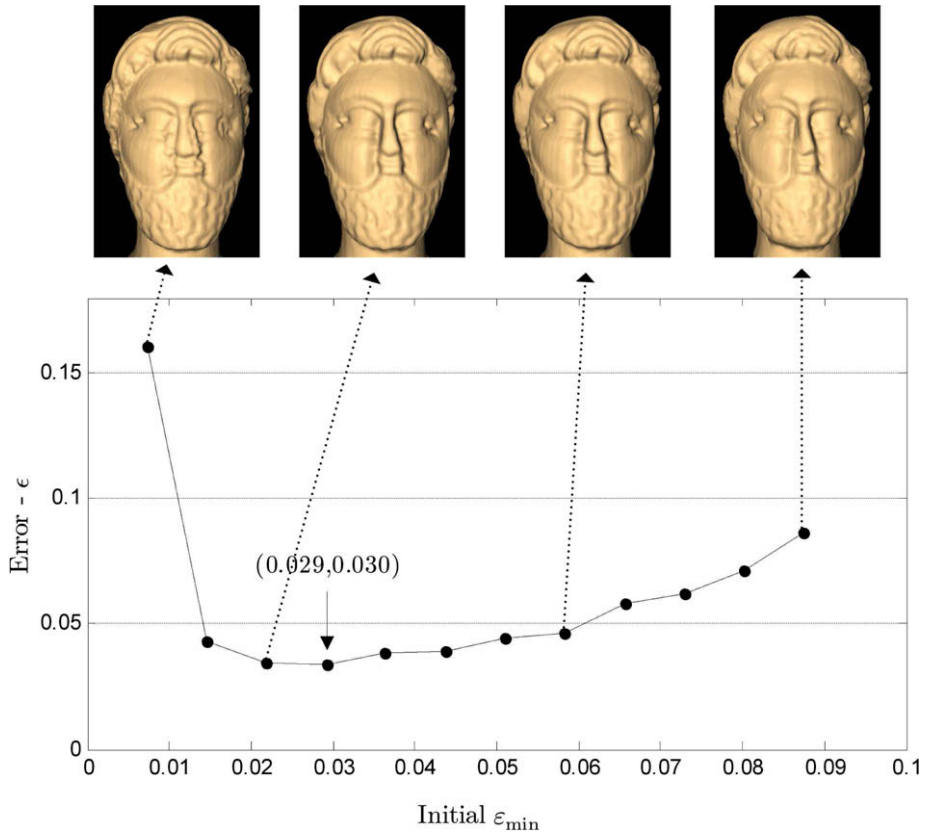

Fig. 16. Reconstruction error ϵ vs. initial $\varepsilon_{\min}$ for the Head object. The images displayed are reconstructions at the highest resolution available, $\varepsilon_{\min} = 0.007$, starting with the specified values of initial $\varepsilon_{\min}$. Our operating point is (0.029, 0.030) as pointed on the plot.

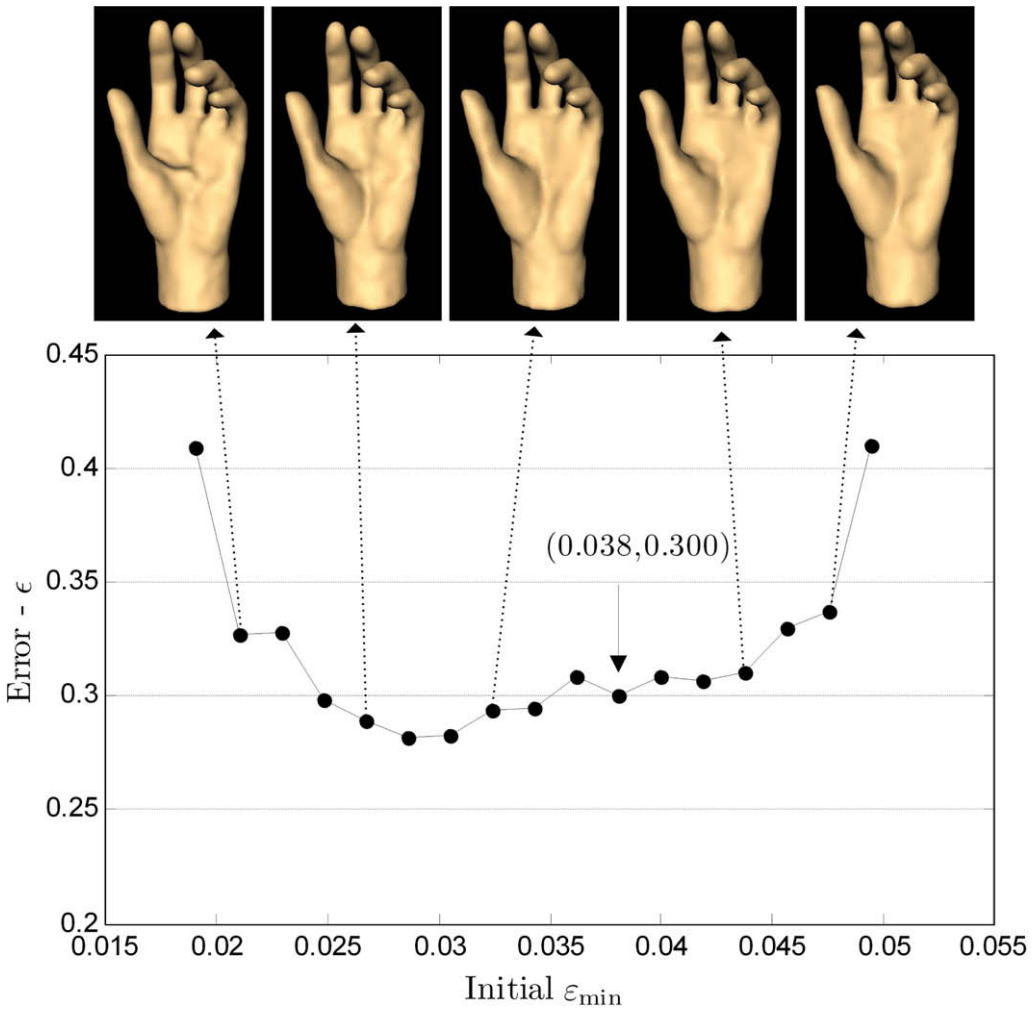

Fig. 17. Reconstruction error ϵ vs. initial $\varepsilon_{\min}$ for the Hand object. The images displayed are reconstructions at the highest resolution available, $\varepsilon_{\min} = 0.019$, starting with the specified values of initial $\varepsilon_{\min}$. Our operating point is (0.038,0.3) as pointed on the plot.

ments for the Head and Hand objects, where we plot the reconstruction error ϵ as a function of the initial $\varepsilon_{\min}$ and display some of the reconstruction results obtained at the highest available resolutions, $\varepsilon_{\min} = 0.007$ and $\varepsilon_{\min} = 0.019$, respectively. We have chosen these two objects for these experiments since they exhibit different properties in terms of the quality of the range dataset and the overall shape complexity. On the provided figures we observe that the initial $\varepsilon_{\min}$ can be chosen safely within a sufficiently large interval, which is (0.022,0.058) and (0.027,0.044) for the Head and Hand objects, respectively. Within these intervals, the reconstructions are all good and differences are hardly noticeable. At the low extreme of these intervals, the protrusions start to appear on the surface, whereas at the high extreme some shape details, e.g., the nose on the Head and the fingers on the Hand, start to get eroded. We also observe that the interval of the initial $\varepsilon_{\min}$ for the Hand object is smaller than the interval for the Head since the range dataset is less dense and more deficient in the former case. Moreover, the fingers of the Hand pose difficulties for reconstruction at low resolutions. Note also that, in both cases, our choices of the initial $\varepsilon_{\min}$ adequately address the trade-off between computational complexity and reconstruction accuracy.

We set the window size parameters, W_1 and W_2, of the carver update process, based on the chosen value of the initial $\varepsilon_{\min}$ and the resolution of the range data. For each reconstruction, during the initial carver assignment procedure, we find, for each triangle of the deformable mesh, the scan-order neighborhood of the lines of sight intersecting the triangle. The average size, $N_1 \times N_2$, of this neighborhood, which is computed over all the triangles, can then be used to set the parameters W_1 and W_2. We set the parameters as $W_1 = 2N_1$ and $W_2 = 2N_2$ so that the window approximately covers all the carver lines within the neighborhood of the triangle to be updated. This setting is used in all our reconstruction experiments and yields always good results since the deformable model proceeds slowly (its speed bounded by $\varepsilon_{\min}/2$) and smoothly. Note also that when the resolution is increased by some factor during coarse-to-fine deformation, the parameters W_1 and W_2 are decreased in the same proportion for computational efficiency.

For the smoothing parameters λ and μ, we have used the setting $\lambda = 0.6307$ and $\mu = -0.6732$, as suggested in [34]. Recall from Section 4.2 that this setting results in a pass-band frequency $k_{PB} \approx 0.1$ for the transfer function, $f(k) = (1 - \lambda k)(1 - \mu k)$, of the smoothing filter. To show the optimality of this selection in our case, in Fig. 18, we plot the reconstruction error ϵ as a function of the pass-band frequency $k_{PB}, 0 < k_{PB} < 1$, for the Head and Hand objects. For each k_{PB} value, we compute the corresponding values of λ and μ by using $k_{PB} = \frac{1}{\lambda} + \frac{1}{\mu}$ and $f(1) = -f(2)$ so as to have a fast and stable filter (see Section 4.2 and [34] for details). We observe that the optimal setting in both cases is around $k_{PB} = 0.1$ and the reconstruction performance is not very sensitive to the choice of k_{PB} around this operating point. But as the value of k_{PB} is increased

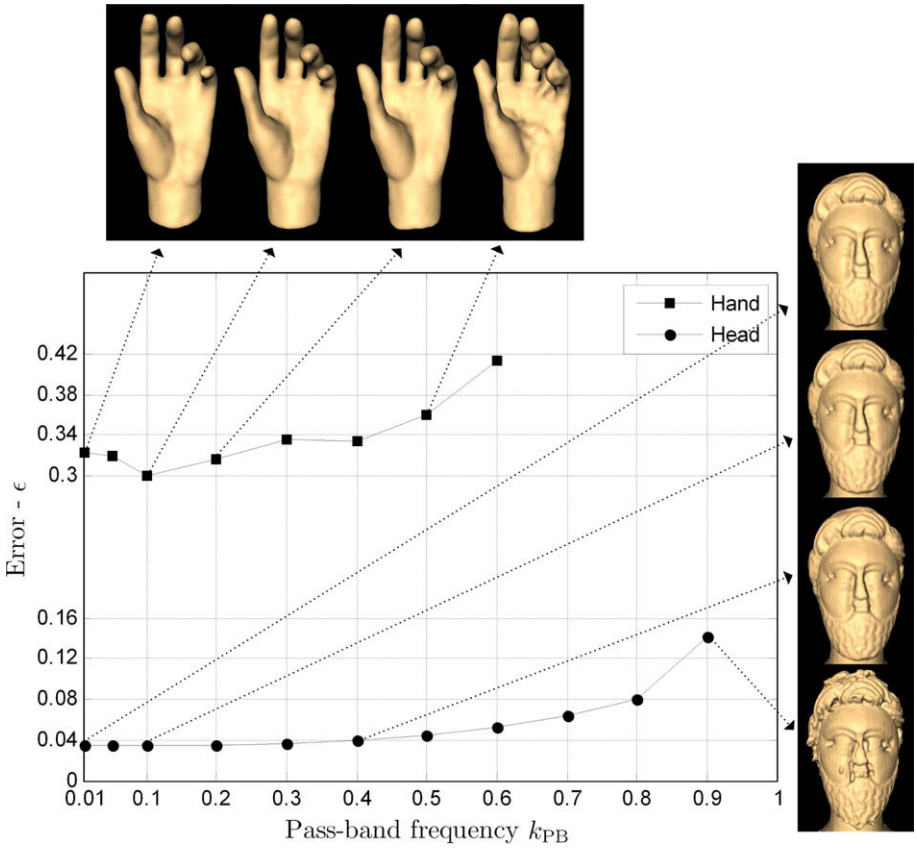

Fig. 18. Reconstruction error ϵ vs. smoothing parameter k_{PB} for the Head and Hand objects. The images displayed are reconstructions at the highest resolution available, using the specified values of k_{PB}. Our operating point is at $k_{PB} \approx 0.1$, which corresponds to $\lambda = 0.6307$ and $\mu = -0.6732$.

too much, protrusions start to appear on the reconstructions. For the Hand object, the algorithm does not even converge for values $k_{PB} > 0.6$. In [34], it is conjectured that the values of k_{PB} from 0.01 to 0.1 produce good fairing results and our findings are in accordance with this observation.

7. Conclusion

We have presented a surface reconstruction method based on mesh deformation that fuses silhouette geometry and range data to construct cavity-sensitive and hole-free object models. The algorithm starts with an initial silhouette model, that describes the object visual hull, and then deforms it to amend missing cavities using the triangulated range data. As a by-product, the fusion technique produces reconstructions of the object surface at increasing levels of detail.

The proposed technique results in aesthetically pleasant and topologically correct mesh models which are always more accurate as compared to silhouette-only and range-data-only reconstructions. The experiments show that it is possible to produce robust and reliable high resolution reconstructions even in the presence of severe occlusions and deficient range data, using a simple low-cost single-stripe rotational scanner. This is thanks to the employed coarse-to-fine strategy, which not only gives robustness but also significantly improves the computational efficiency. By using a slightly more intricate scanner than the one we employed for optical triangulation in our experiments (simply incorporating some translational movement of the laser projector for example), it would be possible to obtain watertight models even with better accuracy.

Acknowledgement

This work has been supported by the European Network of Excellence 3DTV (http://www.3dtv-research.org).

References

[1] B. Curless, M. Levoy, A volumetric method for building complex models from range images, in: Proceedings of SIGGRAPH'96, 1996, pp. 303–312.
[2] R. Whitaker, A level-set approach to 3D reconstruction from range data, International Journal of Computer Vision 29 (3) (1998) 203–231.
[3] S. Tosovic, R. Sablatnig, M. Kampel, On combining shape from silhouette and shape from structured light, in: Proceedings of 7th Computer Vision Winter Workshop, 2002, pp. 108–118.
[4] T. Terauchi, Y. Oue, K. Fujimura, A flexible 3D modeling system based on combining shape-from-silhouette with light-sectioning algorithm, in: International Conference on 3D Digital Imaging and Modeling (3DIM'05), 2005, pp. 196–203.
[5] Y. Yemez, C.J. Wetherilt, A volumetric fusion technique for surface reconstruction from silhouettes and range data, Computer Vision and Image Understanding 105 (1) (2007) 30–41.
[6] L.P. Kobbelt, T. Bareuther, H. Seidel, Multiresolution shape deformations for meshes with dynamic vertex connectivity, Computer Graphics Forum (Eurographics'00) 19 (2000) C249–C260.
[7] J. Park, G.N. DeSouza, A.C. Kak, Dual-beam structured-light scanning for 3D object modeling, in: International Conference on 3D Digital Imaging and Modeling (3DIM'01), 2001, pp. 65–72.
[8] C. Liska, R. Sablatnig, Adaptive 3D acquisition using laser light, in: Proceedings of Czech Pattern Recognition Workshop, 2000, pp. 111–116.
[9] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, D. Fulk, The Digital Michelangelo Project: 3D scanning of large statues, in: Proceedings of SIGGRAPH'00, 2000, pp. 131–144.
[10] S. Rusinkiewicz, O. Hall-Holt, M. Levoy, Real-time 3D model acquisition, in: Proceedings of SIGGRAPH'02, 2002, pp. 438–446.
[11] A. Fasano, M. Callieri, P. Cignoni, R. Scopigno, Exploiting mirrors for laser stripe 3D scanning, in: International Conference on 3D Digital Imaging and Modeling (3DIM'03), 2003, pp. 243–250.

[12] J. Davis, S.R. Marschner, M. Garr, M. Levoy, Filling holes in complex surfaces using volumetric diffusion, in: International Symposium on 3D Data Processing, Visualization, and Transmission, 2002, pp. 428–438.
[13] Y. Chen, G. Medioni, Description of complex objects from multiple range images using an inflating balloon model, Computer Vision and Image Understanding 61 (3) (1995) 325–334.
[14] H. Zhao, S. Osher, R. Fedkiw, Fast surface reconstruction using the level set method, in: IEEE Workshop on Variational and Level Set Methods in Computer Vision (VLSM'01), 2001, pp. 194–201.
[15] J. Carr, R.B. ad J.B. Cherrie, T. Mitchell, W. Fright, B. McCallum, T. Evans, Reconstruction and representation of 3D objects with radial basis functions, in: Proceedings of SIGGRAPH'01, 2001, pp. 67–76.
[16] T. Masuda, Filling the signed distance field by fitting local quadrics, in: Proceedings of 3D Data Processing, Visualization and Transmission, 2004, pp. 1003–1010.
[17] R. Sagawa, K. Nishino, K. Ikeuchi, Adaptively merging large-scale range data with reflectance properties, IEEE Transactions on PAMI 27 (3) (2005) 392–405.
[18] H. Qin, C. Mandal, B.C. Vemuri, Dynamic catmull-clark subdivision surfaces, IEEE Transactions on Visualization and Computer Graphics 4 (3) (1998) 215–229.
[19] C.H. Chien, J.K. Aggarwal, Identification of 3D objects from multiple silhouettes using quadtrees/octrees, Computer Vision, Graphics and Image Processing 36 (2–3) (1986) 256–273.
[20] R. Szeliski, Rapid octree construction from range sequences, Computer Vision, Graphics and Image Processing 58 (1) (1993) 23–32.
[21] W. Niem, J. Wingbermuhle, Automatic reconstruction of 3D objects using a mobile camera, Image and Vision Computing 17 (2) (1999) 125–134.
[22] Y. Yemez, F. Schmitt, 3D reconstruction of real objects with high resolution shape and texture, Image and Vision Computing 22 (2004) 1137–1153.
[23] W. Wojciech, C. Buehler, S.G.R. Raskar, L. McMillan, Image-based visual hulls, in: Proceedings of SIGGRAPH'00, 2000, pp. 369–374.
[24] Y. Matsumoto, K. Fujimura, T. Kitamura, Shape-from-silhouette/stereo and its application to 3D digitizer, in: International Conference on Discrete Geometry for Computing Imagery, 1999, pp. 177–190.
[25] K.N. Kutulakos, S.M. Seitz, A theory of shape by space carving, International Journal of Computer Vision 38 (3) (2000) 199–218.
[26] G. Vogiatzis, P.H.S. Torr, R. Cipolla, Multi-view stereo via volumetric graph-cuts, in: IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 391–398.
[27] A. Hornung, L. Kobbelt, Robust and efficient photo-consistency estimation for volumetric 3D reconstruction, in: European Conference on Computer Vision (ECCV'06), 2006, pp. 179–190.
[28] C.H. Esteban, F. Schmitt, Silhouette and stereo fusion for 3D object modeling, Computer Vision and Image Understanding 96 (3) (2004) 367–392.
[29] J. Isidoro, S. Sclaroff, Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints, in: IEEE International Conference on Computer Vision, 2003, pp. 1335–1342.
[30] S.N. Sinha, M. Pollefeys, Multi-view reconstruction using photo-consistency and exact silhouette constraints: a maximum-flow formulation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 349–356.
[31] Y. Furukawa, J. Ponce, Carved visual hulls for image-based modeling, in: European Conference on Computer Vision (ECCV'06), 2006, pp. 564–577.
[32] Y. Sahillioğlu, Y. Yemez, A surface deformation framework for 3D shape recovery, in: Workshop on Multimedia Content Representation, Classification and Security, 2006, pp. 570–577.
[33] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, International Journal of Computer Vision 1 (4) (1988) 321–332.
[34] G. Taubin, A signal processing approach to fair surface design, in: Proceedings of SIGGRAPH'95, 1995, pp. 315–358.
[35] Z.J. Wood, P. Schröder, D. Breen, M. Desbrun, Semi-regular mesh extraction from volumes, in: Proceedings of Visualization'00, 2000, pp. 275–282.
[36] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle, Mesh optimization, in: Proceedings of SIGGRAPH'93, 1993, pp. 19–26.
[37] Y. Duan, J. Hua, H. Qin, Interactive shape modeling using Lagrangian surface flow, Visual Computer 21 (2005) 279–288.
[38] J. Foley, A.V. Dam, Fundamentals of Interactive Computer Graphics, Addison-Wesley, 1984.