Multiresolution Implicit Object Modeling

Augusto Sarti, Stefano Tubaro

Dipartimento di Elettronica e Informazione – Politecnico di Milano
Piazza L. Da Vinci 32, 20133 Milano, Italy
Email: Augusto.Sarti@polimi.it, Stefano.Tubaro@polimi.it

Abstract

In this paper we discuss two image-based 3D modeling methods based on a multi-resolution evolution of a volumetric function's levelset. In the former the role of the levelset implosion is to fuse ("sew" and "stitch") together several partial reconstructions (depth maps) into a closed model. In the latter the levelset's implosion is steered directly by the texture mismatch between views. Both solutions share the characteristic of operating in an adaptive multiresolution fashion, in order to boost up computational efficiency and robustness.

1 Introduction

A 3D manifold can be generally defined and represented either *explicitly* as an atlas (juxtaposition of partially overlapping local charts), or *implicitly* as the set of points that satisfy a nonlinear constraint in the 3D space (level set of a volumetric function). Similarly, image-based modeling of 3D objects can be envisioned as based on either one of the above two representations. In the former case, a global object model is obtained as a complex "patchworking" of simple local reconstructions (typically depth maps), while in the latter the object surface is described as a level set of an appropriate volumetric function.

As we may expect, an atlas-based 3D modeling method deals with topological complexity with a "divide-and-conquer" strategy, which simplifies the local shape estimation process. The price to pay for this simplification, however, is in the complexity of the steps that are necessary to fuse the local reconstruction into a global closed one (registration, fusion and hole-mending). An implicit surface representation, on the other hand, tends to be quite insensitive to topological complexity, as it may accommodate self-occluding surfaces, concavities, surfaces of volumes with holes (e.g. doughnuts, objects with handles, etc.), or even multiple objects. However, their volumetric nature requires a more redundant data structure.

In this paper we discuss two image-based modeling methods that exploit the key features of a levelset-based approach to deal with complex topological structures. The former ("indirect modeling") tries to overcome intrinsic topological difficulties related to an "atlas-based" approach using a volumetric approach to surface fusion. The latter ("direct method") skips the partial modeling step and uses the images to steer the implosion of the levelset in such a way to obtain the object model in a robust and fast way.

2 Implicit Surface Modeling

A closed surface γ can be expressed in implicit form as

$$\gamma = \{\mathbf{x} | \, \psi(\mathbf{x}) = 0\}$$

where $\psi(\mathbf{x})$ is a volumetric function whose absolute value in $\mathbf{x}$ is given the distance d between $\mathbf{x}$ and the surface, and its sign depends on whether the point $\mathbf{x}$ is inside or outside the surface. Adopting the signed distance as a volumetric function is known to simplify the computation of the surface's differential properties of orders 1 and 2:

- the surface normal can be computed as the gradient $\nabla\psi$ and is a unit vector;
- the surface curvature can be computed as a divergence of the form $\nabla \cdot \nabla\psi$.

In order to model a surface in implicit form, we can proceed [4] by defining a temporally evolving volumetric function whose levelset zero "sweeps" the whole volume of interest until it takes on the desired shape under the influence of some properly defined "external action". The levelset evolution is defined by the Hamilton-Jacobi PDE, which can be

discretized into the update equation

$$\psi(\mathbf{x}, t + \Delta t) = \psi(\mathbf{x}, t) - |\nabla \psi(\mathbf{x}, t)|\, F(\mathbf{x}) \Delta t \quad ,$$

where the velocity function $F(\mathbf{x})$ is bound to be orthogonal to the levelset zero and can be quite arbitrarily defined in order to steer the front propagation toward a desired shape. Terms that may appear into its expression are:

local curvature – which promotes a maximally smooth implosion of the surface

distance from 3D data – which promotes data fitting

inertia – which promotes topological changes (object splitting or generation of holes)

texture agreement – which maximizes the similarity between the appearence of the modeled surface and its available views

Besides such terms, we are free to define new velocity terms that attribute the surface evolution some desired behavior.

3 Indirect Modeling

A common way to build a complete 3D object model consists of combining several simpler surface patches through a 3D "patchworking" process. In order to do so, we need a preliminary registration phase, in which all the available surface patches are correctly positioned and oriented with respect to a common reference frame; and a fusion process, which consists of merging all surface patches together into one or more closed surfaces. One rather standard registration strategy is the Iterative Closest Point [2] algorithm, which consists of minimizing the mean square distance between overlapping portions of the surface, using an iterative procedure. As for surface fusion, in this Section we propose and test an approach that is able to seamlessly "sew" the surface overlaps together, and reasonably "mend" all the holes that remain after surface assembly (usually corresponding to non-visible surface portions).

This "atlas" approach to 3D modeling is suitable for $2D\frac{1}{2}$ modeling solutions such as image-based depth estimation techniques, range cameras, and laser-scanners. The depth maps produced by such devices could be made of a several non-connected surface patches, as occlusions and self-occlusions tend to generate depth discontinuities [1]. Such surfaces usually need a lengthy assembly process in order to become a complete and closed surface.

3.1 Surface Fusion

As anticipated in Section 2, our fusion process is based on the temporal evolution of the zero level set a volumetric function [3, 4]. The velocity function that steer the front evolution accounts for two contrasting needs: that of following the motion by curvature and that of honoring the data (registered surface patches).

A surface is said to follow the motion by curvature when the velocity field that describes the surface deformation is normal to the surface itself and its magnitude is proportional to the local curvature (with sign). Indeed, if the motion were purely by curvature, a surface would tend to deflate completely and disappear, while becoming progressively smoother and smoother (Fig. 1 above). The need to honor the available range data prevents this complete implosion from taking place (Fig. 1 below).

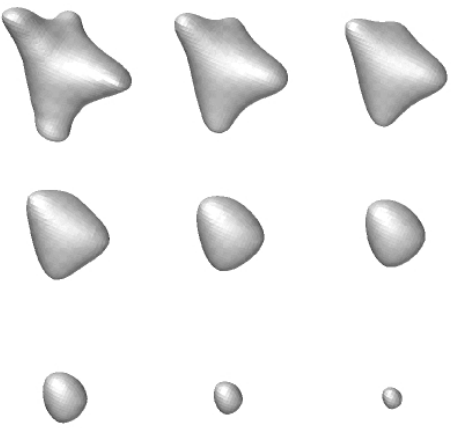

Figure 1: Motion by curvature: the surface deflates in a maximally smooth fashion until it disappears.

In order to implement this implosion-inhibition mechanism, we need to redefine the velocity field associated to the update equation that describes the zero level-set propagation. This velocity is bound to be orthogonal to the propagating front, and its

amplitude is set to

$$F(\mathbf{x}) = F_1(K(\mathbf{x})) + \alpha \frac{K_M(\mathbf{x})}{K(\mathbf{x})} F_2(d(\mathbf{x}))$$

where K is the levelset's local curvature, K_M is the local curvature of the facing surface; d is the distance (with sign) between the propagating front and the surface patch; and α is a parameter that balances local smoothness and proximity to the data. Indeed, this formulation assumes that only one surface patch is facing the propagating front.

Notice that the above definition of velocity holds valid only for the points that lie on the propagation front, therefore we need to extend its validity in the whole volume (or at least in the sorrounding points of the surface). The extension of this function needs to be done consistently with the front propatation, meaning that the levelset should evolve with no self-collisions. This can be done quite easily [3] as follows: given a generic point $\mathbf{x}$ not lying on the surface γ, we can search for the point $\mathbf{y}$ on γ that lies the closest to $\mathbf{x}$ and let $F(\mathbf{x}) = F(\mathbf{y})$.

Given a point on the propagating front, the distance from a surface patch is computed from the orthogonal projection of that point onto the surface patch itself (Fig. 2). If no point on the surface patch faces the point on the level-set orthogonally, then the distance function d is computed from the closest point on the border of the patch (within a preassigned range).

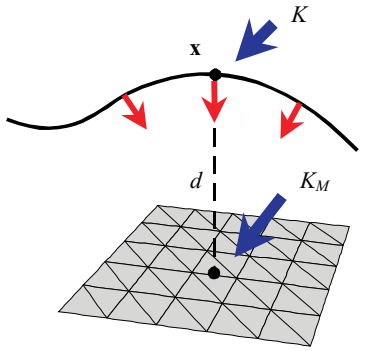

Figure 2: Given a point on the propagating front, its distance from a surface patch is computed from the orthogonal projection onto the target triangle mesh.

When more than one surface patches are facing the propagation front, then the distance function d is computed using the distances from the point on the propagating front and all the orthogonal projections onto the surface patches that face it; the surface orientation; the closeness to the border of the patch (the reliability tends to decrease in the proximity of extremal boundaries); and the mutual occlusion between surface patches.

3.2 Key Features

This approach to surface fusion exhibits a number of desirable characteristics. One of its most appealing features is the fact that it is very robust against topological complexity. In fact, a level-set of a volumetric function is adequate for describing multiple objects of rather arbitrary topology. In addition, it involves a fairly modest computational cost. In fact, in order to obtain high performance at low computational cost, besides updating the volumetric function just in a narrow region around the zero level-set (narrow-band implementation), we operate in a multiresolution fashion (see Fig. 3). This can be achieved by starting with a low-resolution voxset (e.g. a voxset with 10 voxels per side) and letting the front settle down. Then we break down the voxels around the propagating front and resume the front propagation. The operation continues until the final resolution is reached. A key aspect of this process is in the fact that the velocity field that drives the implosion of the level set can be pre-computed on the octree data structure that best fits the available range data.

The resulting model is bound to be a set of closed surfaces, therefore all the modeling "holes" left after mosaicing the partial reconstructions are closed in a topologically sound fashion. In fact, those surface portions that cannot be reconstructed because they are not visible, can sometimes be "patched up" by the fusion process. This ability to "mend" the holes can also be exploited in order to simplify the 3D acquisition session, as it allows us to skip the retrieval of some depth maps.

An interesting aspect of our fusion method is in the possibility to modify the surface characteristics through a processing of the volumetric function. For example, filtering the volumetric function results in a smoother surface model. Finally, the method exhibits a certain robustness against orientation errors, as the non perfect matching of surface borders can be taken care of by the fusion process through a careful definition of the distance function

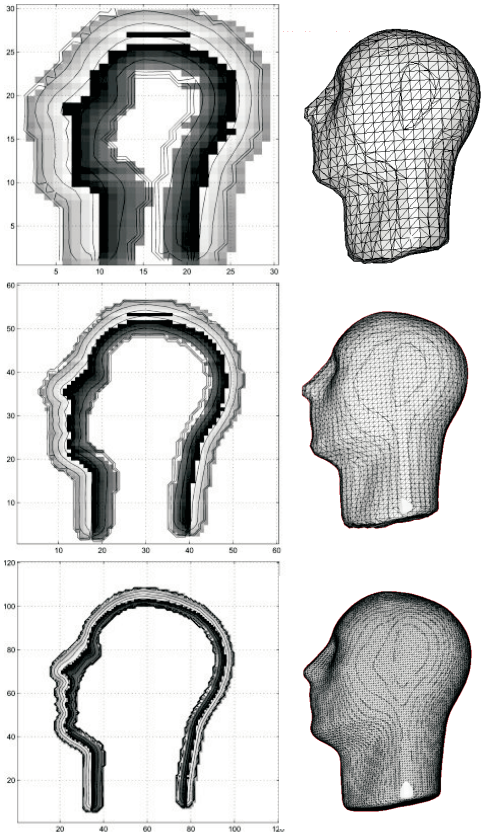

Figure 3: Multiresolution progression of the voxset where the volumetric function is defined.

used in the specification of the volumetric motion field.

3.3 An Example of Application

In order to test the effectiveness of the proposed technique, we applied it to a variety of study cases. A particularly interesting experiment was conducted on an object with a particular topology (a bottle with a handle) that could easily create problems of ambiguities. Any traditional surface fusion approach would, in fact, encounter difficulties in deciding how to complete the surface in the missing regions. Furthermore, besides exhibiting self-occlusion problems, this object puts the multiresolution approach under a severe test. We acquired six depth maps and assembled them together using an ICP algorithm (see Fig. 4). The result was an incomplete model with some accuracy problems in the overlapping regions (at the boundaries of the depth maps). The front evolution is shown in Fig. 5, which results in the (topologically correct) final model of Fig. 6.

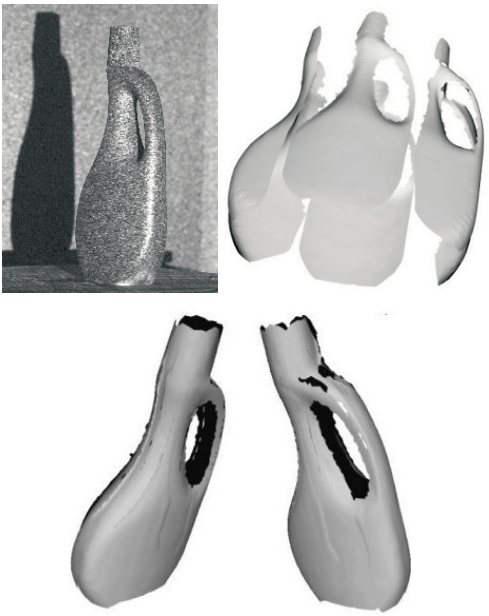

Figure 4: One of the original views (top left). Six unregistered surface patches obtained with stereometric techniques (top right). Two views of the assembled surface patches after registration (bottom): notice the creases due to a non-perfect model overlapping, and the presence of holes in the global model.

4 Direct Modeling

In alternative to using 3D data for the generation of the complete closed object surface, we can use directly the available images. An image-based 3D modeling method that uses an implicit representation of surfaces was recently proposed by Faugeras and Keriven [5]. This modeling approach is based on the temporal evolution of a volumetric function whose zero level-set is a closed surface that represents the surface model as it tends to approximate

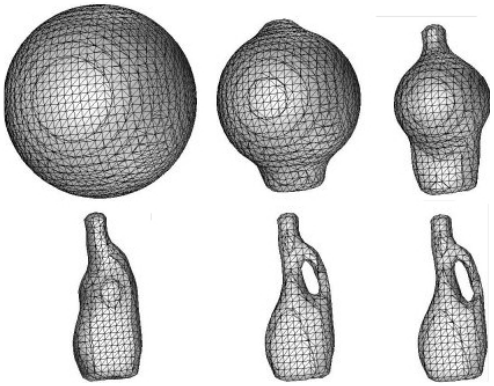

Figure 5: Level set implosion.

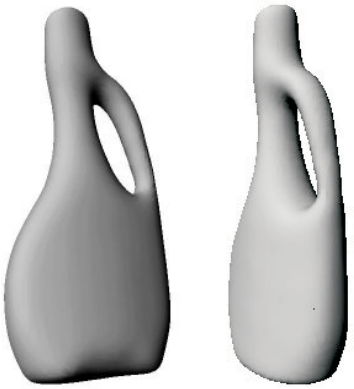

Figure 6: Final 3D model.

the imaged object. This surface, which initially contains the imaged object, evolves by following a motion that is always locally normal to the surface, with a speed that depends on the local surface curvature and to a measurement of the local "texture mismatch" between imaged and "transferred" textures. Transferring an imaged texture onto another view means back-projecting it onto the model and re-projecting it onto the other view. In order to keep the computational complexity at a manageable level, the updating of the volumetric function is only performed within a "narrow band" [3] around the current surface. Our solution, however, significantly generalizes this approach, as it operates in an adaptive multi-resolution fashion, which boosts up the computational efficiency. Multi-resolution, in fact, enables us to quickly obtain a rough approximation of the objects in the scene at the lowest possible voxset resolution. Successive resolution increments allow us to progressively refine the model and add details. In order to do so, we introduce "inertia" in the level-set evolution, which tends to favor topological changes (e.g. the creation of doughnut-like holes in the structure). Finally, through a careful control of the components that steer the level-set evolution (hysteresis, biased quantization, etc.), we are able to recuperate details that were lost at lower resolution levels (surface creases, ridges, etc.).

4.1 Definition of the Velocity Function

One of the terms that contribute to steering the level-set evolution is the "texture mismatch" between imaged and "transferred" textures [6], which is a function of the correlation between homologous luminance profiles [5]. The texture mismatch is

$$\begin{aligned} C(S,\mathbf{n}) &= \int_S \Phi(S,\mathbf{n},v,w)|\mathbf{s}_v \times \mathbf{s}_w|dvdw \\ \Phi &= 1 - \sum_{i,j=1;i\neq j}^{n} \frac{1}{|I_i||I_j|}\langle I_i, I_j\rangle \,. \end{aligned} \quad (1)$$

where $\mathbf{d}\sigma = |\mathbf{s}_v \times \mathbf{s}_w|dvdw$ is the infinitesimal area element of the surface S, associated to the local surface parametrization (v,w) induced by the image coordinate chart; $\mathbf{n}$ is the surface normal; and $I_i(\mathbf{m}_i)$ is the luminance of pixel $\mathbf{m}_i$ in the *i*-th image. This definition of $\mathbf{d}\boldsymbol{\sigma}$ guarantees that the surface representation will be independent on the variables (u,w). The surface patch S through which the luminance transfer occurs is assumed to be a locally planar approximation of the propagating front. Indeed, in order to guarantee that this approximation will mantain a constant quality, the size of this planar patch will change according to the local curvature of the levelset.

The inner product (correlation) between the pair of subimages I_i and I_j is defined as follows:

$$\begin{aligned} \langle I_i, I_j\rangle &= \frac{1}{4pq}\int_{-p}^{p}\int_{-q}^{q}\left(I_i(\mathbf{m}_1+\mathbf{m}) - \overline{I_i}(\mathbf{m}_1)\right)\cdot \\ &\quad \cdot\left(I_j(\mathbf{m}_2+\mathbf{m}) - \overline{I_j}(\mathbf{m}_2)\right)\mathbf{dm} \quad , \end{aligned}$$

where $\mathbf{m}_1$ e $\mathbf{m}_2$ are homologous image points (i.e. image points that correspond to the same point of

the surface model), and

$$\overline{I_k}(\mathbf{m}_k) = \frac{1}{4pq} \int_{-p}^{p} \int_{-q}^{q} I_k(\mathbf{m}_k+\mathbf{m})\mathbf{dm} \ , \ k = 1, 2$$

Although the correlation could be computed between all the viewpoints where there is visibility, only the pair of views with the best visibility is considered. Visibility can be easily checked through a ray-tracing algorithm and measured as a function of the angle between visual ray and surface normal. Notice that normalizing the correlation has a twofold purpose: to limit its range between 0 and 2; and to guarantee that low-energy areas (smooth texture) will have the same range of high-energy (rough texture) areas. Finally, subtracting the average from a luminance profile tends to compensate a non-lambertian behavior of the imaged surfaces.

The velocity function associated to the front propagation is here defined with the twofold need of guaranteeing surface smoothess and consistency between images and final model

$$F(\mathbf{x}) = C(\mathbf{x})\nabla \cdot \nabla\psi + \nabla\Phi \cdot \nabla\psi + k\Phi \ . \quad (2)$$

The first term $C(\mathbf{x})\mathrm{div}\psi$ represents the texture-curvature action and favors a maximally smooth implosion toward a shape that agrees with the available textures. The presence of the texture mismatch cost C, in fact, tends to slow down areas with modest cost, and speed up areas of high cost. Ideally, one would be lead to think that the first term is sufficient to correctly steer the model's evolution, as correct surface regions should have a zero cost, while other reagions are left free to evolve. This, however, is not really true as the cost is rarely equal to zero due to a non-perfect luminance transferral and a non-lambertian radiometric behavior. This causes the front propagation to "trespass" the correct surface. The second term of eq. (2) will tend to contrast this behavior. In fact, in the proximity of the actual surface, the local cost gradient $\nabla\Phi$ is almost parallel (although oppositely oriented) to the propagation front's normal $\mathbf{n} = \nabla\psi$. As a consequence, $\nabla\Phi \cdot \nabla\psi < 0$ tends to discourage the front from propagating beyond the actual surface. Finally, the third element of eq. (2) acts an "inertial" term in order to favor concavities in the final model, provided that a proper dynamic adaptation of k is performed.

4.2 Multiresolution

If the volumetric function that characterizes the level-set is defined on a static voxset of $N \times N \times N$ voxels, the computational complexity of each front propagation step is proportional to N^2, as it is proportional to the surface of the level-set (narrow-band computation). Furthermore, since the velocity F is multiplied by $|\nabla\psi|$ (which is equal to the sampling step), the number of iterations turns out to be proportional to N, with a resulting algorithimc complexity that is proportional to N^3. In order to dramatically reduce this complexity, we developed a multi-resolution approach to level-set evolution. The algorithm starts with a very low resolution level (a voxset of 10–15 voxel per side). When the propagation front converges, the resolution increases and the front resumes its propagation. The process is iterated until we reach the desired resolution. A progressive resolution increment has the desired result of minimizing the amount of changes that each propagation step will introduce in the model, with the result of achieving a better global minimum of the cost function. Furthermore, the number of iterations will be dramatically reduced (from N to $\log N$) with respect to a fixed-resolution approach, with an algorithmic complexity that turns out to be proportional to $N^2 \log N$.

Indeed, starting from a low-resolution voxset, we need to prevent the algorithm from losing details at that resolution or to make sure that the algorithm will be able to recover the lost details. In fact, one has to keep in mind that the motion by curvature tends to dominate over the other terms, therefore some of the details of the object may totally disappear. In order to prevent this from happening, we use a method based on thresholding the local curvature with a hyperbolic tangent function. This guarantees a smoother behavior than a simpler clipping function.

In spite of this smooth thresholding mechanism, in some cases it is not possible to prevent some of the smaller details from disappearing. For this reason, we developed a technique that enables the recovery of lost details before the resolution is increased, which is based on a mechanism of hysteresis in the surface implosion. The idea is to keep track of all the voxels on the zero level-set whose cost Φ is below a certain threshold. After the "implosion" of the level-set, we let the propagation front evolve while driven by a different cost

function that depends on the distance between the surface and such points. This operation makes the surface litterally "climb up" the lost details. As an example of applications, see Fig. 7.

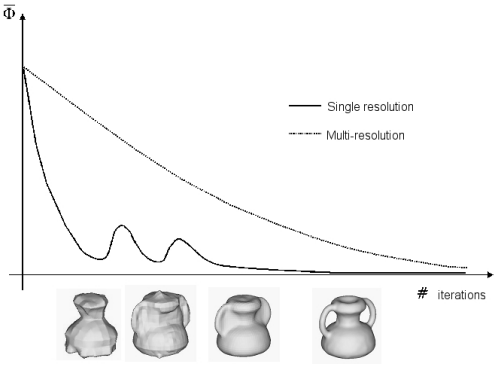

Figure 7: Illustration of the temporal evolution of the cost function (texture mismatch) and of the model. Notice that the cost value suddenly increases at every resolution change, due to the mechanism of recovery of details.

4.3 Examples of application

We tested our approach on several subjects acquired with a camera moving around them. The method proved to be remarkably robust against topological complexity and lack of segmentation (see Figs. 8 to 14). Notice that the meshes of the final models have been built from volumetric data using a customisation of a classical marching cubes algorithm.

Figure 8: Sequence of original views.

5 Conclusions

In this paper we discuss two image-based 3D modeling methods based on a multi-resolution evolution of a volumetric function's levelset. The former consists of fusing ("sewing" and "stitching") numerous partial reconstructions (depth maps) into a closed model, while the latter consists of steering the levelset's implosion with texture mismatch between views. Both solutions share the characteristic of operating in an adaptive multi-resolution fashion, which boosts up computational efficiency and robustness. Both modeling applications have been written in C++, and run on several SW platforms. Computational times depend on the final resolution and on the topological complexity of the imaged object. Using a PC equipped with a Pentium III 800 MHz with a 256MB RAM, the fusion algorithm always completes its task in just a few minutes on voxsets of $128 \times 128 \times 128$ voxels. A bit more intensive is the direct approach, mostly because it does not allow pre-computation of the velocity field. In this case we need a few minutes per resolution level.

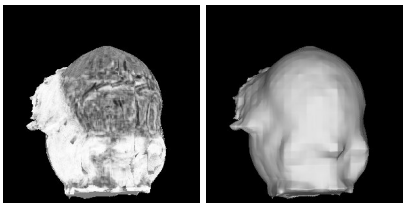

Figure 9: A view of the cost function mapped onto the propagation front. The darker the texture, the heavier the mismatch.

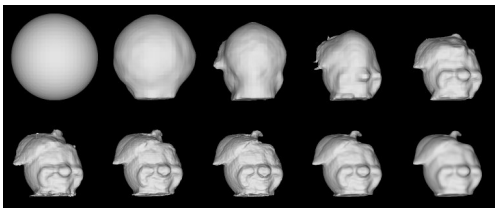

Figure 10: Temporal evolution of the propagation front. The initial volumetric resolution is very modest (in this case the voxset size is 20x20x20), and is not able to account for some topologically complex details of the surface (the fifth frame in lexicographic order is the best one can do at this resolution). As the resolution increases, more details begin to appear, such as the stem of the apple.

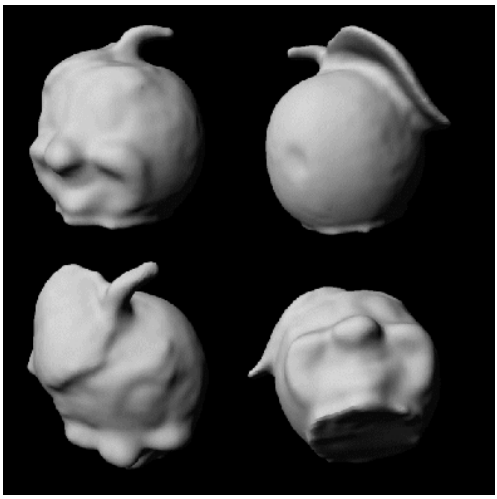

Figure 11: Four views of the final 3D model.

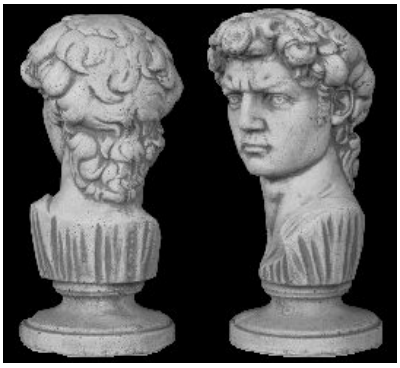

Figure 12: Two of the original views of the subject.

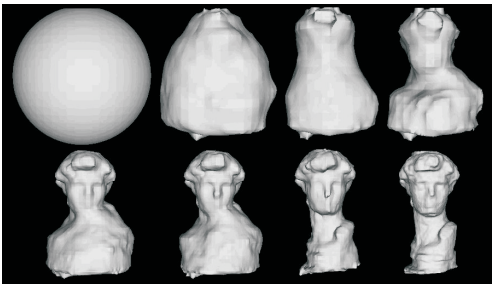

Figure 13: Temporal evolution of the propagation front.

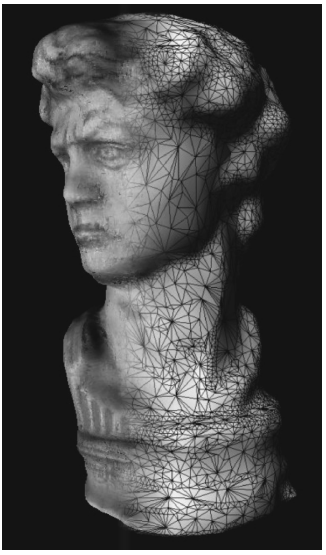

Figure 14: Final 3D model.

References

[1] F. Pedersini, A. Sarti, S. Tubaro: "Visible Surface Reconstruction with Accurate Localization of Object Boundaries", *IEEE Tr. on Circuits and Systems for Video Technology*, Vol. 10, No. 2, March 2000, pp. 278-291.

[2] P.J. Besl, N.D. McKay, "A method for registration of 3-D shapes", *IEEE Tr. on Pattern Analysis and Machine Intelligence*. Vol. 14, No. 2, 1992, pp. 239-256.

[3] R. Malladi, J.A. Sethian, B.C. Vemuri, "Shape Modeling with Front Propagation: A Level Set Approach". *IEEE Tr. on Pattern Analysis and Machine Intelligence*, Feb. 1995, Vol. 17, No. 2, pp. 158-175.

[4] S. Osher, J.A. Sethian, "Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations". *J. Comput. Phys*. Vol. 79, pp. 12-49, 1988.

[5] O. Faugeras, R. Keriven."Variational Principles, Surface Evolution, PDE's, level set Methods, and the Stereo Problem". *IEEE Tr. on Image Processing*, Vol. 7, No. 3, March 1998.

[6] P. Pigazzini, F. Pedersini, A. Sarti, S. Tubaro: "3D Area Matching with Arbitrary Multiview Geometry". *Signal Processing: Image Communications*. Vol. 14, Nos. 1-2, 1998, pp. 71-94.