

Implicit Meshes for Surface Reconstruction

Slobodan Ilic and Pascal Fua

Ecole Polytechnique Fédérale de Lausanne (EPFL)

Computer Vision Laboratory

CH-1015 Lausanne, Switzerland

{Slobodan.Ilic, Pascal.Fua}@epfl.ch¹

Abstract

Deformable 3–D models can be represented either as traditional explicit surfaces, such as triangulated meshes, or as implicit surfaces. Explicit surfaces are widely accepted because they are simple to deform and render, but fitting them involves minimizing a non-differentiable distance function. By contrast, implicit surfaces allow fitting by minimizing a differentiable algebraic distance, but are harder to meaningfully deform and render.

Here we propose a method that combines the strength of both approaches. It relies on a technique that can turn a completely arbitrary triangulated mesh, such as one taken from the web, into an implicit surface that closely approximates it and can deform in tandem with it. This allows both automated algorithms to take advantage of the attractive properties of implicit surfaces for fitting purposes and people to use standard deformation tools they feel comfortable for interaction and animation purposes.

We demonstrate the applicability of our technique to modeling the human upper-body, including face, neck, shoulders and ears, from noisy stereo and silhouette data.

Index Terms

Computer Vision, Reconstruction, Surface Fitting, Modeling, Optimization

I. INTRODUCTION

In the world of Computer Vision and Computer Graphics, 3–D objects tend to be modeled as explicit surfaces such as triangulated meshes or parametric surfaces. Because such representations are intuitive and easy to manipulate, they are widely accepted both by researchers and by graphics designers. These representations, however, are not necessarily ideal for fitting surfaces to potentially noisy and incomplete data such as 3–D points produced by stereo systems or 2–D points from image contours. Fitting typically involves finding the facets that are closest to the 3–D data points or most likely to be silhouette facets, which introduces non-differentiabilities that degrade the convergence properties of most optimizers.

Implicit surfaces are well-suited for simulating physically based deformations [25], [28], [37] and for modeling smooth objects [2], [39]. Because the algebraic distance to an implicit surface is differentiable, they do not suffer from the drawbacks discussed above when it comes to fitting them to 2 and 3–D data [29], [35]. However, they have not gained wide acceptance, in part because they are more difficult to deform and to render than explicit surfaces.

In short, explicit surface representations are well suited for graphics purposes, but less so for fitting and automated modeling. The reverse can be said for implicit surface representations. In this paper, we propose to combine the strengths of both approaches and to avoid their drawbacks by:

- 1) transforming explicit triangulated surfaces into implicit ones, whose shape closely approximates that of the original triangulations;
- 2) deforming the implicit and the explicit surfaces in tandem for fitting and rendering purposes.

We achieve this by attaching a volumetric primitive to each facet of the explicit mesh to turn it into an *implicit mesh* as shown in Fig. 1. Its shape depends *only* on the facet geometry and, when a facet deforms, so does the corresponding primitive. For fitting purposes, we can use any appropriate parameterization of the explicit surface since it uniquely defines that of the implicit one. In this work, we use either Dirichlet Free Form Deformations [26] or PCA weights [4].

Our contribution is therefore an approach to surface reconstruction that lets us take an explicit surface model of arbitrary complexity and regularity, turn it into an implicit mesh, and take advantage of the attractive properties of implicit surfaces for fitting purposes. Because the implicit

surface closely approximates the explicit one and they deform together, the reshaped explicit mesh is also available for rendering and animation. This lets us handle arbitrary triangulations that were not necessarily designed with fitting in mind.

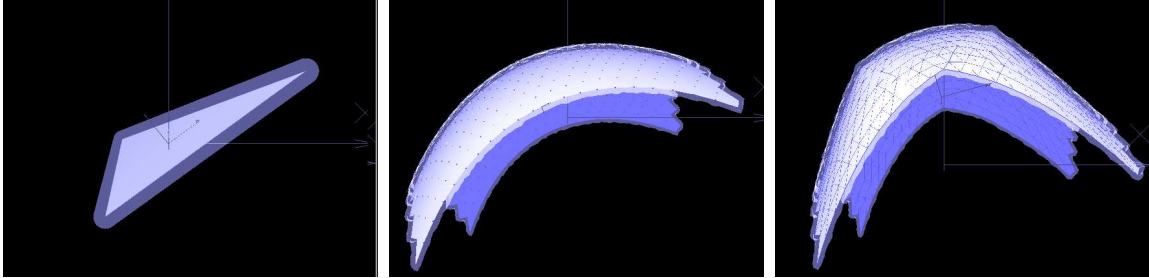


Fig. 1. Converting an explicit surface into an implicit mesh. From left to right: single facet, explicit surface mesh before and after deformation. The implicit meshes that approximate them are shown in transparent light gray.

In the remainder of the paper, we first briefly review earlier approaches. We then introduce our approach to creating implicit meshes and fitting them to image data. Finally we demonstrate its applicability to the complex task of fitting upper-body model that include head, neck, shoulders, and ears, to stereo and silhouette data.

II. Related work

Three-dimensional reconstruction of visible surfaces continues to be an important application of Computer Vision and many approaches relying on full 3-D explicit representations, such as 3-D surface meshes [6], [38], parameterized surfaces [24], [34], local surfaces [14], particle systems [36] and PDE-based models [10], [20] have been proposed. In the Computer Graphics world, there has also been a great deal of work on fitting parametric surfaces, such as B-spline patches [11], [21], and subdivision surfaces [16], [23], to unorganized 3D data. They are typically used to reconstruct surfaces from relatively clean laser-scanned data without using a predefined model. Some of these methods can automatically retrieve the structure from unorganized sets of data points [11], [16]. However, when dealing with very noisy and incomplete data such as the stereo disparity maps we use, they are less than ideal. Furthermore, tangent plane or G^1 continuity constraints have to be enforced at the boundaries between patches, which may result in an ill-posed constrained least-square problem.

There has also been sustained interest in the use of volumetric primitives [28], [38] and implicit surface representations [8], [27], [29], [32], [35] for fitting purposes. Most of these methods have been tailored for specific shapes such as the human body. However, more generic methods that rely on implicit algebraic surface splines have also been investigated [2]. These representations are based on both the parametric and implicit nature of B-spline basis functions. Because, complex shapes may require very many patches, fitting such a model to 3-D data would involve a large number of control parameters to optimize. Implicit surfaces can be deformed by twisting, bending, and tapering the space in which the model lives using a suitable warping function [3], [5], [31], [40]. However, these deformations are limited to surfaces with both parametric and implicit representations, such as spheres or cylinders.

Radial basis functions (RBF) based implicit surfaces [12], [13], [39] and Moving Least-Square (MLS) surfaces [1], [22], [32] have proved extremely effective for tasks such as interpolating scattered data, especially when obtained using laser scanners. However, they are not designed to handle the kind of noise one encounters when fitting surfaces to image data extracted using purely passive techniques such as stereo. Furthermore, an RBF surface is controlled by the position of the RBF centers and by the off-surface points, required to eliminate trivial solutions. Fitting them then requires computing the RBF weights which uniquely determine the surface. This entails an increase in the number of the variables to be optimized during surface fitting, which is suboptimal in the presence of noisy data because it tends to result in an over-parameterized model and ill-posed fitting problem that requires sophisticated regularization constraints to be solved. We are not aware of any existing technique that can do this. The same can be said for MLS surfaces that achieve smooth interpolation by joining local patches, defined on a local reference domain. By contrast, we will show that because the shape of our implicit meshes only depends on the 3-D position of the underlying explicit mesh, we can easily use well established techniques to reduce the number of optimization variables.

In short, both explicit and implicit approaches to 3-D modeling have their strengths and weaknesses. Explicit surfaces are easy to deform and render, but are not ideal for fitting purposes because of the non-differentiability of the distance function. Implicit surfaces do not suffer from this problem [29], [35], but it is not always easy to control their shape in an intuitively pleasing way. To combine the strengths of both approaches, it is therefore important to be able to go back and forth between the two kinds of representations, which is what we propose here.

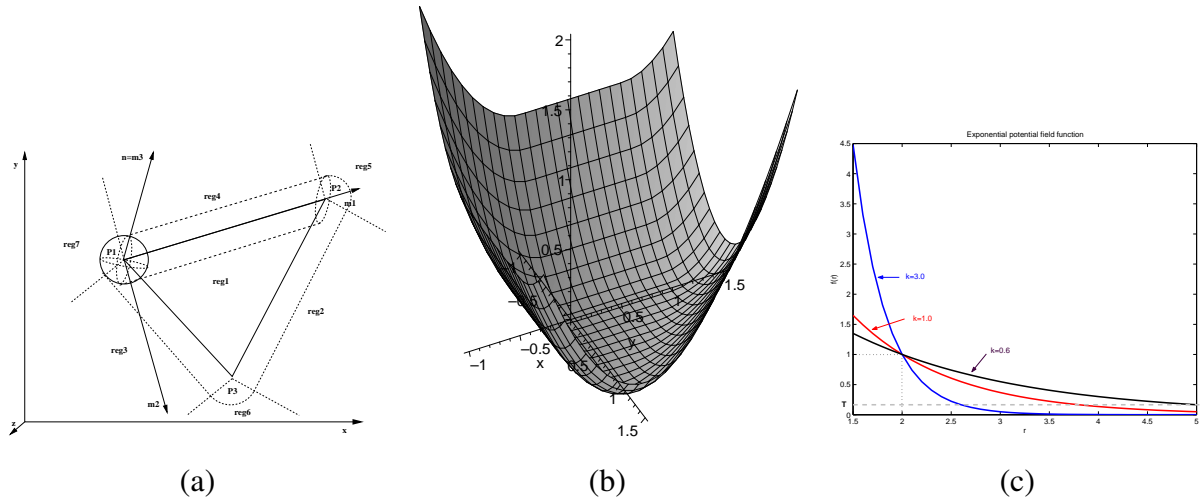


Fig. 2. Distance function for triangular metaballs. (a) Seven-region segmentation of the facet plane used in Eq. 1. We also represent the local coordinate system $\{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$. (b) Distance function $d(\mathbf{x})$ for a standardized facet with vertices $P_1 = \{0, 0, 0\}$, $P_2 = \{1, 0, 0\}$ and $P_3 = \{0, 1, 0\}$ when \mathbf{x} moves in the $z = 0$ plane. Note that it is smooth across the facet edges. (c) Exponential potential field function, showing how the smoothing parameter k controls the range of influence of the primitives and, thus, the amount of smoothing.

III. FROM EXPLICIT TO IMPLICIT MESHES

To create an implicit mesh that can deform in tandem with an explicit one, we define an implicit surface that closely approximates the explicit mesh and whose deformations depend only on the motion of its vertices.

To this end, we attach a volumetric primitive, or metaball, to each facet. This can be done in two different ways. The simplest is to use spherical primitives as was done in [18], which works best for fairly regular and high-resolution meshes. The more general approach to accurately approximating arbitrarily low-resolution or irregular meshes we advocate here is to use the triangular metaballs depicted by Fig. 1. These are defined in terms of a C^1 distance function d that closely approximates the orthogonal distance to the whole facet as a piecewise-polynomial function and is computed as follows.

Let \mathbf{x} be a 3-D point and F a facet with vertices P_1 , P_2 , and P_3 . To compute $d(\mathbf{x})$, the distance of \mathbf{x} to F , we first define the seven-region partition of the facet plane depicted by Fig. 2(a) and compute $P_{\mathbf{x}}$, the orthogonal projection of \mathbf{x} into that plane. When \mathbf{x} projects inside the facet, $P_{\mathbf{x}}$ belongs to $reg1$ and we take $d(\mathbf{x})$ to be the squared orthogonal distance to the plane. If \mathbf{x} projects outside of the facet but in in the bands perpendicular to the edges, $P_{\mathbf{x}}$ falls

$$d(\mathbf{x}) = \begin{cases} \left(\frac{\mathbf{n} \bullet (\mathbf{x} - \mathbf{P}_1)}{\|\mathbf{n}\|} \right)^2 & \text{if } P_{\mathbf{x}} \in \text{reg1} \\ \frac{\|(\mathbf{x} - \mathbf{P}_i) \bullet \mathbf{e}\|^2}{\|\mathbf{e}\|^2} & \text{if } P_{\mathbf{x}} \in \text{reg2}, \text{reg3}, \text{reg4}, P_i \in \{P_2, P_3, P_1\} \\ \|\mathbf{x} - \mathbf{P}_i\|^2 & \text{if } P_{\mathbf{x}} \in \text{reg5}, \text{reg6}, \text{reg7}, P_i \in \{P_2, P_3, P_1\} \end{cases} \quad (1)$$

within regions *reg2*, *reg3* or *reg4*. We then take $d(\mathbf{x})$ to be the squared Euclidean distance from the closest facet edge. In the remaining cases, $P_{\mathbf{x}}$ falls within regions *reg5*, *reg6* or *reg7* and we take $d(\mathbf{x})$ to be the Euclidean distance to the closest vertex. We therefore write where \mathbf{n} is the facet normal and $\mathbf{e} \in \{P_2P_3, P_1P_3, P_1P_2\}$ is the direction vector of the edge closest to $P_{\mathbf{x}}$. Fig. 2(b) depicts $d(\mathbf{x})$ for a standardized facet with vertices $P_1 = \{0, 0, 0\}$, $P_2 = \{1, 0, 0\}$ and $P_3 = \{0, 1, 0\}$ when \mathbf{x} moves in the $z = 0$ plane. Inside the triangle, the distance is zero. Outside, it is a quadric that derivatives are zero along the edges and at the vertices, which results in smooth blending across regions.

Note that the distance of a point to a facet's edge that appears on the second line of Eq. 1 is the cylindrical distance to that edge. Similarly, the distance to a vertex that appears on the third line of Eq. 1 is the spherical distance to that vertex. Intuitively, a triangular metaball can be understood as being made of two planes, one on each side of the explicit facet, that blend seamlessly with three implicit cylinders whose axes are aligned with the edges and three implicit spheres centered at the vertices. The cylinders and spheres are represented by dotted lines in Fig. 2(a). In a companion report [19], we formalize these observations in terms of a matrix representation, which we then use to formally prove that the distance of Eq. 1 is C^1 with respect to the 3-D coordinates of both the vertices and data points.

Finally, the distance function can be incorporated in the following potential field function

$$f_i(\mathbf{x}) = \exp(-k(d_i(\mathbf{x}) - d_0^2)) \quad , \quad (2)$$

where d_0 represents the constant thickness of the implicit surface. Again, the total field is the sum of the individual metaball fields, which yields the final expression of the implicit surface as the set of points $\mathbf{x} \in R^3$ such that

$$F(\mathbf{x}) = T - \sum_{i=1}^N \exp(-k(d_i(\mathbf{x}) - d_0^2)) \quad . \quad (3)$$

where T is a given iso-value. Note that the isosurfaces of the potential field, and therefore the shape of the implicit mesh, are relatively unaffected by changes in the resolution of the explicit surface. This is because the isosurfaces tend to be parallel to the facet planes and their distances to the facets depend only on the d_0 and k parameters of Eq. 2, over which we have full control, as opposed to facet sizes.

In practice, both the d_0 thickness and k smoothing parameters of Eq. 2 influence the smoothness and accuracy of the implicit mesh. Increasing d_0 or decreasing k tends to smooth the implicit surface at the cost of increasing the volume they enclose. Our goal is therefore to find the best possible compromise between accuracy and smoothness as a function of k and d_0 . To quantify the influence of these two parameters, we introduce quantitative measures of smoothness and accuracy that depend on the thickness of the volume enclosed by the implicit mesh.

- **Surface waviness** = $w(k, d_0)$: Average ratio of minimal and maximal volume thickness evaluated respectively at the center of gravity of the facets and at their vertices.
- **Surface thickness** = $t(k, d_0)$: Average volume thickness measured at the same locations as those used to evaluate waviness.

In practice, we constrain d_0 to be less than 10% of the average edge length and seek values of d_0 and k such that

$$w(k, d_0) > 1.0 - w_{max} \quad , \quad (4)$$

$$t(k, d_0) < d_0 + t_{max} \quad , \quad (5)$$

where w_{max} and t_{max} are two user specified thresholds [19]. Note that for generic models such as the ones of Fig. 3(a), this computation needs only be performed once and the optimal values of k and d_0 stored and reused for all subsequent fits to image data.

IV. OPTIMIZATION FRAMEWORK

In this section, we introduce the framework we have developed to fit *generic models* such as the ones of Fig. 3(a,c) to the image data, which here comes in the form of 3-D point clouds from stereo and 2-D silhouette points from occluding contours.

A. Reparametrizing the Implicit Meshes

The shape of our implicit meshes depends only on the position of the 3-D vertices of the corresponding explicit meshes. In theory, for fitting purposes, we could optimize the shape with

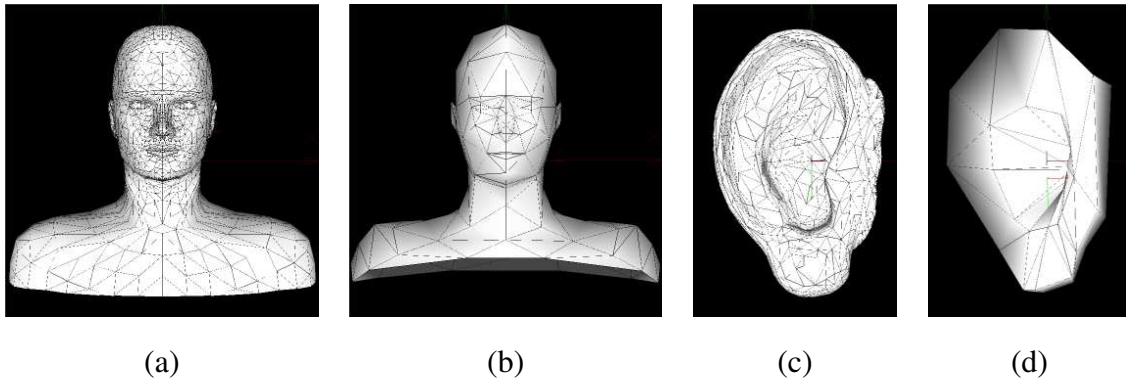


Fig. 3. Surface and control triangulations. (a,c) The generic low-resolution triangulation we use for upper-body and ear modeling. (b,d) A subset of vertices serve as DFFD control points. They are themselves triangulated to impose the regularization constraints of Section IV.

respect to the coordinates of these vertices. However, because image data is very noisy, this would amount to over-parameterization and would not yield good results. In earlier work [17], we showed that introducing Dirichlet Free Form Deformations (DFFDs) is an effective way to handle this problem when fitting explicit meshes.

Unlike other Free Form Deformation methods [7], [31], DFFDs do not require the control points to lay on a regular rectangular grid. This is achieved by replacing the usual rectangular local coordinates by generalized natural neighbor coordinates [33]. This gives us the ability to place control points at arbitrary locations. Given such a set of control points $\{P_1, \dots, P_N\}$ the surface shape is entirely described by the state vector $\mathbf{S} = \{\Delta P_1, \dots, \Delta P_N\}$ formed by concatenating the displacements of all the control points with respect to their original positions.

This approach naturally extends to implicit meshes since the distance $d(\mathbf{x})$ of Eq. 1 depends only the 3-D coordinates of \mathbf{x} and on the corresponding facet's vertices displacements, that are themselves linear functions of the ΔP_i displacements introduced above. We therefore rewrite the distance function of Eq. 1 as $d(\mathbf{x}, \mathbf{S})$, where \mathbf{S} is the state vector defined in the previous paragraph. Similarly, when considering all the facets together, we rewrite the field potential function F of Eq.3 as

$$F(\mathbf{x}, \mathbf{S}) = T - \sum_{i=1}^N \exp(-k(d_i(\mathbf{x}, \mathbf{S}) - d_0^2)) , \quad (6)$$

where \mathbf{x} is a point in R^3 and d_i is the distance to facet F_i .

In this fashion, we have parametrized both the explicit and the implicit surface in terms of the \mathbf{S} state vector. As discussed in the following section, this will allow us to deform both representations in tandem to fit the corresponding surface to image data by minimizing a differentiable objective function.

B. Objective Function for Implicit Meshes

In the case of implicit meshes, we use the image data to write n_{obs} *observation equations* of the form

$$F(\mathbf{x}_i, \mathbf{S}) = \epsilon_i, \quad 1 \leq i \leq n_{obs}, \quad (7)$$

where F is the field function of either Eq. 6, \mathbf{x}_i one of the data points, \mathbf{S} the state vector, and ϵ_i the corresponding residual. ϵ_i is the algebraic distance to the implicit mesh and should be as small as possible. Fitting therefore implies minimizing

$$\chi^2 = \mathbf{v}^t W \mathbf{v} \quad (8)$$

where $\mathbf{v} = [\epsilon_1, \dots, \epsilon_{n_{obs}}]$ is the vector of residuals and W a diagonal weight matrix associated to the observations. In practice, our system must be able to deal with observations coming from different sources. To guarantee that their influences are commensurate, we assign a weight w_{type_i} to each kind of observation, where $type_i$ is the nature of the observation, and minimize

$$\chi^2 = 1/2 \sum w_{type_i} \epsilon_i^2, \quad (9)$$

where the w_{type_i} are chosen so that the contribution to the objective function gradients of all the observations of a particular kind are of similar magnitudes [15].

Because there are both noise and potential gaps in the image data, we found it necessary to introduce a regularization term that guarantees smooth deformation of the generic model. This is achieved by triangulating the control points as shown in Fig. 3(b,d) and introducing a deformation energy E_D that approximates the sum of the square of the derivatives of displacements across the control surface. By treating the control triangulation facets as C^0 finite elements, we can write

$$E_D = \Delta_x^T K \Delta_x + \Delta_y^T K \Delta_y + \Delta_z^T K \Delta_z \quad (10)$$

where K is a stiffness matrix and Δ_x, Δ_y and Δ_z are the vectors of the x, y and z coordinates of the control vertices' displacements. The term we actually optimize becomes

$$E_T = \chi^2 + \lambda_D E_D , \quad (11)$$

where λ_D is a small positive constant. Because F is differentiable everywhere, so is E_T . Furthermore, this formulation allows us to treat stereo and silhouette data in a similar way.

V. RESULTS

To validate our approach, we focus on using stereo and silhouette data because they are complementary sources of information. Stereo works well on textured surfaces facing the camera but fails where the view direction and the surface normal is close to being orthogonal, which is exactly where silhouettes provide robust information.

We first illustrate the behavior of our implicit meshes on upper body data, including head, neck, shoulders, and ear. We compare our triangular implicit meshes against traditional explicit meshes and discuss fitting accuracy and computational complexity issues.

A. Fitting to Stereo and Silhouette Data

In the example of Fig. 4 we use as input an initially uncalibrated 11-frame video sequence in which the camera moves around a static subject. We first used a model-driven bundle-adjustment technique [15] to compute the relative motion and, thus, register the images. We then ran a maxflow graph-cut algorithm [30] to derive disparity maps from consecutive image and produce the clouds of 3-D points depicted by images (b,d) in the figure's middle row. We used snakes to outline the silhouettes shown as white lines in the top row. We also picked five 2-D points—outside corners of the eyes, corners of the mouth and tip of the nose—to compute rotations and translations that roughly line up the generic model with the images. As shown in images (a,c,e) of the figure's middle row, even though the face is relatively well aligned, the generic shoulders are fairly far from the real ones. The bottom row of the figure depicts the fitted and textured model whose projections are now correct in all views. This shows that the recovered shape is geometrically accurate even at places where the surface slants away from the cameras and, therefore, where stereo fails. Note that these texture-mapped views were generated using standard OpenGL tools to render the explicit surface that was deformed in tandem with the

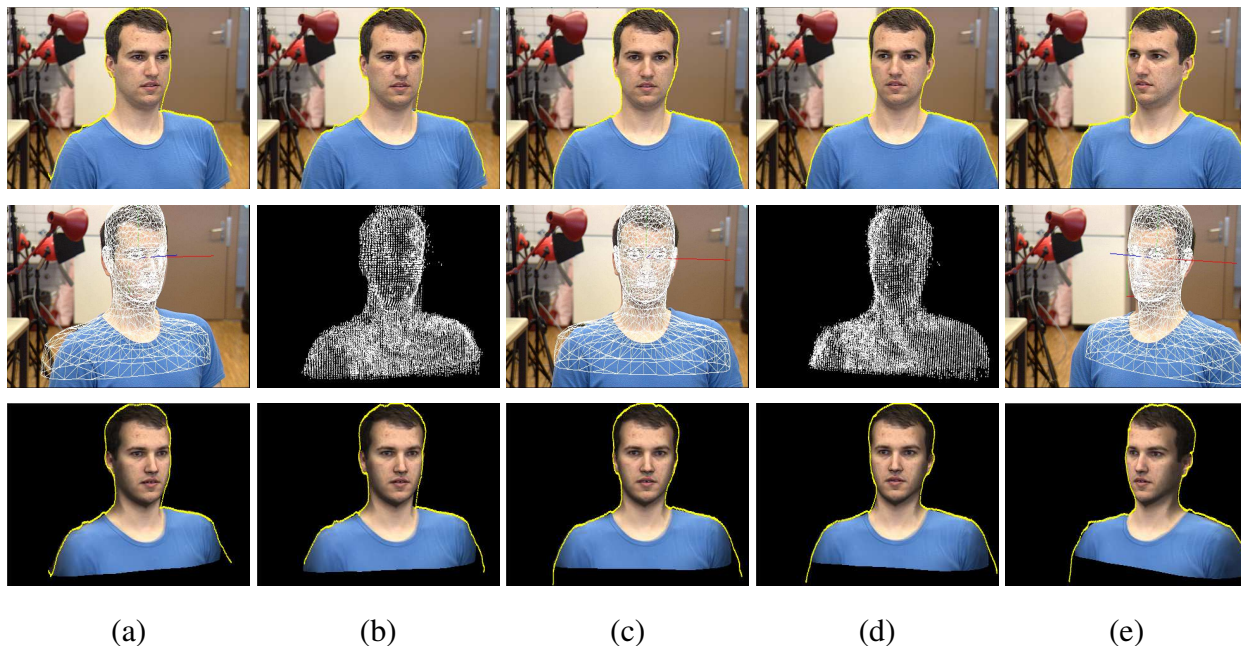


Fig. 4. Reconstruction from an uncalibrated video sequence. Top row: 5 of 7 images from a short video sequence with overlaid silhouettes for the neck and shoulders. Middle row: (a,c,e) Projection of the initial 3-D wireframe model on the image (b,d) Projection of the cloud of 3-D points computed using a maxflow graph-cut stereo algorithm. Bottom row: Textured reconstructed models obtained by using a triangular implicit mesh model for the upper body.

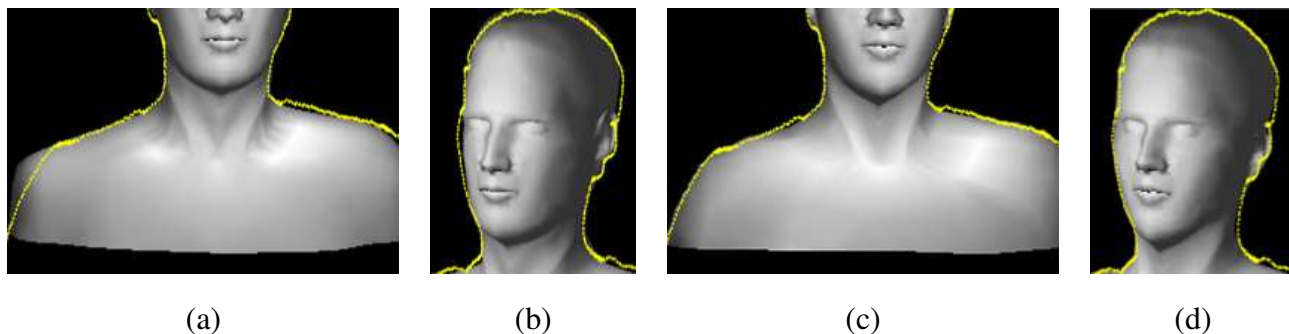


Fig. 5. Comparing explicit and implicit approaches to fitting our model to the stereo and silhouette data of Fig. 4. We show magnified front view of the shoulders and side view of the face of the shaded model with the occluding contours overlaid as white lines. (a,b) Using explicit surfaces yields a poor fit around the shoulders and the face. (c,d) Implicit surfaces yields a 3-D surface whose projection matches the occluding contours much better.

implicit one. In other words, having both kinds of representation simultaneously available at the same time spared us the need to use sophisticated and time consuming implicit surface rendering techniques.

The results of Fig. 4 were obtained by attaching triangular primitives to the low resolution model of Fig. 3(a). In Fig. 5, we compare the results of fitting the model to the same data using either our implicit surface formalism or a more traditional scheme. In both cases, the meshes were parameterized in terms of the DFFD control points of Fig. 3(b). When not using implicit surfaces, we minimize an objective function that is designed to constrain facets near the occluding contour to be seen edge-on [19]. More precisely, given a silhouette point and the line-of-sight it defines, we look for a facet that is almost parallel to it and such that there is a 3-D point along this line for which the Euclidean distance from the point to the facet is small. If such a facet exists, we introduce in the objective function a term proportional to the square of this distance. Minimizing the objective function then tends to make the distance even smaller and results in the facet producing an occluding contour that goes through the silhouette point. This explicit formulation is much more sensitive than the implicit one to the presence of large facets that degrade the accuracy and yields noticeably worse results near the occluding contours.

In a second example, we consider a human ear, whose shape is far more complex. As shown in Fig. 6(a), we projected textured light on an ear and acquired a stereo pair of images, which allowed us to compute a fairly dense disparity map. We then outlined occluding contours in both images and fitted the model of Fig. 3(c), which we found on the web, to this stereo and silhouette data, both with and without using our implicit surface formalism. Again, as shown in Fig. 6(d), it is only when using our implicit surfaces to perform the fit that the model’s occluding contours end up lining up correctly with the silhouettes.

In the final example of Fig. 8, we demonstrate the versatility of our approach by replacing the DFFD models we have used so far by high resolution morphable face models parameterized in terms of PCA weights [4]. In earlier work [9], we showed that they can be fitted to short video sequences using point correspondences. Here, by simply turning them into implicit meshes and retaining the same parameterization, we can further refine them using silhouette information. Note that even though the model, with its 75972 vertices and 150958 facets, is very complex, fitting was performed in a reasonable time of approximately 10 min on a 2.8GHz PC.

B. Accuracy and Computational Complexity

To quantify fitting error and computational complexity, we started with Ear_0 (1167 vertices, 1620 facets), the model of Fig. 3(c). We then subdivided its facets to obtain three additional

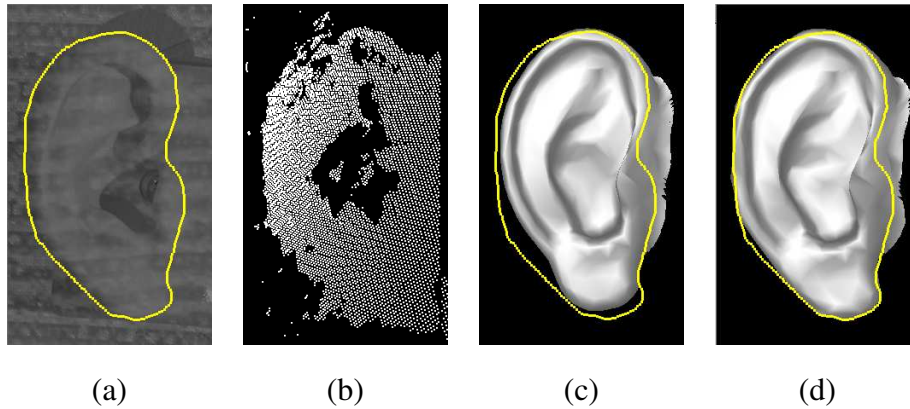


Fig. 6. Ear modeling. (a) One image of the stereo pair with overlaid silhouettes. (b) Corresponding cloud of 3D points. (c) Shaded ear model fitted using explicit surfaces. (d) Shaded ear model fitted using implicit surfaces. Note that the model's occluding contours match the actual silhouettes much better.

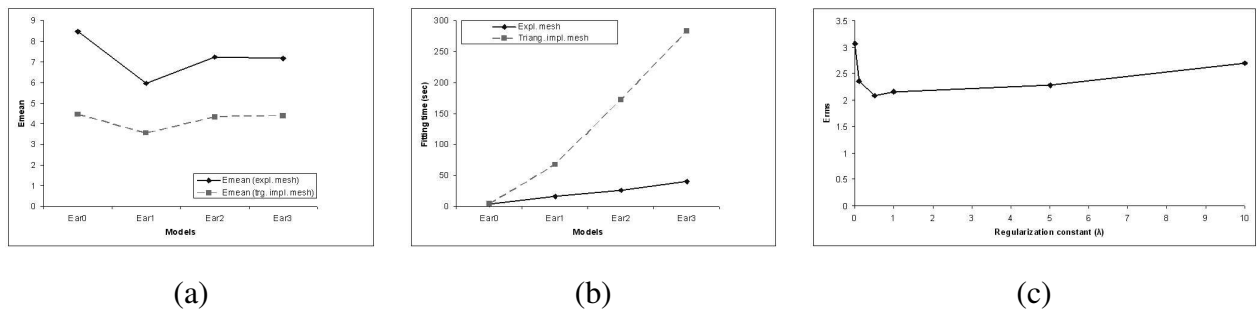


Fig. 7. Measured fitting errors and computation times as a function of increasing resolution. (a) Mean distance in millimeters of explicit and implicit mesh to the silhouette data. (b) Computation times for explicit and implicit mesh fitting on a 2.6GHz PC in seconds. (c) Influence of the regularization constant λ on the root mean square error of the fit.

models of increasing complexity: Ear_1 (3954 vertices, 6480 facets), Ear_2 (14388 vertices, 25920 facets) and Ear_3 (54696 vertices, 103680 facets). Fig. 7(a) depicts the mean distance of the lines of sight defined by the silhouette points to the models fitted with or without using implicit meshes. When using them, the accuracy varies little with mesh resolution and, as observed before, is much better than the one obtained without them. As shown in Fig. 7(b), there is a computational price to be paid for using implicit meshes. Note however that, because the result is fairly insensitive to mesh resolution, there is no advantage to subdividing the mesh and one need therefore not incur this penalty. Furthermore, even for the model with 103680 facets, the computation time remains manageable on a modern machine.

To gauge the influence of the regularization constant λ_D of Eq. 11, in Fig. 7(c), we plot the

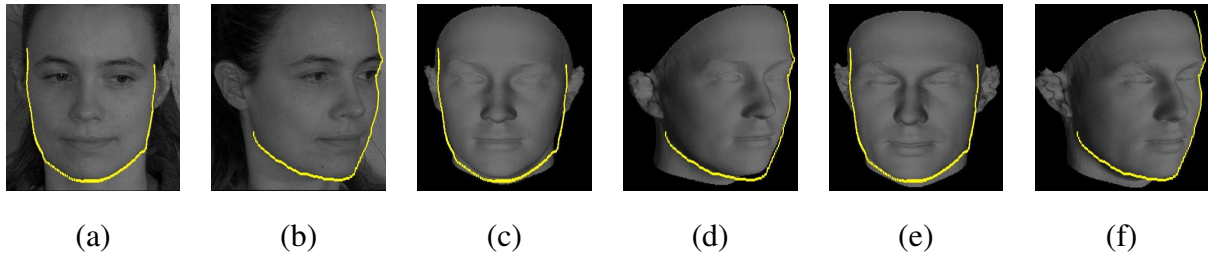


Fig. 8. Head modeling using high resolution morphable models. (a,b) Two images from a short video sequence with outlined silhouettes. (c,d) Recovered face shape using only correspondences. Note that the shape of the occluding contours is somewhat inaccurate. (e,f) Improved shape using silhouette information.

value of $\sqrt{(\sum w_{type_i} \epsilon_i^2 / \sum w_{type_i})}$, the root mean square residual of our fit using the notations of Eq. 9. For very small values of λ_D , the influence of the regularization term is insufficient to “convexify” the problem and the optimizer tends to get trapped in meaningless local minima. For very large values of λ_D , the model becomes too stiff again resulting again in increased errors. However, between these extremes, there is a large range in which the result is fairly insensitive to the exact value of λ_D , making it easy to pick an appropriate value.

VI. CONCLUSION

We have presented an approach to combining explicit and implicit surface representations that allows us to take advantage of the strengths of both. To this end, we have developed a technique for creating *implicit meshes* from explicit ones by attaching triangular primitives to their facets. These primitives are defined in such a way that their shape depends only on the 3D location of the mesh vertices, which allows us to simultaneously fit both representations to image data by minimizing a differentiable objective function.

We have chosen to use DFFD control points to parameterize the position of the mesh vertices, which allows us to perform this minimization with respect to a limited number of parameters. Our method, however, is generic as we demonstrated by also applying it to morphable face models [4].

We used the example of upper-body modeling using stereo and silhouette data to demonstrate the power of this approach. The explicit models we used were not tailored for fitting purposes and exhibited both highly irregular facets and a complex topology, none of which had a significant impact on the quality of the fitting.

REFERENCES

- [1] N. Amenta and Y. Kil. Defining point-set surfaces. *Computer Graphics, SIGGRAPH Proceedings*, pages 264–270, July 2004.
- [2] C. Bajaj and I. Ihm. C^1 Smoothing of Polyhedra with Implicit Algebraic Splines. *Computer Graphics, SIGGRAPH Proceedings*, 26(2):79–88, 1992.
- [3] A. H. Barr. Local and global deformations of solid primitives. *Computer Graphics, SIGGRAPH Proceedings*, pages 21–30, September 1984.
- [4] V. Blanz and T. Vetter. A Morphable Model for The Synthesis of 3–D Faces. In *Computer Graphics, SIGGRAPH Proceedings*, Los Angeles, CA, August 1999.
- [5] J. F. Blinn. A Generalization of Algebraic Surface Drawing. *ACM Transactions on Graphics*, 1(3):235–256, 1982.
- [6] I. Cohen, L. D. Cohen, and N. Ayache. Introducing new deformable surfaces to segment 3D images. In *Conference on Computer Vision and Pattern Recognition*, pages 738–739, 1991.
- [7] S. Coquillart. Extended Free-Form Deformation: A sculpturing Tool for 3D Geometric Modeling. *Computer Graphics, SIGGRAPH Proceedings*, 24(4):187–196, 1990.
- [8] M. Desbrun and M.P. Gascuel. Animating Soft Substances with Implicit Surfaces. *Computer Graphics, SIGGRAPH Proceedings*, pages 287–290, 1995.
- [9] M. Dimitrijević, S. Ilić, and P. Fua. Accurate Face Models from Uncalibrated and Ill-Lit Video Sequences. In *Conference on Computer Vision and Pattern Recognition*, Washington, DC, June 2004.
- [10] Ye Duan, Liu Yang, Hong Qin, and Dimitris Samaras. Shape reconstruction from 3d and 2d data using pde-based deformable surfaces. In *European Conference on Computer Vision*, volume 3, pages 238–251, Prague, Czech Republic, May 2004.
- [11] M. Eck and H. Hoppe. Automatic reconstruction of B-spline surfaces of arbitrary topological type. In *Computer Graphics, SIGGRAPH Proceedings*, pages 325–334, 1996.
- [12] J. C. Carr et al. Reconstruction and representation of 3d objects with radial basis functions. In *Computer Graphics, SIGGRAPH Proceedings*, volume 2, 2001.
- [13] J. C. Carr et al. Smooth surface reconstruction from noisy range data. In *ACM GRAPHITE*, pages 119–126, Melbourne, Australia, 2003.
- [14] F. P. Ferrie, J. Lagarde, and P. Whaite. Recovery of Volumetric Object Descriptions from Laser Rangefinder Images. In *European Conference on Computer Vision*, Genoa, Italy, April 1992.
- [15] P. Fua. Regularized Bundle-Adjustment to Model Heads from Image Sequences without Calibration Data. *International Journal of Computer Vision*, 38(2):153–171, July 2000.
- [16] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jun, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise Smooth Surface Reconstruction. In *Computer Graphics, SIGGRAPH Proceedings*, pages 295–302, 1994.
- [17] S. Ilić and P. Fua. Using Dirichlet Free Form Deformation to Fit Deformable Models to Noisy 3-D Data. In *European Conference on Computer Vision*, Copenhagen, Denmark, May 2002.
- [18] S. Ilić and P. Fua. Implicit Mesh Models for Modeling and Tracking. In *Conference on Computer Vision and Pattern Recognition*, Madison, WI, June 2003.
- [19] S. Ilić and P. Fua. Implicit Meshes for Surface Reconstruction. Technical Report IC/2004/25, EPFL, 2004.
- [20] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

- [21] V. Krishnamurthy and M. Levoy. Fitting smooth surfaces to dense polygon meshes. In *Computer Graphics, SIGGRAPH Proceedings*, pages 313–324, 1996.
- [22] D. Levin. Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization Edited by Brunnett, Hamann and Mueller, Springer-Verlag*, pages 37–49, 2003.
- [23] N. Litke, A. Levin, and P. Schröder. Fitting subdivision surfaces. In *Proceedings of the conference on Visualization '01*, pages 319–324. IEEE Computer Society, 2001.
- [24] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(441-450), 1991.
- [25] D. Metaxas and D. Terzopoulos. Shape and Nonrigid Motion Estimation through Physics-Based Synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1991.
- [26] L. Moccozet and N. Magnenat-Thalmann. Dirichlet Free-Form Deformation and their Application to Hand Simulation. In *Computer Animation*, 1997.
- [27] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level Partition of Unity Implicits. In *Computer Graphics, SIGGRAPH Proceedings*, volume 22(2), pages 463–470, 2003.
- [28] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:715–729, 1991.
- [29] R. Plänkers and P. Fua. Articulated Soft Objects for Multi-View Shape and Motion Capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.
- [30] S. Roy and I.J. Cox. A Maximum-Flow Formulation of the N-camera Stereo Correspondence Problem. In *International Conference on Computer Vision*, pages 492–499, Bombay, India, 1998.
- [31] T.W. Sederberg and S.R. Parry. Free-Form Deformation of Solid Geometric Models. *Computer Graphics, SIGGRAPH Proceedings*, 20(4), 1986.
- [32] C. Shen, J. F. O’Brien, and J. R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. In *Proceedings of ACM SIGGRAPH 2004*. ACM Press, August 2004.
- [33] R. Sibson. A Vector Identity for the Dirichlet Tessellation. In *Math. Proc. Cambridge Philos. Soc.*, pages 151–155, 1980.
- [34] E. M. Stokely and S. Y. Wu. Surface parameterization and curvature measurement of arbitrary 3-d objects: five practical methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):833–839, August 1992.
- [35] S. Sullivan, L. Sandford, and J. Ponce. Using geometric distance fits for 3–d. object modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1183–1196, December 1994.
- [36] R. Szeliski and D. Tonnesen. Surface Modeling with Oriented Particle Systems. In *Computer Graphics, SIGGRAPH Proceedings*, volume 26, pages 185–194, July 1992.
- [37] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:703–714, 1991.
- [38] D. Terzopoulos and M. Vasilescu. Sampling and reconstruction with adaptive meshes. In *Conference on Computer Vision and Pattern Recognition*, pages 70–75, 1991.
- [39] G. Turk and J.F. O’Brien. Shape transformation using variational implicit functions. In *Computer Graphics, SIGGRAPH Proceedings*, volume 33, pages 335–342, 1999.
- [40] B. Wyvill and K. van Overveld. Warping as a modelling tool for csg/implicit models. In *Shape Modelling Conference, University of Aizu, Japan*, pages 205–214. IEEE Society Computer Press ISBN0-8186-7867-4, March 1997. invited.