FROM EXPLICIT TO IMPLICIT SURFACES FOR VISUALIZATION, ANIMATION AND MODELING

Slobodan Ilic, Pascal Fua
Computer Vision Lab, Swiss Federal Institute of Technology (EPFL)*
Slobodan.Ilic, Pascal.Fua@epfl.ch

KEY WORDS: Vision, Implicit Surfaces, Reconstruction, Modeling.

ABSTRACT

Deformable 3–D models are used extensively in Computer Graphics and Computer Vision for Visualization, Animation and Modeling. They can be represented either as traditional explicit surfaces, such as triangulated meshes, or as implicit surfaces.

On one hand, explicit surfaces can easily be manually deformed by users, such as graphics designers, either directly by moving the mesh vertices or indirectly using a Free Form Deformation approach. On the other hand, implicit representations are well-suited both for physically-based simulations and for modeling noisy image-data. For example, to fit a cloud of 3–D points, implicit representations allow fitting without search because one only needs to evaluate a differentiable field function at every data point, instead of searching for the facets that are closest to the data points.

In this paper we propose a method that can turn an explicit surface into an implicit shell, which closely approximates its shape and can deform in tandem with it. This allows both graphics designers to deform and reshape the implicit surface by manipulating explicit surfaces using standard deformation techniques and automated fitting algorithms to take advantage of the attractive properties of implicit surfaces.

We demonstrate the applicability of our technique for upper body—head, neck and shoulders—modeling and animation.

1 INTRODUCTION

In the world of Computer Graphics, 3D objects tend to be modeled as explicit surfaces such as spline patches or triangulated meshes. Because such representations are intuitive and easy to manipulate, they are widely accepted among graphics designers. As a result, hardware implementations of polygon rendering have become commonplace. These representations, however, are not necessarily ideal for fitting surfaces to data such as 3D points produced by laser-scanners and stereo systems or 2D points from image contours. This stems from the fact that fitting typically involves finding the facets that are closest to the 3–D data points or most likely to be silhouette facets. This requires searching, which is slow, and dealing with the non-differentiability of the distance function, which degrades the convergence properties of most optimizers.

Implicit surfaces known in the literature under different names: *Blobby Molecules(Blinn, 1982), Soft Objects(Wyvill and Wyvill, n.d.) and MettaBalls(Nishimura and al., 1985)* took substantial attention in both Computer Graphics and Computer Vision communities. They are well-suited for simulating physically based processes and for modeling smooth objects. Because the algebraic distance to an implicit surface is computed by evaluating a differentiable function, they do not suffer from the drawbacks discussed above when it comes to fitting them to 2 and 3D data (Sullivan et al., 1994, Plänkers and Fua, 2001a, Desbrun and Gascuel, 1995). 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 very well suited for graphics purposes, but less so for fitting and automated

modeling. The reverse can be said of implicit surface representations. In this paper, we propose to combine the strengths of both approaches and to avoid their drawbacks by:

1. transforming explicit surfaces into implicit surfaces, which we call *implicit shells*, whose shape closely approximates that of the original triangulations as depicted by Fig. 2(d, e, f)

2. deforming the implicit and the explicit surfaces in tandem for fitting and rendering purposes as shown in Fig. 2(g, h, i)

To create the implicit shells we attach spherical or triangular metaballs to each facet of the explicit mesh. The parameters of those metaballs are a function of the facet geometry. As a result, when a facet deforms, so does the corresponding metaball. Particularly we choose Dirichlet Free Form Deformations (Moccozet and Magnenat-Thalmann, 1997, Ilic and Fua, 2002) for deforming the implicit and explicit models in tandem, but the way how implicit shells are created allows using of any other deformation method including direct mesh manipulation or indirect mesh manipulation using set of control points. This means that any FFD, or B-spline based approach of deforming meshes can be used to deform their implicit shells. However, some other indirect methods for explicit surface deformation, such as PCA parmetrization (Blanz and Vetter, 1999), can be used to deform our implicit shells.

Our contribution is therefore an approach to surface fitting that allows us to take an arbitrary explicit surface model, for example one that has been obtained from the web and was not designed with fitting in mind, turn it into an implicit shell, and deform it to obtain an optimal least-square
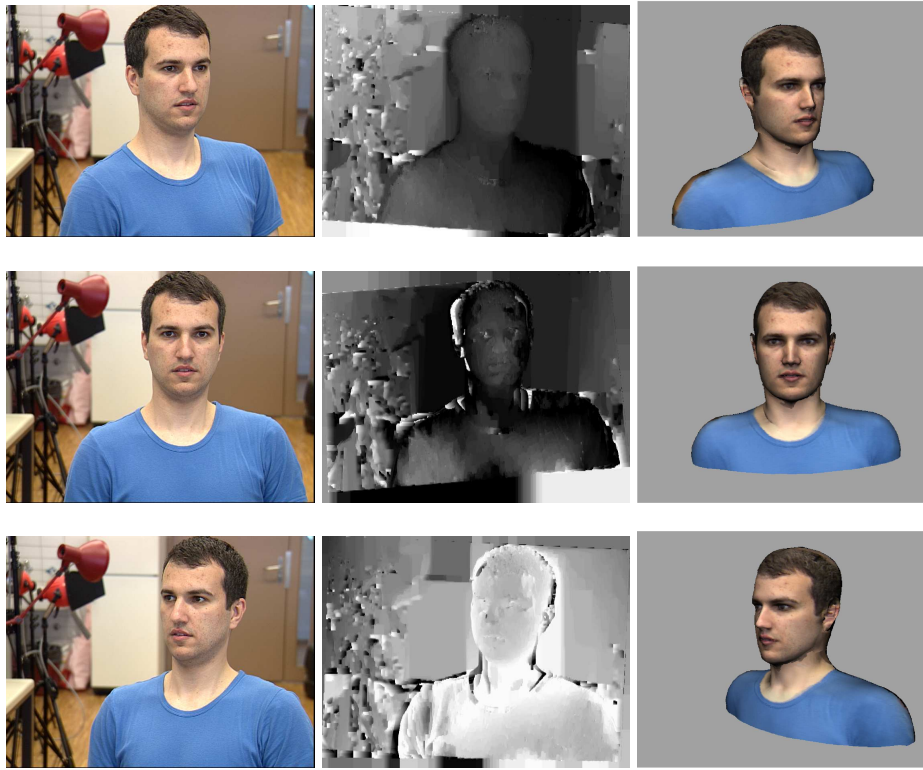

Figure 1: *Reconstruction from an uncalibrated video sequence. Left colum: 3 of 6 images from a short video sequence. Middle colum: Disparity maps extracted from consecutive image pairs using correlation-based stereo. Right coloum: Reconstructed and textured models obtained by using an explicit model for the head and an implicit mesh model for the neck and shoulders.*

fit to new experimental data using a few well-chosen control points. It can then be instantly used, modified or blended with other objects. In Fig. 1, we demonstrate our approach in the complex case of head, neck and shoulders modeling from images. We fit the head separately from the neck and shoulders to image-data, blend them, and, finally, animate them.

In the remainder of the paper, we first briefly review earlier approaches. We then introduce our approach to creating implicit shells and deforming them. Finally, we describe our optimization framework and demonstrate the applicability of our framework to upper-body modeling.

2 PREVIOUS WORK

2.1 Fitting Explicit Surfaces

Three-dimensional reconstruction of visible surfaces continues to be an important goal of the computer vision research community and many approaches relying on full 3–D explicit representations have been proposed, such as 3–D surface meshes (Cohen et al., 1991, Terzopoulos and Vasilescu, 1991), parameterized surfaces (Stokely and Wu, 1992, Lowe, 1991), local surfaces (Ferrie et al., 1992), and particle systems (Szeliski and Tonnesen, 1992). In earlier work (Ilic and Fua, 2002), we showed that Dirichlet Free Form Deformations (DFFDs) (Moccozet and Magnenat-Thalmann, 1997) could be used to robustly fit explicit surfaces to noisy stereo data because they let us parametrize

the surface using a very small number of control points. Unlike the original Free Form Deformations (FFDs) (Sederberg and Parry, 1986) and most of their successors (Coquillart, 1990, Kalra et al., 1992, Chang and Rockwood, 1994), DFFDs do not require the control points to lie on a regular rectangular grid. This is achieved by replacing the standard rectangular local coordinates by generalized natural neighbor coordinates, also known as Sibson coordinates (Sibson, 1980). It gives us the ability to place control points at arbitrary locations rather than on a regular lattice, and thus, much greater flexibility. In practice, control points are taken to be on the surface triangulation, with a denser distribution where the surface curvature is high. DFFD has local deformation property that makes deformation process intuitive and it is desirable for fitting as well. In this paper, we extend this idea to also deforming implicit surfaces.

In the Computer Graphics world, there has also been a great deal of work on fitting parametric surfaces, such as B-spline patches or subdivision surfaces (M. Eck, 1996), to 3D data. They are typically used to produce models from relatively clean laser-scanner data. B-spline patches are easily controlled using a set of control points and widely accepted for CAD modeling purposes. However, when dealing with very noisy data such as the stereo data of Fig. 1, B-spline methods that require many patches to achieve the required modeling precision, and consequently many control points, could be hard-pressed to preserve even G^1 continuity among patches.

2

2.2 Fitting Implicit Surfaces

There has also been sustained interest in the use of volumetric primitives (Kakadiaris and Metaxas, 1996, Terzopoulos and Vasilescu, 1991, Pentland and Sclaroff, 1991) and implicit surface representations (Desbrun and Gascuel, 1995, Sullivan et al., 1994, Plänkers and Fua, 2001b) for fitting purposes. These methods, however, are tailored for specific shapes such as the human body and its skeleton and there is no generally accepted way to deform generic implicit surfaces.

A popular way to deform implicit surfaces is to twist, bend, and taper the space in which the model lives by choosing a suitable warping function (Blinn, 1982, Barr, 1984, Wyvill and van Overveld, 1997). However, these deformations are limited to parametric surfaces, such as spheres or cylinders, and there is no way to warp the space in a free form manner. In (Bardinet et al., 1998), simple superquadrics are parametrized using conventional FFDs for automatic heart reconstruction and deformation from medical images. Here the FFDs ability to deform parametric surfaces has been exploited, but only to reshape a single primitive. Our proposed implicit shells coupled with DFFDs go much further by allowing us to deform completely generic implicit surfaces. In spirit, the implicit shells are related to the earlier distance surfaces (Bloomenthal and Shoemake, 1991). However, in this earlier work, the problems associated to bulges created by metaballs blending into each other are handled by a convolution mechanism that looses the algebraic nature of the distance function and makes the distance surfaces impractical for the kind of fitting we perform.

Radial basis functions (RBF) (J. C. Carr, 2001, J. C. Carr and Beatson, 1997, Turk and O'Brien, 1999) are an interesting alternative to soft objects or metaballs (Wyvill and Wyvill, n.d., Nishimura and al., 1985). The shape of the resulting surface, however, is controlled not only by the position of the RBF centers but also by the RBF weights that have no geometric interpretation, which makes this approach also unsuitable for in-tandem deformation of explicit and implicit surface.

2.3 Explicit vs Implicit

In short, both approaches to 3–D modeling have their strengths and weaknesses for the purpose of fitting noisy image-data. Explicit surfaces are easy to deform and to render using well known computer graphics techniques, but as discussed earlier, are not ideal for fitting purposes. Implicit surfaces are better suited for least-squares style fitting because they can be used to define differentiable objective functions (Sullivan et al., 1994, Plänkers and Fua, 2001a). However, unless one uses either a single geometric primitive or a set of such primitives attached to some kind of skeleton, it is relatively difficult to control their shape in an intuitively pleasing way. As a result users such as graphics designers tend to prefer explicit models. It is therefore important to be able to go back and forth between the two kinds or representations.

3 IMPLICIT SHELLS FROM EXPLICIT MESHES

To create an implicit surface model that can deform in tandem with the explicit surface, we must address two problems:

1. Creating an implicit shell that closely approximates the shape of the initial explicit mesh,

2. Controlling the object shape, in both its explicit and implicit forms, using the same set of parameters.

Our approach is depicted by Fig. 2. We now discuss its components.

3.1 Explicit Surface Deformation

We have shown in earlier work that introducing DFFD control points is an effective way to deform explicit meshes (Ilic and Fua, 2002). These points can be distributed freely in space and every surface triangulation point is influenced by certain subset of control points. The magnitudes of these influences, known as Sibson coordinates (Sibson, 1980), are computed before the optimization starts (Moccozet and Magnenat-Thalmann, 1997). The displacement of each surface triangulation point is the linear combination of the displacements of the control points that influence it. Let $P_1, \ldots, P_N \in R^3$ be the set of control points and Q be a subset influencing surface triangulation point p, $Q \subset P$, $Q = \{P_k\}$, where $k = 0, \ldots, N_c$. The elements of Q are the natural neighbors of p and their influence is expressed by the Sibson coordinates u_k. Let the control points from Q be displaced from their initial positions by ΔP_k, $k = 0, \ldots, N_c$. The new position of the surface triangulation point becomes:

$$p^{new} = p + \sum_{k=0}^{N_c} u_k \Delta P_k, P_k \in Q \qquad (1)$$

with $\sum_{k=0}^{N_c} u_k = 1$ and $u_k > 0$.

3.2 From Explicit to Implicit Surfaces

To create implicit shell of the explicit triangular mesh we convert each mesh' facets to a metaball primitive. Metaball primitives we consider here are:

1. spherical metaballs

2. triangular metaballs

Spherical metaball is an isosurface of the scalar field generated from the skeleton which is the center of the sphere circumscribed around the facet with its center on the facet as shown in Fig. 2(b). Triangular metaball is an isosurface of the scalar field generated from the skeleton which is now whole facet instead of the point as in the case of spherical

metaball. The parameters for the metaball include the position of the skeleton, and the potential filed function depending on the distance from the skeleton. This is depicted in Fig. 2(c).

Spherical metaballs are very simple and suitable for fairly regular meshes, while the triangular metaballs are more complex but suitable for completely generic meshes. Triangular metaballs provide much closer approximation of the explicit mesh what is discussed bellow.

3.2.1 Spherical Metaballs

First way of building the implicit shell whose shape approximates the initial explicit mesh is done as follows. We circumscribe a spherical metaball primitive around each facet of the surface triangulation in such a way that the sphere center lies on the facet, as it is shown in the Fig 2 (b). In fact spherical metaball is an isosurface of the scalar field depending on the distance from the field generating point, also called skeleton. For the spherical metaball skeleton is the center of the triangle and the distance is simple Euclidean distance of the points in space from the center of the sphere. We can express the radius and center of the spherical metaball in function of triangle vertices:

$$\vec{rC} = \frac{3\vec{rG} - \vec{rH}}{2} \qquad (2)$$

The center of the sphere circumscribed around the triangle depends on the triangle's center of the gravity $\vec{rG}$ and the orthocenter $\vec{rH}$. Finally, we have the distance from the center of the sphere to be the potential field function:

$$f(x, y, z) = exp(-k(\|\vec{r}\| - \|\vec{r_0}\|)) \qquad (3)$$

$$\|\vec{r}\| = \sqrt{(x - rC_x)^2 + (y - rC_y)^2 + (z - rC_z)^2} \qquad (4)$$

$$\|\vec{r_0}\| = \sqrt{(P_{1x} - rC_x)^2 + (P_{1y} - rC_y)^2 + (P_{1z} - rC_z)^2} \qquad (5)$$

where $|\vec{r}|$ is the distance of any 3D point from the sphere center and $|\vec{r_0}|$ is the radius of the spherical metaball. If we now consider the complete mesh consisting of a number of the triangles we get the definition of the implicit surface which approximates the mesh:

$$F(x, y, z) = T - \sum_{i=0}^{N} exp(-k(\|\vec{r}\| - |\vec{r_{0i}}|)$$

Where T is the threshold of the potential field meaning that changing this value we can obtain different isosurface of the potential field. Usually we set T to be equal one, so that

all the points on the surface have potential filed value equal zero, inside surface values smaller then zero and outside values greater then zero.

Ideally, to get a smooth implicit surface, the explicit mesh should have equally sized facets. In practice, the smaller the facets, the smaller the spheres circumscribed around them, and the closer the resulting implicit mesh approximates the initial explicit mesh. We have therefore found experimentally that subdividing the explicit mesh until all the facets are small enough is sufficient to produce visually pleasing results. Furthermore, as will be discussed bellow, the number of control parameters does not depend on the number of primitives and there is no significant computational penalty in so doing. Fig. 2(d, e) depicts the conversion of a simple triangulation patch into an implicit mesh.

3.2.2 Triangular Metaballs

Previous approach assumes that the explicit mesh has small, approximately equally sized facets so that the obtained implicit surface closely approximates explicit one. To overcome this limitation and to better approximate explicit mesh we use following approach. We create the implicit shells in such a way that instead of considering point, where the center of the sphere, is the skeleton around which the potential field is created, we use the whole triangle as the skeleton and compute potential filed around it in function of the distance from that primitive. In the Fig. 2(c) you can see metaball created around the triangle which we call triangular metaball.

The distance function is the Euclidean distance from the triangle expressed as function which defines distance either from the plane if the point projects on the triangle or the distance from the line or point if the point project outside the triangle. In this case it is necessary to consider seven different regions. According to which region projection of the considered 3D point $\vec{P}$ belongs to we can write following distance function:

$$\|\vec{r}\| = \begin{cases} \left|\frac{\vec{n}\,\vec{P} - D}{\|n\|}\right| & , & region 1 \\[2mm] \frac{\left\|(\vec{r} - \vec{P_2}) \times \vec{a}\right\|}{\|\vec{a}\|} & , & region 2 \\[2mm] \frac{\left\|(\vec{r} - \vec{P_1}) \times \vec{c}\right\|}{\|\vec{c}\|} & , & region 3 \\[2mm] \frac{\left\|(\vec{r} - \vec{P_2}) \times \vec{b}\right\|}{\|\vec{b}\|} & , & region 4 \\[2mm] \left\|\vec{P} - \vec{P_2}\right\| & , & region 5 \\[2mm] \left\|\vec{P} - \vec{P_3}\right\| & , & region 6 \\[2mm] \left\|\vec{P} - \vec{P_1}\right\| & , & region 7 \end{cases} \qquad (6)$$

Finally, distance function can be incorporated in the same potential field function as used for spherical metaballs:

$$F(x, y, z) = T - \sum_{i=0}^{N} exp(-k(\|\vec{r}\| - d_0)) \qquad (7)$$
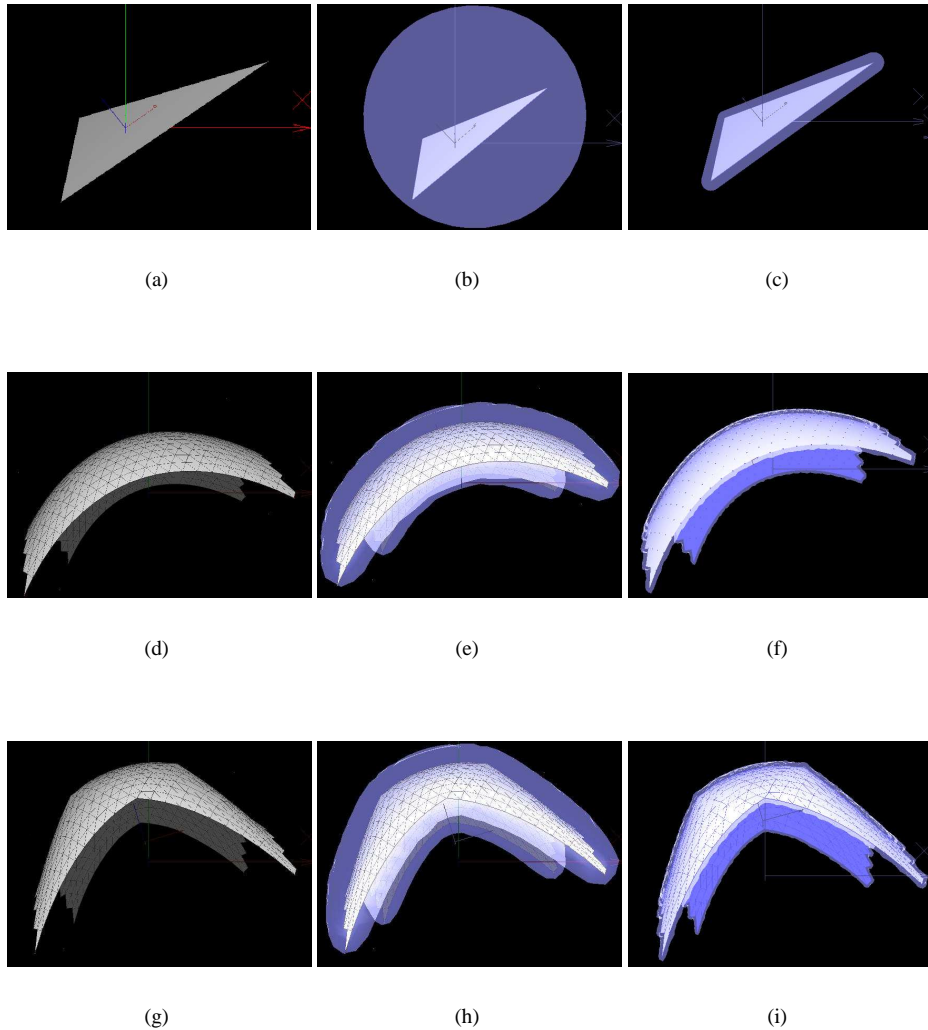
4

Figure 2: *Converting an explicit surface into an implicit shell. Upper row: Single triangle as the part of the explicit mesh (a), converted to the spherical metaball shown transparent (b) and to the triangular metaball (c). Middle row: Initial explicit surface (d), converted to the transparent implicit shell with the spherical metaballs (e) and converted to the implicit shell using triangular metaballs (f). Bottom row: Deformed explicit mesh (g), and corresponding deformed implicit shells shown transparent around the explicit one with spherical metaballs (h) and with the triangular metaballs (i).*

where d_0 is the fixed distance from the triangle, where potential field of the triangular metaball is zero. Having control over the parameter d_0 we can approximate the explicit mesh with the arbitrary thin implicit surface. This approach to conversion allow us to convert any mesh, for example one taken from the web, to the implicit shell and use it for automatic fitting or blending.

3.2.3 Implicit Shells Parametrization and Deformation
Our goal is to deform the implicit mesh in tandem with the explicit mesh. Explicit mesh is deformed using set of control points. In our case these are DFFD control points, but in general it can be any set of control points which control shape of the mesh, such as FFD or B-spline and NURBS patch control points. All this methods allow that mesh vertices, can be expressed as a function of the control points. Since both, spherical metaball and triangular metaball parameters depend only on the vertices of the

triangle it means that they depend also on control points. We can therefore write the field function F that defines the implicit shell as

$$F(\mathbf{x}, P_1, .., P_N) = T - \sum_{i=1}^{n} f(r(\mathbf{x}, P_1, P_2, .., P_N), r_i)$$

(8)

where $\mathbf{x}$ is a point in R^3, f is the exponential field functions discussed below, $r(x, P_1, P_2, ..., P_N)$ is the Euclidean distance to the primitive i, and r_i is the radius of primitive. For the spherical primitives radius of the primitive also depends on the control points, while for the triangular metaballs this is fixed value labeled as d_0 in Eq. 7.
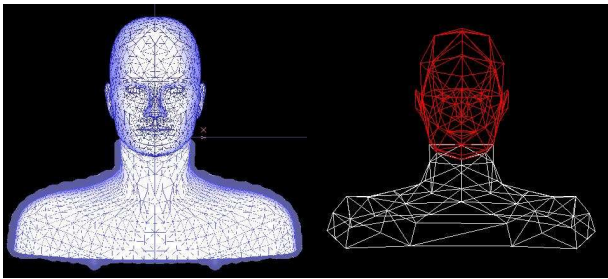
Figure 3: *Generic model of the upper body and corresponding control mesh. Left: Complete model surface triangulation, with rigid head as a mesh and deformable neck-shoulders converted to the implicit surface. Right: Complete control mesh*

4 OPTIMIZATION FRAMEWORK

Our goal is to deform the implicit mesh so that it conforms to the image data which is made of 3–D points derived from stereo and silhouette information. In standard least-squares fashion, for each data-point $\mathbf{x}_i$, we write an observation equation of the form

$$d(\mathbf{x}_i, S) = obs_i^{type} + \epsilon_i, 1 \le i \le nobs \qquad (9)$$

with weight w_i^{type}, where $type$ is one of the possible types of observations we use, S is a state vector that defines the surface shape, d is the distance from the point to the surface, and ϵ_i is the deviation from the model. In practice, we take $d(\mathbf{x}_i, S)$ to be the algebraic distance of $\mathbf{x}$ to the implicit surface defined by the field function of F of Eq.8 and we minimize the weighted sum of the squares of the deviations. To ensure that the minimization proceeds smoothly, the system automatically computes the w_i^{type} weights so that the different kinds of observations have commensurate influence(Ilic and Fua, 2002).

4.1 Parametrization and Regularization

In theory we could take the parameter vector S to be the vector of all x, y, and z coordinates of the surface triangulation. However, because the image data is very noisy, we would have to impose very strong regularization constraints. This is why we chose to use the DFFD approach to deforming the surface instead and introduce *control triangulations* such as the one dpicted by Fig. .3. Their vertices are points located at characteristic places on the model and serve as DFFD control points. This ability to place the control points at arbitrary locations is what sets DFFDs apart from all other kinds of FFDs. The control triangulation facets are used to introduce the regularization constraint discussed below. In our scheme, we take the state vector S to be the vector of 3-D displacements of DFFD control points(Ilic and Fua, 2002).

Because there are both noise and gaps in the image data, we still found it necessary to introduce a small regularization term. Since, we expect the deformation between the initial shape and the original one to be smooth, this can be done by preventing deformations at neighboring vertices of the control mesh to be too different. This is enforced by introducing a deformation energy E_S 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 write

$$E_S = \Delta_x^t K \Delta_x + \Delta_y^t K \Delta_y + \Delta_z^t K \Delta_z \qquad (10)$$

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:

$$E = \sum_{1 \le i \le nobs} w_i^{type} \epsilon_i^2 + \lambda_D E_D$$

where λ_D is a small positive constant.

4.2 Stereo and Silhouette Observations

In this work, we concentrate on combining stereo and silhouette data. Because the field-function F of Eq.8 is both well-defined and differentiable, the observations and their derivatives can be computed both simply and without search.

3–D Point Observations   Disparity maps are used to compute clouds of noisy 3–D points such as those of Fig.1. Each one is used to produce one observation of the kind described by Eq. 9. Minimizing the corresponding residuals tends to force the fitted surface to be as close as possible to these points. Because of the long range effect of the exponential field function in the error function F of Eq.8, the fitting succeeds even when the model is not very close to the data. Also, during least-squares optimization, an error measure that approaches zero instead of becoming even greater with growing distance has the effect of filtering outliers.

Silhouettes Observations   A silhouette point in the image defines a line of sight tangential to the surface. Let θ be an element of the state vector. For each value θ, we define the implicit surface:

$$S(\theta) = \left\{ \mathbf{x} \in R^3, F(\mathbf{x}, \theta) = 1.0 \right\} \qquad (11)$$

Let $\mathbf{x}(\theta)$ be the point on the line of sight where it is tangential to $S(\theta)$. By definition, it must satisfy the two constraints:

1. The point is on the surface, therefore $F(\mathbf{x}(\theta), \theta) = 1.0$.

2. The normal to $S(\theta)$ is perpendicular to the line of sight at $\mathbf{x}(\theta)$.

We integrate silhouette observations into our framework by performing, before each minimization, a search along the line of sight to find the point that has the lowest field value and, therefore, satisfies the second constraint. It is then used to add one of the observations described by Eq. 9to enforce the first constraint.
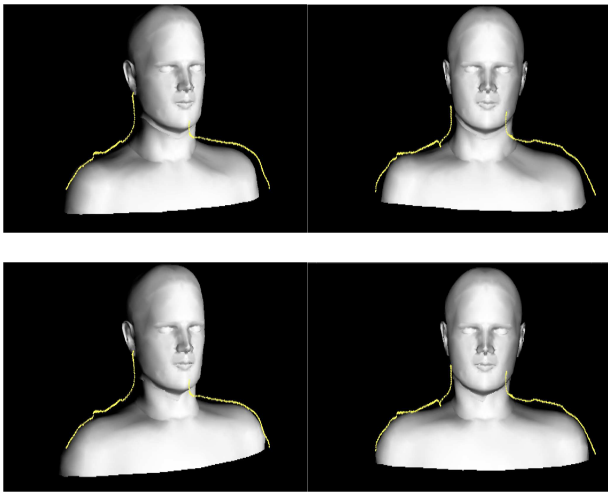
6

Figure 4: *Reconstructed shaded model of a person with overlaid silhouettes. Top row: Reconstructed model of the person using stereo alone viewed using the same perspective as that of the original images from Figure 1 and with overlaid silhouettes extracted from original images. Bottom row: Equivalent results using both stereo and silhouette data.*

5 RESULTS

We first demonstrate our technique on modeling people's neck and shoulders, and then animate them.

Neck and Shoulder Modeling   Here we recover person using real stereo and silhouette data reconstructed from an initially uncalibrated 6–frame video sequence in which the camera was moving around a static subject. In the left side colum of Fig. 1, we show the first, middle and last frames of the sequence. We used snakes to extract the silhouettes. In the absence of calibration information, we used a model-driven bundle-adjustment technique (Fua, 2000) to compute the relative motion and, thus, register the images. We then used a max-flow algorithm to derive from consecutive image pairs disparity maps, such as those shown in the middle colum of Fig. 1. Fig. 4 depicts reconstruction results of the neck and shoulders obtained either by using stereo alone or by using both stereo and silhouettes. In both cases, the head was reconstructed separately using our earlier DFFD-based method (Ilic and Fua, 2002). Notice that it is only when we combine both information sources that we get a model that projects correctly in all the views. This shows that its shape is geometrically correct even at places where the surface slants away from the cameras and, therefore, where stereo fails. Note that the texture-mapped views of Fig. 1 and the shaded views of Fig. 4 were generated by moving the initial explicit surface to match the deformed implicit shell, thereby underlining the importance to go back and forth from the explicit to the implicit representation.

Animation results   Finally, we show that automatically obtained results can be easily manually manipulated. Result of the optimization process is optimal position of the control points which define the shape of the object we model
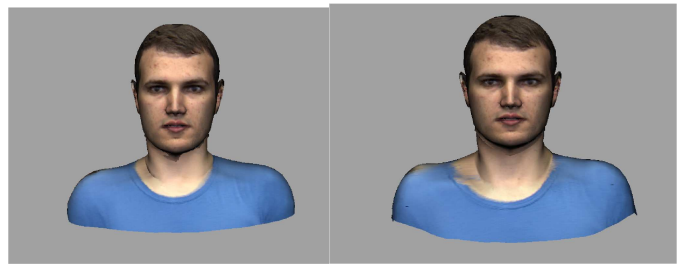

Figure 5: *Deformation results. Left image: Reconstructed model of the person in its initial position. Right image: The same person with the both sholders rised simultaniously.*

from images. These displacement of control point can be used to further deform the object. In the Fig. 5(left) we show the person after reconstruction and in the Fig. 5(right) person with both shoulders rised. This is obtained by simply selecting and moving some control points on the sholders.

6 CONCLUSION

We have presented an approach to switching from explicit surfaces to implicit that allows us to take advantage of the strengths of both kinds of approaches. To this end, we have proposed a technique for creating implicit shells in such a way that their shape depend only on the explicit surfaces' shape and that they are both parametrized in the same way. Particularly we choose Dirichlet Free Form Deformations for deforming the implicit and explicit models in tandem, but the way how implicit shells are created allows using of any other deformation method including direct mesh manipulation or indirect mesh manipulation using set of control points. This means that any FFD, or B-spline based approach of deforming meshes can be used to deform their implicit shells. However, some other indirect methods for explicit surface deformation, such as PCA parmetrization, can be used to deform our implicit shells.

We used the example of upper-body modeling using stereo and silhouette data to demonstrate the power of this approach. The explicit model we started from was not tailored for fitting purposes has man facets and a complex topology, neither of which has a significant impact on the quality of the fitting or the complexity of the computation.

Our next step will be to explore the use of this method for tracking upper body motion form monocular video sequences what should automatically produce animation parameters of the model. We expect this to result in a completely generic for modeling and animation from images.

REFERENCES

Bardinet, E., Cohen, L. and Ayache, N., 1998. A Parametric Deformable Model to Fit Unstructured 3D Data. Computer Vision and Image Understanding 71(1), pp. 39–54.

Barr, A. H., 1984. Local and global deformations of solid primitives. pp. 21–30.

Blanz, V. and Vetter, T., 1999. A Morphable Model for The Synthesis of 3–D Faces. In: Computer Graphics, SIGGRAPH Proceedings, Los Angeles, CA.

Blinn, J. F., 1982. A Generalization of Algebraic Surface Drawing. ACM Transactions on Graphics 1(3), pp. 235–256.

Bloomenthal, J. and Shoemake, K., 1991. Convolution Surfaces. SIGGRAPH 25(4), pp. 251–256.

Chang, Y. and Rockwood, A. P., 1994. A Generalized de Casteljau Approach to 3D Free-form Deformation. pp. 257–260.

Cohen, I., Cohen, L. D. and Ayache, N., 1991. Introducing new deformable surfaces to segment 3D images. In: Conference on Computer Vision and Pattern Recognition, pp. 738–739.

Coquillart, S., 1990. Extended Free-Form Deformation: A sculpturing Tool for 3D Geometric Modeling. 24(4), pp. 187–196.

Desbrun, M. and Gascuel, M., 1995. Animating Soft Substances with Implicit Surfaces. pp. 287–290.

Ferrie, F. P., Lagarde, J. and Whaite, P., 1992. Recovery of Volumetric Object Descriptions from Laser Rangefinder Images. In: European Conference on Computer Vision, Genoa, Italy.

Fua, P., 2000. Regularized Bundle-Adjustment to Model Heads from Image Sequences without Calibration Data. International Journal of Computer Vision 38(2), pp. 153–171.

Ilic, S. and Fua, P., 2002. Using Dirichlet Free Form Deformation to Fit Deformable Models to Noisy 3-D Data. In: European Conference on Computer Vision, Copenhagen, Denmark.

J. C. Carr, R. K. B., 2001. Reconstruction and Representation of 3D Objects with Radial Basis Functions. pp. 67–76.

J. C. Carr, W. R. F. and Beatson, R. K., 1997. Surface Interpolation with Radial Basis Functions for Medical Imaging. IEEE Transactions on Medical Imaging 16(1), pp. 96–107.

Kakadiaris, I. and Metaxas, D., 1996. Model based estimation of 3d human motion with occlusion based on active multi-viewpoint selection. In: Conference on Computer Vision and Pattern Recognition, San Francisco, CA.

Kalra, P., Mangili, A., Thalmann, N. M. and Thalmann, D., 1992. Simulation of Facial Muscle Actions Based on Rational Free Form Deformations. In: Eurographics.

Lowe, D. G., 1991. Fitting parameterized three-dimensional models to images. IEEE Transactions on Pattern Analysis and Machine Intelligence.

M. Eck, H. H., 1996. Automatic reconstruction of B-spline surfaces of arbitrary topological type. pp. 325–334.

Moccozet, L. and Magnenat-Thalmann, N., 1997. Dirichlet Free-Form Deformation and their Application to Hand Simulation.

Nishimura, H. and al., 1985.

Pentland, A. and Sclaroff, S., 1991. Closed-form solutions for physically based shape modeling and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 13, pp. 715–729.

Plänkers, R. and Fua, P., 2001a. Articulated Soft Objects for Video-based Body Modeling. In: International Conference on Computer Vision, Vancouver, Canada, pp. 394–401.

Plänkers, R. and Fua, P., 2001b. Tracking and Modeling People in Video Sequences. Computer Vision and Image Understanding 81, pp. 285–302.

Sederberg, T. and Parry, S., 1986. Free-Form Deformation of Solid Geometric Models.

Sibson, R., 1980. A vector identity for the Dirichlet Tessellation. In: Math. Proc. Cambridge Philos. Soc., pp. 151–155.

Stokely, E. M. and Wu, S. Y., 1992. Surface parameterization and curvature measurement of arbitrary 3-d objects: five practical methods. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(8), pp. 833–839.

Sullivan, S., Sandford, L. and Ponce, J., 1994. Using geometric distance fits for 3–d. object modeling and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(12), pp. 1183–1196.

Szeliski, R. and Tonnesen, D., 1992. Surface Modeling with Oriented Particle Systems. In: Computer Graphics, SIGGRAPH Proceedings, Vol. 26, pp. 185–194.

Terzopoulos, D. and Vasilescu, M., 1991. Sampling and reconstruction with adaptive meshes. In: Conference on Computer Vision and Pattern Recognition, pp. 70–75.

Turk, G. and O'Brien, J. F., 1999. Shape transformation using variational implicit surfaces. pp. 335–342.

Wyvill, B. and van Overveld, K., 1997. Warping as a modelling tool for csg/implicit models. In: Shape Modelling Conference, University of Aizu, Japan, IEEE Society Computer Press ISBN0-8186-7867-4, pp. 205–214. inivited.

Wyvill, G. and Wyvill, B., n.d. Data Structure for Soft Objects.