MODEL-BASED SHAPE FROM SILHOUETTE
A Solution Involving a Small Number of Views

J. F. Menudet, J. M. Becker, T. Fournel and C. Mennessier
Laboratoire Hubert Curien, UMR CNRS 5516 - Université Jean Monnet, 18 rue Pr Benoît Lauras, Saint-Etienne, France

Keywords: Visual Hull, Shape From Silhouette, Contour, Radial Basis Function, Deformable Model, Non-Rigid Registration.

Abstract: This article presents a model-based approach to Shape From Silhouette reconstruction. It is formulated as a problem of 3D-2D non-rigid registration: a surface model is deformed until it correctly matches the detected silhouettes in the images. An efficient and reliable solution is proposed, based on a Radial Basis Function deformation driven by control points located on the contour generators of the 3D model. Unlike previous methods relying on non-linear optimization techniques, the proposed method only requires a linear system solving. Another advantage of this model-based approach is to produce a surface representation of the visual hull. Moreover, the introduction of shape priors allows to reduce in a dramatic way the number of views required to obtain a realistic reconstruction. Application to human body modeling is given.

1 INTRODUCTION

Image-based modeling aims at the acquisition of a 3D model from multiple images of a real world object. It is a simple and low cost approach compared to other methods such as laser scanner or structured-light projection. Some examples of image-based modeling techniques are Structure From Motion (SFM) (Hartley and Zisserman, 2004) and Shape From Shading (Prados and Faugeras, 2005). Another popular technique is Shape From Silhouette (SFS) which produces the *visual hull* of the object, an outer approximation of the 3D shape, consistent with a set of silhouettes detected in several images (Laurentini, 1994). Apparent contours give strong clues about the 3D shape of an object. Furthermore, they are far more easier to find than point correspondences required for example in SFM. Nevertheless, the accuracy of the visual hull is directly related to the number of input images as well as the different locations of the camera. Another well-known intrinsic limitation of SFS is the impossibility to reconstruct concavities.

Typically, a good SFS reconstruction requires 15 to 30 views of the object (depending on its complexity), with apparent contours of the object detected in each view. Automatic detection is possible in some conditions (almost uniform background) but in most cases, a human supervision is necessary. Hence, the number of views is a critical issue, strongly influencing the overall processing time. An efficient and accurate SFS method with a small number of views is therefore highly desirable.

When less data is available, the inherent loss of accuracy can be compensated by a priori knowledge as accounted for in inverse problems literature. If the object to reconstruct is not totally unknown, SFS could also benefit from this approach. The principle of a deformable model is precisely well suited to incorporate shape prior. For example, the human body topology is always the same and its shape is approximately known a priori. A generic model is therefore easy to construct. The main problem is the expression of the model deformation to match the data, i.e. the apparent contours in the context of SFS.

The method presented in this article is space deformation driven by silhouettes' fitting: the projection of the model after this deformation should coincide exactly with the silhouettes. Our main contribution is a linear solution to the underlying problem of 3D/2D non-rigid registration, unlike prior works

F. Menudet J., M. Becker J., Fournel T. and Mennessier C. (2007).
MODEL-BASED SHAPE FROM SILHOUETTE - A Solution Involving a Small Number of Views.
In *Proceedings of the Second International Conference on Computer Vision Theory and Applications - IU/MTSV*, pages 379-386
Copyright © SciTePress

based on non-linear optimization framework. It is therefore an efficient and reliable (no local minima) technique. This is achieved by the displacements of some control points located on the contour generators of the surface model. Radial Basis Function (RBF) is then used to construct a space deformation interpolating these displacements. A special case of RBF deformation is the Thin Plate Spline warping presented in (Bookstein, 1989). In section 2, a review of related works is presented. Then, section 3 details the different steps of the proposed method: computation of the control points, estimation of their displacements and construction of the space deformation. Finally, some results are shown in section 4. We illustrate our approach throughout the paper with human body modeling, believed to be a relevant and important application of our method.

2 RELATED WORK

A large literature covers the problem of the representation and construction of the visual hull. Two categories of methods can be distinguished: surface-based and volume-based approaches. The former is based on the intersection of the visual cones generated by the apparent contours. Intuitive at first, it is actually difficult to implement for complex objects but some efficient algorithms exist for polyhedral reconstruction (Franco and Boyer, 2003). Because of the complexity of surface-based approach, volumetric reconstruction is widely used (Szeliski, 1993). The space of interest is first discretized by a voxel representation. Then, each voxel is projected on the different image planes and voxels falling completely outside of any silhouette are removed. This simple approach has nevertheless some drawbacks. First, a surface representation is often preferred (for rendering purpose for example). Secondly, voxel representation is only an approximation of the exact visual hull due to quantization error. Decreasing the voxel size obviously reduces the problem at the expense of rapidly growing computational and storage costs (octree hierarchies are therefore often used).

Model-based reconstruction of objects has recently received great attention. It has been used to incorporate complementary data (mainly silhouette and texture). In (Isidro and Sclaroff, 2003), a surface mesh is iteratively deformed in order to obtain a photometrically-consistent solution while enforcing silhouette constraint. A similar snake-like technique is described in (Hernandez and Schmitt, 2004): the texture driven force deforming the model is here computed with a multi-stereo correlation voting approach and a gradient vector flow diffusion (Xu and Prince, 1998). A silhouette force is added to constraint the model. A method combining explicit (mesh) and implicit (metaball) surface has been proposed in (Ilic and Fua, 2006): the model is fitted to silhouette and stereo data with a Dirichlet Free Form Deformation incorporated in an optimization scheme. Another solution (Sullivan and Ponce, 1998) in the case of pure SFS (when only silhouette data is available) is based on an initial spline surface deformed to match the silhouette. Again, it relies on an optimization framework (gradient-descent) subject to local minima and long computation time. In the field of human body modeling, the most closely related work to ours is (Hilton et al., 2000) which deforms a generic human model in order that its silhouettes approximate the given ones. Is is a fast technique but the model deformation is restricted to affine transformation. Another different approach is statistical shape model as used in (Fleute and Lavallée, 1999) but the training phase (inherent to this kind of model) is by itself a problem.

3 MODEL-BASED SHAPE FROM SILHOUETTE

3.1 Overview of the Proposed Method

The main idea is to compute some control points on the contour generators of the 3D model and move them with the objective that their projection match the silhouettes. RBF finally provides a smooth deformation of the 3D space (hence of the embedding model), based on the displacement of these control points. Recall that for a given camera position, a contour generator (or rim) of a surface $\mathcal{S}$ is the curve on $\mathcal{S}$ which projects to the apparent contour in the image (Rosten and Drummond, 2003). Moreover, surface normals along rim curve are orthogonal to the viewing direction (see figure 1).

Let us now detail the different steps of the proposed method. Apparent contours are assumed detected in calibrated images (Zhang, 2000), i.e. intrinsic and extrinsic parameters of the camera are known. Vectors and matrices are respectively typeset using bold and sans-serif fonts, e.g. vector $\mathbf{P}$ and matrix K. Vectors are always column vectors such as $\mathbf{P} = [X\,Y\,Z]^{\mathrm{T}}$. The cross-product of two 3D vectors is denoted by $\times$ and the symbol $\sim$ means equality up to a non zero scale factor.

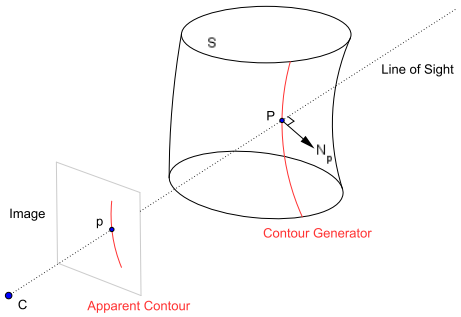

Figure 1: Contour generator (or rim) on a surface is the locus of points where surface normal is orthogonal to the line of sight.

3.2 Initial Registration

The first step is a registration of the 3D model in the world coordinate system. It may be based on few landmarks (at least 3) located on the model and identified in various images. 3D reconstruction of the image points is possible thanks to the known camera parameters. Finally, a transformation between the model landmarks and the reconstructed points can be computed and applied to the model. With 3 landmarks, only a similarity (rotation, translation plus scaling) can be estimated. When more landmarks are available, affine or projective transformation is a valid alternative, bringing a first deformation of the model in addition to the registration of the coordinate systems.

3.3 Computation of Control Points on the 3d Model

We want to place some control points on the rims of the 3D model. The first step is the computation of the rims, a classical problem occurring for example in non-photorealistic rendering (Hertzmann and Zorin, 2000). Different techniques have been proposed, depending on the way the surface is represented. Implicit surfaces are probably the most convenient representation for rim computation: an efficient algorithm based on the solution of an ordinary differential equation is available (Rosten and Drummond, 2003). Nevertheless, implicit surfaces are difficult to render and their deformation is not intuitive. Surfaces represented by triangular meshes are therefore preferred for their versatility and fast rendering ability.

We have adopted the solution presented in (Hertzmann, 1999), well suited for triangular meshes representing smooth surfaces (via subdivision technique for example). Let $\mathbf{N_v}$ be the surface normal at a vertex $\mathbf{V}$ of the mesh. Let $\mathbf{C}$ be the position of the camera center. As mentioned above (see figure 1), rim curve is the locus of surface points $\mathbf{P}$ whose normal is perpendicular to the line of sight:

$$\mathbf{N_p}.(\mathbf{P}-\mathbf{C}) = 0. \quad (1)$$

For each vertex of the mesh, it is possible to compute the normalized dot product between surface normal and viewing direction:

$$d_v = \frac{\mathbf{N_v}.(\mathbf{V}-\mathbf{C})}{\|\mathbf{N_v}\|\,\|\mathbf{V}-\mathbf{C}\|}. \quad (2)$$

The surface is assumed continuous, hence if two vertices $\mathbf{V_1}$ and $\mathbf{V_2}$ forming an edge of the mesh have different signs for values d_v, then there must be a rim point on this edge - that is, a zero crossing of the dot product function (see figure 2). The location of this zero crossing along the edge can be estimated with a linear interpolation between the 2 vertices using values d_{V_1} and d_{V_2}. The rim curve is then constructed by connecting the rim points located on edges sharing a common vertex.

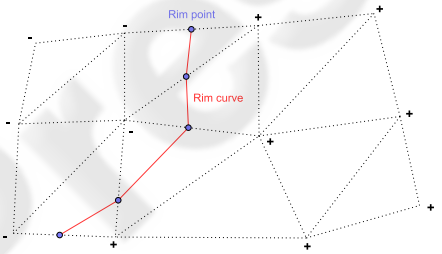

Figure 2: The sign of the dot product between surface normal and viewing direction is computed at each vertex of the mesh. A change of signs between two adjacent points allows to identify an edge with a rim point (blue circle) which can be connected to create the rim curve (red line).

This algorithm is applied to the different locations of the camera. The control points are taken as the visible rim points previously obtained. Note that only the visibility of these points has to be checked. It is significantly simpler than the removal of hidden portions of curves, as it would be required in a computer graphics application. The result of this algorithm clearly depends on the resolution of the mesh: the smaller the triangles, the better the rim estimation and the higher the number of control points. But in return, the number of triangles also influences the computation time. A multi-resolution approach would be clearly an efficient solution. Subdivision surface precisely allows to refine locally the mesh around the initial rim curve. In this way, only edges of the newly created triangles are tested, speeding up the computation . This process is repeated until the desired resolution is reached. As will be seen in the sequel (see

section 3.5), a large number of control points (more than a few thousand) is actually undesirable.

3.4 Estimation of Control Points Displacements

The present issue is to find for every previously computed control point $\mathbf{P}$, the displacement toward the corresponding point $\mathbf{Q}$. The projection of $\mathbf{Q}$ must be located on the apparent contours, but an infinite number of points satisfy this constraint. Two observations will guide us to compute $\mathbf{Q}$:

- Obviously, $\mathbf{Q}$ is located on the optical rays passing through silhouette pixels.
- To ensure a coherent deformation of the model, control points $\mathbf{P}$ should be moved along the surface normal $\mathbf{N_P}$. This is a common assumption (Hernandez and Schmitt, 2004) which has the advantage to preserve the initial quality of the mesh (especially an even repartition of its vertices).

Therefore, we propose the following idea to estimate $\mathbf{Q}$ given a rim point $\mathbf{P}$ on the 3D model: compute λ such that the point $\mathbf{Q} = \mathbf{P} + \lambda \mathbf{N_p}$ projects to the apparent contour (see figure 3).

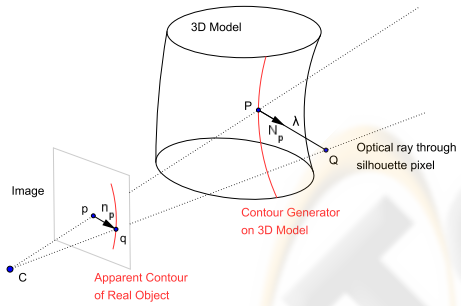

Figure 3: Estimation of $\mathbf{Q}$, the rim point on the real object corresponding with the rim point $\mathbf{P}$ of the 3D model.

Because the camera is calibrated, coordinates $[X\,Y\,Z]^{\mathrm{T}}$ of $\mathbf{P}$ are known in the 3D camera frame. For the same reason, homogeneous coordinates of its projection $\mathbf{p}$ can also be determined from the intrinsic parameters matrix K (Zhang, 2000):

$$\mathbf{p} = [u\,v\,1]^{\mathrm{T}} \sim \mathsf{K}\mathbf{P}. \tag{3}$$

λ can be computed in two steps:

1. Starting from the point $\mathbf{p}$ in image, find the point $\mathbf{q} = [u'\,v'\,1]^{\mathrm{T}}$ on apparent contour situated in the direction of $\mathbf{n_p}$ (the projection of $\mathbf{N_p}$).
2. Solve for λ the equation

$$\mathsf{K}(\mathbf{P} + \lambda \mathbf{N_p}) \sim \mathbf{q} \tag{4}$$

which can be rewritten

$$\mathsf{K}(\mathbf{P} + \lambda \mathbf{N_p}) \times \mathbf{q} \sim \mathbf{p} \times \mathbf{q} + \lambda \mathbf{n_p} \times \mathbf{q} = \mathbf{0}, \tag{5}$$

giving 3 scalar equations for finding λ. One of them may be preferred to avoid the smallest components of $\mathbf{n_p} \times \mathbf{q}$.

3.5 Space Deformation

3.5.1 Problem Statement

The previous steps define only the displacements $\mathbf{P_i} \rightarrow \mathbf{Q_i}$ of N control points whereas a deformation of space is required: an underlying $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ mapping

$$S(\mathbf{P}) = [s_x(\mathbf{P})\, s_y(\mathbf{P})\, s_z(\mathbf{P})]^{\mathrm{T}} \tag{6}$$

has to be estimated. It would allow to interpolate the displacement of any 3D point, especially those of the 3D model. The problem is to find the three $\mathbb{R}^3 \rightarrow \mathbb{R}$ interpolant functions (s_x, s_y, s_z) such that

$$\begin{aligned} s_x(\mathbf{P_i}) &= Q_i^x \\ s_y(\mathbf{P_i}) &= Q_i^y \qquad \text{for } i = 1,\ldots,N \\ s_z(\mathbf{P_i}) &= Q_i^z \end{aligned} \tag{7}$$

with $\mathbf{Q_i} = \left[Q_i^x\, Q_i^y\, Q_i^z\right]^{\mathrm{T}}$. There are obviously an infinite number of functions verifying the interpolation conditions (7). The problem is to find the "best one".

3.5.2 Radial Basis Function Interpolation

Radial Basis Function (RBF) is a popular technique to solve this problem of scattered data interpolation (Carr et al., 2003). In general, suppose that the values f_i of a $\mathbb{R}^d \rightarrow \mathbb{R}$ function f are known at N arbitrary points $\mathbf{c_i} \in \mathbb{R}^d$: the problem is to find an interpolant function $s(\mathbf{x})$ such that $s(\mathbf{c_i}) = f_i = f(\mathbf{c_i})$ for $i = 1,\ldots,N$. A RBF is a function s of the form

$$s(\mathbf{x}) = p(\mathbf{x}) + \sum_{i=0}^{N} w_i \phi(\|\mathbf{x} - \mathbf{c_i}\|) \tag{8}$$

where $p(\mathbf{x})$ is a polynomial with a low degree m and $\phi(r)$ is the basis function defined for $r \in [0, +\infty[$. The w_i's and $\mathbf{c_i}$'s are respectively the *weights* and the *centers* of the RBF. Some common RBFs are:

- gaussian: $\phi(r) = e^{-cr^2}$ (and optional polynomial)
- multiquadric: $\phi(r) = \sqrt{r^2 + c^2}$ with $p(\mathbf{x}) = a_1$
- thin-plate spline (d=2): $\phi(r) = r^2 \ln(r)$ with $p(\mathbf{x}) = a_1 + a_2 x + a_3 y$
- biharmonic spline (d=3): $\phi(r) = r$ with $p(\mathbf{x}) = a_1 + a_2 x + a_3 y + a_4 z$
- triharmonic spline (d=3): $\phi(r) = r^3$ with $p(\mathbf{x})$ a trivariate quadratic polynomial

Depending on the choice of ϕ and p, the resulting function s will have different properties. For example, splines give smooth interpolant functions because they minimize energy in second (thin-plate, biharmonic) or third (triharmonic) derivative of s (Bookstein, 1989; Carr et al., 2003). Note also that a closely related technique to RBF is kriging with spline interpolation as a special case (Trochu, 1993). Mainly used in geostatistics, its goal is to find the best linear unbiased estimator of a random function (best means with minimal variance). Interestingly, in this framework, the basis functions ϕ are interpreted as a spatial correlation between the data which yields some insights on their choice.

The interpolation problem is reduced now to the computation of the weights $\mathbf{w} = [w_1 \dots w_N]^{\mathrm{T}}$ and coefficients $\mathbf{a} = [a_1 \dots a_k]^{\mathrm{T}}$ of

$$p(\mathbf{x}) = a_1 p_1(\mathbf{x}) + \dots + a_k p_k(\mathbf{x}) \tag{9}$$

where $\{p_1, \dots, p_k\}$ is a basis for polynomials of degree at most m (for example $\{1, x, y\}$ in the case of the 2D thin-plate spline). Let $\mathbf{f}$ be the vector $[f_1 \dots f_N]^{\mathrm{T}}$, Φ the $N \times N$ matrix with $\Phi_{\mathrm{ij}} = \phi(\|\mathbf{c_i} - \mathbf{c_j}\|)$ and P the $N \times k$ matrix with $\mathsf{P}_{\mathrm{ij}} = p_j(\mathbf{c_i})$. The interpolation conditions can then be rewritten in matrix form:

$$\begin{bmatrix} \Phi & \mathsf{P} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{a} \end{bmatrix} = \mathbf{f}. \tag{10}$$

This system is underdetermined ($N + k$ unknowns and N equations). Therefore, the so-called "side conditions" are imposed on $\mathbf{w}$ to ensure that the transformation has square integrable second derivatives:

$$\mathsf{P}^{\mathrm{T}} \mathbf{w} = \mathbf{0}. \tag{11}$$

(10) and (11) can be gathered into the following symmetric linear system:

$$\begin{bmatrix} \Phi & \mathsf{P} \\ \mathsf{P}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}. \tag{12}$$

The weights $\mathbf{w}$ and polynomial coefficients $\mathbf{a}$ are the solution of (12) and completely define the interpolant function $s(\mathbf{x})$. The different choices of $\phi(r)$ and $p(\mathbf{x})$ presented above guarantee an invertible system under very mild conditions on the centers' locations (not aligned for thin-plate spline while gaussian and multiquadric place no restrictions). Nevertheless, using a large number of data points can eventually give rise to a ill-conditioned matrix in equation (12). Moreover, basis functions with non-compact support (e.g. polyharmonic splines) yield a dense system whose solution requires $O(N^3)$ operations.

3.5.3 Radial Basis Function Deformation

As said before, we are looking for 3 functions (s_x, s_y, s_z) describing a space deformation. These 3 RBFs share the same centers (namely the N control points $\mathbf{P_i}$), hence the matrix of system (12) is identical for the 3 functions which saves computation time. Because a smooth transformation is desired, biharmonic or triharmonic splines seem appropriate. The latter have been used in free-form modeling because of their predictable and intuitive behavior (Botsch and Kobbelt, 2005). We adopt it for the same reason, using generally less than 1000 control points.

An interesting variant of the interpolation problem comes from the introduction of a regularization term (i.e. smoothing), yielding approximation rather than interpolation. It may be a better choice when the reliability of the control points is low. The solution to this approximation scheme is very similar to the interpolation one: the only modification is the addition of a parameter λ in the diagonal of the matrix Φ which is therefore replaced by $\Phi + \lambda \mathrm{I}$. The parameter λ controls the smoothness of the transformation: with $\lambda = 0$, we obtain the interpolant transformation whereas higher values of λ bring smoother transformations, until the weights $\mathbf{w}$ vanish when $\lambda \to \infty$ (simple polynomial transform, e.g. affine).

3.5.4 Progressive Deformation

The proposed deformation scheme relies on the fundamental assumption that the deformed rims coincide with the rims of the transformed model. This assumption looks very reasonable because the deformation is smooth and normal to the surface. However, it is not strictly true, especially when a large deformation is applied. In other words, the rims computed after the model deformation may not be projected exactly on the apparent contours in images. This drawback occurs especially when the initial model is far from the real object. Moreover, in this case the computed rim points are usually misplaced on the initial surface, i.e. they do not correspond with the apparent contours of the real object. It can generate some artifacts in the space deformation.

We overcome both problems with a progressive and regularized space deformation. It is achieved with successive RBF deformations steered by a decreasing smoothing parameter λ. After each deformation, control points and their displacements are recomputed. Therefore, their reliability increases as the surface is getting closer to the target shape, justifying the progressive decrease of λ. At the last step, λ is set to 0 to obtain exact interpolation. In our current implementation, the number of steps (namely 6) and the successive values of λ are fixed a priori; affine transformation ($\lambda \to \infty$) is forced during the first 2 iterations, then $\lambda = (1, 0.1, 0.01, 0)$ for the last 4 steps. Note that these values are related to the overall size of

the model. Here, the control points are scaled to fit in a unit cube before forming the RBF system.

4 APPLICATION TO HUMAN BODY MODELING

4.1 Experimental Setup

We demonstrate an application of our method with the modeling of a mannequin female torso based on only 4 images (see figure 4). These images (1600×1200 pixels) were acquired with a digital still camera. The photographs are only taken from the right side and front side of the dummy. The apparent contours in the different views are found in a semi-automatic way: an initial contour is hand drawn, then deformed using a snake method (Xu and Prince, 1998). All images are calibrated using a plane-based calibration method (Zhang, 2000) (calibration grid not shown in images).

4.2 Initialization

We use a generic model of female torso, represented by a mesh of 9000 triangles. The initial affine registration of this model (see section 3.2) is computed by pointing at 4 anatomical landmarks in the images (seen as green dots in figure 4). Several views of the 3D model after this first registration step are presented in the figure 5. The projection in the images of the model vertices is shown in figure 4. At this step, volume, position and orientation of the model are rather good but the matching of silhouettes is not respected. Especially, breast, waist and spinal shapes are not coherent with the images.

4.3 Results

The rim curves are computed on the initial surface model (black curves in figure 5), according to the calibration parameters. The same figure shows the displacements of 344 control points estimated from the apparent contours (green segments). The control points are well distributed all around the shape though the photographs are taken from restricted viewpoints. The displacement of the control points can be significant in some areas (up to 4 cm).

The progressive deformation described previously is finally applied. The resulting surface model after space deformation is presented in figure 6 and its projection in the 4 images is shown in figure 7. Note how the model perfectly matches the silhouettes (especially the shape of the waist). Nevertheless, the projection of the final model in two images not used in the deformation reveals some differences (especially around the right shoulder blade) between the reconstructed and the actual shape (see figure 8). Indeed, deformable model method is obviously not a complete substitute for image data that could give, for example, the shape of the shoulder blade. Accuracy has been manually assessed by measuring some circum-

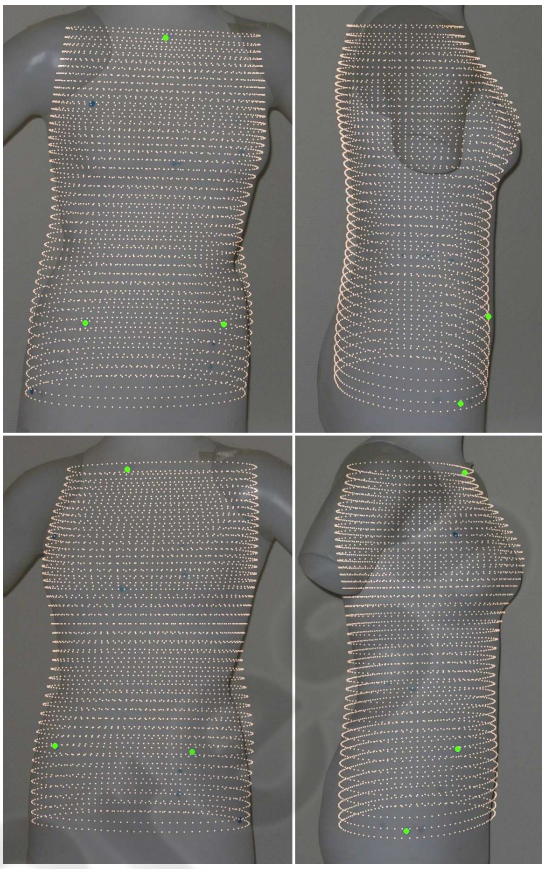

Figure 4: The 4 images of a dummy torso with the anatomical landmarks used to compute the initial registration (green dots). Projection of the initial model is superimposed.

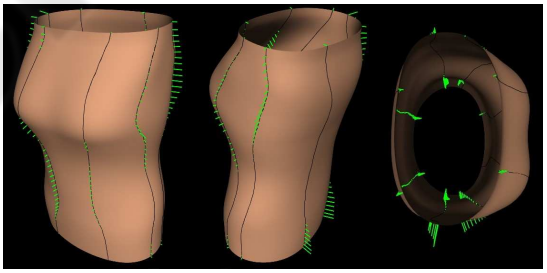

Figure 5: Three different views of the initial 3D model. The computed rim curves are shown in black. The green segments represent the displacements of control points used to computed the RBF deformation.

ferences on both real object and 3D model. A 3% relative mean error of has been found, probably due to camera parameters and manual measurements uncertainties.

The computation time is only 3 seconds on a computer equipped with a P4 3GHz processor and 1024 MB of RAM. This time is based on a basic C++ implementation, but there is a lot of place for improvement of speed. The most time consuming tasks are rims computation and RBF system solving. In both cases, more efficient algorithms are available (Hertzmann and Zorin, 2000; Botsch and Kobbelt, 2005).

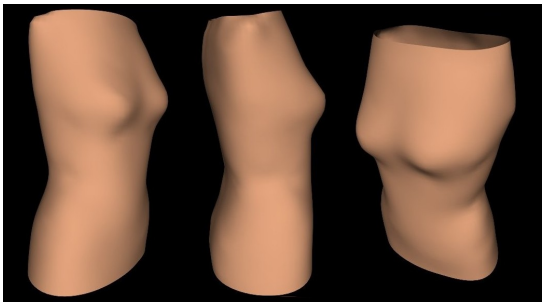

Figure 6: Different views of the final 3D model, i.e. after the space deformation.

4.4 Influence of the Initial Model

In another experiment, the generic female torso model has been replaced by a cylinder. The goal is to evaluate the influence of the initial surface. The 3D shape obtained after deformation is shown in figure 9. The final result is globally similar to the previous surface and its projection in images still matches the apparent contours. The main problem is the shape of the chest which is not satisfying, especially the loss of the concave part between the breasts. However, this is not surprising since it is a well-known limitation of SFS. The resulting shape is however rather acceptable, given the small number of images. It confirms that our model-based SFS is able to bring a significant improvement of quality when the model is well chosen.

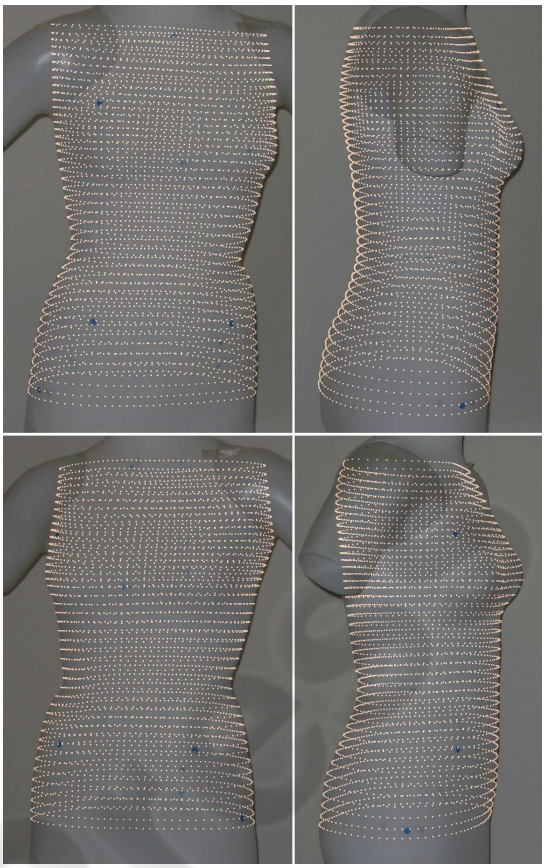

Figure 7: Projection in the images of the final 3D model.

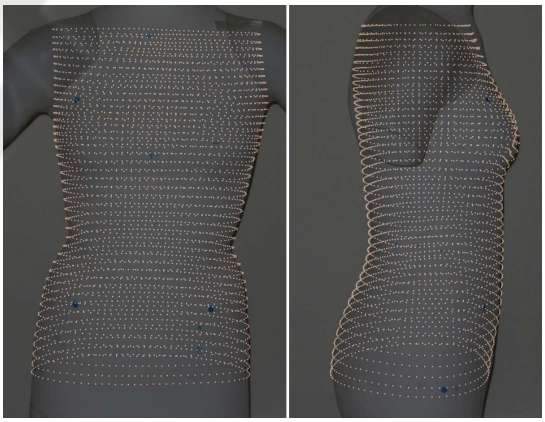

Figure 8: Projection of the final 3D model in two additional images, not used during the deformation.

5 CONCLUSION

A model-based Shape From Silhouette technique has been presented. The result is a surface representation of the visual hull, more convenient than volumetric representation. Moreover, the introduction of a shape prior allows to reduce the number of input images while keeping a good reconstruction quality. Our main contribution is a reliable and efficient solution to the underlying problem of 3D/2D non-rigid registration between the 3D model and the 2D silhouettes. The core idea is that the apparent contours give sufficient strong clues to construct a progressive deformation of space based on Radial Basis Function.

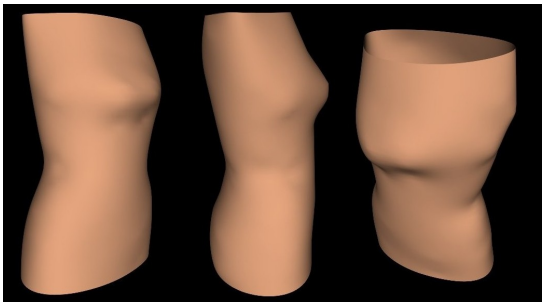

Figure 9: The 3D shape obtained from the deformation of a cylinder.

Unlike previous works, the method does not use non-linear optimization strategies: a simple linear system has to be solved. Efficiency is closely connected with the size of this system, i.e. the number of control points used to drive the deformation. Our method is clearly intended for smooth objects, hence a few hundred of control points are sufficient. Another factor influencing the computation time is the number of images but the method is designed to keep it small. The main limitations are the availability of a generic 3D model of the object and the requirement of some landmarks for the initial registration. A point to improve is the decreasing scheme of the smoothing parameter used in the deformation. We plan to find a dynamic adjustment of this sequence, based on an analysis of the displacements of control points.

Human body is a good subject for our method. We have demonstrated the modeling of a human torso with as few as 4 images. A classical SFS technique would need at least 20 images to produce such a realistic 3D model. The next step is the modeling of the whole body. The work presented in (Hilton et al., 2000) may, for example, be adapted to fit our approach. A lot of applications in medical, garment or virtual reality fields would be possible.

REFERENCES

Bookstein, F. (1989). Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585.

Botsch, M. and Kobbelt, L. (2005). Real-time shape editing using radial basis functions. In *Eurographics05*.

Carr, J., Beatson, R., McCallum, B., Fright, W., McLennan, T., and Mitchell, T. (2003). Smooth surface reconstruction from noisy range data. In *International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia (Graphite 2003)*.

Fleute, M. and Lavallée, S. (1999). Nonrigid 3-d/2-d registration of images using statistical models. In *MICCAI*, pages 138–147.

Franco, J. and Boyer, E. (2003). Exact polyhedral visual hulls. In *British Machine Vision Conference*.

Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry In Computer Vision*. Cambridge University Press, second edition.

Hernandez, C. and Schmitt, F. (2004). Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3):367–392.

Hertzmann, A. (1999). Introduction to 3d non-photorealistic rendering: Silhouettes and outlines. In *Non-Photorealistic Rendering, ACM SIGGRAPH Course Notes*.

Hertzmann, A. and Zorin, D. (2000). Illustrating smooth surfaces. In *ACM SIGGRAPH 2000*.

Hilton, A., Beresford, D., Gentils, T., Smith, R., Sun, W., and Illingworth, J. (2000). Whole-body modelling of people from multi-view images to populate virtual worlds. *Visual Computer: International Journal of Computer Graphics*, 16:411–436.

Ilic, S. and Fua, P. (2006). Implicit meshes for surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):328–333.

Isidro, J. and Sclaroff, S. (2003). Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints. In *IEEE International Conference on Computer Vision, 2003.*, volume 2, pages 1335–1342.

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

Prados, E. and Faugeras, O. (2005). Shape from shading: a well-posed problem? In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 870–877.

Rosten, E. and Drummond, T. (2003). Rapid rendering of apparent contours of implicit surfaces for real-time tracking. In *British Machine Vision Conference*.

Sullivan, S. and Ponce, J. (1998). Automatic model construction and pose estimation from photographs using triangular splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1091–1097.

Szeliski, R. (1993). Rapid octree construction from image sequences. *CVGIP: Image Understanding*, 58(1):23–32.

Trochu, F. (1993). A contouring program based on dual kriging interpolation. *Engineering with Computers*, 9:160–177.

Xu, C. and Prince, J. L. (1998). Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7:359–369.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334.