Fast and Controllable 3D Modelling from Silhouettes

Mukta Prasad, Andrew Zisserman and Andrew W. Fitzgibbon[2]

Robotics Research Group, University of Oxford [2]Microsoft Research, Cambridge

Abstract

We show how a 3D model of a complex curved object can be easily extracted from a single 2D image. A user-defined silhouette is the key input; and we show that finding the smoothest 3D surface which projects exactly to this silhouette can be expressed as a quadratic optimization, a result which has not previously appeared in the large literature on the shape-from-silhouette problem. For simple models, this process can immediately yield a usable 3D model; but for more complex geometries the user will wish to further shape the surface. We show that a variety of editing operations—which can be defined either in the image or in 3D—can also be expressed as linear constraints on the 3D shape parameters. We extend the system to fit higher genus surfaces. Our method has several advantages over the system of Zhang et al. *[ZDPSS01] and over systems such as* SKETCH *and* Teddy.

Categories and Subject Descriptors (according to ACM CCS): I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—modeling and recovery of physical attributes.

1. Introduction

The relentless increase in demand for 3D content has inspired researchers to devise techniques which allow models to be acquired directly from the real world. Computer vision techniques permit automatic reconstruction from multiple photos in some cases, but since *Façade* [DTM96], the value of user input in extracting models from images has been clear. We are interested in single-view reconstruction: given a single photo of a curved object, recover a textured 3D model with the minimum of user input in simple cases, and enough controllability to make good models of complex objects.

In particular, this paper is a development of two classes of 3D modellers: the $2\frac{1}{2}$D reconstruction from images of Zhang *et al.* [ZDPSS01] and silhouette-based modellers such as SKETCH [ZHH96] and *Teddy* [IMT99]. The former allows the user to mark up a 2D image with 3D hints such as surface normals and specified depths, and finds the smoothest Monge patch (or "range image") model which satisfies those constraints. The strong feature of the system is that the model is defined by an energy minimization, so that the model returned is always the smoothest possible, given the user's supplied constraints. Furthermore, because the energy is convex, a globally optimal solution is guaranteed, and can be obtained very quickly on modern computers. The weaknesses are twofold: first, the Monge patch representation is rather restrictive, representing just one side of the 3D model, and because its shape is difficult to control near the silhouette, it is difficult to create a realistic model by just stitching the model and its reflection. The second weakness is that a considerable user input is required, even for simple models.

Given that for many models, one of the strongest indicators of 3D shape is the silhouette, it is natural to seek a system which allows a fully 3D rather than $2\frac{1}{2}$D representation, and to constrain the 3D model to project to the silhouette. Two strands of previous research have addressed the problem of reconstruction from silhouettes in a single view; which we shall refer to as the "variational" and "sketching" methods.

Variational methods cast the problem as energy minimization. The objective is to find the minimal energy surface which projects to the image silhouette. Terzopoulos [TWK87] provided the first application of this scheme for image-based modelling, but used an iterative algorithm to "inflate" a 3D mesh until its image projection met the silhouette. The primary failings of this approach are: (a) the difficulty of maintaining mesh consistency during the interactions; and (b) the tendency to stop in local minima when fitting complex silhouettes. In a sense, the local minima are the real problem: the globally optimum energy rarely corresponds to a mangled mesh, so if it can be found, the mesh is consistent. Many strategies have been proposed to correct the consistency problem (a), we believe this paper is the first work to cleanly cast the formulation in a way that solves (b)

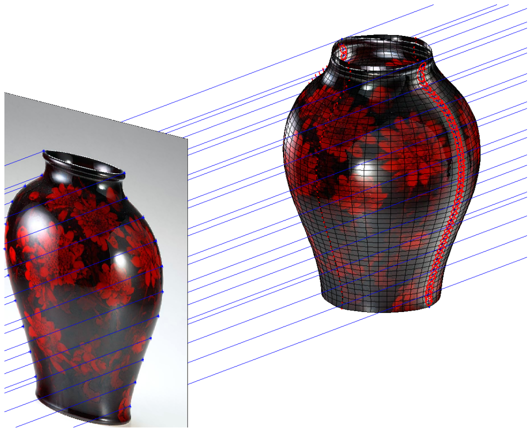

Figure 1: *We "lift" 3D models from images with a simple user interface. For simple models, the object silhouette provides all necessary constraints, while a number of simple editing operations allow more complex models to be created.*

in the general 3D case, although the $2\frac{1}{2}$D solution has been known since [Sze90].

The *sketching* systems do not have a problem with local optima. In particular, *Teddy* defines a special-purpose algorithm for the inflation of a user-defined contour to produce a 3D model. The algorithm works well for a variety of contours, and might make a good initial estimate for the iterative variational approaches. The disadvantage of *Teddy*'s inflation is that for complex silhouettes, the algorithm introduces artifacts, and it is not defined for surfaces of nonzero genus. Karpenko *et al.* [KHR02] address these deficiencies by expressing the 3D shape as the zero level set of a volumetric potential field. Their approach allows more complex silhouettes to be dealt with cleanly, and can force subsequent modifications of the object to continue to obey the silhouette constraint, which is difficult with *Teddy*. However, the sketchers overconstrain the problem, and make commitments to the surface shape which are difficult to revoke with subsequent editing operations. In contrast, we wish to impose the minimal possible constraint —the surface's perpendicularity to the viewing direction at the silhouette—and allow variational smoothness to do the rest. Solving a linear system to do the same is our biggest USP.

2. Constrained variational surfaces

We seek a 3D surface $\mathbf{r} : [0,1]^2 \mapsto \mathbb{R}^3$, which is as smooth as possible while obeying user-specified constraints. The surface is a function $\mathbf{r}(u,v) = [x(u,v), y(u,v), z(u,v)]^\top$. Smoothness is defined in terms of an energy on the surface, and we follow [Sze90, ZDPSS01] and use the thin plate energy

$$E(\mathbf{r}) = \int_0^1 \int_0^1 \|\mathbf{r}_{uu}\|^2 + 2\|\mathbf{r}_{uv}\|^2 + \|\mathbf{r}_{vv}\|^2 \mathrm{d}u\mathrm{d}v \quad (1)$$

Without constraints on the surface, the global minimum of $E(\mathbf{r})$ is the singularity $\mathbf{r}(u,v) = \mathbf{0}$. However, the imposition of constraints generates more interesting shapes. Constraints we consider may take several forms, as follows (the k subscripts indicate that in general we will have several simultaneous constraints, indexed by k):

- **Position** constraints are of the form $\mathbf{r}(u_k, v_k) = \mathbf{p}_k$ for known values of $u_k, v_k, \mathbf{p}_k$.
- **Normal** constraints require the surface normal at (u_k, v_k) to equal a supplied normal $\mathbf{n}_k$. The normal to $\mathbf{r}$ at a point is the unit vector along $\mathbf{r}_u \times \mathbf{r}_v$. Imposing the constraint as the pair of constraints linear in $\mathbf{r}$

$$\mathbf{n}_k \cdot \mathbf{r}_u(u_k, v_k) = 0 \quad (2)$$
$$\mathbf{n}_k \cdot \mathbf{r}_v(u_k, v_k) = 0 \quad (3)$$

  will allow the global optimum to be found.
- **Partial position** constraints act on just one component of $\mathbf{r}$, for example $z(u_k, v_k) = z_k$ constrains only the z coordinate at (u_k, v_k).

Subsequent constraints we shall discuss can be expressed in terms of these basic primitives, so we shall describe the discrete representation which we optimize.

The surface is represented by three $m \times n$ matrices, X, Y, Z, representing discretizations of the components $x(u,v), y(u,v), z(u,v)$ of $\mathbf{r}$. When solving for the surface, we shall reshape the matrices columnwise into vectors $\mathbf{x}, \mathbf{y}, \mathbf{z}$, and stack those into a single vector of unknowns $\mathbf{g}$. As a notational convenience, let reshape$(\cdot)$ convert column vectors to matrices or vice versa as appropriate, so $\mathbf{x} = \text{reshape}(X)$. In appropriate units, the central difference approximation to the first derivative x_u is $X_u = \frac{1}{2}(X(i+1) - X(i-1))$ and may be represented [ZDPSS01] as a large $(mn \times mn)$ sparse matrix C_u, so that $X_u = \text{reshape}(C_u \,\text{reshape}(X))$. Second derivatives are similarly represented by C_{uv} etc, so the energy function (1) in discrete form becomes

$$\varepsilon(\mathbf{x}) = \mathbf{x}^\top (C_{uu}^\top C_{uu} + 2C_{uv}^\top C_{uv} + C_{vv}^\top C_{vv})\mathbf{x} \quad (4)$$
$$E(\mathbf{g}) = \varepsilon(\mathbf{x}) + \varepsilon(\mathbf{y}) + \varepsilon(\mathbf{z}) \quad (5)$$
$$= \mathbf{g}^\top C\mathbf{g} \quad (6)$$

which is quadratic in $\mathbf{g}$ and may therefore be solved for using any number of reliable methods [ZDPSS01]. Linear constraints such as the position or normal constraints remain linear in $\mathbf{g}$, and are easily represented as a separate constraint equation $A\mathbf{g} = \mathbf{b}$. Our major contribution is to show that the silhouette constraint may be represented in a way that is linear in the unknowns $\mathbf{g}$.

The silhouette constraint The above is quite well known. However the constraints are hard to apply because the (u,v) parameters must be supplied for each constraint. Zhang *et al.* avoid this problem by restricting the surface to a Monge patch (graph surface, range image) of the form $\mathbf{r}(u,v) = (u, v, z(u,v))$. For single-view reconstruction, constraints are supplied in the image (e.g. the depth at pixel (x,y)) which immediately yields u and v.

For general 3D surfaces this mapping is not available, and bootstrapping the constraint input is difficult. However, we

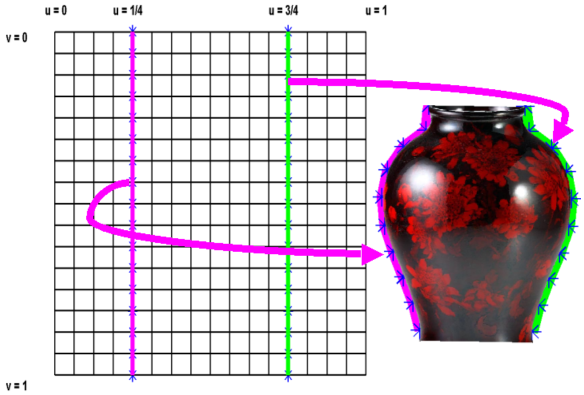

Figure 2: *In the continuous problem, if the contour generator is continuous (but not necessarily planar), we are at liberty to choose the uv parameter curves which project to the silhouette. Here we illustrate the choice $u = \frac{1}{4}$ for the left contour generator and $u = \frac{3}{4}$ for the right. In the discrete domain this yields the* ***global optimum*** *of the shape-from-silhouette problem, while all previous approaches have found local minimizers.*

show that the object's silhouette is a curve for which the mapping may readily be obtained. The object silhouette (see [Koe90] for genial discussions of these terms) is the image of the *contour generator* (CG), which is the set of 3D points on the surface at which the viewing direction is in the tangent plane. For convenience, we shall assume orthographic projection along the Z axis, and that the surface does not exhibit self-occlusions along the silhouette (i.e. that we can see all of the CG). These assumptions are often adequate in practice, but we shall discuss later how they can be removed.

The CG is a curve on the surface, or equivalently its *domain* is a curve in the (u,v) parameter space. If that curve is $\mathbf{d} = \{\mathbf{d}_t = (u_t, v_t) | 0 \leq t \leq 1\}$ then the CG is $\mathbf{c}_t = \mathbf{r}(u_t, v_t)$. We are given an image silhouette $\mathbf{s} = \{\mathbf{s}_t | 0 \leq t \leq 1\}$ (i.e. $\mathbf{s}$ is the infinite set of 2D points on the silhouette and we are assuming it has magically been parameterized by the same t as the CG). At each point on $\mathbf{s}$, we can compute the unit normal (n_x, n_y) to the 2D curve. Under orthographic projection along Z, the 3D surface normal $\mathbf{n}$ at any point on the CG must have Z component zero, and thus the 3D normal at t is given by $\mathbf{n}_t = (n_x, n_y, 0)$. This means we know the surface normal at any point on the contour generator, so the (infinite) set of linear constraints which force the silhouette of the 3D surface $\mathbf{r}$ to coincide with $\mathbf{s}$ are

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \mathbf{r}(u_t, v_t) = \mathbf{s}_t \qquad \text{[Projection]} \qquad (7)$$

$$\mathbf{n}_t^\top \mathbf{r}_u(u_t, v_t) = 0 \qquad \text{[Normal]} \qquad (8)$$

$$\mathbf{n}_t^\top \mathbf{r}_v(u_t, v_t) = 0 \qquad \text{[Normal]} \qquad (9)$$

amounting to four linear constraints on $\mathbf{r}$ for each $t \in [0,1]$.

The simple observation we can then make is to view the energy $E(\mathbf{r})$ as an approximation to surface curvature, which is invariant to parametrization. If we were minimizing curvature, we would be at liberty to choose any reparametrization of (u,v) without changing the energy. We use this freedom to *define* any curve in (u,v) to be the domain of the contour generator and fit subject to that constraint. For example, for objects of either cylinder or torus topology, the CG can be the curves $u = \frac{1}{4}, u = \frac{3}{4}$. Figure 2 illustrates this choice.

Of course, because we are not minimizing curvature, parametrization *does* matter, so it is necessary to make some modifications before the above scheme will work. These modifications control inflation and spillage.

Inflation A pure implementation of the above optimization produces a surface which does indeed project to the silhouette but for which $z(u,v) = 0 \; \forall u, v$. In order to avoid this solution, and to shape the surface away from the silhouette, two sorts of inflation constraint are useful. The simplest is a partial position constraint which requires the surface to pass through the $z = \pm 1$ planes. Imposing $z(0, \frac{1}{2}) = -1$ and $z(\frac{1}{2}, \frac{1}{2}) = 1$ amounts to two linear constraints which yield inflated models such as that in figure 3a.

Inflation curves are pairs of 2D curves drawn on the image (see figure 3c) which represent the silhouette of a surface of revolution (SOR). The inflation constraint they impose is to assign the z values along the (image of the) axis of the SOR, i.e. several constraints of the form $z(x_k, y_k) = z_k$. Unlike the silhouette constraint, we do not have the freedom to assign an arbitrary (u,v) parameter curve to this constraint, so the system is solved first with the silhouette constraint only, yielding a surface $\mathbf{r}'(u,v)$, which yields a mapping from (u,v) to image (x,y), which is inverted to produce mappings $u'(x,y), v'(x,y)$. The system is re-solved with the constraints $z(u'(x_k,y_k), v'(x_k,y_k)) = z_k$. Examples are shown in figure 3c.

Spillage The silhouette constraint guarantees that the surface is *locally* consistent with the image silhouette, but does not prevent unconstrained parts of the surface spilling out into the background. However because new constraints added to the system do not undo the silhouette constraint, spillage is trivially corrected by dragging any points which are outside the image boundary to the inside. Implementing partial position constraints on x and y only means the spillage is corrected without undue influence on the 3D shape. This is the step illustrated in figure 3b.

Implementation We solve the constrained optimization problem on a 64×64 grid, significantly smaller than that needed for Monge patch reconstruction. Solution using Matlab's sparse backslash operator takes less than one second.

3. Discussion

We have shown how a new formulation of 3D surface reconstruction from silhouettes allows the global minimizer of a

(a) (b) (c)

Figure 3: *Steps in model building for a genus one surface. (a) User supplies silhouettes and marks points of high curvature. Some of the model spills into the background. (b) User drags model points into the interior of the object. The minimization corrects the spill while maintaining silhouette consistency. (c) The user adds inflation curves to improve the 3D shape, creating a narrower spout, and adding a bulge at the base. Overall user input is the silhouette (power-assisted), three spillage drags, and six inflation curves, none of which need to be precise.* Middle and Bottom row*: 3D models with the* (u, v) *curves and constraints.*

thin-plate energy to be found as the solution to a linear system. Previous energy-based approaches have relied on iterative optimization strategies which frequently fell into local minima or mangled the surface mesh; and previous ad-hoc approaches could not guarantee to maintain the silhouette without complex polygon book-keeping.

Although we allow non-planar contour generators, we still assume that all of the CG is visible. Discontinuities caused by self-occlusion (torus swallowtails for example) can be handled by careful cutting of the parameter domain. Also, we can begin to consider optimization of curvature by iterative reweighting a quadratic energy term. This iteration should still be preferable to the Terzopoulos-based aproaches because the silhouette constraints are always obeyed throughout the optimization, and 2D investigations support this.

References

[DTM96] Debevec P. E., Taylor C. J., Malik J.: Modeling and rendering architecture from photographs: A hybrid geometry- and image- based approach. In *Proceedings, ACM SIGGRAPH* (1996), pp. 11–20.

[IMT99] Igarashi T., Matsuoka S., Tanaka H.: Teddy: a sketching interface for 3D freeform design. In *Proc. ACM SIGGRAPH* (1999), pp. 409–416.

[KHR02] Karpenko O., Hughes J. F., R. R.: Free-form sketching with variational implicit surfaces. *Computer Graphics Forum 21*, 3 (2002), 585–594.

[Koe90] Koenderink J.: *Solid Shape*. MIT Press, 1990.

[Sze90] Szeliski R.: Fast surface interpolation using hierarchical basis functions. *IEEE PAMI 12*, 6 (1990), 513–528.

[TWK87] Terzopoulos D., Witkin A., Kass M.: Symmetry-seeking models for 3D object reconstruction. *Proc. ICCV* (1987), 269–276.

[ZDPSS01] Zhang L., Dugas-Phocion G., Samson J., Seitz S.: Single view modeling of free-form scenes. In *Proc. CVPR* (2001), pp. I:990–997.

[ZHH96] Zeleznik R. C., Herndon K. P., Hughes J. F.: Sketch: an interface for sketching 3D scenes. In *Proc. ACM SIGGRAPH* (1996), pp. 163–170.