21

Reverse Engineering using Blending and Adaptive Shape Evolution

Douglas DeCarlo and Dimitris Metaxas
Department of Computer and Information Science
University of Pennsylvania, Philadelphia, PA 19104
dmd@gradient.cis.upenn.edu, dnm@central.cis.upenn.edu

Abstract

We propose a method for reverse engineering that is based on the use of topologically adaptive deformable models. The technique is based on a shape representation scheme which allows two shapes to be combined into a single model. The desired regions of the two shapes are selected, and then merged together forming a blended shape. For reconstruction, blending is incorporated into a deformable model framework. The model automatically adapts to the data, blending when necessary. Hierarchical blending allows multiple blends of a shape to occur, forming an evolution from the initial shape of a sphere to the final shape. Blending also allows the insertion of a hole between arbitrary locations. The models used are globally defined, making the recovered shape a natural symbolic description. We present reconstruction experiments involving shapes of various topologies.

Keywords

Deformable Models, Blending, Adaptive Shape Evolution, Range Data, Computer Vision

1 INTRODUCTION

We propose a method for the extraction of complex shapes from incomplete and sparse range data that is useful in reverse engineering applications. This method is based on our previously proposed shape representation scheme (see DeCarlo and Metaxas (1995)) which allows two parameterized primitives to be combined into a single model, and can be used for automatic surface reconstruction. The desired regions of the two primitives are selected, and then merged together forming a blended shape. For example, a cylinder and sphere can be blended together to form a light-bulb shape.

Many researchers have addressed this problem by creating expressive models with many parameters (see Delingette (1994), Liao and Medioni (1994), Malladi et al (1994), Szeliski et al (1993), Terzopoulos (1988)).

In other work, shape has been represented using a collection of parameters ordered by the level of detail (see Pentland (1991) and Vemuri (1994)). Techniques using superquadrics (e.g., Ferrie et al (1993), Leonardis et al (1994)) were presented to obtain part-level models. The models of Metaxas and Terzopoulos (1991, 1993) represented prominent shape features using global deformations and

M. J. Pratt et al. (eds.), *Product Modeling for Computer Integrated Design and Manufacture*

surface detail using local deformations. Axial blending by DeCarlo and Metaxas (1994) allowed the combination of shape primitives along common axes. In addition to a compact representation, the distinguishing feature of these models was the ability to represent objects with varying topology (such as a sphere and a torus) in a unified way.

In order to be applied to a wide class of objects, a shape model should have no a priori parameterization or imposition of physical properties. Examples of such restrictions can be seen in modal analysis (e.g., Pentland (1991)), where the modal calculation depends on some predefined elastic properties, and in work by DeCarlo and Metaxas (1994) and Metaxas and Terzopoulos (1993) where predefined types of global and local deformations constrain the class of objects that can be represented. Without prior knowledge, such a shape model should have an initial shape that does not favor *any* particular shape–a sphere is a reasonable choice. Furthermore, the associated shape estimation technique should allow the model to evolve based on the shape of the given data, making the representation data driven and not predefined. In the process of model evolution, abrupt changes in the model should be avoided, since such drastic decisions are rarely robust.

The desired shape evolution ability during shape estimation is encompassed in the blending-based shape representation scheme that was first proposed by DeCarlo and Metaxas (1995). In this method, specific regions on each of the two shapes are selected and "glued" together. It is also possible to glue in a hole, permitting the creation of holes through any part of the shape. The application of blending in a hierarchical way provides the ability to perform multiple blends. This arbitrary blending of global shape is a non-trivial generalization of the axial blending presented by DeCarlo and Metaxas (1994). Through a blending hierarchy (where the component shapes are themselves blended), multiple blends of a shape can occur, allowing complex shapes to be constructed from multiple parts. Our blending also allows the insertion of a hole between arbitrary locations.

Besides having wide shape coverage, additional benefits of using this representation become clear during shape reconstruction. For reconstruction, blending is incorporated into a deformable model framework. The model automatically adapts to the data using an automatic decision procedure based on the error of fit (see Metaxas and Terzopoulos 1993), performing blending when necessary. Starting from the initial shape of a sphere, a spatially smooth *evolution* of shape to the final shape is possible—no sudden jumps in the shape occur during fitting. We feel this greatly improves the robustness and stability of reconstruction, since no "drastic" decisions need to be made to alter the model, and initialization of the model is simple.

Any shape with a global shape parameterization can be used as a component shape for blending, allowing integration into existing CAD frameworks. It also should be possible to convert a blended shape into a model using algebraic blending surfaces, as proposed by Hoffmann and Hopcroft (1988). The component shapes we use are globally defined (superquadrics), making the recovered shape a compact symbolic description. This symbolic representation is expressed as a tree where qualitative adjacency relationships are described, and the leaves are shape primitives.

As opposed to all previous part-based shape estimation techniques, our model can be represented as a set of connected components where each component is an integral part of the model. An additional benefit of using globally parameterized models is their robustness to noise and their suitability in estimating the underlying shape of sparse and incomplete data.

In this paper, we briefly introduce the process of blending. Then, we present how blended models can be incorporated into the physics-based estimation framework developed by Metaxas and Terzopoulos (1993, 1991). We demonstrate our technique through a series of experiments involving incomplete range data from objects with various topologies.

2 BLENDED SHAPES

In the following sections, we will describe the process of shape blending. The models used to perform the blending are 3-D surface shape models defined by a global shape $\mathbf{s}$. Locations on the shape $\mathbf{s}$ are identified by material coordinates $\mathbf{u} = (u, v)$ which are specified over a domain Ω. For every 3-D shape (such as a superellipsoid), we have $\mathbf{s} : \Omega \rightarrow \mathbb{R}^3$. An example (in this case, a sphere) is shown in Fig. 1, and shows how the material coordinates (the domain Ω) are "folded up" resulting in the closed shape $\mathbf{s}$.

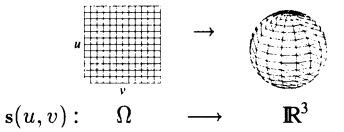


Figure 1 Example shape primitive $\mathbf{s}(\mathbf{u}) = \mathbf{s}(u, v)$

To represent the shape, a mesh of nodes is used, where each node is assigned a unique point in Ω. The edges connecting the nodes (determined by adjacency in Ω) specify the topology. Nodes are merged together where points in Ω map to the same 3-D model location (such as for the poles of a sphere) resulting in a mesh topology that agrees with the topology of the shape primitive.

A *blended* shape is a combination of two component shapes. The desired pieces of each component shape are selected, and are "glued" together smoothly using linear interpolation. The gluing operation performed here is a generalization of the the axial blending presented by DeCarlo and Metaxas (1994). We can blend together two shapes using the following formula:

$$\mathbf{s}(\mathbf{u}) = \mathbf{s}_1(\mathbf{u})\alpha(\mathbf{u}) + \mathcal{R}\Big(\mathbf{s}_2(\mathcal{B}(\mathbf{u}))\Big)\Big(1 - \alpha(\mathcal{B}(\mathbf{u}))\Big), \tag{1}$$

where $\mathbf{s}_1$ and $\mathbf{s}_2$ are the component shapes, as in Fig. 2(a), and $\alpha : \Omega \rightarrow [0, 1]$ is the blending function which controls how each of the component shapes are expressed and how they are glued together. α is specified using the blending regions shown in Fig. 2(b). These blending regions are overlaid on the material coordinate spaces of the component shapes. The portion of these spaces corresponding to the retained surface is shown in white, while the portion which is removed is shown in gray. The retained portions of the surfaces corresponding to these regions are shown in Fig. 2(c). Basically, the blending regions specify where to "cut" the shape (the boundary line), and which part of the shape to keep (white or gray). Ideas similar to these cutting and gluing operations have been used in studying manifold surgery for topology an in the work by Hirsch 1991), where manifolds of arbitrary topology are constructed using a few simple components. In order to glue arbitrary parts of shapes together, they first must be prepared–the mappings $\mathcal{B} : \Omega \rightarrow \Omega$ and $\mathcal{R} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ serve this purpose. For the moment, we will assume that both $\mathcal{B}$ and $\mathcal{R}$ are identity mappings. Finally, $\mathbf{s}$ is the resulting blended shape, shown in Fig. 2(d).

The resulting blended shape $\mathbf{s}$ has a smooth transition between each of its components ($\mathbf{s}_1$ and $\mathbf{s}_2$). For the example blending region given in Fig. 3(a), the corresponding blending function α is shown in (b). Notice how α is 1 where the region is white, 0 where the region is gray, and has a smooth transition from 0 to 1 connecting these two regions. The area on the shape primitive expressed by this

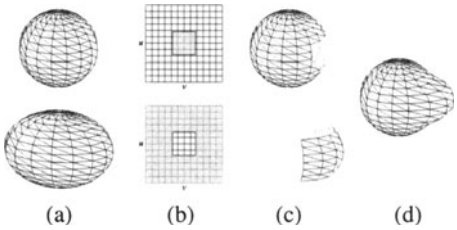


Figure 2 Simple blending example

particular blending region is displayed in Fig. 3(c) as the white region on the shape. A discussion of the implementation of α, as well as the size of the transition region (the "steepness" of the plateau) is given in section 2.3.

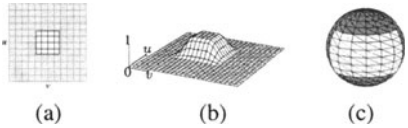


Figure 3 Example blending function

The blending regions of each of the component shapes in Fig. 2(b) line up exactly. In general, the blending region boundary for s_1 does not have such a simple correspondence with the blending region boundary for s_2, since arbitrary locations of the component shapes can be selected. The invertible mapping $\mathcal{B}$ is used to form a simple correspondence. Figs. 4(b) and (c) show how $\mathcal{B}$ maps the blending region for s_2 to correspond with s_1, shown in (a). $\mathcal{B}$ maps the boundary *and* its neighborhood, permitting a correspondence to be performed for those points near the boundary where linear interpolation is performed (where $0 < \alpha < 1$). Each boundary has an associated direction (clockwise or counter-clockwise), which must be preserved by $\mathcal{B}$. Not doing so can produce a non-orientable self-intersecting surface (not a likely property for an estimated shape!).

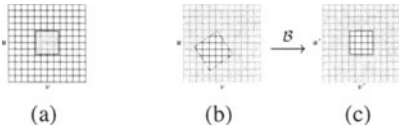


Figure 4 Blending region alignment using $\mathcal{B}$

In addition to the correspondence in material coordinate space performed by $\mathcal{B}$, there must also be a correspondence in 3-D space. In other words, the two components must be spatially aligned before blending is performed. This spatial correspondence is performed by $\mathcal{R}$, a rigid transformation (translation and rotation) applied to s_2. An example of this spatial alignment is shown in Fig. 5, where the result of applying $\mathcal{R}$ is shown in Fig. 5(c). The blending regions used are given in Figs. 4(a) and (b).

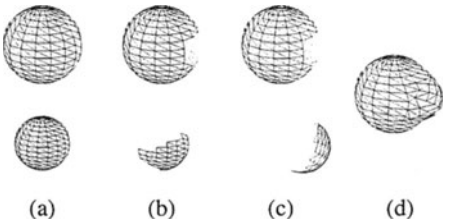


Figure 5 Another blending example

The mesh of the blended result in Fig. 5(d) is a composite mesh formed by combining the meshes in (c). A mesh merging algorithm developed by Rutishauser et al (1994) can be used to perform this merging as a post-processing step to blending. The bottleneck in mesh merging algorithms is the nearest-node computations, which become constant time operations since $\mathcal{B}$ provides correspondence between meshes.

2.1 Hole additions and indentations

The addition of a hole requires two blending regions—two parts of a shape are cut out, and a hole is glued in its place. Fig. 6(a) shows a sphere with two parts removed. The blended regions for this sphere are also displayed, showing how two regions in Ω are removed. Fig. 6(b) shows the hole of a torus. Two separate cuts are made around the torus hole, and the resulting hole is glued into the sphere, forming the torus shown in Fig. 6(c).

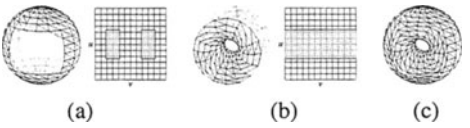


Figure 6 Hole addition example

Unlike the holes in the work of DeCarlo and Metaxas (1994) which could only be added between the poles of a sphere, the holes presented here are general, and can be added between any two locations of a blended object. We will see in section 3.1 how the hole additions are performed automatically during estimation.

Indentations can also be produced using blending. Fig. 7(a) shows a cylinder with part of the top removed by the given blending region. Fig. 7(b) shows a shape which is turned inside-out by having a negative axis length, producing a closed surface with the surface on the inside. Removing part of the surface reveals the inside, which can be used for making a cavity. Blending together the cylinder and inverted cylinder produces the cup shown in Fig. 7(c).

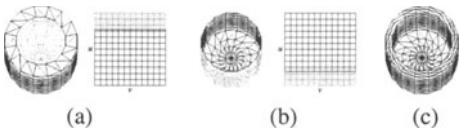


Figure 7 Indentation formation

2.2 Hierarchy of blended shapes

In the sections above, the blending described is performed on simple shapes. Without additional machinery, we can perform blending on component shapes which are already blended. This results in a tree structure, where the leaves of the tree are shape primitives, and the internal nodes of the tree are blended shapes. Since $\mathcal{B}$ is invertible, it can be used to find corresponding points on any two component shapes anywhere in this tree structure.

Since holes can be added anywhere, this means that a closed object of any topology can be represented. We will see an example of the fitting of a two-holed object in section 4. We can produce a symbolic description of a blended shape using this hierarchy. For each of the experiments performed in section 4, a symbolic description is provided. Such a description may be useful for recognition purposes.

2.3 Representation of blending regions

For the applications in this paper, the blending regions are defined as four-sided regions in Ω space. Each region has 6 parameters, c_u, c_v, p_{1_u}, p_{1_v}, p_{2_u} and p_{2_v} (where the vector p_1 is linearly independent of p_2), as shown in Fig. 8(a). For blending regions which include a pole of $\mathbf{s}$, a slightly different formulation will be needed.

There are also additional parameters for each blended shape–a translation and rotation to specify $\mathcal{R}$ (6 parameters), the parameter d which controls the extent of the transition region of α (plateau "steepness"), and h which restricts the blending function range to $[h, 1]$. h is used to control the magnitude of the contribution of $\mathbf{s}_2$ to the resulting shape (h is used to smoothly add a hole in section 3.1).

A straightforward basis transformation operation maps points in Ω to points in the basis (p_1, p_2). We will denote the point $\mathbf{u} \in \Omega$ expressed in this basis as $\bar{\mathbf{u}}$. Points on the boundary of the blending region have $\|\bar{\mathbf{u}}\| = 1$, and points on the interior have $\|\bar{\mathbf{u}}\| < 1$. $\mathcal{B}$ is constructed by transforming a value of $\mathbf{u}$ to $\bar{\mathbf{u}}$ using the blending region for $\mathbf{s}_2$, and then back using the blending region for $\mathbf{s}_1$. α is computed from $\bar{\mathbf{u}}$ by setting α to 0 where $\|\bar{\mathbf{u}}\| < 1 - d$, to 1 where $\|\bar{\mathbf{u}}\| > 1 + d$, and to an intermediate value using the Hermite polynomial $H(x) = 3x^2 - 2x^3$ for $x \in [0, 1]$ to produce a C^1 smooth piecewise polynomial surface.

(a) (b)

Figure 8 (a) Blending region representation (b) Region of force equilibration

3 RECONSTRUCTION AND EVOLUTION

Blending is incorporated into the physics-based framework of Metaxas and Terzopoulos (1993) to accomplish the shape reconstruction. This requires the collection of parameters of the model into generalized coordinates $\mathbf{q_s}$:

$$\mathbf{q_s} = (\mathbf{q_{s_1}}^\top, \mathbf{q_{s_2}}^\top, \mathbf{q}_b^\top, \mathbf{q_T}^\top)^\top. \tag{2}$$

$\mathbf{q_{s_1}}$ and $\mathbf{q_{s_2}}$ are the parameters of each of the component shapes, $\mathbf{q}_b$ are the parameters that specify α, $\mathcal{B}$ and $\mathcal{R}$, and $\mathbf{q_T}$ are parameters of global parameterized deformations (such as bending), which can be added on top of a blended shape. For a hierarchical blended shape, each of the component shapes $\mathbf{q_{s_1}}$ and $\mathbf{q_{s_2}}$ can either be a primitive, or a child blended shape which will have the form given by (2).

When fitting the model to data, the goal of shape reconstruction is to recover $\mathbf{q}$. The application of forces to the surface of the model deforms it into the shape represented by the data as shown by Terzopoulos and Metaxas (1991).

The dynamics framework requires the computation of $\mathbf{J_s} = \partial \mathbf{s}/\partial \mathbf{q_s}$, the Jacobian for the shape $\mathbf{s}$. $\mathbf{J_s}$ "converts" the applied forces into generalized forces, which deform the global shape. The addition of blending alters the computation of $\mathbf{J_s}$. In particular, from (1):

$$\mathbf{J_s} = \left[\alpha(\mathbf{u})\mathbf{J_{s_1}} \;\middle|\; \Big(1 - \alpha(\mathcal{B}(\mathbf{u}))\Big)\mathcal{R}_{\text{rot}}(\mathbf{J_{s_2}}) \;\middle|\; \mathbf{J}_b\right], \tag{3}$$

where $\mathbf{J_{s_1}} = \partial \mathbf{s_1}/\partial \mathbf{q_{s_1}}$ is the Jacobian for $\mathbf{s_1}$, $\mathbf{J_{s_2}} = \partial \mathbf{s_2}/\partial \mathbf{q_{s_2}}$ is the Jacobian for $\mathbf{s_2}$, and $\mathbf{J}_b = \partial \mathbf{s}(\mathbf{u})/\partial \mathbf{q}_b$ is the Jacobian for the parameters of the blending function, and reflects how the global shape $\mathbf{s}$ changes with respect to the blending function parameters $\mathbf{q}_b$. $\mathcal{R}_{\text{rot}}$ is the rotational component of $\mathcal{R}$. From (1), we find that

$$\mathbf{J}_b = \mathbf{s_1}(\mathbf{u})\frac{\partial \alpha(\mathbf{u})}{\partial \mathbf{q}_b} - \mathcal{R}\Big(\mathbf{s_2}(\mathcal{B}(\mathbf{u}))\Big)\frac{\partial \alpha(\mathcal{B}(\mathbf{u}))}{\partial \mathbf{q}_b} \tag{4}$$

Given the implementation of α presented in section 2.3, we can construct a piecewise derivative of each of the blending parameters in $\mathbf{q}_b$.

3.1 Model evolution

Our initial model in our estimation system is always a sphere (represented using a single superellipsoid). During reconstruction, blending is automatically performed based on forces from the data points and

the current shape. The addition of each blend splits the model, requiring the specification of blending regions. Repeated application of blending causes an *evolution* from a sphere to the final model.

For a given shape model and data, the estimation process eventually reaches a steady state–all applied forces equilibrate or vanish. Forces equilibrate due to the inability of the model to reach a shape that represents the data. Fig. 8(b) shows a hypothetical situation where forces have equilibrated (the ellipsoid cannot produce the tapered shape represented by this data).

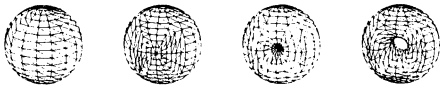
A region growing algorithm (where a region consists of nodes, and connectivity is defined by the mesh) is applied to the mesh to isolate regions on the shape with equilibrating forces, such as the bold region in Fig. 8(b). Using equilibrated forces is similar to finding residual data points in the work of Liao and Medioni (1994). Once such a region is located, the current model is automatically *split*, and is replaced with a blended model where $\mathbf{s}_1$ and $\mathbf{s}_2$ are initially equal. The boundary of both blending regions is initialized to the region of equilibration, and $\mathcal{R}$ is initialized to the identity transformation.

This operation changes only the representation of the model, *not* the geometry of the shape. The benefit of performing this representational split is that the shape can now deform to better capture the deviation in the data. According to (3), the blending function has the desirable effect of localizing the effect of a force to the appropriate shape component. The forces that were previously pulling on $\mathbf{s}_1$ in the blending region now cause the deformation of $\mathbf{s}_2$.

The addition of a hole to the model requires the proximity detection of different parts of the shape. Surface self-intersection detection methods such as in the work by Manocha and Canny (1991) can be used. Once such a proximity is detected, a blended shape is automatically constructed which combines the previous model with a hole that is aligned to join the two areas in proximity.

Holes do not necessarily form between two locations on the same component shape. If the hole is deep, it is very common for the hole to form between two indentations formed due to the presence of the hole in the data, as seen in the experiment in Figs. 13(f)-(g). The blending regions for $\mathbf{s}_1$ are computed from the results of the proximity detection. A superquadric toroid is used for $\mathbf{s}_2$ using the blending region from Fig. 6(b). $\mathcal{R}$ must also be specified with the hole aligned with the proximate regions on $\mathbf{s}_1$. This is performed by a separate dynamic fitting process where the positions of boundary regions of $\mathbf{s}_1$ pull on a model containing the boundary regions of $\mathbf{s}_2$. A bending deformation can be added to the hole to allow bent holes.

Fig. 9 shows the transformation from a shape model shown in (a) (in this case, a sphere) to include a hole (d). Fig. 9(b) shows the model which is constructed when the hole blend is initially created. The torus hole is not expressed in the resulting shape since $h = 1$, and the topology of the hole mesh is that of a sphere. During the fitting process, if a hole is present in the data, data forces change h to have the value 0, shown in Fig. 9(c). When $h = 0$, the topology of the hole mesh becomes that of a torus, and the result is a torus with a closed hole, which can now open as shown in Fig. 9(d). When $h > 0$, the torus hole is not permitted to open, since doing so produces an open surface.



(a) (b) (c) (d)

Figure 9 Hole evolution

It is by these two processes–splitting and hole addition, which produce the evolution of the model

from a sphere to the final model when combined with the physics-based deformation. Neither of these processes cause abrupt changes in the geometry of the model. It is interesting to note that the number of parameters required for a blended shape exceeds that of a CSG model. Blending requires the explicit specification of blending region boundaries, while CSG uses object interpenetration. The benefit of this specification is seen during estimation, where a prior segmentation of data is not necessary. Instead, the evolution methods described above provide a good initial guess at a part segmentation. The physics-based estimation technique then finds a set of parameters (including those which specify the blending regions) which better represent the data, to form the parts of the model.

4 EXPERIMENTS

In the following three fitting experiments, we show the results of using our shape estimation system using superquadric ellipsoids and toroids. Fig. 10 shows information on each of the experiments including the source and size of the set, the number of parameters in the final model, the number of iterations taken by the solver, and the resulting mean squared error (MSE). Each iteration takes approximately 1/2 second on a 100 MHz SGI R4000.

Data	Source	Points	MSE	Parm	Iter
block/cylinder	MSU (column2)	1034	1.4%	29	190
mug	MSU (cup1)	1207	2.8%	68	372
two holed	CAD generated	893	1.0%	83	511

MSU: data from the Michigan State University PRIP database

Figure 10 Experiment data and statistics

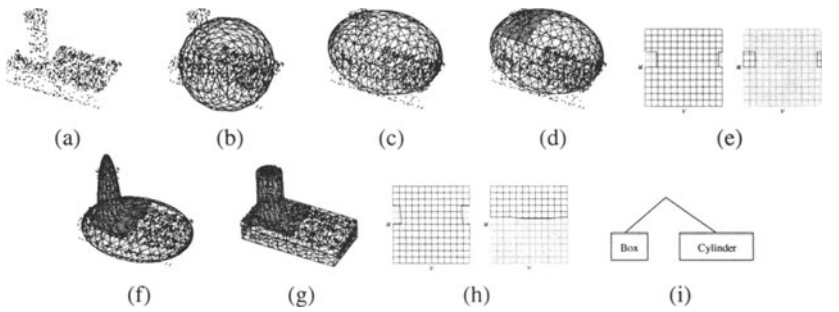


Figure 11 Fitting of a block and cylinder

In each of the three examples, the data set and initial model of a sphere is shown. The first fit shown is a rough fit—by fitting the axis parameters of the initial model. After the final result, a symbolic representation of the object is shown (with manually generated names).

Fig. 11(d) shows the first adaptation of the shape model, where a blending region was added (in

gray). The blended region added is shown in (e). After fitting the new shape model, the shape in (f) is formed, followed by the final fit in (g), and the final blending regions in (h).

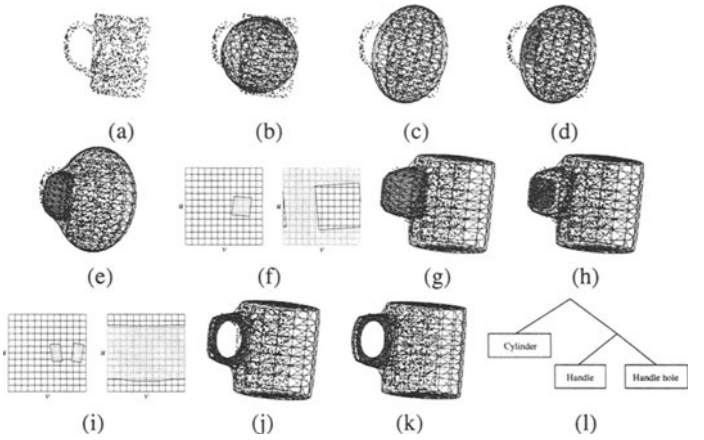


Figure 12 Fitting of a mug

The fitting of a mug is shown in Fig. 12. The blending region which corresponds to the mug handle forms in (d), and after rough fitting is shown in (e) with the corresponding blending region in (f). After further fitting of the handle in (g), a hole blend is added in (h) with blending region (i). After rough fitting and hole opening (j), the final fit is obtained (k).

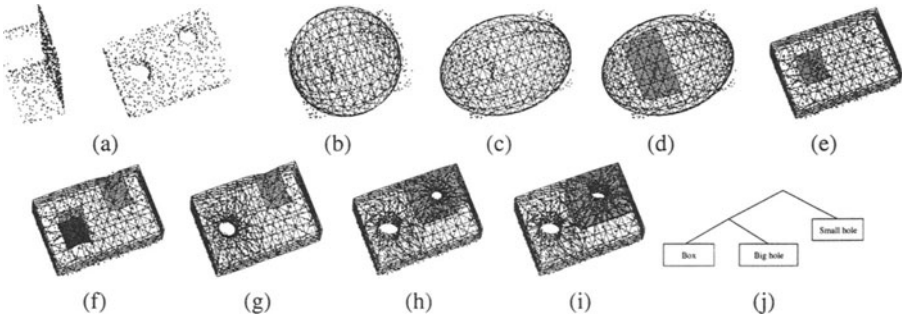


Figure 13 Fitting of a two holed object

Finally, the fitting of an object with two holes is shown in Fig. 13. An indentation blending region for one of the holes is formed in (d), and after initial fitting becomes indented (e). After several more blends, the shape evolves two additional indentations (f). After further fitting, a hole has formed in (g). The second hole has formed in (h), and the final fitted result is shown in (i).

4.1 Examination of stability

We first examine the effect of using different range-scanner views of the same object. In this case, three different views of a two-holed object (synthesized data) are examined. The data in Fig. 14(a) and (d) show views where the hole interiors are visible. The resulting estimated blended shapes, shown in (b) and (e), respectively, appear similar in shape, while clearly the hierarchies formed, shown in (c) and (f) and qualitatively different (the order of the big and small holes in the trees are reversed). This is due to the different order of decisions made in the model evolution (see section 3.1) during estimation. The next example, shown in Fig. 14(g), is a view which contains very little interior hole data. As a result, only an indentation forms, as seen in (h) and (i).

Fig. 15(a),(d), and (g) show data for a box shaped object with a hole, and a varying sized cylindrical attachment. As seen by the fitting results in (b),(e) and (h), the resulting shapes fit the data well, yet their hierarchies (c),(f) and (i) can sometimes vary as in the earlier example.

When using a global description of shape, one would expect representational changes to occur (such as the tree ordering differences seen here). For our method, representational changes can occur when the size of extracted parts of the object change size or relative position.

4.2 User-interaction

We can give the user some control over how the hierarchy forms by allowing them to select a subset of data that is currently active during estimation. Fig. 16 shows how the user can de-select a portion of the data, resulting in the data shown in (a). The cylindrical part of the object has been removed from the full data set, shown in (d) (this is also the same data set as in Fig 15(a)). The fit to the restricted data set is shown in Fig. 16(b), with the corresponding hierarchy in (c). As would be expected, the cylinder blend is not formed. At this point, the full data set is used to complete the fit, shown in Fig. 16(e). The resulting final hierarchy, shown in (f) is different from that shown in Fig. 15(c), due to the user interaction.

5 CONCLUSIONS AND FUTURE WORK

The blending method we have presented, when combined with the deformable model shape estimation framework, allows the smooth evolution from the initial model of a sphere to the final model. Over the course of the reconstruction, the representation of the model automatically adapts to improve its fit of the data. The model is split when necessary, and holes in the model can be added between arbitrary locations.

The adaptation of the model is performed in a way that does not include any abrupt geometric changes in the model. Since the model is based on global parameterizations, the technique is robust to noise and can be used in sparse and incomplete data. We have presented shape estimation results for objects with varying topology whose shape could not be estimated compactly and robustly with previous shape models and techniques.

As in many other frameworks, physics-based estimation with blending uses parameter scheduling (see DeCarlo and Metaxas (1994)) to allow stable fitting of a large number of parameters. A more qualitative scheme for performing this scheduling should be developed.

Hole removal using a "fission" process (such as turning a donut into a croissant) is also possible

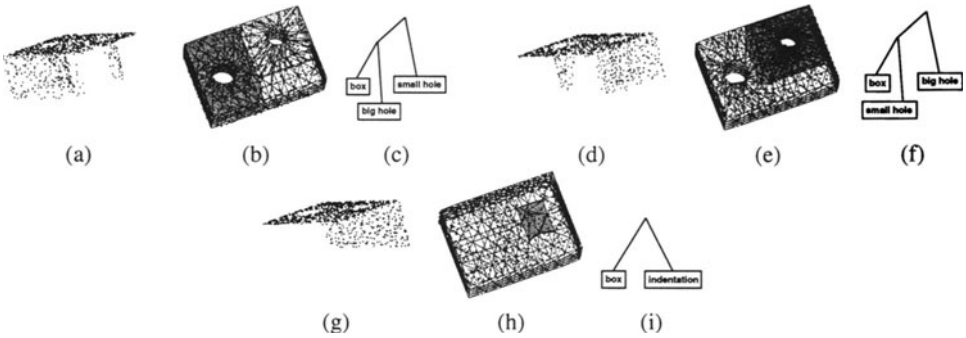


Figure 14 Various 3D range scanner views of two holed object

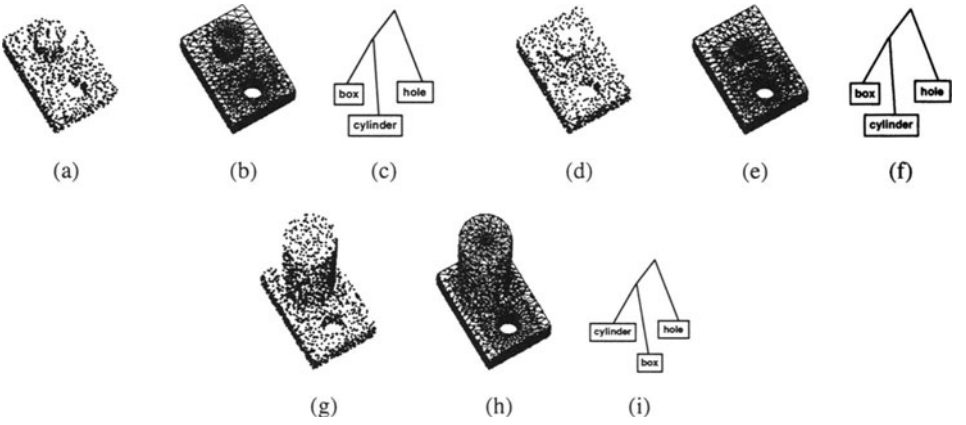


Figure 15 Various cylinder sizes in box/cylinder/hole object

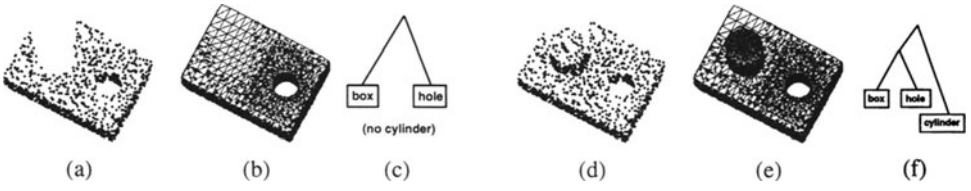


Figure 16 User disabling part of data to alter hierarchy

using blending. This would allow the framework to easily back out of a hole decision that seemed reasonable upon its application.

The physics-based estimation framework used has been shown to be relatively insensitive to small changes in data and orientation due to its global nature. For blending, small changes such as these can produce a different final model, but one that is qualitatively the same. Large changes, on the other hand, are likely to cause a different part breakdown. This, and its effects on recognition applications, require further investigation.

REFERENCES

I. Biederman. Recognition-by-components: a theory of human image understanding. *Psychological Review*, 94:115–147, April 1987.

T. Binford. Visual perception by computer. In *IEEE Conference on Systems and Control*, December 1971.

D. DeCarlo and D. Metaxas. Blended deformable models. In *Proceedings CVPR '94*, pages 566–572, 1994.

D. DeCarlo and D. Metaxas. Adaptive shape evolution using blending. In *Proceedings ICCV '95*, pages 834–839, 1995.

H. Delingette. Simplex meshes: A general representation for 3D shape reconstruction. In *Proceedings CVPR '94*, pages 856–859, 1994.

F. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes and superquadrics: Geometry from the bottom up. *IEEE Pattern Analysis and Machine Intelligence*, 15(8):771–784, 1993.

A. J. Hanson. Hyperquadrics: smoothly deformable shapes with convex polyhedral bounds. *Computer Vision, Graphics, and Image Processing*, 44:191–210, 1988.

M. W. Hirsch. *Differentiable Topology*. Springer-Verlag, 1991.

A. Leonardis, F. Solina, and A. Macerl. A direct recovery of superquadric models in range images using recover-and-select paradigm. In *Proceedings ECCV '94*, pages 309–318, 1994.

C. Liao and G. Medioni. Simulataneous segmentation and approximation of complex patterns. In *Proceedings CVPR '94*, pages 617–623, 1994.

R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Pattern Analysis and Machine Intelligence*, 1994, to appear.

D. Manocha and J. F. Canny. A new approach for surface intersection. *International Journal of Computational Geometry and Applications*, pages 491–516, 1991.

D. Marr and K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings Royal Society London*, 200:269–294, 1978.

D. Metaxas and D. Terzopoulos. "Shape and Nonrigid Motion Estimation Through Physics-based Synthesis," *IEEE Pattern Analysis and Machine Intelligence*, 15(6):580–591, June 1993.

A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *IEEE Pattern Analysis and Machine Intelligence*, 13(7):715–729, 1991.

M. Rutishauser, M. Stricker, and M. Trobina. Merging range images of arbitrarily shaped objects. In *Proceedings CVPR '94*, pages 573–580, 1994.

F. Solina and R. Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Pattern Analysis and Machine Intelligence*, 12(2):131–147, 1990.

R. Szeliski, D. Tonnesen, and D. Terzopoulos. Modeling surfaces of arbitrary topology with dynamic particles. In *Proceedings CVPR '93*, pages 82–87, 1993.

D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Pattern Analysis and Machine Intelligence*, 13(7):703–714, 1991.

D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.

B. C. Vemuri and A. Radisavljevic. Multiresolution stochastic hybrid shape models with fractal priors. *ACM Transactions on Graphics*, 13(2):177–207, 1994.