RuCCS TR-37
August, 1997

Using Aspect Graphs to Control the Recovery and Tracking of Deformable Models

Sven Dickinson
Center for Cognitive Science and
Department of Computer Science
Rutgers University
sven@cs.rutgers.edu

Dimitri Metaxas
Department of Computer and
Information Science
University of Pennsylvania
dnm@graphics.cis.upenn.edu

Technical Report #37
Center for Cognitive Science (RuCCS)
Rutgers University, Busch Campus
Psychology Building Addition
New Brunswick, NJ 08903

Using Aspect Graphs to Control the Recovery and Tracking of Deformable Models[*]

Sven J. Dickinson
Department of Computer Science and
Rutgers Center for Cognitive Science (RuCCS)
Rutgers University
New Brunswick, NJ 08903

Dimitri Metaxas
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104-6389

Abstract

Active or deformable models have emerged as a popular modeling paradigm in computer vision. These models have the flexibility to adapt themselves to the image data, offering the potential for both generic object recognition and non-rigid object tracking. Because these active models are underconstrained, however, deformable shape recovery often requires manual segmentation or good model initialization, while active contour trackers have been able to track only an object's translation in the image. In this paper, we report our current progress in using a part-based aspect graph representation of an object [14] to provide the missing constraints on data-driven deformable model recovery and tracking processes.

[*] Appears in *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 11, No. 1, February, 1997, pp 115–142. (special issue containing selected papers from the Workshop on Spatial Computing: Representation, Interpretation and Applications, Curtin University of Technology, Perth, Western Australia, December 2–3, 1995.)

1 Introduction

In the computer vision community, active or deformable models have emerged as a popular modeling paradigm, and have been applied to both the problems of shape recovery and shape tracking. The approaches to shape recovery are exemplified by the class of deformable or active model recovery techniques, in which a model contour (in 2-D) or surface (in 3-D) adapts itself to the image data under the influence of "forces" exerted by the image data [18, 26, 27, 25]. As shown in Figure 1, points on the model are "pulled" towards corresponding (e.g., closest) data points in the image, with the integrity of the model often maintained by giving the model physical properties such as mass, stiffness, and damping. Having such flexible models is critical in an object recognition system, particularly when object models are more generic and do not specify exact geometry.

As powerful as these data-driven, deformable model recovery techniques are, they are not without their limitations. Their success relies on both the accuracy of initial image segmentation and initial placement of the model given the segmented data. For example, such techniques often assume that the bounding contour of a region belongs to the object, a problem when the object is occluded. Furthermore, focusing only on an object's silhouette assumes 3-D models with rotational symmetry, i.e., no surface discontinuities, e.g., [25]. In addition, such techniques often require a manual segmentation of an object into parts to which models are fitted, e.g., [26]. If the models are not properly initialized, a canonical fit may not be possible, e.g., [23]. These limitations are a consequence of using such unconstrained models.

Data-driven, deformable models have also been applied to the problem of tracking both 2-D and 3-D shapes. As shown in Figure 2, a properly initialized model in one frame is placed in a subsequent frame and, provided the motion is small between the two frames, will change its position and shape to align itself with the data in the new frame. These data-driven approaches to shape tracking track the silhouette of a blob in 2-D (or surface of a blob in 3-D), e.g., [18, 6, 24]. Although 2-D translation can be recovered and, in some cases, translation in depth (e.g., [5]), lack of any model information prevents the recovery of rotation in depth and the detection of occlusion.

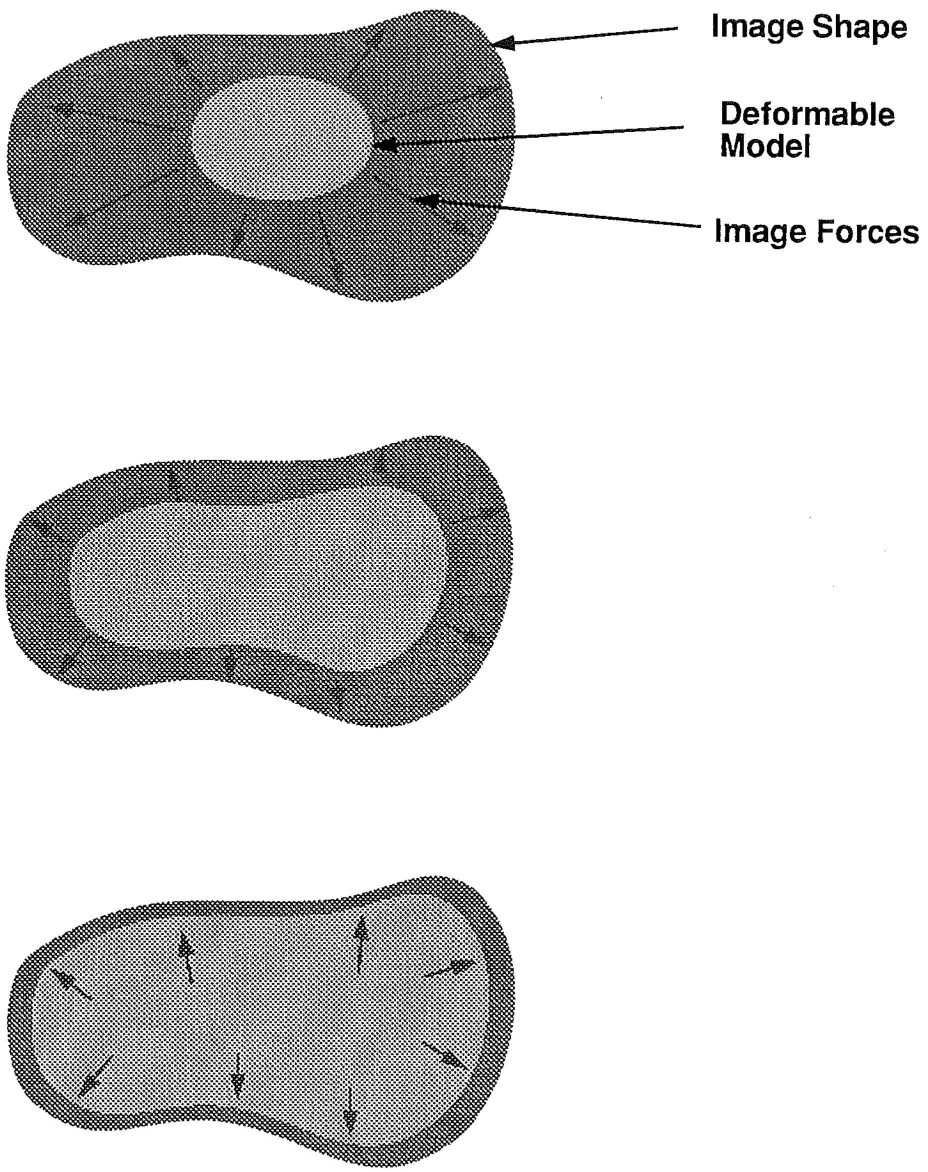

Figure 1: Data-Driven Shape Recovery

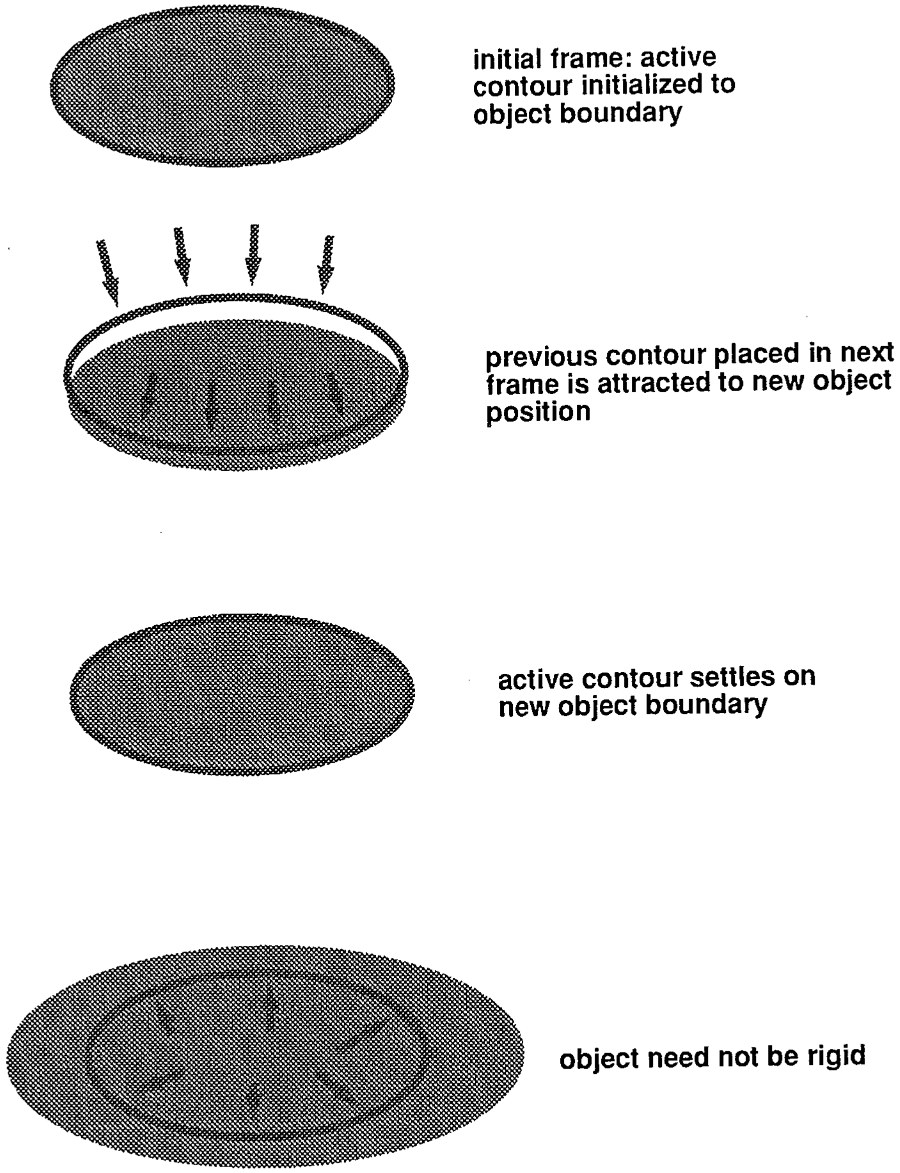
initial frame: active contour initialized to object boundary
previous contour placed in next frame is attracted to new object position
active contour settles on new object boundary
object need not be rigid

Figure 2: Data-Driven Object Tracking

In this paper, we show how an object representation integrating object-centered volumetric part models and viewer-centered part aspects, introduced in [14], can be used to provide strong constraints on the recovery and tracking of 3-D shape using deformable models. We review our current progress in a number of areas, including the recovery of 3-D deformable models from both 2-D and 3-D data, and both the qualitative and quantitative tracking of 3-D shape from 2-D data. An emerging theme thoughout the discussion will be the use of an aspect-based shape description to provide a number of powerful constraints whose absence limits current work in the recovery and tracking of deformable models. In the following sections, we review the object representation, and show how its application to both shape recovery and tracking can overcome the limitations of the deformable model shape recovery and tracking approaches described above.

2 A Parts-Based Aspect Representation

In this section, we briefly review a representation which models an object's 3-D shape in terms of a set of qualitatively-defined volumetric parts [14]. This representation, combining both object-centered and viewer-centered models, forms the backbone of our qualitative shape recovery and object recognition (both top-down and bottom-up) paradigms, reported in [16, 15, 9, 10]. In the following sections, we will see how this same representation can be used to constrain the recovery and tracking of deformable models from both 2-D and 3-D image data.

The hybrid representation we use to describe objects draws on two prevalent representation schools in the computer vision community. The first school is called object-centered modeling, whereby three-dimensional object descriptions are invariant to changes in their position and orientation with respect to the viewer. The second school is called viewer-centered modeling, whereby an object description consists of the set of all possible views of an object, often linked together to form an aspect graph. Object-centered models are compact, but their recognition from 2-D images requires making 3-D inferences from 2-D features. Viewer-centered models, on the other hand, reduce the recognition problem from three dimensions down to two, but incur the cost of having to store many different views for each object.

In order to meet the goals of qualitative object modeling and matching, we first model objects as object-centered constructions of volumetric parts chosen from some arbitrary, finite set of part classes [14]. It is at the volumetric part modeling level, that we invoke the concept of viewer-centered modeling. Traditional aspect graph representations of 3-D objects model an entire object with a set of aspects (or views), each defining a topologically distinct view of an object in terms of its visible surfaces [19]. Our approach differs in that we use aspects to represent a (typically small) set of volumetric parts from which objects appearing in our image database are constructed, rather than representing the entire object directly.

Our goal is to use aspects to recover the 3-D volumetric parts that make up the object in order to carry out a recognition-by-parts procedure, rather than attempting to use aspects to recognize entire objects. The advantage of this approach is that since the number of qualitatively different volumes is generally small, the number of possible aspects is limited and, more important, *independent* of the number of objects in the database. By having a sufficiently large set of volumetric part building blocks, and by assuming that objects appearing in the image database can be composed from this set, our training phase, which computes the part views, is independent of the contents of the image database.

The disadvantage of our hybrid representation is that if a volumetric part is occluded from a given 3-D viewpoint, its projected aspect in the image will also be occluded. We must therefore accommodate the matching of occluded aspects, which we accomplish by use of a hierarchical representation we call the *aspect hierarchy*. The aspect hierarchy consists of three levels, consisting of the set of *aspects* that model the chosen volumes, the set of component *faces* of the aspects, and the set of *boundary groups* representing all subsets of contours bounding the faces. The ambiguous mappings between the levels of the aspect hierarchy are captured in a set of upward and downward conditional probabilities, mapping boundary groups to faces, faces to aspects, and aspects to volumes [8]. The probabilities are estimated from a frequency analysis of features viewed over a sampled viewing sphere centered on each of the volumetric classes.

The representation for aspects has a tremendous impact on the coverage of the 3-D part classes. If aspects encode a precise specification of angles between lines, curvature, etc.,

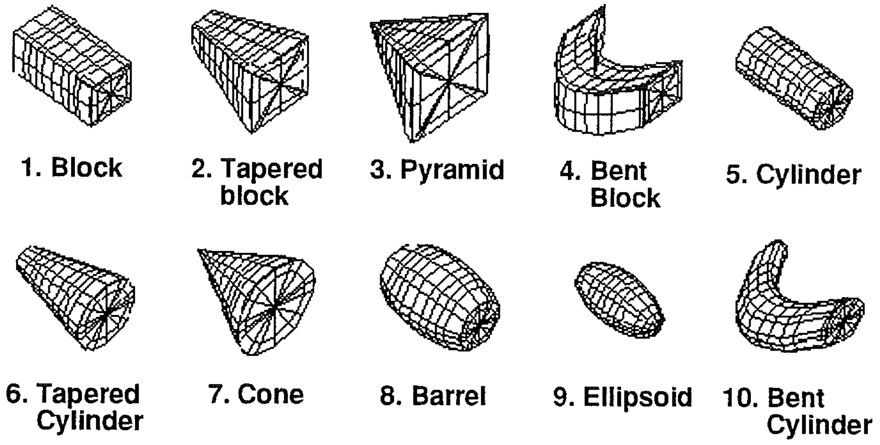

Figure 3: The Ten Modeling Primitives

then even slightly stretching or bending a volumetric part, for example, would give rise to a new set of aspects. To ensure that our volumetric part classes are invariant to minor 3-D shape deformations, we make our aspects invariant to minor 2-D shape deformations. Thus, faces and boundary groups encode qualitative relationships (e.g., cotermination, parallelism, and symmetry) between qualitatively-defined contours (e.g., straight, convex, and concave), while aspects simply encode adjacencies between labeled faces.

For the experiments reported in this paper, we have selected a set of ten volumetric part classes, illustrated in Figure 3, while Figure 4 illustrates a portion of the corresponding aspect hierarchy. To construct objects, the primitives are attached to one another with the restriction that any junction of two primitives involves exactly one distinct surface from each primitive.

In an unexpected, or bottom-up, recognition framework, the aspect hierarchy is used to recover faces, aspects, and finally volumes from a region-segmented image, as shown in Figure 5, while in an expected, or top-down recognition framework, the aspect hierarchy is used to direct a Bayesian search strategy mapping target objects to target faces in the image, as shown in Figure 6. Details of these strategies will not be presented here, and can be found in [16, 15, 9, 10].

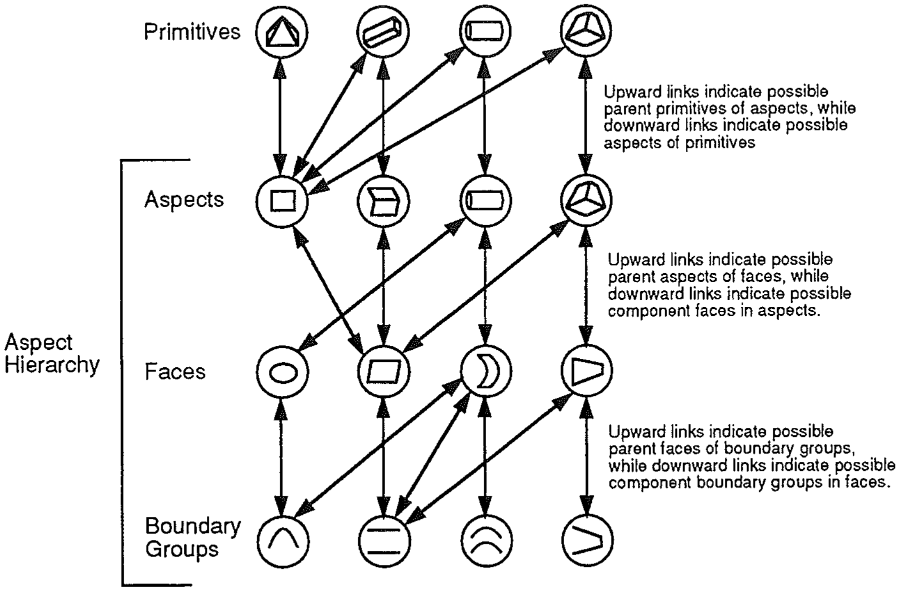
Primitives
Aspects
Faces
Boundary Groups
Aspect Hierarchy
Upward links indicate possible parent primitives of aspects, while downward links indicate possible aspects of primitives
Upward links indicate possible parent aspects of faces, while downward links indicate possible component faces in aspects.
Upward links indicate possible parent faces of boundary groups, while downward links indicate possible component boundary groups in faces.

Figure 4: The Aspect Hierarchy

3 Shape Recovery from a 2-D Image

In [16, 15, 9, 10], we outlined techniques for recovering and recognizing 3-D objects from a single 2-D image. Although the technique segments the scene into a set of qualitatively-defined parts, no metric information is recovered for the parts nor is the 3-D position and orientation of the parts recovered. For problems such as subclass recognition, where finer shape distinctions are necessary, and grasping, where accurate localization is critical for gripper placement, these qualitative recognition strategies do not recover sufficient metric shape information.

In this section, we describe a technique whereby the recovered qualitative shape is used to constrain the physics-based recovery of a deformable quantitative model from the recovered image contours. As shown in Figure 7, distances between a recovered aspect and a projected model aspect are converted to 2-D image forces. These forces, in turn, are mapped to a set of generalized forces which deform the model and bring its projection into alignment with the recovered aspect. The technique: 1) ensures that only data used to infer object shape will exert forces on the model; 2) is not sensitive to model initialization; 3) is able to recover shapes with surface discontinuities; and 3) uses qualitative shape knowledge to constrain shape recovery. Details of the algorithm can be found in [21, 12].

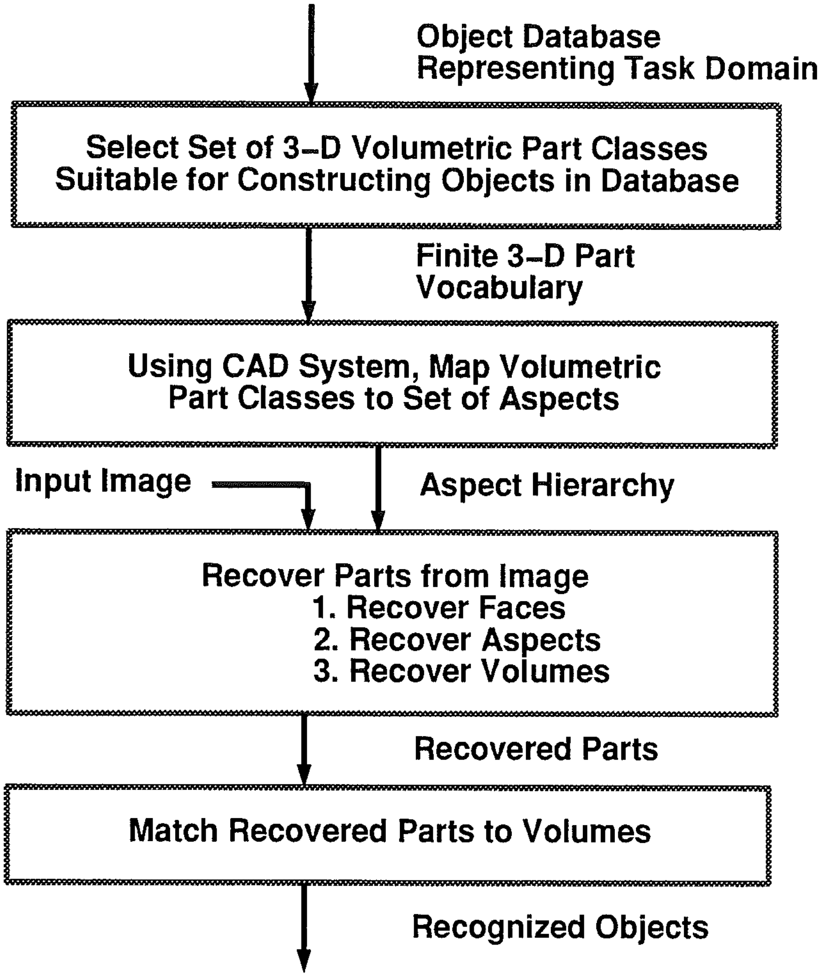

Figure 5: Unexpected Object Recognition

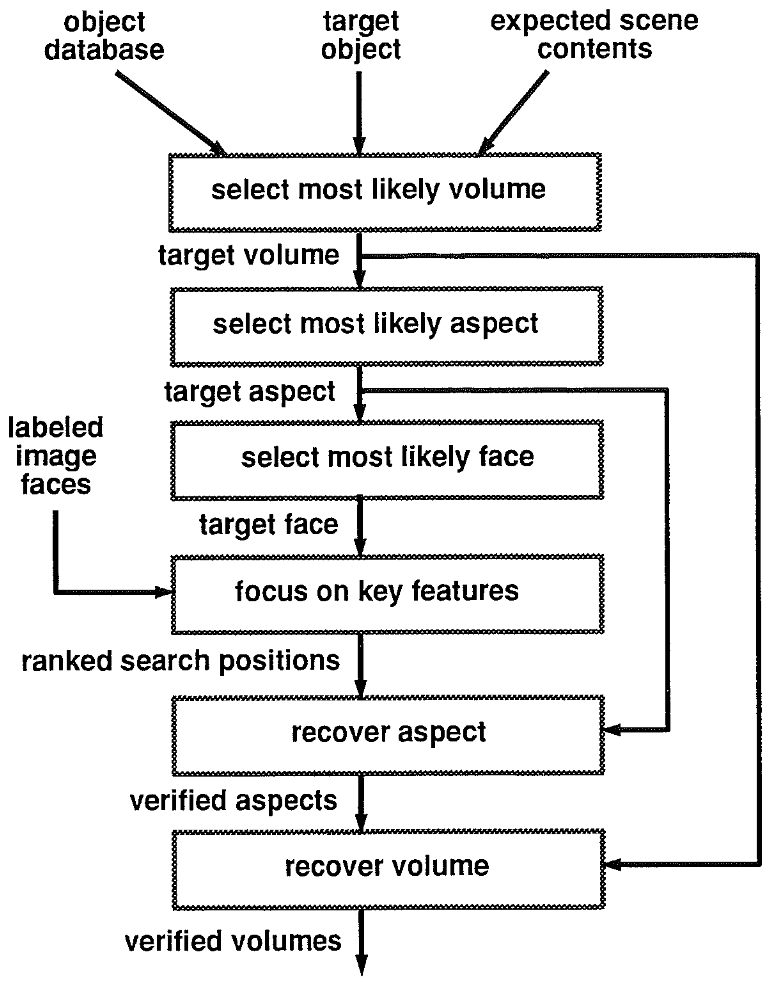

Figure 6: Expected Object Recognition

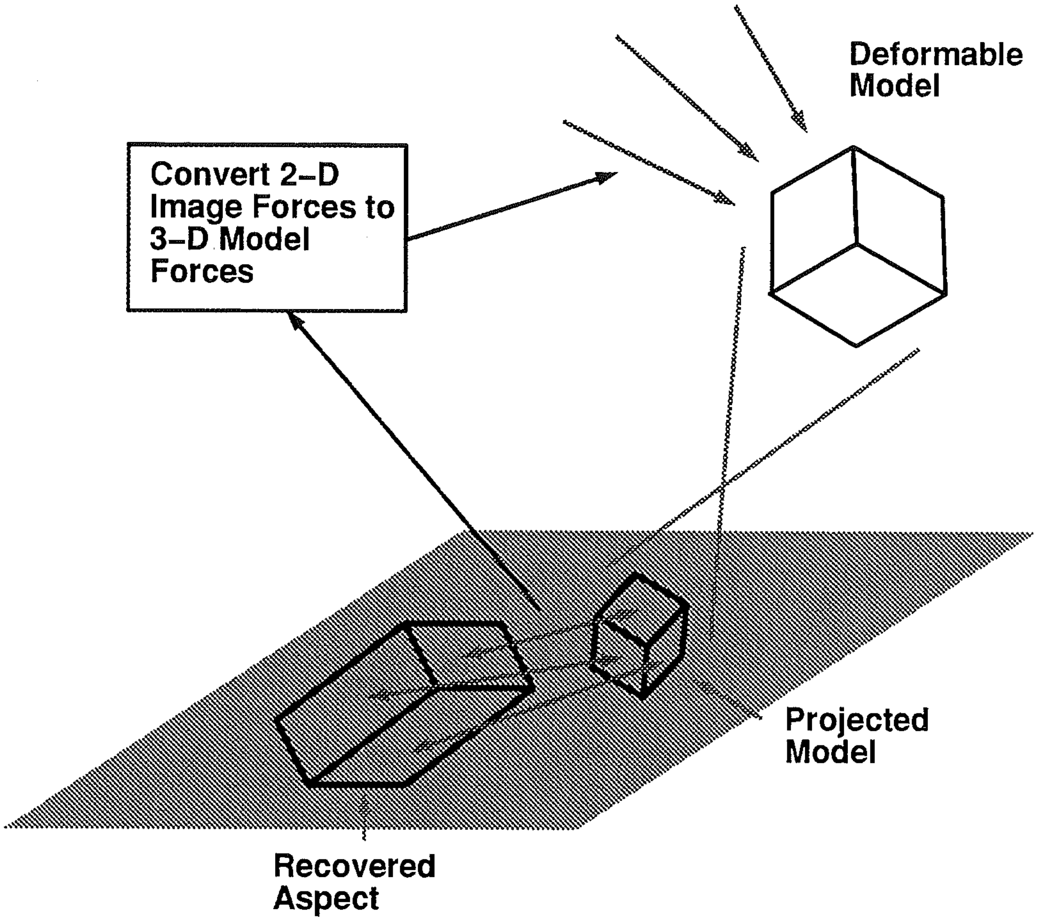

Figure 7: Using Qualitative Shape to Constrain Physics-Based Deformable Shape Recovery

3.1 The Geometry of the Deformable Model

Geometrically, the quantitative shape models that we recover are closed surfaces in space whose intrinsic (material) coordinates are $\mathrm{u} = (u, v)$, defined on a domain Ω [25, 12]. The positions of points on the model relative to an inertial frame of reference Φ in space are given by a vector-valued, time-varying function of u:

$$\mathbf{x}(\mathrm{u}, t) = (x_1(\mathrm{u}, t), x_2(\mathrm{u}, t), x_3(\mathrm{u}, t))^\top \tag{1}$$

where $^\top$ is the transpose operator. We set up a noninertial, model-centered reference frame ϕ [20], and express these positions as:

$$\mathbf{x} = \mathbf{c} + \mathbf{R}\mathbf{p}, \tag{2}$$

where $\mathbf{c}(t)$ is the origin of ϕ at the center of the model, and the orientation of ϕ is given by the rotation matrix $\mathbf{R}(t)$. Thus, $\mathbf{p}(\mathrm{u}, t)$ denotes the canonical positions of points on the model relative to the model frame. We further express $\mathbf{p}$ as the sum of a reference shape $\mathrm{s}(\mathrm{u}, t)$ (global deformation) and a displacement function $\mathbf{d}(\mathrm{u}, t)$ (local deformation):

$$\mathbf{p} = \mathrm{s} + \mathbf{d}. \tag{3}$$

We define the global reference shape as

$$\mathrm{s} = \mathbf{T}(\mathbf{e}(\mathrm{u}; a_0, a_1, \ldots); b_0, b_1, \ldots). \tag{4}$$

Here, a geometric primitive $\mathbf{e}$, defined parametrically in u and parameterized by the variables a_i, is subjected to the *global deformation* $\mathbf{T}$ which depends on the parameters b_i. Although generally nonlinear, $\mathbf{e}$ and $\mathbf{T}$ are assumed to be differentiable (so that we may compute the Jacobian of s) and $\mathbf{T}$ may be a composite sequence of primitive deformation functions $\mathbf{T}(\mathrm{e}) = \mathrm{T}_1(\mathrm{T}_2(\ldots \mathrm{T}_n(\mathrm{e})))$. We concatenate the global deformation parameters into the vector

$$\mathbf{q}_s = (a_0, a_1, \ldots, b_0, b_1, \ldots)^\top. \tag{5}$$

Even though our technique for defining $\mathbf{T}$ is independent of the primitive $\mathbf{e} = (e_1, e_2, e_3)^\top$ to which it is applied, we will use superquadric ellipsoid primitives due to their suitability in vision applications.

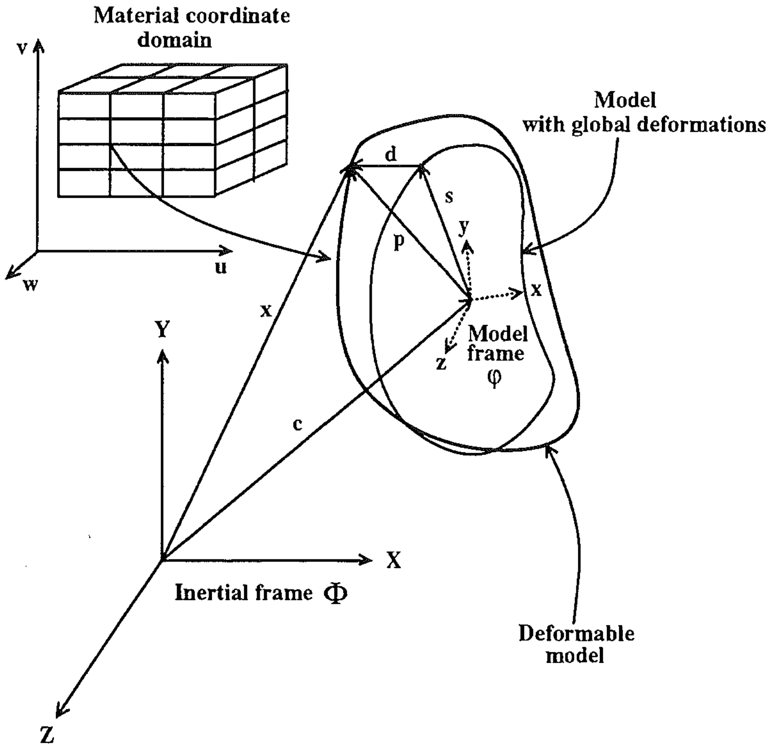

Figure 8: Geometry of Deformable Models

We first consider the case of superquadric ellipsoids [1], which are given by the following formula:

$$\mathbf{e} = a \begin{pmatrix} a_1 {C_u}^{\epsilon_1} {C_v}^{\epsilon_2} \\ a_2 {C_u}^{\epsilon_1} {S_v}^{\epsilon_2} \\ a_3 {S_u}^{\epsilon_1} \end{pmatrix}, \tag{6}$$

where $-\pi/2 \leq u \leq \pi/2$ and $-\pi \leq v < \pi$, and where ${S_w}^{\epsilon} = \mathrm{sgn}(\sin w)|\sin w|^{\epsilon}$ and ${C_w}^{\epsilon} = \mathrm{sgn}(\cos w)|\cos w|^{\epsilon}$, respectively. Here, $a \geq 0$ is a scale parameter, $0 \leq a_1, a_2, a_3 \leq 1$ are aspect ratio parameters, and $\epsilon_1, \epsilon_2 \geq 0$ are "squareness" parameters.

We then combine linear tapering along principal axes 1 and 2, and bending along principal axis 3 of the superquadric $\mathbf{e}$[1] into a single parameterized deformation $\mathbf{T}$, and express the

[1] These coincide with the model frame axes x, y and z respectively.

reference shape as:

$$\mathbf{s} = \mathbf{T}(\mathbf{e}, t_1, t_2, b_1, b_2, b_3) = \left(\begin{array}{l} (\frac{t_1 e_3}{a a_3 w} + 1)\ e_1 + b_1\ cos(\frac{e_3 + b_2}{a a_3 w} \pi b_3) \\ (\frac{t_2 e_3}{a a_3 w} + 1)\ e_2 \\ e_3 \end{array} \right), \tag{7}$$

where $-1 \leq t_1, t_2 \leq 1$ are the tapering parameters in principal axes 1 and 2, respectively; b_1 defines the magnitude of the bending and can be positive or negative; $-1 \leq b_2 \leq 1$ defines the location on axis 3 where bending is applied; and $0 < b_3 \leq 1$ defines the region of influence of bending. Our method for incorporating global deformations is not restricted to only tapering and bending deformations. Any other deformation that can be expressed as a continuous parameterized function can be incorporated in our global deformation in a similar way.

We collect the parameters in s into the parameter vector:

$$\mathbf{q}_s = (a, a_1, a_2, a_3, \epsilon_1, \epsilon_2, t_1, t_2, b_1, b_2, b_3)^\top. \tag{8}$$

The above global deformation parameters are adequate for quantitatively describing the ten modeling primitives shown in Figure 3. In the following section, we describe how these global deformation parameters, describing a volume's quantitative shape, are recovered from an image. In cases where local deformations **d** are necessary to capture object shape details, we use the finite element theory and express the local deformations as

$$\mathbf{d} = \mathbf{S}\mathbf{q}_d, \tag{9}$$

where **S** is the shape matrix whose entries are the finite element shape functions, and $\mathbf{q}_d$ are the model's nodal local displacements [20].

3.2 Simplified Numerical Simulation

When fitting the quantitative model to visual data, our goal is to recover $\mathbf{q} = (\mathbf{q}_c^\top, \mathbf{q}_\theta^\top, \mathbf{q}_s^\top, \mathbf{q}_d^\top)^\top$, the vector of degrees of freedom of the model. The components $\mathbf{q}_c$, $\mathbf{q}_\theta$, $\mathbf{q}_s$, and $\mathbf{q}_d$, are the translational, rotational, global deformation, and local deformation degrees of freedom, respectively. Our approach carries out the coordinate fitting procedure in a physics-based way. We make our model dynamic in **q** by introducing mass, damping, and a deformation strain energy. This allows us, through the apparatus of Lagrangian dynamics, to arrive at a set of equations of motion governing the behavior of our model under the action of externally applied forces.

The Lagrange equations of motion take the form [25]:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{g}_q + \mathbf{f}_q, \tag{10}$$

where $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ are the mass, damping, and stiffness matrices, respectively, where $\mathbf{g}_q$ are inertial (centrifugal and Coriolis) forces arising from the dynamic coupling between the local and global degrees of freedom, and where $\mathbf{f}_q(\mathrm{u}, t)$ are the generalized external forces associated with the degrees of freedom of the model. If it is necessary to estimate local deformations in (10), we tessellate the surface of the model into linear triangular elements.

For fast interactive response, we employ a first-order Euler method to integrate (10).[2] However, in fitting a model to static data, we simplify these equations by setting both $\mathbf{M}$ and $\mathbf{K}$ to zero, yielding a model which has no inertia and comes to rest as soon as all the applied forces vanish or equilibrate.

3.3 Applied Forces

In the dynamic model fitting process, the data are transformed into an externally applied force distribution $\mathbf{f}(\mathrm{u}, t)$. We convert the external forces to generalized forces $\mathbf{f}_q$ which act on the generalized coordinates of the model [25]. We apply forces to the model based on differences between the model's projected points and points on the recovered aspect's contours. Each of these forces is then converted to a generalized force $\mathbf{f}_q$ that, based on (10), modifies the appropriate generalized coordinate in the direction that brings the projected model closer to the data. The application of forces to the model proceeds in a face by face manner. Each recovered face in the aspect, in sequence, affects particular degrees of freedom of the model. In the case of occluded volumes, resulting in both occluded aspects and occluded faces, only those portions (boundary groups) of the regions used to infer the faces exert external global deformation forces on the model.

[2] In Section 6, we will see how Equation (10) is also used in object tracking.

3.4 Model Initialization

One of the major limitations of previous deformable model fitting approaches is their dependence on model initialization and prior segmentation [27, 25, 23]. Using our qualitative shape recovery process as a front end, we first segment the data into parts, and for each part, we identify the relevant non-occluded data belonging to the part [16, 15, 9, 10]. In addition, the extracted qualitative volumes explicitly define a mapping between the image faces in their projected aspects and the 3-D surfaces on the quantitative models. Moreover, the extracted volumes can be used to immediately constrain many of the global deformation parameters. For example, from the qualitative shape classes, we know if a volume is bent, tapered, or has an elliptical cross-section.

Although the initial model can be specified at any position and orientation, the aspect that a volume encodes defines a qualitative orientation that can be exploited to speed up the model fitting process. Sensitivity of the fitting process to model initialization is also overcome by independently solving for the degrees of freedom of the model. By allowing each face in an aspect to exert forces on only one model degree of freedom at a time, we remove local minima from the fitting process and ensure correct convergence of the model.

3.5 Examples

To illustrate the fitting stage, consider the contours belonging to the recovered tapered cylinder, shown in Figure 9. Having determined during the qualitative shape recovery stage that we are trying to fit a deformable superquadric to a tapered cylinder, we can immediately fix some of the parameters in the model. In addition, the qualitative shape recovery stage provides us with a mapping between faces in the image and physical surfaces on the model. For example, we know that the elliptical face maps to the top of the tapered cylinder, while the body face maps to the side of the tapered cylinder. For the case of the tapered cylinder, we will begin with a (superquadric) cylinder model and will compute the forces that will deform the cylinder into the tapered cylinder appearing in the image. Assuming that the x and y dimensions are equal, we compute the following forces:

1. The cylinder is initially oriented with its z axis orthogonal to the image plane. The first step involves computing the centroid of the elliptical image face (known to correspond to the top of the cylinder). The distance between the centroid and the projected center of the cylinder top is converted to a force which translates the model cylinder. Figure 9(a) shows the image contours corresponding to the lamp shade and the cylinder following application of this force. Figure 9(b) shows a different view of the image plane, providing a better view of the model cylinder.

2. The distance between the two image points corresponding to the extrema of the principal axis of the elliptical image face and two points that lie on a diameter of the top of the cylinder is converted to a force affecting the x and y dimensions with respect to the model cylinder. Figures 9(c) and 9(d) show the image and the cylinder following application of this force.

3. The distance between the projected model contour corresponding to the top of the cylinder and the elliptical image face corresponds to a force affecting the orientation of the cylinder. Figures 9(e) and 9(f) show the image and the cylinder following application of this force. This concludes the application of forces arising from the elliptical image face, i.e., top of the tapered cylinder.

4. Next, we focus on the image face corresponding to the body of the tapered cylinder to complete the fitting process. The distance between the points along the bottom rim of the body face and the projected bottom rim of the cylinder corresponds to a force affecting the length of the cylinder in the z direction. Figures 9(g) and 9(h) show the image and the cylinder following application of this force.

5. Finally, the distance between points on the sides of the body face and the sides of the cylinder corresponds to a force which tapers the cylinder to complete the fit. Figures 9(i) and 9(j) show the image and the tapered cylinder following application of this force.

As shown in the above example, the recovered aspect plays a critical role in constraining the fitting process. tracking.

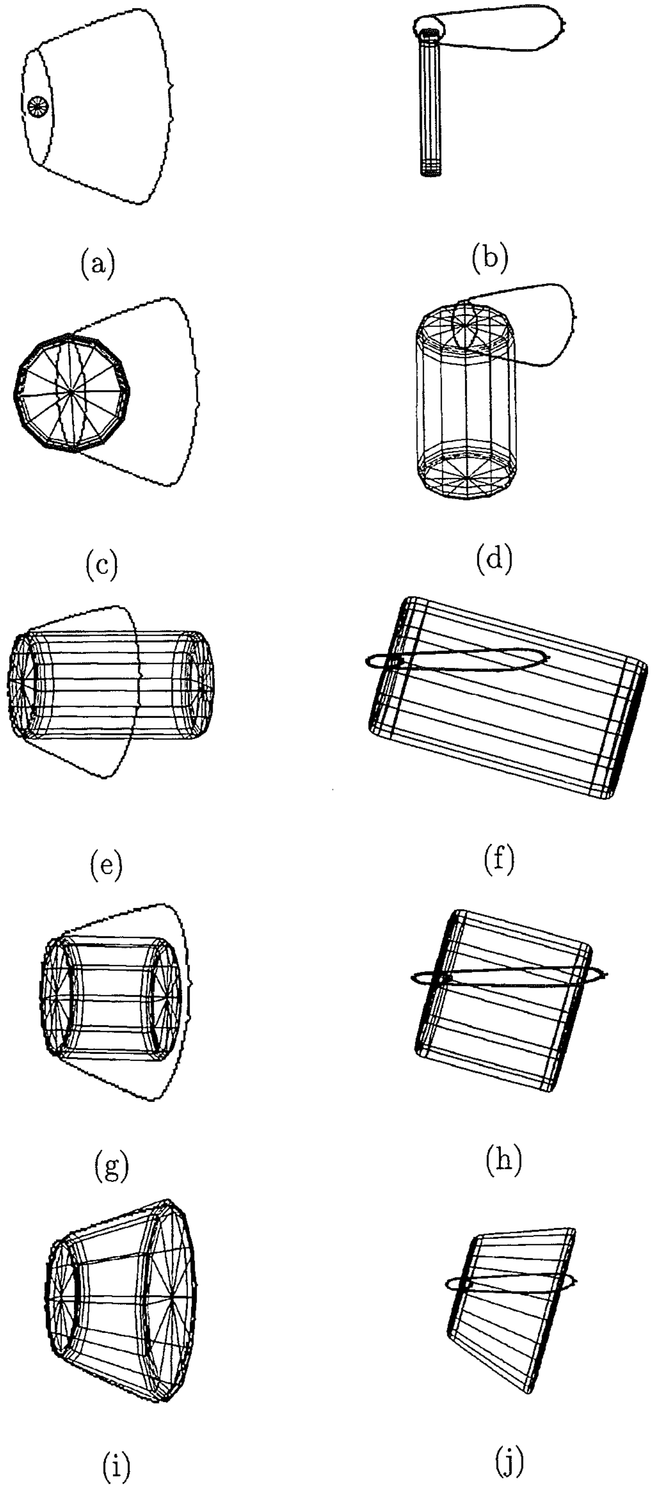

Figure 9: Quantitative Shape Recovery for Lamp Shade

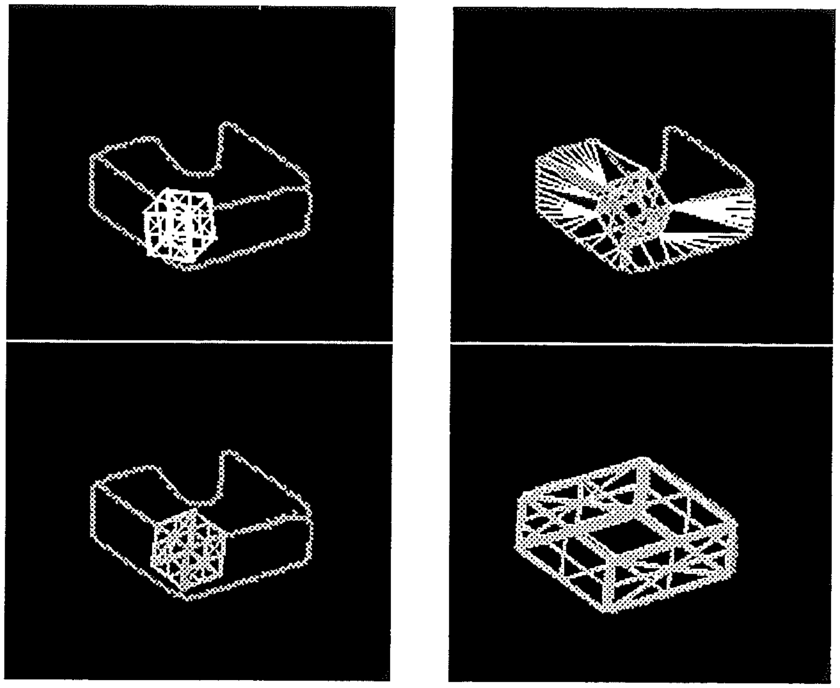

Figure 10: Sequence of steps in the recovery of a deformable superquadric block from a recovered occluded aspect of a block.

In Figure 10, we show a set of snapshots extracted from the recovery of a block volume from a partial aspect recovered from an image of a block. Although one of the faces (top face) has been corrupted due to both shadow and occlusion, only those portions of its bounding contour that were used to match the recovered aspect's component face actually exert forces on the deformable model. Only by encoding qualitative shape information in the models (aspects) can we decide which contours belong to the object we are trying to recover.

4 Shape Recovery from a 3-D Image

The aspect hierarchy was originally introduced as a representation to support 3-D object recognition from 2-D images. By having faces in the aspect hierarchy represent 3-D surfaces instead of 2-D projections of 3-D surfaces, the aspect hierarchy can now be used to constrain the recovery and recognition of 3-D objects from range data. Furthermore, by adding face attributes such as mean and Gaussian curvature, we can effectively prune many of the mappings from boundary groups to faces, faces to aspects, and aspects to volumes. We call the new aspect hierarchy, the *range aspect hierarchy* [13].