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.

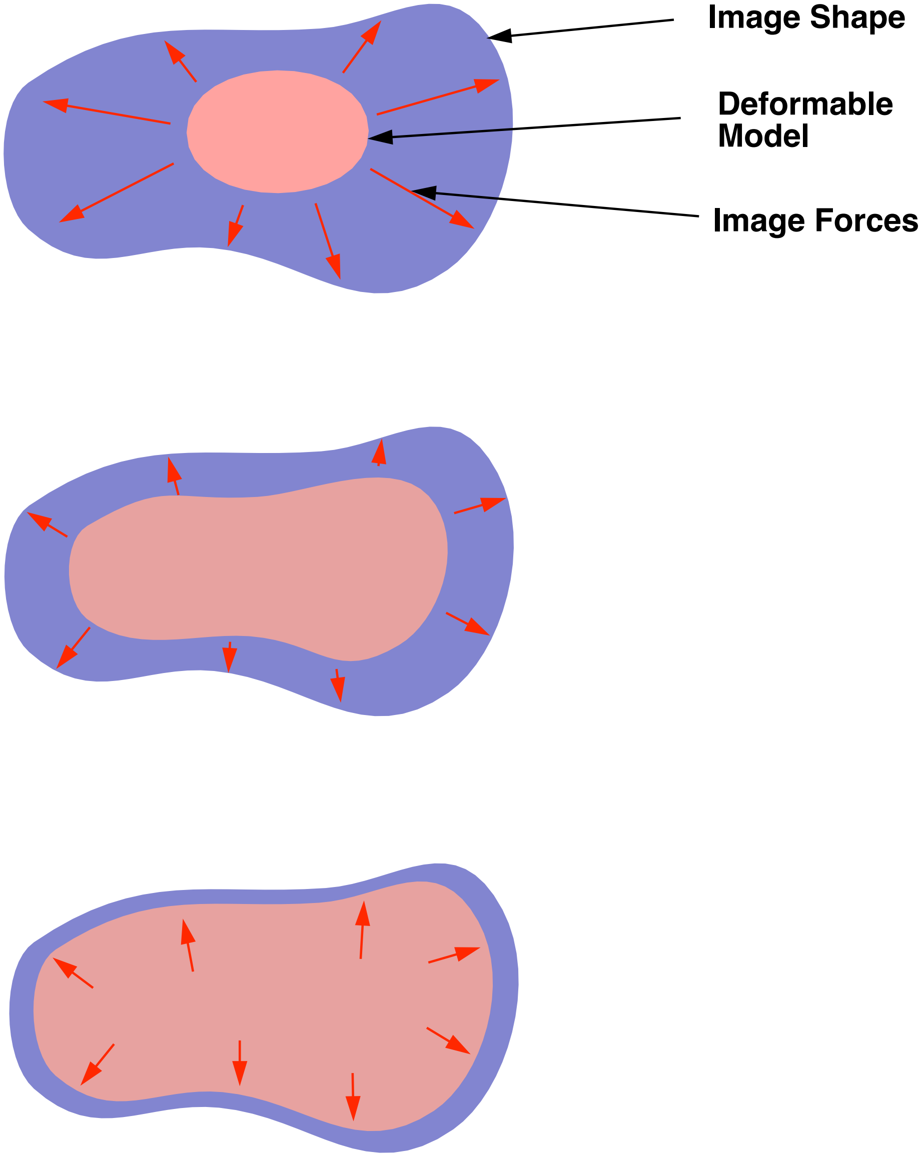


Figure 1: Data-Driven Shape Recovery

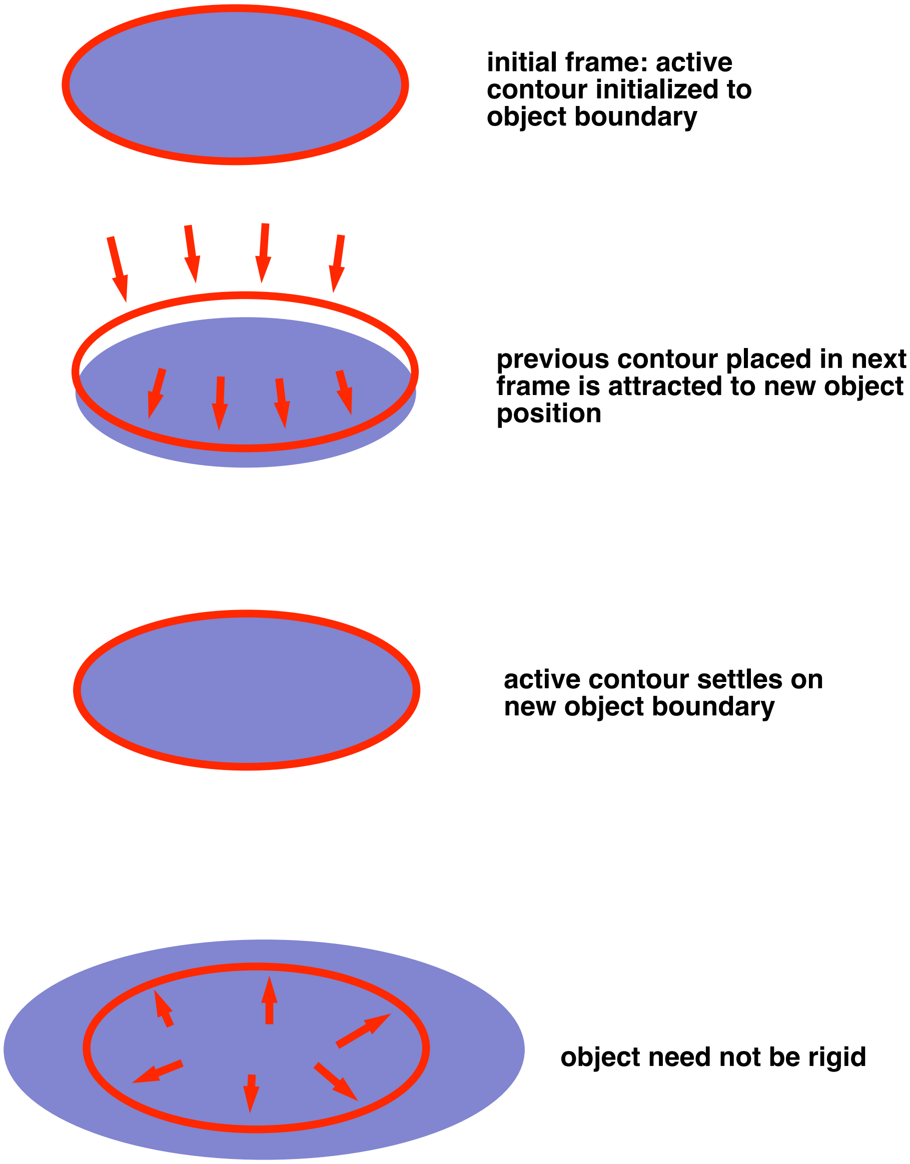

initial frame: active contour initialized to object boundary
previous contour placed in next frame is attracted to new object position
active contour settles on new object boundary
object need not be rigid

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.,

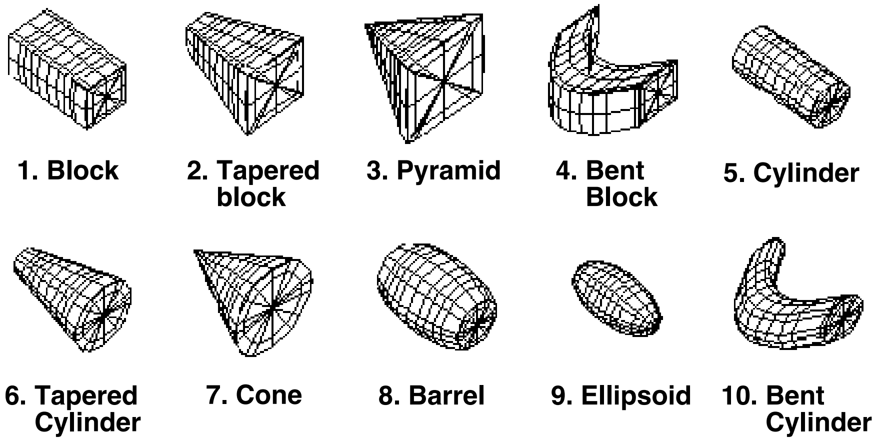


Figure 3: The Ten Modeling Primitives

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].

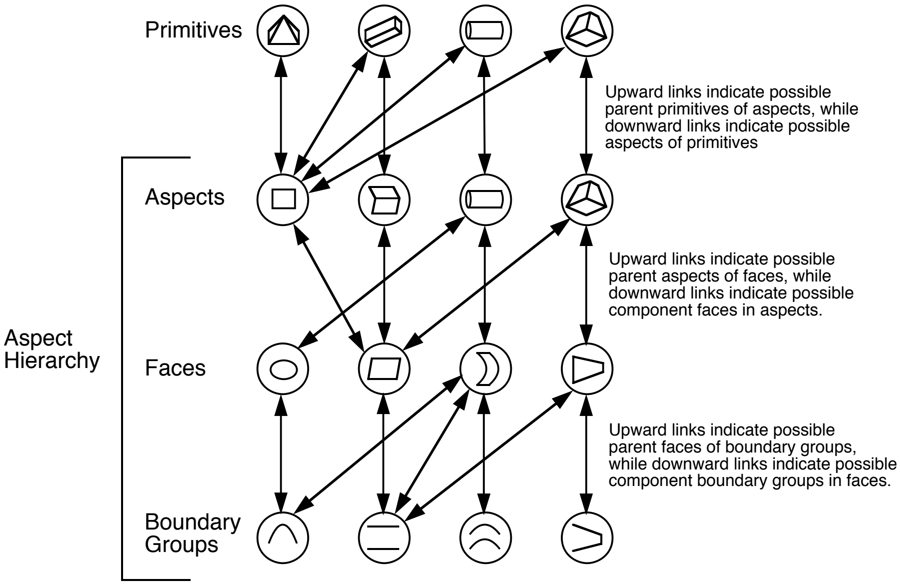

Primitives
Aspects
Faces
Boundary Groups
Aspect Hierarchy
Upward links indicate possible parent primitives of aspects, while downward links indicate possible aspects of primitives
Upward links indicate possible parent aspects of faces, while downward links indicate possible component faces in aspects.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

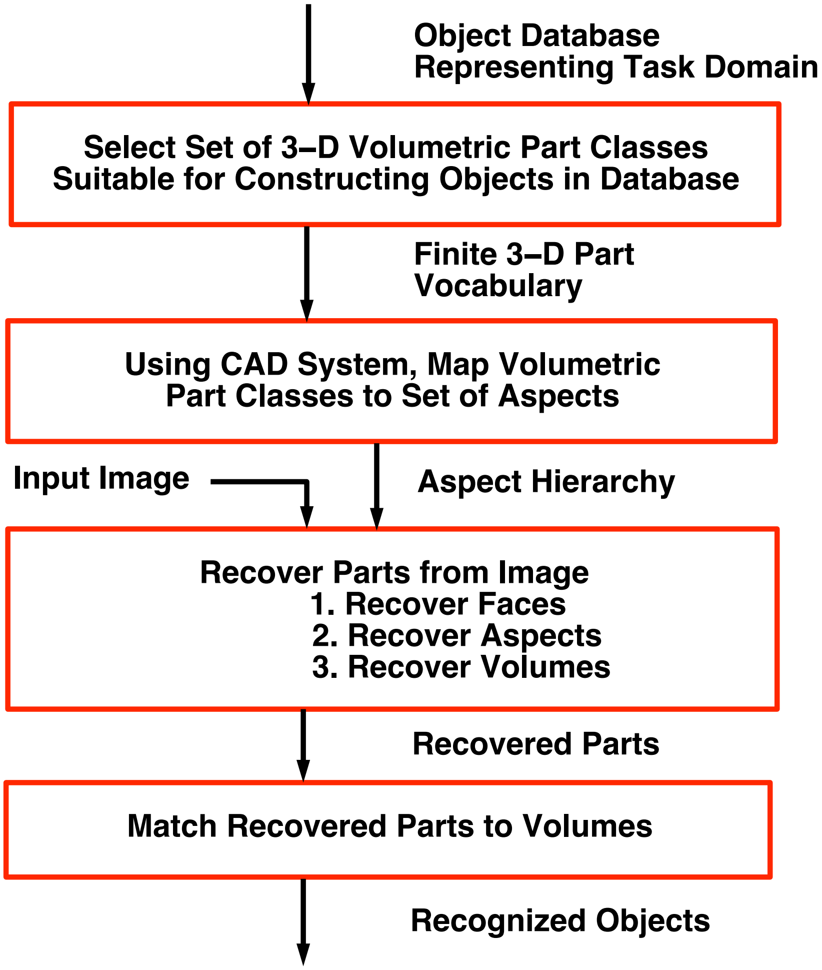


Figure 5: Unexpected Object Recognition

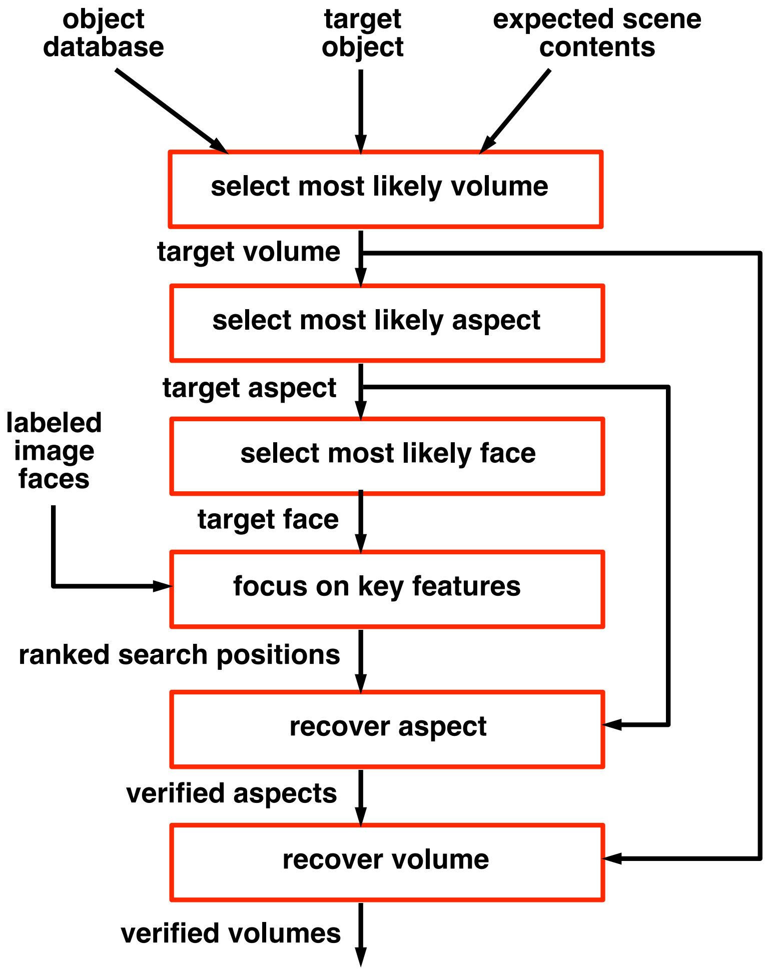


Figure 6: Expected Object Recognition

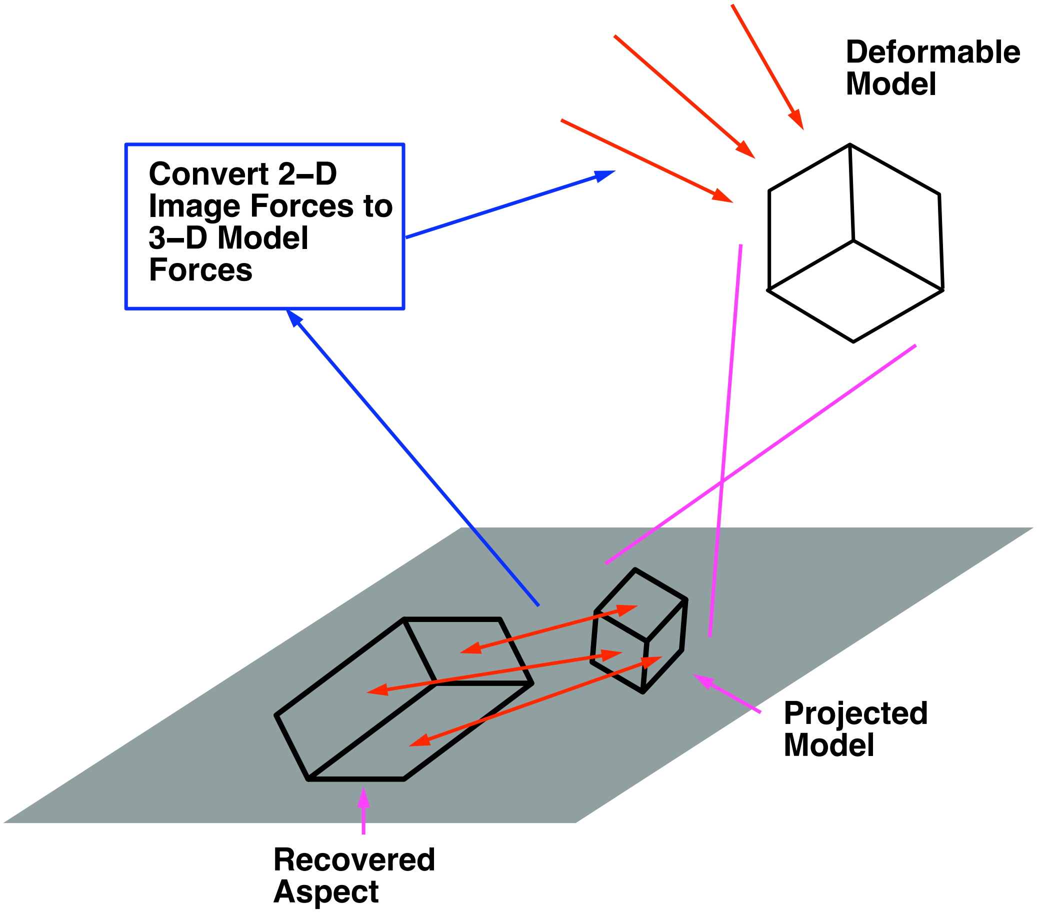


Figure 7: Using Qualitative Shape to Constrain Physics-Based Deformable Shape Recovery

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 $\mathbf{s}(\mathrm{u}, t)$ (global deformation) and a displacement function $\mathbf{d}(\mathrm{u}, t)$ (local deformation):

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

We define the global reference shape as

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

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 $\mathbf{s}$) and $\mathbf{T}$ may be a composite sequence of primitive deformation functions $\mathbf{T}(\mathbf{e}) = \mathbf{T}_1(\mathbf{T}_2(\ldots \mathbf{T}_n(\mathbf{e})))$. 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.

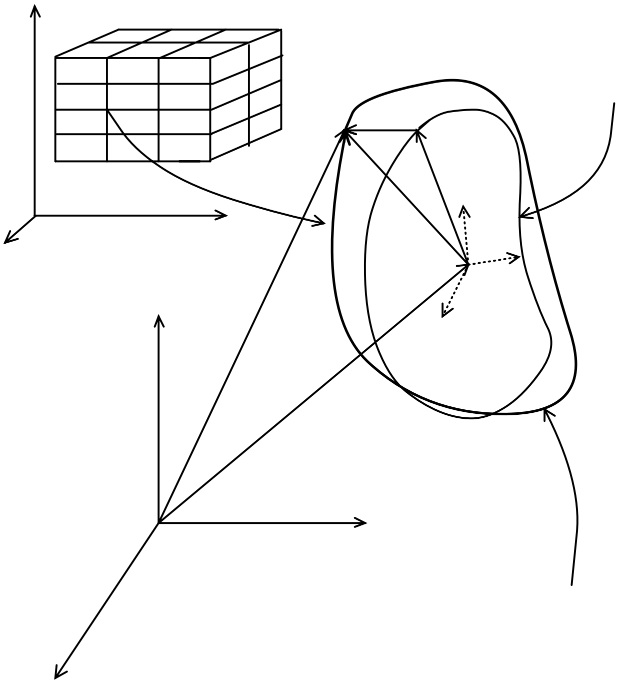


Figure 8: Geometry of Deformable Models

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 ${S_w}^{\epsilon} = \operatorname{sgn}(\sin w)|\sin w|^{\epsilon}$ and ${C_w}^{\epsilon} = \operatorname{sgn}(\cos w)|\cos w|^{\epsilon}$, 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:

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

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 $\mathbf{s}$ 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 $\mathbf{d}$ 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 $\mathbf{S}$ 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 $\mathbf{q}$ 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.

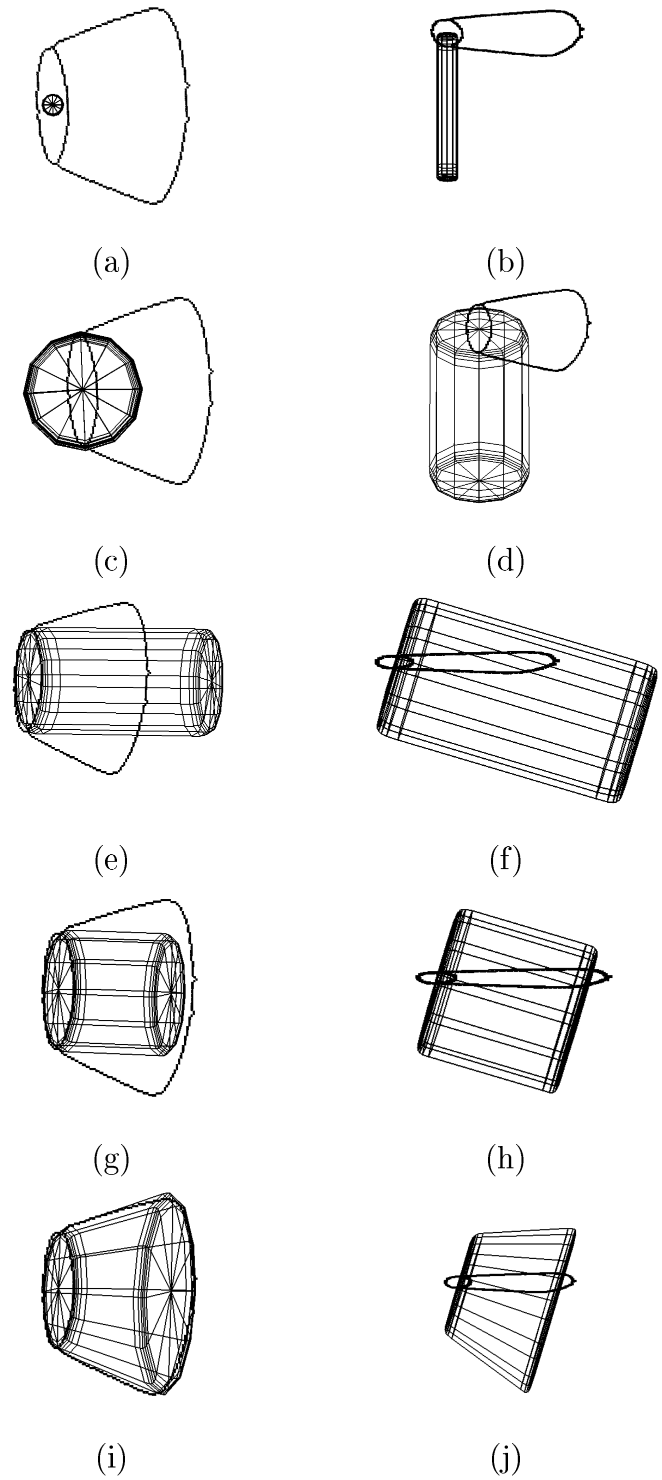
2. The distance between the two image points corresponding to the extrema of the principal axis of the elliptical image face and two points that lie on a diameter of the top of the cylinder is converted to a force affecting the x and y dimensions with respect to the model cylinder. Figures 9(c) and 9(d)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



(a) (b)

(c) (d)

(e) (f)

(g) (h)

(i) (j)

Figure 9: Quantitative Shape Recovery for Lamp Shade

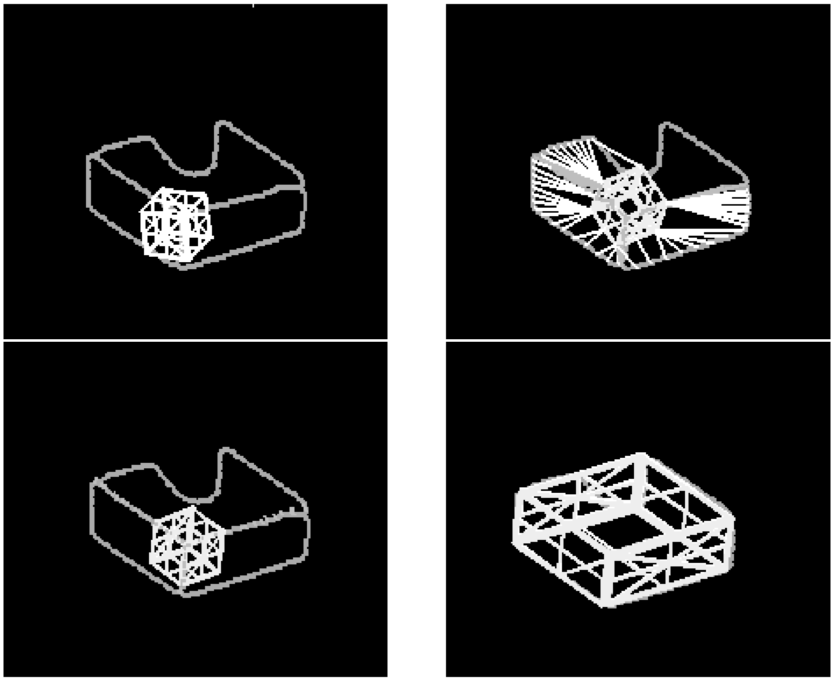


Figure 10: Sequence of steps in the recovery of a deformable superquadric block from a recovered occluded aspect of a block.

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].

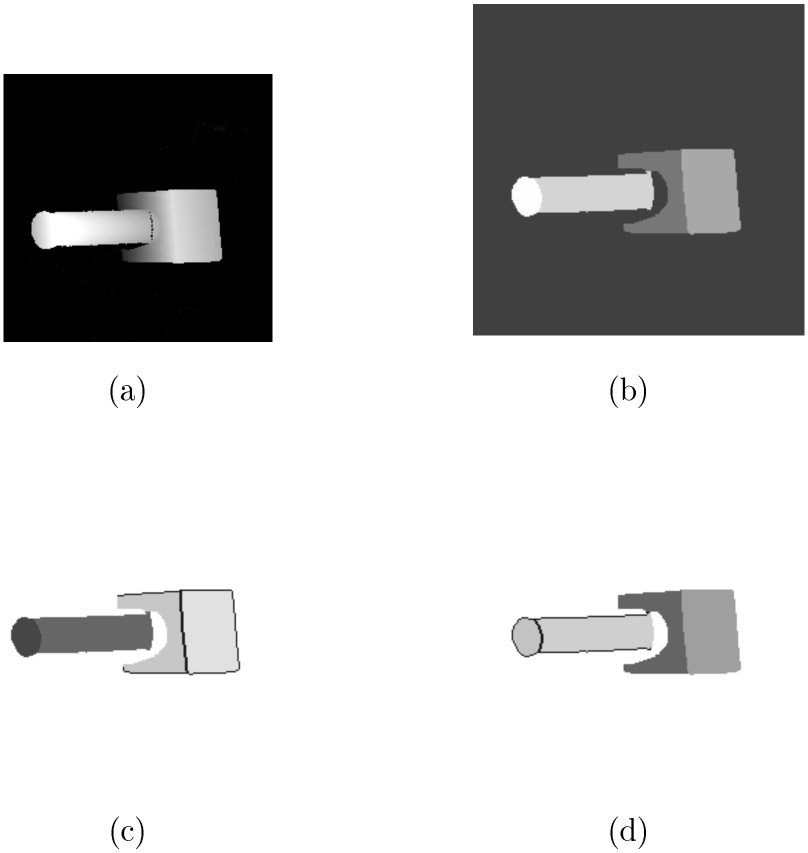


Figure 11: Qualitative Shape Recovery: (a) range image, (b) region segmentation, (c) recovered qualitative block, and (d) recovered qualitative cylinder

We demonstrate our approach by recovering the two parts in the range image shown in Figure 11(a)[3]; the region segmented image is shown in Figure 11(b). For this example, we invoked the expected object recognition mode to first search for the best instance of a block. Figure 11(c) shows the highlighted aspect recovered for the block; only those contours used to infer the block are highlighted in the image. Note that the most probable aspect for the block (containing three faces) was not recovered; however, the next most probably aspect (containing two faces) was recovered and used to locate the block. Figure 11(d) shows the highlighted aspect recovered for the cylinder.

For each of the two recovered qualitative volumes, we now proceed to show the results of using the recovered qualitative shape to constrain the fitting of a deformable model to

[3] The image was captured using a Technical Arts Scanner at the Michigan State University's PRIP Laboratory.

the original range data. In Figures 12(a-c), we show the initial, an intermediate, and the final frame in the sequence of model deformations taking the initial model to its final shape describing the block. For clarity, only the recovered contours belonging to the block's recovered aspect are shown in the sequence. Next, in Figures 12(d-g), we show the initial, two intermediate, and the final frame in the sequence of model deformations taking the initial model to its final shape describing the cylinder. Finally, in Figure 12(h), we show the two fitted parts together from a different viewpoint.

The fitting constraints provided by the recovered qualitative shape allows the deformable model fitting process to be invariant to initial position, orientation, and shape of the model. Unfortunately, relying on the correspondence between recovered image faces and model surfaces means that the recovery process is sensitive to region segmentation errors. However, by only allowing high-scoring volumes to constrain the fitting process, the chances of letting any region segmentation problems affect the fitting process is low. In fact, low-scoring volumes can be used to guide the sensor to acquire a higher-scoring volume [10].

5 Qualitative Shape Tracking

There are two ways in which an object can be tracked. If we have identified the object in the image from the qualitative shapes of its parts, we would like to be able to qualitatively track the object as it moves, for example, from "front" to "side" to "back" without knowing the exact geometry of the object. Alternatively, if we have used the recovered qualitative shape to recover the exact geometry of the object, then we would also like to be able to track the object's exact position and orientation, and possibly its shape if it is non-rigid.

In this section, we describe our approach to qualitative shape tracking, while in the next section, we address the problem of quantitative shape tracking. Our approach to qualitative object tracking, as shown in Figure 13, combines a symbolic tracker and an image tracker [11]. Just as we used a qualitative shape model to govern a data-driven shape recovery process, we will use the same qualitative shape model to govern a data-driven shape tracking process.

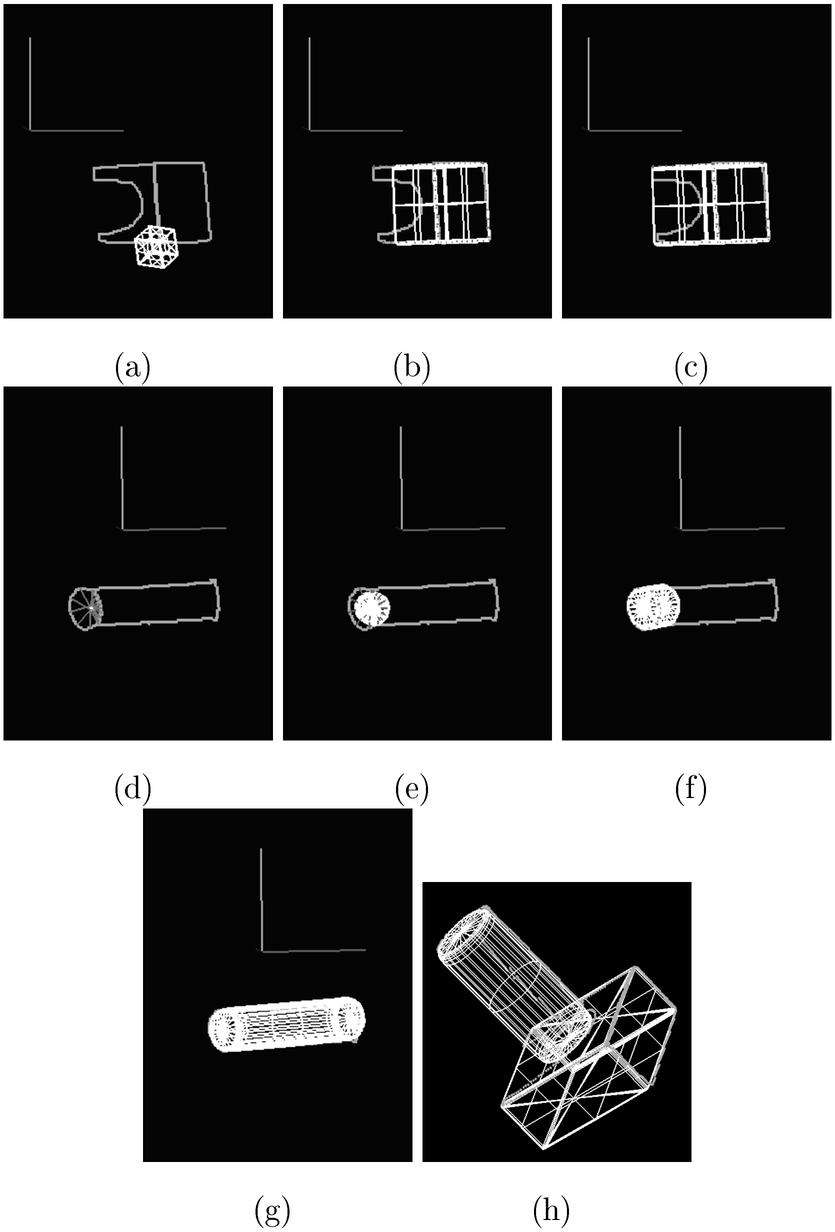


Figure 12: Selected Frames from the Part Fitting Sequence: (a) Initial Block Frame; (b) Intermediate Block Frame; (c) Final Block Frame; (d) Initial Cylinder Frame; (e) Intermediate Cylinder Frame; (f) Second Intermediate Cylinder Frame; (g) Final Cylinder Frame; (h) Recovered Object from a Different Viewpoint.

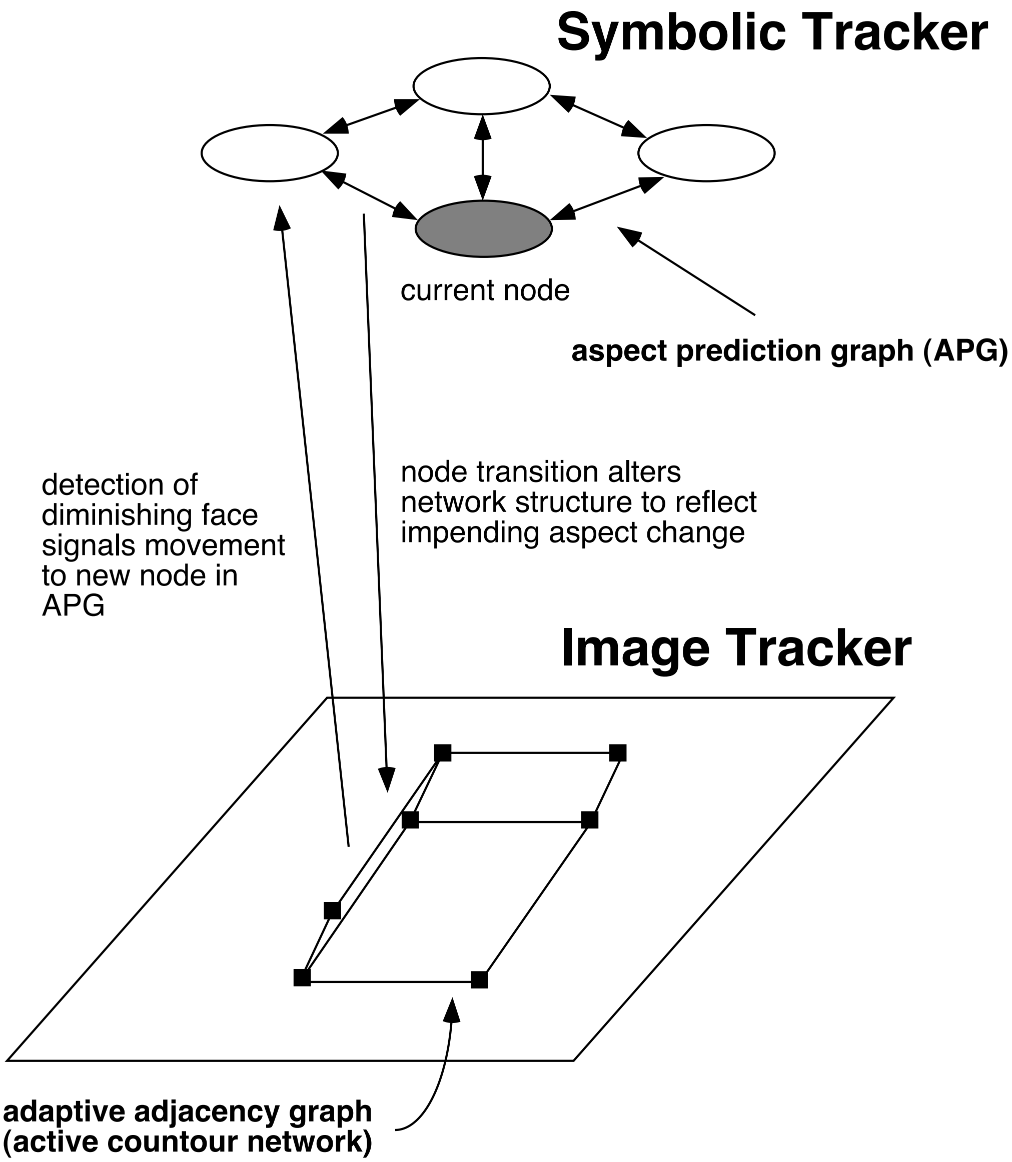

Symbolic Tracker
current node
aspect prediction graph (APG)
detection of diminishing face signals movement to new node in APG
node transition alters network structure to reflect impending aspect change
Image Tracker
adaptive adjacency graph (active countour network)

Figure 13: Qualitative Object Tracking

5.1 Image Tracker

The image tracker employs a representation called an adaptive adjacency graph, or AAG. The AAG is initially created from the contours belonging to a recovered aspect, and consists of a network of active contours (snakes) [18]. The active contours are controlled by internal and external forces and change their shape and location so as to minimize their total energy defined as:

$$E_{contour} = \int_0^1 (E_{int}(v(s))ds + E_{image}(v(s))ds + E_{con}(v(s))ds) \tag{11}$$

$$v(s) = \text{a parametric representation of the contour}$$

E_{int} represents internal energy corresponding to effects of contour bending or spanning discontinuities, and represents membrane and thin plate behaviors of the contour. External energy consists of two terms, E_{image} and E_{con}. The first term represents forces exerted by the image, while the second represents external constraints. E_{con} represents external energy imposed by a user in the form of repelling forces or constraints attaching a point of the snake to a particular location in the image or to a point on another snake. Desired effects of snake dynamics are achieved by selecting the relative weights of factors controlling the snake.

E_{image}, or the force image, is created by a sequence of morphological smoothing and gradient operators. The gradient image is thresholded and the result is blurred with a Gaussian. The standard deviation, s, of the Gaussian kernel determines the size of the zone of influence (potential well) around each image feature. Selection of s depends on the expected density of features in the image and the disparity between successive images.

The AAG encodes the topology of the network's regions, as defined by minimal cycles of contours. Contours in the AAG can deform subject to both internal and external (image) forces while retaining their connectivity at nodes. Connectivity of contours is achieved by imposing constraints (springs) between the contour endpoints. If an AAG detected in one image is placed on another image that is slightly out of registration, the AAG will be "pulled" into alignment using local image gradient forces.

The basic behavior of the AAG is to track image features while maintaining connectivity of the contours and preserving the topology of the graph. This behavior is maintained as long as the positions of active contours in consecutive images do not fall outside the zones of influence of tracked image features. This, in turn, depends on the number of active contours, the density of features in the image, and the disparity between successive images. If either the tracked object or the camera moves between successive frames, the observed scene may change due to disappearance of one of the object faces. The shape of the region corresponding to the disappearing face will change and eventually the size of the region will be reduced to zero. The image tracker monitors the sizes and shapes of all regions in the AAG and detects such events. When such an event is detected, a signal describing the event is sent to the symbolic tracker.

5.2 Symbolic Tracker

The symbolic tracker tracks movement from one node to another in a representation called the aspect prediction graph [10]. Each of the nodes in this representation, derived from an aspect graph [19] and the aspect hierarchy, represents a topologically different viewpoint of the object, while arcs between nodes specify the visual events or changes in image topology between nodes. The role of the symbolic tracker is to:

1. Determine which view or aspect of the object is currently visible (current node).

2. Respond to visual events detected by the image tracker by predicting which node (aspect) will appear next (target node).

3. From the visual event specification defined by the current and target nodes, add or delete structure from the active contour network (predictions).

4. If aspect predictions cannot be verified by the image tracker, or visual event predictions cannot be recognized by the symbolic tracker, the symbolic tracker must be able to bootstrap the system to relocate itself in the aspect prediction graph.

5.3 Visual Event Recognition

The symbolic tracker specifies the criteria for which a visual event will be detected by the image tracker. Currently, we use region area as the single event criteria. If at any time during the image tracking of an aspect, one or more of its faces' areas falls below some threshold, we interpret that to mean that the face is undergoing heavy foreshortening and will soon disappear. When a region's area drops below the threshold, the image tracker sends a signal to the symbolic tracker. Given its current position (node) in the aspect prediction graph, the symbolic tracker compares the outgoing arcs, or visual events, with the events detected by the image tracker. The arc in the aspect prediction graph matching the observed visual event defines a transition to a new aspect.

The transition between the current aspect and the predicted aspect defines a set of visual events in terms of the faces in the aspect defined by the current APG node. If one or more faces disappear from the current aspect to the predicted aspect, the symbolic tracker directs the image tracker to delete those contours from the adaptive adjacency graph which both belong to the disappearing faces and are not shared by any remaining faces. Alternatively, if one or more new faces are expected to appear, the symbolic tracker directs the image tracker to add structure to the adaptive adjacency graph. Since the symbolic tracker knows along which existing contours new faces should appear, it can specify between which nodes in the adaptive adjacency graph new contours should be added.

5.4 Example

In Figure 14, we demonstrate our tracking technique on a sequence of images taken of a rotating block. Note that for the first frame, the AAG was created from the recovered aspect. For subsequent frames, a blurred, thresholded, gradient image is used to exert external forces on the AAG. Moving left to right, top to bottom, we can follow the AAG as it tracks the image faces. When the foreshortened face's area falls below a threshold, the visual event is signaled to the symbolic tracker. Consequently, the nodes and contours belonging to the disappearing faces are removed while nodes and contours belonging to the face predicted to appear are added. Note that in order to ensure that new contours and old contours do not "lock on" to the same image gradient ridge, the contours are automatically "pulled apart", so that they will converge to the correct edges in the image. We are currently investigating the use of repulsion forces that would more effectively prevent network contours from converging.

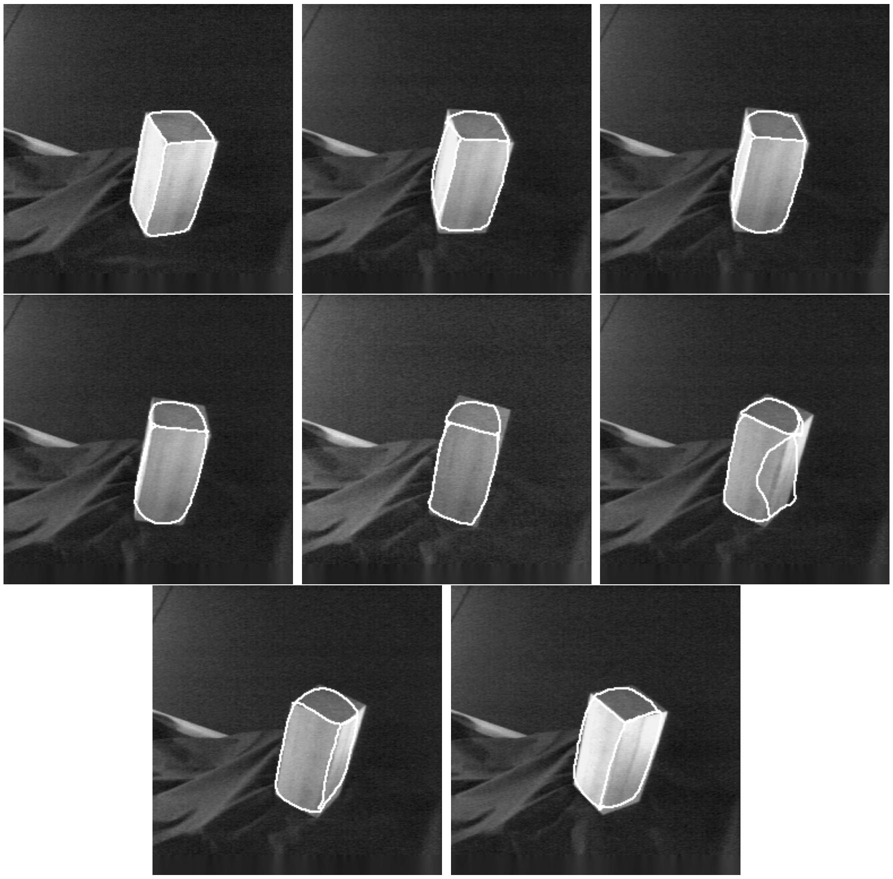


Figure 14: Tracking a Rotating Block. There were 11 images in the sequence with 10 iterations of the AAG per image, for a total of 110 snapshots of the AAG. Working left to right and top to bottom, we show snapshots 1, 23, 32, 41, 70, 82, 90, and 110. Note that when the disappearing face is detected (70), the new face is predicted and contours are added (82). The added contours are automatically "pulled apart" to ensure that they do not converge to the same image edge; final position of the new edge is shown in frame 90.

6 Quantitative Shape Tracking

Our approach to quantitative tracking [3, 4] makes use of our frameworks for qualitative and quantitative shape recovery described in previous sections, as well as a physics-based framework for quantitative motion estimation [22]. To be able to track multiple objects, initialization of the models is performed in the first frame of the sequence based on our quantitative shape recovery process. For successive frames, the qualitative shape recovery process can be avoided in favor of a physics-based model updating process requiring only a gradient computation in each frame, as shown in Figure 15. Assuming small deformations between frames, local forces derived from stereo images are sufficient to update the positions, orientations, and shapes of the models in 3-D; no costly feature extraction or correspondence is necessary.

6.1 Tracking and Prediction

Kalman filtering techniques have been applied in the vision literature for the estimation of dynamic features [7] and rigid motion parameters [17, 2] of objects from image sequences. We use a Kalman filter to estimate the object's shape and motion in a sequence of images. This allows us to predict where the object will appear in the image at some future time, thereby increasing the likelihood of the projected model falling within the local gradient field.

We incorporate a Kalman filter into our dynamic deformable model formulation by treating the model's Lagrangian equations of motion (10) as system models. Based on the use of the corresponding extended Kalman filter, we perform tracking by updating the model's generalized coordinates $\mathbf{q}$ according to the following equation:

$$\dot{\hat{\mathbf{u}}} = \mathbf{F}\hat{\mathbf{u}} + \mathbf{g} + \mathbf{P}\mathbf{H}^\top \mathbf{V}^{-1}(\mathbf{z} - \mathbf{h}(\hat{\mathbf{u}})), \qquad (12)$$

where $\mathbf{u} = (\dot{\mathbf{q}}^\top, \mathbf{q}^\top)^\top$ and matrices $\mathbf{F}, \mathbf{H}, \mathbf{g}, \mathbf{P}, \mathbf{V}$ are associated with the model dynamics, the error in the given data, and the measurement noise statistics [22]. Since we are measuring local short range forces directly from the image potential, the term $\mathbf{z} - \mathbf{h}(\hat{\mathbf{u}})$ represents the 2-D image forces. Using the above Kalman filter, we can predict at every step the expected

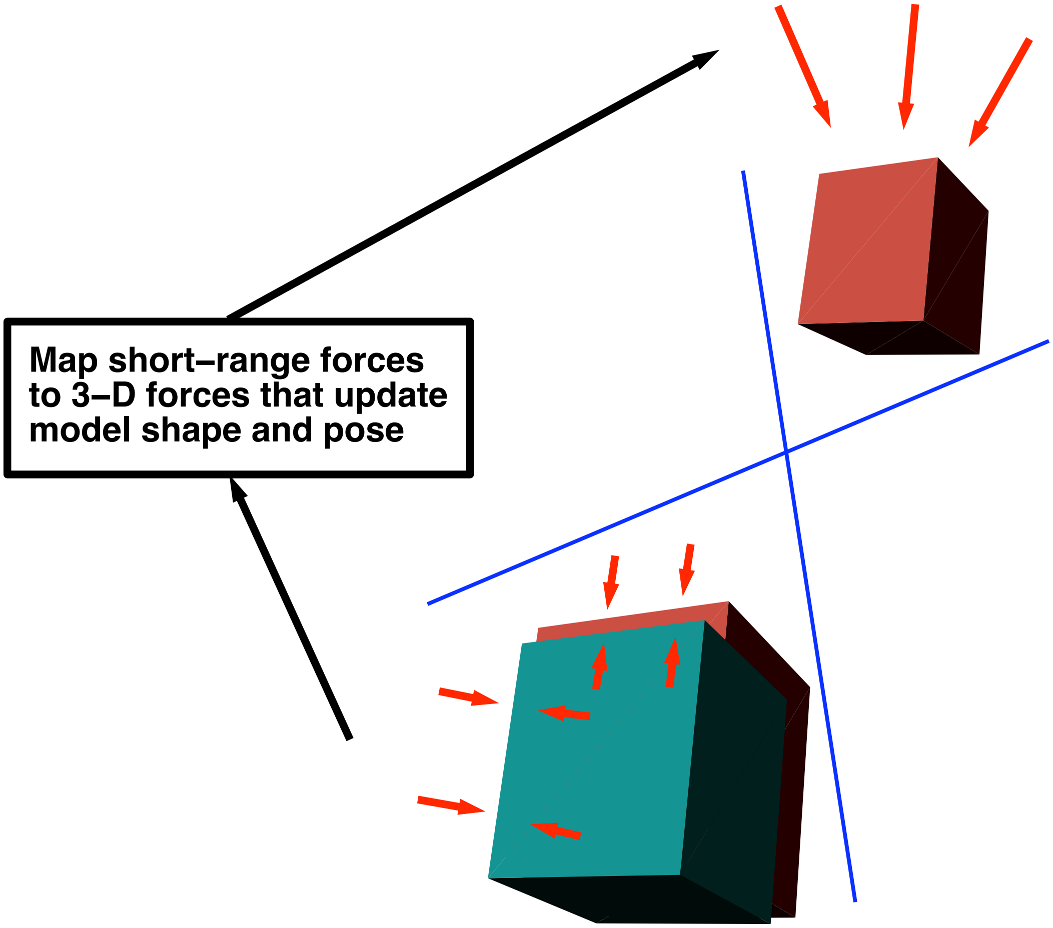

Map short–range forces to 3–D forces that update model shape and pose

Figure 15: Quantitative Object Tracking

location of the data in the next image frame, based on the magnitude of the estimated parameter derivatives $\dot{\mathbf{q}}$.

6.2 Computing Forces on the Model

Only those nodes on the model surface that are visible should respond to image forces. A modal node is made active if: 1) it lies on the occluding contour of the model from that viewpoint [27], or 2) the local surface curvature at the node is sufficiently large and the node is visible. Visibility of the nodes can be determined in two ways. Since we have a 3-D model, we can easily test the visibility of each node on the model, turning off those nodes that are self-occluded. A more elegant approach involves the symbolic tracker used in the qualitative tracker. By maintaining which aspect prediction graph node is visible, the symbolic tracker can activate only those nodes corresponding to visible faces. Determining which aspect is visible can be computed directly from knowledge of the model's exact pose. Alternatively, we can also pursue a data-driven approach to determining which aspect is visible. Analogous to our qualitative tracker, local image events can be used to detect a change in aspect with the aspect governing which nodes on the model are active (visible). In this case, a sudden vanishing of forces along a contiguous set of projected model nodes belonging to a face would signify an aspect change.

We must also deal with occlusion due to both known and unknown objects passing in front of the object being tracked. If the occluding object geometry and pose is known, then node visibility of the tracked object can be easily computed. Forces at occluded nodes can be simply turned off until they become visible again. For occlusion by an unknown (untracked) object, we can monitor changes in the image forces exerted on the tracked model's nodes. If the local forces at a particular node suddenly vanish or greatly increase, this erratic behavior can be used to suggest local occlusion, resulting in deactivation of those nodes. The Kalman filter can still maintain the track based on prior motion as well as other active nodes until the object becomes disoccluded.

6.3 Examples

We demonstrate our approach in a series of tracking experiments involving real stereo image sequences. Figure 16(a) shows the first pair of stereo images. The initial pose and shape of both objects are recovered using a stereo extension of our quantitative shape recovery algorithm. The objects are subsequently tracked using only image gradient forces, shown as a set of blurred edges in the image. Figures 16(b-f) show snapshots of the two objects being tracked with the wire-frame models overlaid on the image potential. This example illustrates our ability to track an object when a known object partially occludes it. The nodes on either model which are determined to be occluded (through either self-occlusion or occlusion by another known model) are deactivated until they become visible.

In the second experiment, we consider a sequence of stereo images (24 frames) of a scene containing multiple objects, including a two-part object. Figure 17 shows the initial stereo images of the multi-object scene. Figure 18(a) shows the initialized models using the same technique as before. Figure 18(b) shows the image potentials at an intermediate time frame where the aspects of some parts have changed and some parts have become partially occluded. Figures 18(c-f) show that each object is still successfully tracked under these circumstances with the individual part models overlaid on the image potentials.

7 Conclusions

Data-driven, deformable models allow models to adapt their shapes according to the image data. This is in stark contrast to typical CAD-based vision systems which attempt to model the exact geometry of an object. However, by effectively weakening the models to such an extent, they are no left with enough constraints to support segmentation, recognition, or tracking beyond simple translation. In this paper, we have shown how a qualitative object representation, integrating both object-centered and viewer-centered models, provides the missing constraints necessary to overcome these problems. Specifically, our probabilistic part-based aspect hierarchy:

1. provides the missing constraints on physics-based deformable model recovery, allowing more accurate shape recovery from both 2-D and 3-D data,

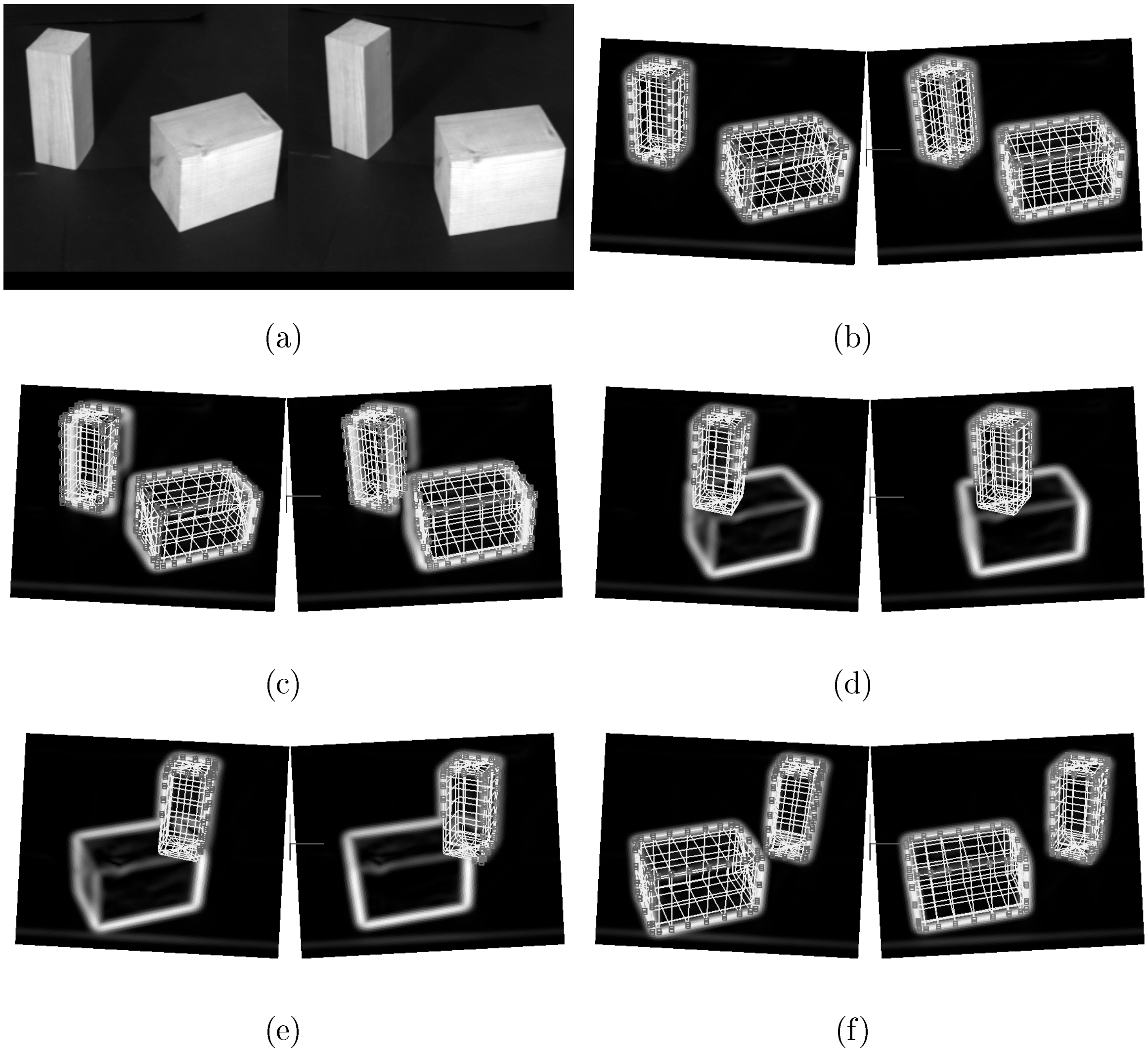


Figure 16: Tracking two independently moving blocks in a sequence of stereo images: (a) stereo pair, (b) initialized models, (c) start of occlusion, (d) taller block partially occluded (occluding model not shown), (e) taller block becoming disoccluded (occluding model not shown), (f) end of occlusion. Note that only the active model nodes are marked, with occluded nodes at the bottom of the taller block unmarked.

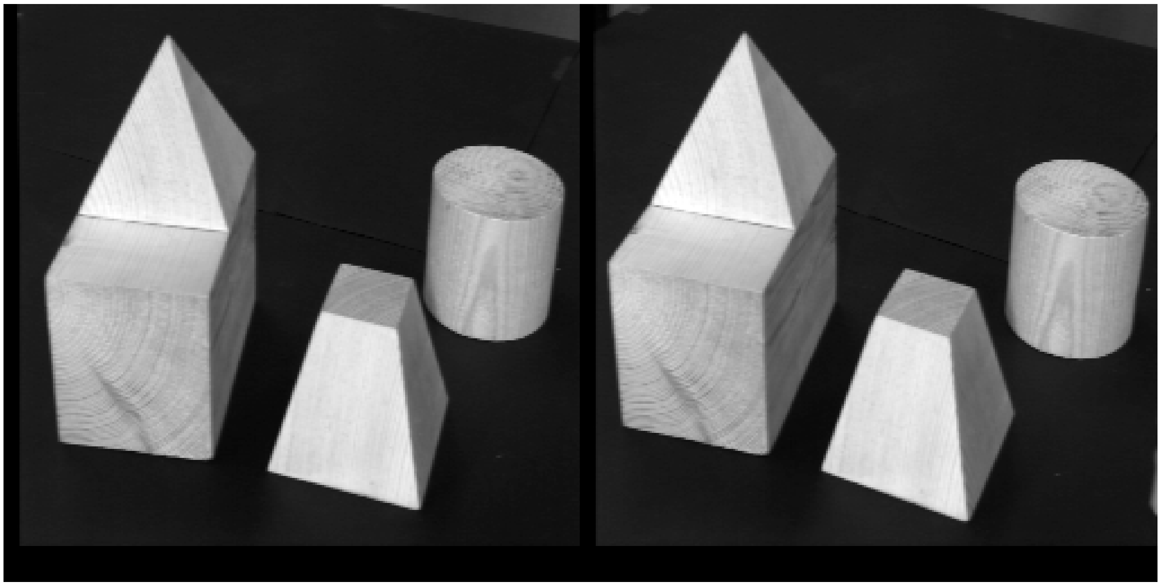


Figure 17: Initial stereo images of the multi-object scene.

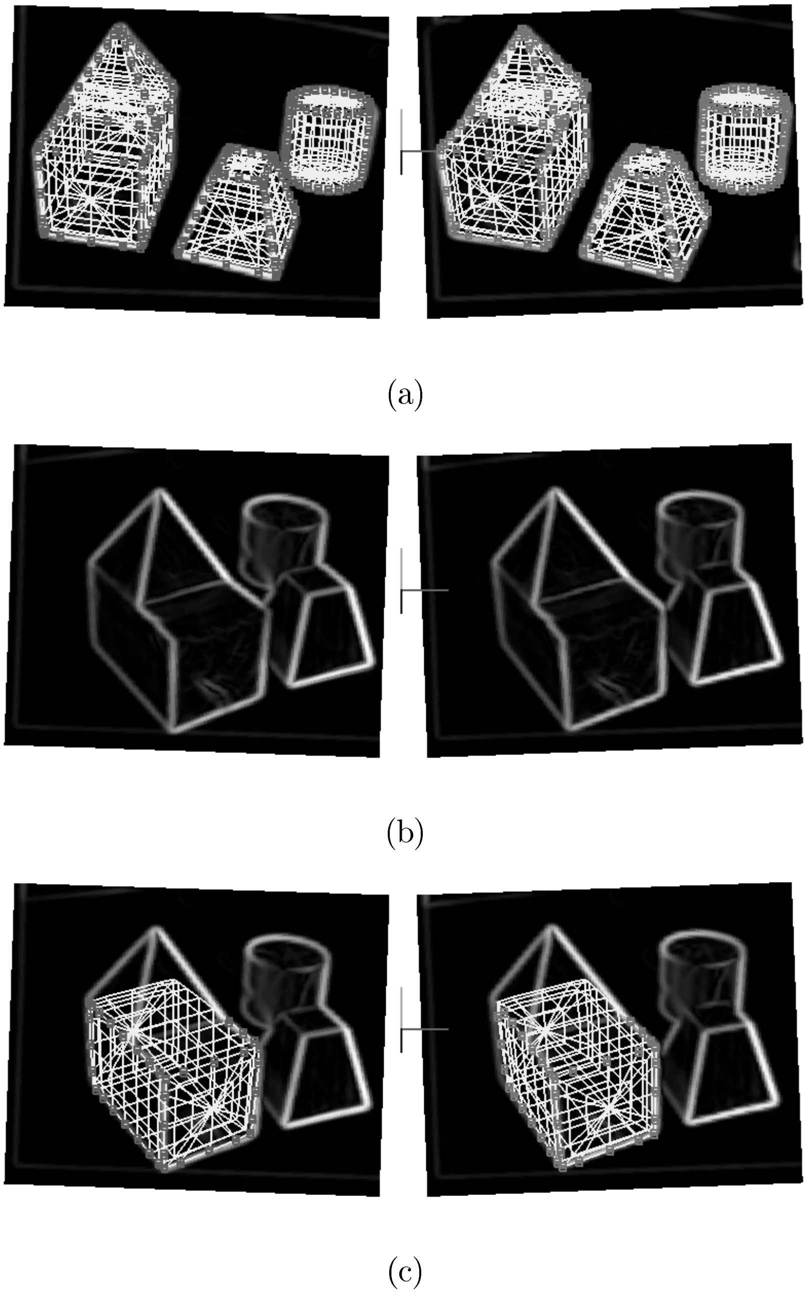
2. provides a control mechanism for an active contour network that can qualitatively track an object's translation *and* rotation in depth, and

3. provides a control mechanism that can control node activation when quantitatively tracking an object's motion and shape.

The results reported here are still preliminary, with much work remaining. The techniques are still sensitive to region segmentation performance, and the objects used in the experiments are simplified. Our goal has been to explore a number of closely-related object recognition behaviors that must be addressed by an active agent in a dynamic environment. Our object representation has so far provided a common framework for novel algorithms for these and other behaviors (e.g., active object recognition [10]). We continue to refine these algorithms while at the same time attempting to work with more complex scenes containing more realistic objects.

8 Acknowledgements

The support of the National Science Foundation (Sven Dickinson: NSF CAREER IRI-9623913; Dimitri Metaxas: NSF IRI-93-09917, NSF CAREER IRI-9624604) is gratefully

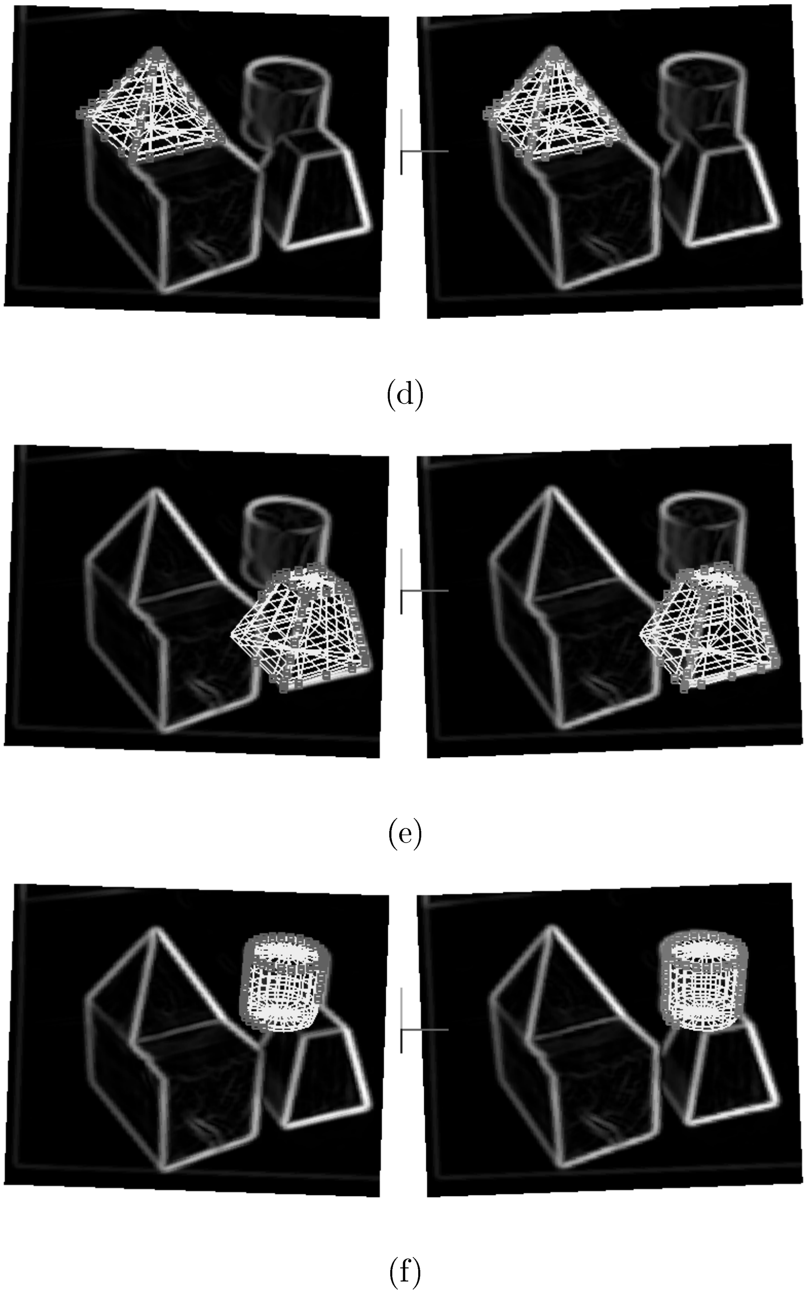


(a)

(b)

(c)

Figure 18: Tracking multiple objects in a sequence of stereo images: (a) initialized models, (b) image potentials of an intermediate frame (both occlusions and visual events have occurred), (c) each object part correctly tracked with part model overlaid on the image potentials. Note that only the active model nodes are marked.



(d)

(e)

(f)

Figure 18: (Cont'd) Tracking multiple objects in a sequence of stereo images: (d-f) each object part correctly tracked with part models overlaid on the image potentials. Note that only the active model nodes are marked.

acknowledged. Dimitri Metaxas would also like to gratefully acknowledge the support of the Army Research Office (ARO-DAAH04-96-1-007).

References

[1] A. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1:11–23, 1981.

[2] T. J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-D motion estimation from a monocular image sequence. *IEEE Transactions on Aerospace and Electronic Systems*, 26(4):639–656, 1990.

[3] M. Chan, D. Metaxas, and S. Dickinson. A new approach to tracking 3-D objects in 2-D image sequences. In *Proceedings, AAAI '94*, Seattle, WA, August 1994.

[4] M. Chan, D. Metaxas, and S. Dickinson. Physics-based tracking of 3-D objects in 2-D image sequences. In *Proceedings, 12 International Conference on Pattern Recognition*, pages 326–330, Jerusalem, Israel, October 1994.

[5] R. Cipolla and A. Blake. Motion planning using image divergence and deformation. In A. Blake and A. Yuille, editors, *Active Vision*, pages 189–201. MIT Press, 1992.

[6] R. Curven, A. Blake, and R. Cipolla. Parallel implementation of lagrangian dynamics for real-time snakes. In *Proceedings, British Machine Vision Conference (BMVC '91)*, pages 27–35, September 1991.

[7] R. Deriche and O. Faugeras. Tracking line segments. *Image and Vision Computing*, 8(4):261–270, 1990.

[8] S. Dickinson. The recovery and recognition of three-dimensional objects using part-based aspect matching. Technical Report CAR-TR-572, Center for Automation Research, University of Maryland, 1991.

[9] S. Dickinson. Part-based modeling and qualitative recognition. In A. Jain and P. Flynn, editors, *Three-Dimensional Object Recognition Systems*, Advances in Image Communication and Machine Vision Series. Elsevier, Amsterdam, 1993.

[10] S. Dickinson, H. Christensen, J. Tsotsos, and G. Olofsson. Active object recognition integrating attention and viewpoint control. In *Proceedings, ECCV '94*, Stockholm, Sweden, May 1994.

[11] S. Dickinson, P. Jasiobedzki, H. Christensen, and G. Olofsson. Qualitative tracking of 3-D objects using active contour networks. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, June 1994.

[12] S. Dickinson and D. Metaxas. Integrating qualitative and quantitative shape recovery. *International Journal of Computer Vision*, 13(3):1–20, 1994.

[13] S. Dickinson, D. Metaxas, and A. Pentland. Constrained recovery of deformable models from range data. In *Proceedings, 2nd International Workshop on Visual Form*, Capri, Italy, May 1994.

[14] S. Dickinson, A. Pentland, and A. Rosenfeld. A representation for qualitative 3-D object recognition integrating object-centered and viewer-centered models. In K. Leibovic, editor, *Vision: A Convergence of Disciplines.* Springer Verlag, New York, 1990.

[15] S. Dickinson, A. Pentland, and A. Rosenfeld. From volumes to views: An approach to 3-D object recognition. *CVGIP: Image Understanding*, 55(2):130–154, 1992.

[16] S. Dickinson, A. Pentland, and A. Rosenfeld. 3-D shape recovery using distributed aspect matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):174–198, 1992.

[17] E. D. Dickmanns and Volker Graefe. Applications of dynamic monocular machine vision. *Machine Vision and Applications*, 1:241–261, 1988.

[18] M. Kass, A. Witkin, and D. Terzopolous. Snakes: Active contour models. *Internation Journal of Computer Vision*, 1(4):321–331, 1988.

[19] J. Koenderink and A. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[20] D. Metaxas. Physics-based modeling of nonrigid objects for vision and graphics. *Ph.D. thesis, Dept. of Computer Science, Univ. of Toronto*, 1992.

[21] D. Metaxas and S. Dickinson. Integration of quantitative and qualitative techniques for deformable model fitting from orthographic, perspective, and stereo projections. In *Proceedings, Fourth International Conference on Computer Vision*, Berlin, Germany, May 1993.

[22] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, June 1993.

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

[24] D. Terzopolous and R. Szeliski. Tracking with kalman snakes. In A. Blake and A. Yuille, editors, *Active Vision*, pages 3–21. MIT Press, 1992.

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

[26] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-seeking models and 3d object recovery. *International Journal of Computer Vision*, 1:211–221, 1987.

[27] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motion. *Artificial Intelligence*, 36:91–123, 1988.