Integrating Qualitative and Quantitative Object Representations in the Recovery and Tracking of 3-D Shape*

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

Data-driven models such as active contours in 2-D and deformable surfaces in 3-D have become prevalent in the computer vision community, particularly in the areas of shape tracking and shape recovery. They provide an important alternative to typical model-based recognition and tracking approaches that assume knowledge of the exact geometry of the object. Despite the power of these approaches, data-driven models often encode too little model information. As a consequence, active 3-D model recovery schemes often require manual segmentation or good model initialization, and active contour trackers have been able to track only an object's translation in the image. To overcome these problems requires bridging the representational gap between overconstrained geometric models and underconstrained active models. In previous work, we introduced a qualitative object representation integrating object-centered and viewer-centered models. In this paper, we first show how this representation provides the missing constraints on the recovery of quantitative 3-D deformable models from 2-D images. We then show how this same representation provides the missing constraints needed to qualitatively track an object's rotation in depth or to quantitatively track an object's pose.

*To appear in: L. Harris and M. Jenkin (eds.), *Computational and Psychophysical Mechanisms of Visual Coding*, Cambridge University Press, New York, NY.

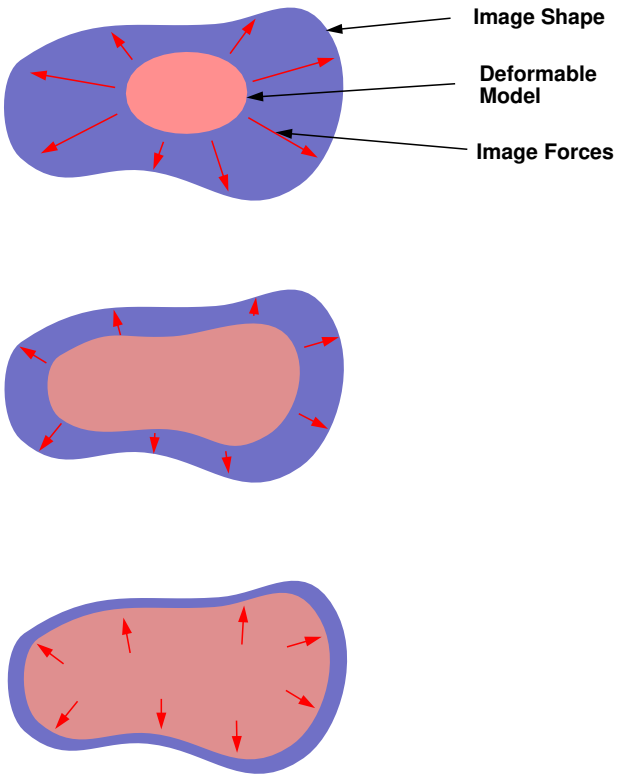

Figure 1: Data-Driven Shape Recovery

1 Introduction

In the computer vision community, there exists a continuum of approaches to recovering two- and three-dimensional shape from two- and three-dimensional images. This continuum is bounded by two extremes. At one end lie the purely data-driven approaches to shape recovery, while at the other end lie the purely model-driven approaches. Moreover, this continuum also applies to object tracking where, at one end, model-driven trackers assume knowledge of exact object geometry, while at the other end, data-driven trackers simply track the translation of an object's outline in the image. Although these purely data-driven and model-driven paradigms are extremely powerful, each of these two schools suffers from serious limitations. The solution, we believe, is a class of techniques for shape recovery and tracking whose underlying representations are intermediate between the two extremes.

The purely data-driven 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 [21, 36, 37, 35]. 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.

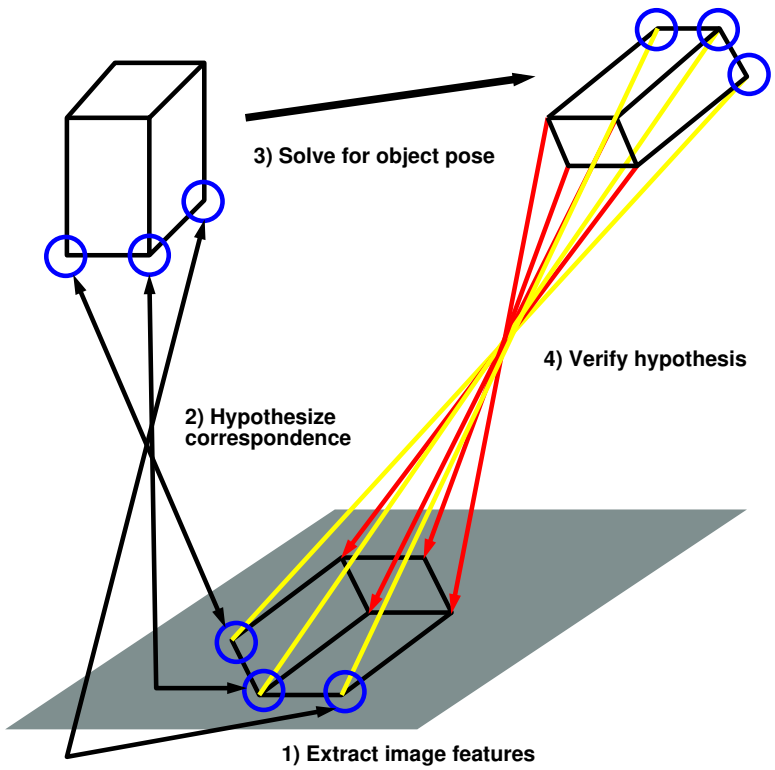

Figure 2: Model-Driven Object Recognition

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., [35]. In addition, such techniques often require a manual segmentation of an object into parts to which models are fitted, e.g., [36]. If the models are not properly initialized, a canonical fit may not be possible, e.g., [32]. These limitations are a consequence of using such unconstrained models.

At the other extreme lie the purely model-driven approaches to shape recovery, in which the exact geometry of the object is captured in a model, e.g., [26, 20, 24, 38]. As shown in Figure 2, simple image features such as corners or changes in curvature are paired with similar features on a 3-D model to yield a set of hypothesized correspondences. In order to verify a given hypothesis, the model is transformed to bring the chosen model features into alignment with their corresponding image features. Because the correspondence is weak, other features belonging to the aligned model must be projected into the image and compared to other image features. If there is sufficient agreement between the two, the object is "recognized." In this case, shape recovery is essentially provided by the model in the form of a 2-D template extracted from a 3-D model.

For machine vision applications, where the number of object models is small and exact object geometry is known, this approach is highly effective, requiring the extraction of simple, robustly-

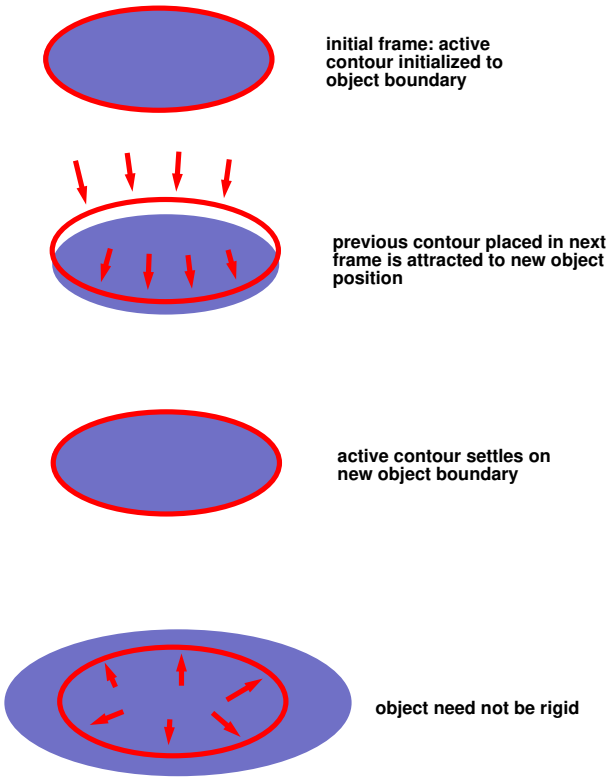

Figure 3: Data-Driven Object Tracking

recoverable image features and offering insensitivity to occlusion. However, for large databases, combinatorial complexity renders the search intractable. Furthermore, since the approach is dominated by the verification of local features, such as lines or corners, the approach is very sensitive to minor changes in the shapes of the objects. For example, if the dimensions or curvature of an object part changes, a new object model must be added.

In the tracking domain, a similar continuum exists. Purely data-driven approaches, as shown in Figure 3, track the silhouette of a blob in 2-D (or surface of a blob in 3-D), e.g., [21, 7, 34]. Although 2-D translation can be recovered and, in some cases, translation in depth (e.g., [6]), lack of any model information prevents the recovery of rotation in depth. At the other extreme, as shown in Figure 4, purely model-driven approaches can track an object's six degrees of freedom accurately, but require an exact specification of the object's geometry, and cannot support non-rigid object tracking, e.g., [27, 18, 39, 40].

In this paper, we describe both shape recovery and shape tracking paradigms that attempt to close the gap between underconstrained data-driven approaches and overconstrained model-driven approaches. The critical component of our approach is a parts-based object representation that combines both object-centered and viewer-centered 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 data-driven and model-driven techniques described above.

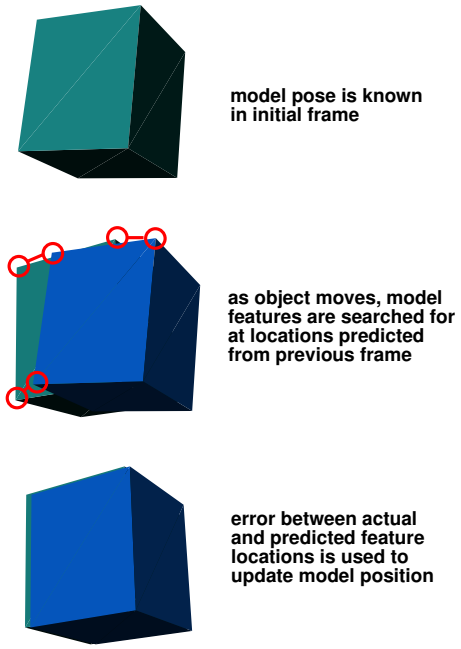

Figure 4: Model-Driven Object Tracking

2 Bridging the Representational Gap

In this section, we first describe a representation which models an object's 3-D shape in terms of a set of qualitatively-defined volumetric parts. This representation, combining both object-centered and viewer-centered models, will not only form the backbone of our qualitative shape recovery and tracking paradigms, but will be used to govern our quantitative shape recovery and tracking paradigms. We therefore include in this section a description of our quantitative shape model.

2.1 Qualitative Shape Modeling

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 qualitatively-defined volumetric parts chosen from some arbitrary, finite set [14]. The part classes are qualitative in the sense that they are invariant to degree of curvature, relative dimensions, degree of tapering, etc. Their choice was inspired by a set of

volumetric shapes (called geons) proposed by the psychologist, Biederman, as a set of shapes which the human visual system could quickly recover from a 2-D image, and which were rich enough to describe a large number of everyday objects [2]. Unlike Biederman, however, we do not restrict our representation to geons; we borrow only the notion of a finite set of qualitatively-defined volumetric parts used to build objects.

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 [22]. 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. Consequently, 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 [9]. The probabilities are estimated from a frequency analysis of features viewed over a sampled viewing sphere centered on each of the volumetric classes.

To demonstrate our techniques for shape recovery, object recognition, and tracking, we have selected an object representation similar to that used by Biederman [2], in which the Cartesian product of contrastive shape properties gives rise to a set of volumetric primitives called geons. For our investigation, we have chosen three properties including cross-section shape, axis shape, and cross-section size variation [14]. The values of these properties give rise to a set of ten primitives (a subset of Biederman's geons), modeled using Pentland's SuperSketch 3-D modeling tool [31], and illustrated in Figure 5. Figure 6 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.

2.2 Quantitative Shape Modeling

The qualitative object models described in the previous section will play a critical role in both the recovery of quantitative 3-D shape models from an image and the quantitative tracking of 3-D shape models in an image sequence. Geometrically, these quantitative shape models are closed surfaces in space whose intrinsic (material) coordinates are $\mathrm{u} = (u, v)$, defined on a domain Ω [35, 12]. The positions of points on the model relative to an inertial frame of reference Φ in space are given by a

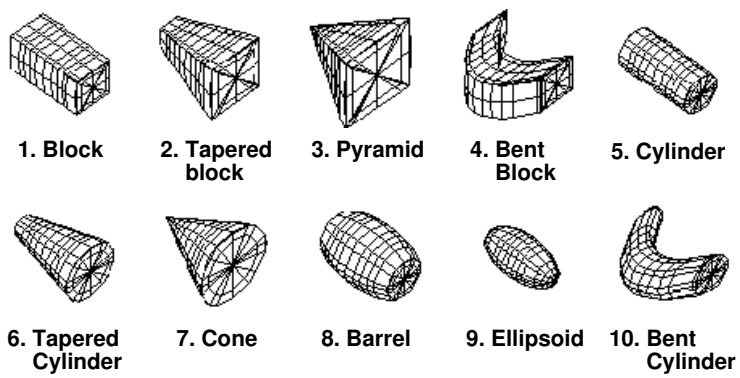

Figure 5: The Ten Modeling Primitives

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 [28], 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.

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

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

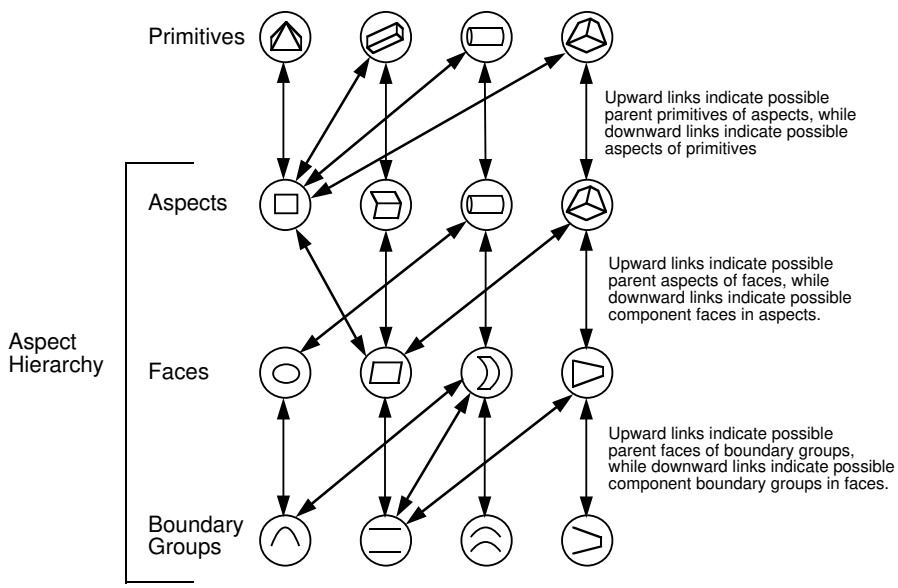

Figure 6: The Aspect Hierarchy

where $-\pi/2 \leq u \leq \pi/2$ and $-\pi \leq v < \pi$, and where ${S_w}^\epsilon = \text{sgn}(\sin w)|\sin w|^\epsilon$ and ${C_w}^\epsilon = \text{sgn}(\cos w)|\cos w|^\epsilon$, 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 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 5. 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

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

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

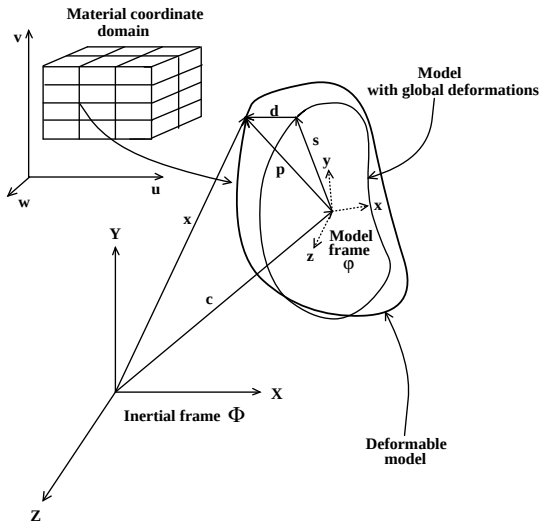

Figure 7: Geometry of Deformable Models

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 [28].

3 Recovering 3-D Shape

Identifying or recognizing the object's class may require only that we recover the coarse shape of the object. However, if we need to recover a more accurate, quantitative shape description for subclass recognition or for grasping, then a qualitative description is insufficient. In the following two subsections, we first outline an approach to qualitative shape recovery using the aspect hierarchy [16, 15, 10]. Next, we show how the recovered aspects, along with the recovered qualitative shape, can be used to constrain a physics-based deformable model recovery process that will yield the quantitative shapes of the object's parts [12, 13].

3.1 Qualitative Shape Recovery

An analysis of the conditional probabilities in the aspect hierarchy [16, 10] suggests that for 3-D modeling primitives which resemble the commonly used generalized cylinders, superquadrics, or geons, the most appropriate image features for recognition appear to be image regions, or faces. Moreover, the utility of a face description can be improved by grouping the faces into the more complex aspects, thus obtaining a less ambiguous mapping to the primitives and further constraining their orientation. Only when a face's shape is altered due to primitive occlusion or intersection should we descend to analysis at the contour or boundary group level. Our approach, therefore, first segments the input image into regions and then determines the possible face labels for each region. Next, we assign aspect labels to the faces, effectively grouping the faces into aspects. Finally, we map the aspects to primitives and extract primitive connectivity.

The first step in recovering a set of faces is a region segmentation of the input image. We begin by applying Saint-Marc, Chen, and Medioni's edge-preserving adaptive smoothing filter to

the image [33]. Next, we apply a fast region segmentation algorithm due to Kristensen and Nielsen [23], resulting in a region label image. In this method, a queue-based technique is used to merge pixels which differ by less than a similarity threshold. The comparison is based on simple pooled statistics for the regions. Given scale information about the objects in the field of view, an additional threshold is used to reject small regions.

From the resulting 2-D region label image, we build a *region topology graph*, in which nodes represent regions and arcs specify region adjacencies. Each node (region) encodes the 2-D bounding contour of a region as well as a mask which specifies pixel membership in the region. From the region topology graph, each region is characterized according to the qualitative shapes of its bounding contours. The steps of partitioning the bounding contour and classifying the resulting contours are performed simultaneously using a minimal description length algorithm [25]. From a set of initial candidate contour breakpoints (derived from a polygonal approximation), the algorithm considers all possible groupings of the inter-breakpoint contours. The best partitioning is chosen as the grouping with the minimum description length based on how well lines and elliptical arcs can be fit to the segment groups in terms of the cost of coding the various segments. The result is a *region boundary graph* representation for a region, in which nodes represent bounding contours, and arcs represent relations between the contours, including cotermination, parallelism, and symmetry.[2]

Once we have established a description for each image region, the next step is to match that description against the faces in the aspect hierarchy using an interpretation tree search [19]. Descriptions that exactly match a face in the aspect hierarchy will be given a single label with probability 1.0. For region boundary graphs that do not match due to occlusion or segmentation errors, we descend to an analysis at the boundary group level and match subgraphs of the region boundary graph to the boundary groups in the aspect hierarchy. Each subgraph that matches a boundary group generates a set of possible face interpretations (labels), each with a corresponding probability defined by the non-zero conditional probabilities mapping boundary groups to faces in the aspect hierarchy. The result is a *face topology graph* in which each node contains a set of face labels (sorted by decreasing order of probability) associated with a given region.

3.1.1 Unexpected Object Recognition

In an unexpected object recognition domain, in which there is no a priori knowledge of scene content, each face in the face topology graph (recall that there may be many faces at each node) is used to infer a set of aspect hypotheses, using the non-zero conditional probabilities mapping faces to aspects in the aspect hierarchy. A search through the space of aspect hypotheses for a covering of the regions of the image is guided by a heuristic based on the conditional probabilities in the aspect hierarchy [16, 15]. During the search process, aspect verification, like face matching, is accomplished through the use of an interpretation tree search [19]. Once a set of aspects has been recovered, each aspect is used to infer one or more volume hypotheses based on the non-zero conditional probabilities mapping aspects to volumes in the aspect hierarchy. This time, we search through the space of volume hypotheses until we find a set of volumes that is consistent with the objects in the database [16, 15].

[2]See Dickinson et al. [16] for a discussion on how parallelism and symmetry are computed.

3.1.2 Expected Object Recognition

In an expected or top-down object recognition domain, in which we are searching for a particular object or part, we use the aspect hierarchy as an attention mechanism to focus the search for an aspect at appropriate regions in the image. This technique was applied to the top-down recognition of multipart objects in [13]. Moving down the aspect hierarchy and guided by a Bayesian utility measure, target objects map to target volumes which, in turn, map to target aspect predictions which, in turn, map to target face predictions. Those faces in the face topology graph whose labels match the target face prediction provide an ordered (by decreasing probability) set of ranked search positions at which the target aspect prediction can be verified. If the mapping from a verified aspect to a target volume is ambiguous, this attention mechanism can be used to drive an active recognition system which moves the camera to obtain a less ambiguous view of the volume [13]. Finally, it should be noted that for either top-down (expected) or bottom-up (unexpected) volume recovery, each recovered volume encodes the aspect in which it it viewed; the aspect, in turn, encodes the faces that were used in instantiating the aspect, while each face specifies those contours in the image used to instantiate the face.

3.1.3 Example

To illustrate our approach to qualitative shape recovery and recognition, consider the image of a table lamp, as shown in Figure 8; the results of the bottom-up (unexpected) qualitative shape recovery algorithm are shown in Figure 9. At the top, the image window contains the regions extracted from the image, along with the region (face) numbers. To the left is a window describing the recovered primitives (primitive covering). The mnemonics, PN, PL, and PP, refer to primitive number (simply an enumeration of the primitives in the covering), primitive label (see Figure 5), and primitive probability, respectively. The mnemonics AN, AL, AP, and AS refer to the aspect number (an enumeration), aspect label (see [16]), aspect probability, and aspect score (how well aspect was verified), respectively. The mnemonics FN, FL, FP, and PS refer to face number (in image window), face label (see [16]), face probability, and corresponding primitive attachment surface (see [16]), respectively, for each component face of the aspect. The search window indicates the status of the aspect and primitive covering searches, along with the recognized object (table lamp) and a goodness of fit. There were seven objects in the database.

3.2 Quantitative Shape Recovery

In the previous section, we outlined a technique for recognizing a 3-D object 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, the above qualitative recognition strategy does 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 (described in Section 2.2) from the recovered image contours. As shown in Figure 10, distances between a recovered aspect and a projected model aspect are converted to 2-D image forces. These forces, in turn, are mapped

Figure 8: Image of a Table Lamp (256 x 256)

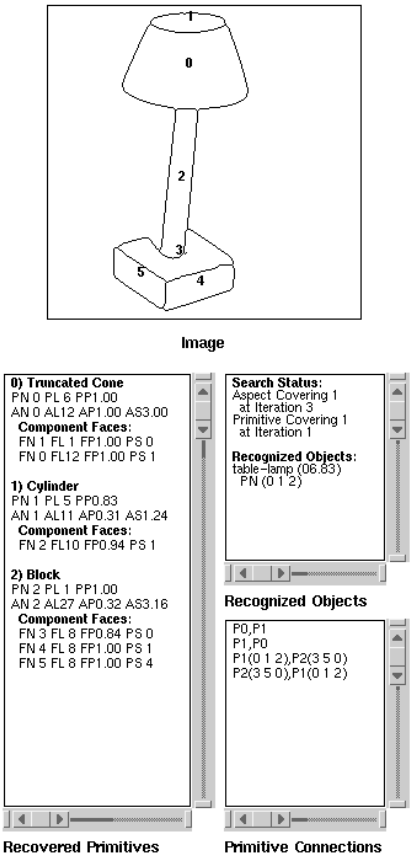

Figure 9: Recovered Qualitative Primitives

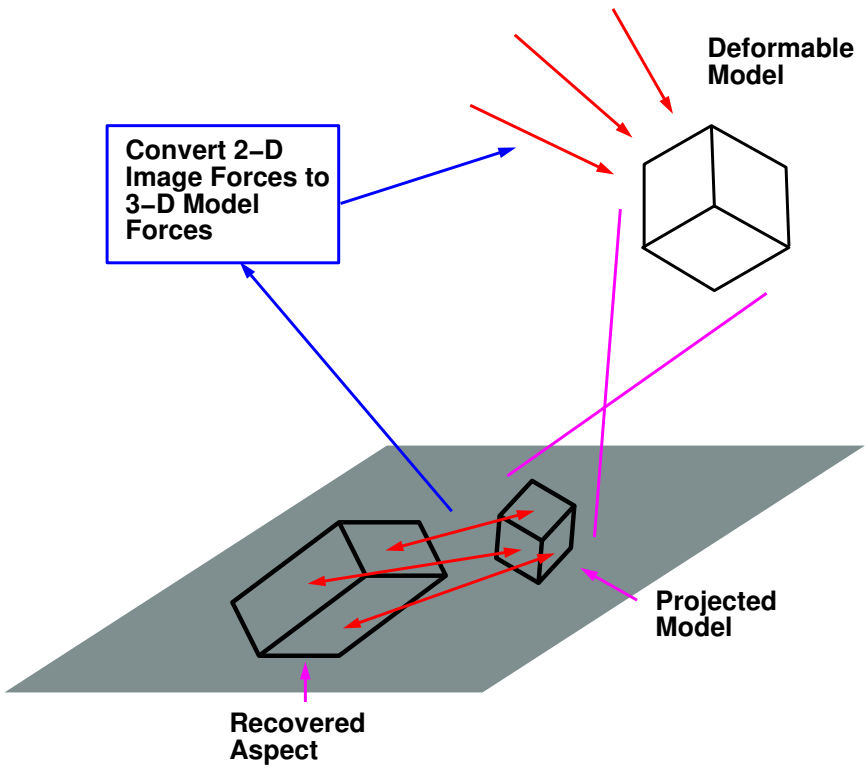

Figure 10: Using Qualitative Shape to Constrain Physics-Based Deformable Shape Recovery

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 [29, 12], while an extension of the technique to shape recovery from range data can be found in [13].

3.2.1 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 [35]:

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

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).[3] 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.

3.2.2 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 [35]. 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.

3.2.3 Model Initialization

One of the major limitations of previous deformable model fitting approaches is their dependence on model initialization and prior segmentation [37, 35, 32]. Using the 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. 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.

3.2.4 Example

To illustrate the fitting stage, consider the contours belonging to the recovered lamp shade shown in Figure 9. Having determined during the qualitative shape recovery stage that we are trying to fit a deformable superquadric to a truncated cone, we can immediately fix some of the parameters in the model. In addition, the qualitative shape recovery stage provides us with a mapping between

[3] In Section 4, we will see how Equation (10) is also used in object tracking.

faces in the image and physical surfaces on the model. For example, we know that the elliptical face (FN 1, in Figure 9) maps to the top of the truncated cone, while the body face (FN 0) maps to the side of the truncated cone. For the case of the truncated cone, we will begin with a cylinder model (superquad) and will compute the forces that will deform the cylinder into the truncated cone 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 11(a) shows the image contours corresponding to the lamp shade and the cylinder following application of this force. Figure 11(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 11(c) and 11(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 11(e) and 11(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 truncated cone.

4. Next, we focus on the image face corresponding to the body of the truncated cone 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 11(g) and 11(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 11(i) and 11(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. In the next section we will examine two object trackers and show how the aspect also plays a critical role in object tracking.

4 Tracking 3-D Shape

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

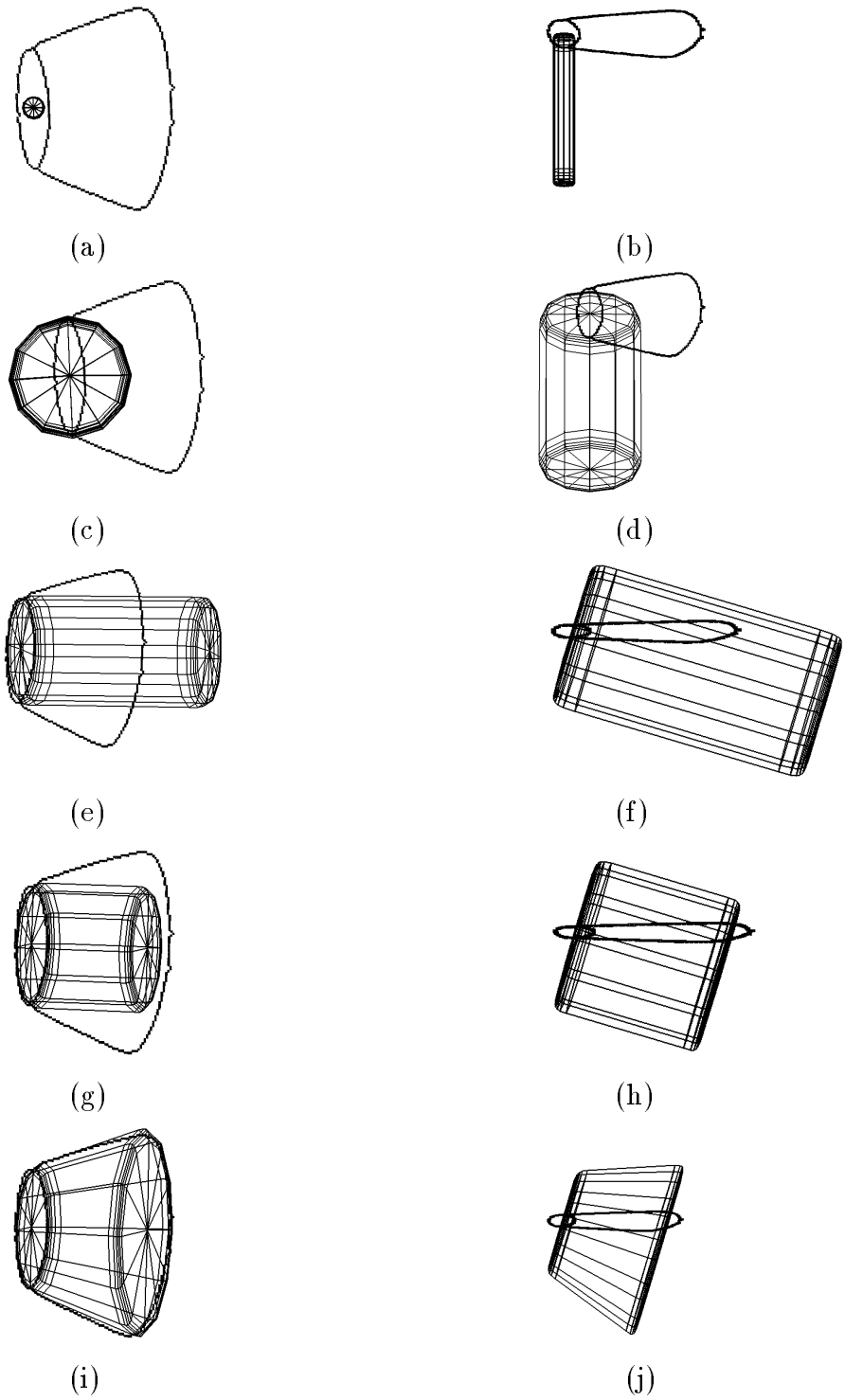

Figure 11: Quantitative Shape Recovery for Lamp Shade

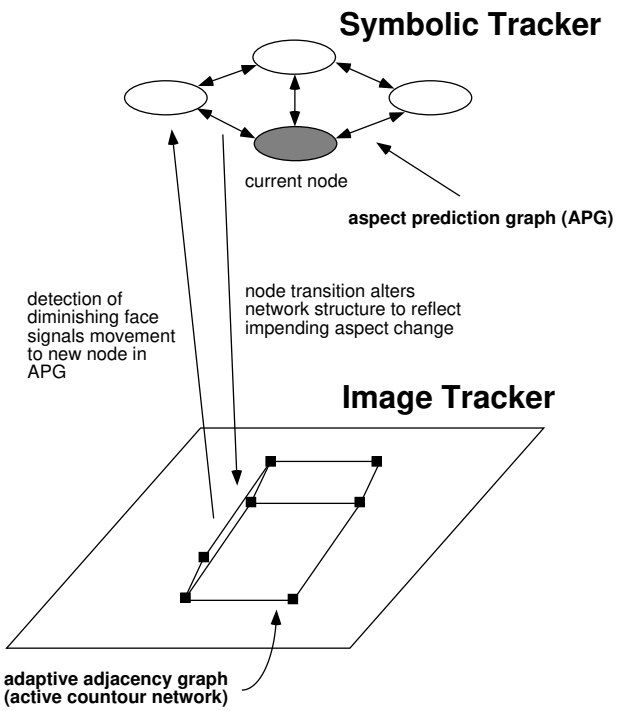

Figure 12: Qualitative Object Tracking

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 the following subsections, we outline our approach to each of these problems.

4.1 A Qualitative Tracker

Our approach to qualitative object tracking, as shown in Figure 12, 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.

4.1.1 Image Tracker

The image tracker employs a representation called an adaptive adjacency graph, or AAG. The AAG is initially created from a recovered aspect, and consists of a network of active contours (snakes) [21]. In addition, 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 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.

4.1.2 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 [22] 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 predicted aspects 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.

4.1.3 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.

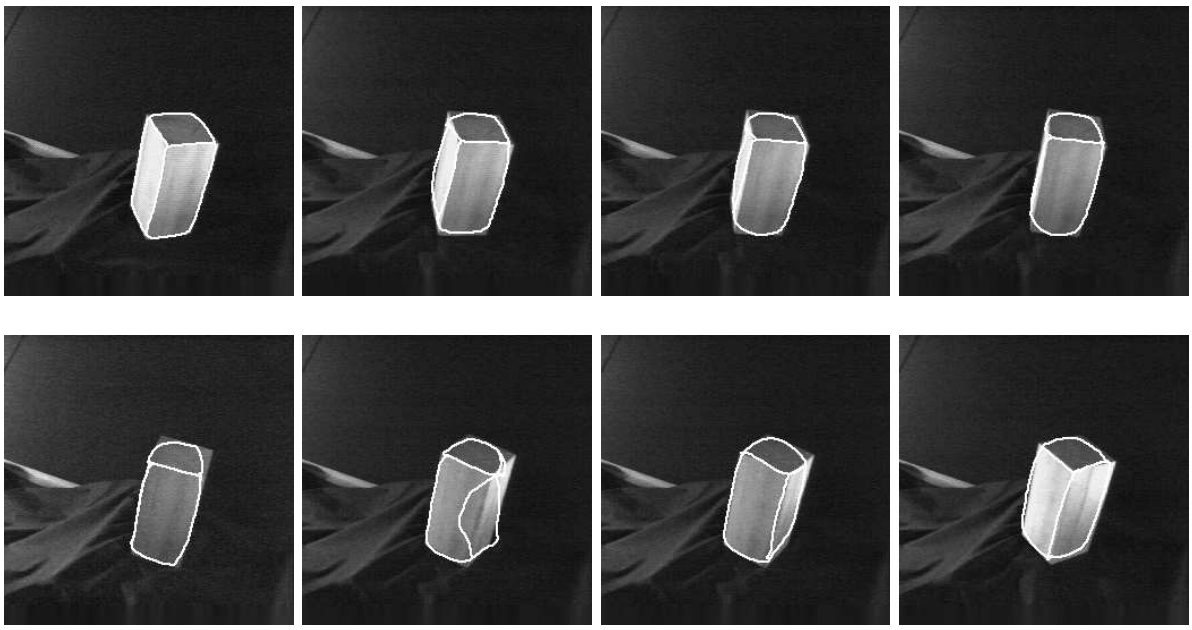

Figure 13: Tracking a Rotating Block. There were 11 images in the sequence with 10 iterations of the AAG per image, for a total of 110 snapshots of the AAG. Working left to right and top to bottom, we show snapshots 1, 23, 32, 41, 70, 82, 90, and 110. Note that when the disappearing face is detected (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.

4.1.4 Example

In Figure 13, 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.

4.2 Quantitative Object Tracking

Our approach to quantitative tracking [4, 5] 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 [30]. 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

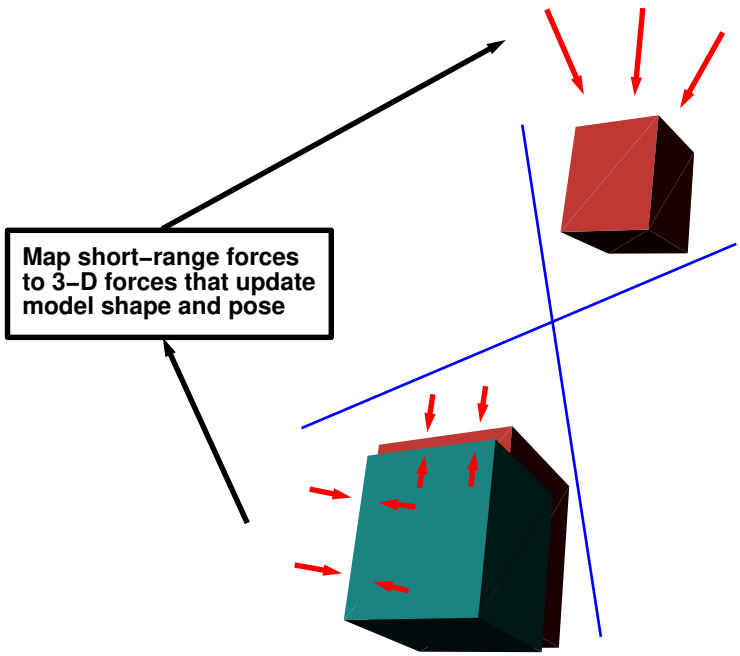

Figure 14: Quantitative Object Tracking

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 14. 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.

4.2.1 Tracking and Prediction

Kalman filtering techniques have been applied in the vision literature for the estimation of dynamic features [8] and rigid motion parameters [17, 3] 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:

$$\dot{\hat{\mathbf{u}}} = \mathbf{F}\hat{\mathbf{u}} + \mathbf{g} + \mathbf{P}\mathbf{H}^\top \mathbf{V}^{-1} \left(\mathbf{z} - \mathbf{h}(\hat{\mathbf{u}})\right), \tag{11}$$

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

4.2.2 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 [37], 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.

4.2.3 Examples

We demonstrate our approach in a series of tracking experiments involving real stereo image sequences. Figure 15(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 15(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 16 shows the initial stereo images of the multi-object scene. Figure 17(a) shows the initialized models using the same technique as before. Figure 17(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 17(c-f) show that each object is still successfully tracked under these circumstances with the individual part models overlaid on the image potentials.

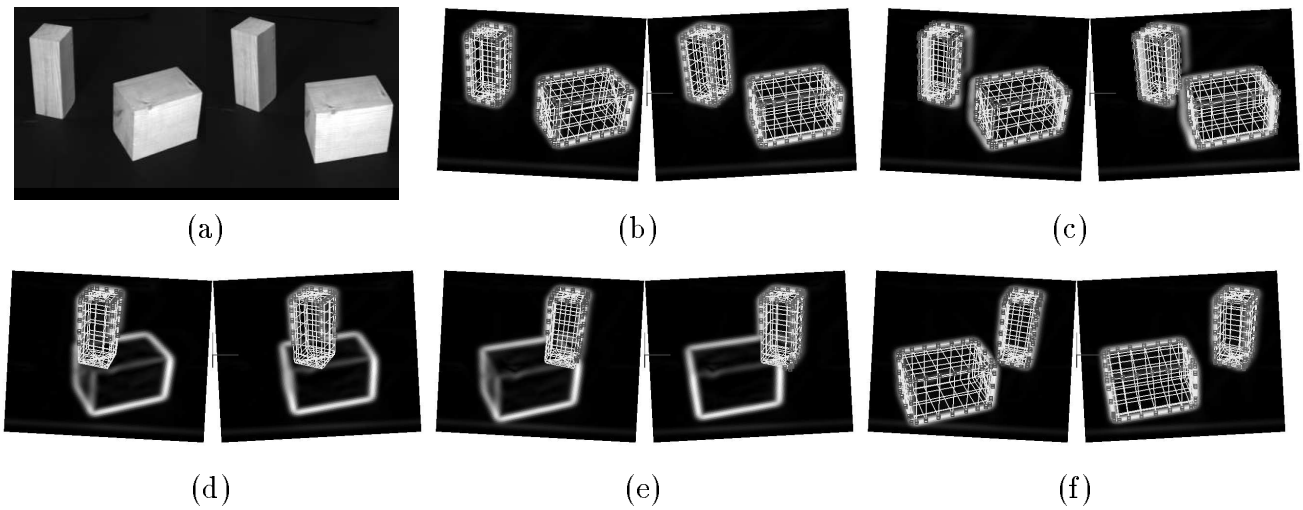

Figure 15: Tracking two independently moving blocks in a sequence of stereo images: (a) stereo pair, (b) initialized models, (c) start of occlusion, (d) taller block partially occluded (occluding model not shown), (e) taller block becoming disoccluded (occluding model not shown), (f) end of occlusion. Note that only the active model nodes are marked, with occluded nodes at the bottom of the taller block unmarked.

5 Conclusions

We have shown how the representational gap between underconstrained data-driven shape recovery and overconstrained model-driven object recognition can be bridged using an intermediate representation. Our part-based probabilistic aspect hierarchy combines the advantages of object-centered and viewer-centered modeling, offering a unifying representation that: 1) encodes sufficient shape information to support part segmentation and generic object recognition based on coarse shape, 2) provides the missing constraints on physics-based deformable model recovery, allowing more accurate shape recovery *if needed*, 3) provides a control mechanism for an active contour network that can qualitatively track an object's translation *and* rotation in depth, and 4) 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.

References

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

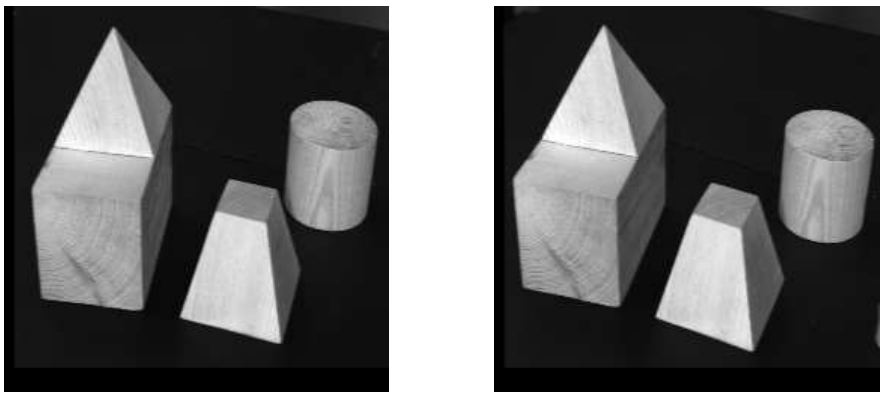

Figure 16: Initial stereo images of the multi-object scene.

[2] I. Biederman. Human image understanding: Recent research and a theory. *Computer Vision, Graphics, and Image Processing*, 32:29–73, 1985.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[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.

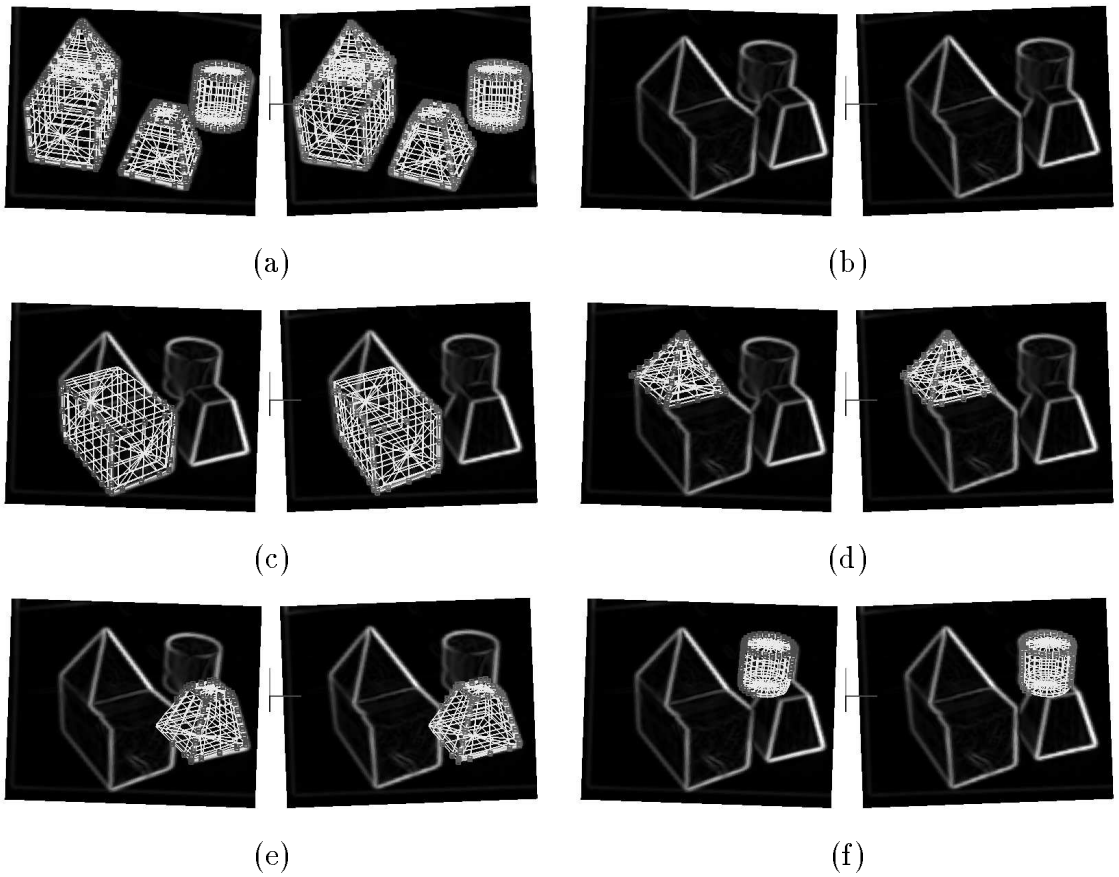

Figure 17: Tracking multiple objects in a sequence of stereo images: (a) initialized models, (b) image potentials of an intermediate frame (both occlusions and visual events have occurred), (c-f) each object part correctly tracked with part models overlaid on the image potentials. Note that only the active model nodes are marked.

[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] D. Gennery. Visual tracking of known three-dimensional objects. *International Journal of Computer Vision*, 7(3), 1990.

[19] W. Grimson and T. Lozano-Pérez. Model-based recognition and localization from sparse range or tactile data. *International Journal of Robotics Research*, 3(3):3–35, 1984.

[20] D. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212, 1990.

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

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

[23] S. Kristensen and H. Nielsen. 3d scene modeling for robot navigation. *M.SC. Thesis*, October 1992.

[24] Y. Lamdan, J. Schwartz, and H. Wolfson. On recognition of 3-D objects from 2-D images. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 1407–1413, Philadelphia, PA, 1988.

[25] M. Li. Minimum description length based 2-D shape description. Technical Report CVAP114, Computational Vision and Active Perception Lab, Royal Institute of Technology, Stockholm, Sweden, October 1992.

[26] D. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers, Norwell, MA, 1985.

[27] D. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991.

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

[29] 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.

[30] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):580–591, June 1993.

[31] A. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986.

[32] 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.

[33] P. Saint-Marc, J.-S. Chen, and G. Medioni. Adaptive smoothing: A general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):514–529, 1991.

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

[35] 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.

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

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

[38] D. Thompson and J. Mundy. Model-directed object recognition on the connection machine. In *Proceedings, DARPA Image Understanding Workshop*, pages 93–106, Los Angeles, CA, 1987.

[39] G. Verghese, K. Gale, and C. Dyer. Real-time, parallel tracking of three-dimensional objects from spatiotemporal sequences. In V. Kumar, P.S. Gopalakrishnan, and L.N. Kanal, editors, *Parallel Algorithms for Machine Intelligence and Vision*. Springer-Verlag, New York, 1990.

[40] J. Wu, R. Rink, T. Caelli, and V. Gourishankar. Recovery of the 3D location and motion of a rigid object through camera image (an extended kalman filter approach). In *International Journal of Computer Vision*, volume 3, pages 373–394, 1989.