Int'l Journal of Computer Vision, Vol. 20, No. 3, Dec, 1996 (final version).

FORMS: A Flexible Object Recognition and Modeling System

Song Chun Zhu [1] and A.L. Yuille[2].

Abstract

We describe a flexible object recognition and modeling system (FORMS) which represents and recognizes animate objects from their silhouettes. This consists of a model for generating the shapes of animate objects which gives a formalism for solving the inverse problem of object recognition. We model all objects at three levels of complexity: (i) the primitives, (ii) the *mid-grained shapes*, which are deformations of the primitives, and (iii) objects constructed by using a grammar to join mid-grained shapes together. The deformations of the primitives can be characterized by principal component analysis or modal analysis. When doing recognition the representations of these objects are obtained in a bottom-up manner from their silhouettes by a novel method for skeleton extraction and part segmentation based on deformable circles. These representations are then matched to a database of prototypical objects to obtain a set of candidate interpretations. These interpretations are verified in a top-down process. The system is demonstrated to be stable in the presence of noise, the absence of parts, the presence of additional parts, and considerable variations in articulation and viewpoint. Finally, we describe how such a representation scheme can be automatically learnt from examples.

[1] Division of Applied Sciences, Harvard University. Cambridge, MA 02138.
[2] The Smith-Kettlewell Eye Research Institute. 2232 Webster Street, San Fran., CA 94115.

1 Introduction

This paper proposes a novel method for representing and recognizing flexible objects from their silhouettes. We will be specifically interested in animate objects such as people, hands, animals, leaves, fish and insects. The modeling and recognition of such flexible objects is made difficult by the following factors: (i) the silhouettes of these objects will vary greatly with their articulation and the observer's viewpoint, so techniques such as linear combinations of views [38] or viewpoint interpolation [32] seem inapplicable, (ii) such objects rarely contain salient features, such as corners or straight lines, which often play a large role in recognizing rigid objects [13], [19] [16], (iii) such objects do not seem to possess geometric invariants of the type recently exploited for recognizing certain classes of rigid objects [25]. In short, there will be considerable variation in the silhouettes of the objects. The representation, therefore, must be flexible enough to capture these variations and the recognition system must be sophisticated enough to take them into account. The representation must also be simple, in the sense of depending on a small number of parameters, and be suitable for statistical analysis, reasoning and learning.

The representation must also help capture the intuitive concept of similarity between shapes. Although there exist many mathematical similarity measures, none of them seem adequate for capturing human intuitions [23]. In FORMS the similarity measure is based on the statistical variations of the representations of the shapes.

Our approach builds on three important themes in object recognition. The first is the attempt to represent objects in terms of elementary parts, such as generalized cylinders [3], [7] [21] [9] [2]. The second is the use of deformable templates and deformable models [12], [40] [37], [33], [29], [15]. The third is the effort to solve recognition in a bottom-up/top-down loop using specific knowledge of the models to resolve ambiguities ¡occurring in the bottom-up process [24].

The general architecture of our approach is shown in figure 1. Objects are made by joining parts (or *mid-grained shapes*) together using grammatical models. These parts are generated by deforming elementary primitive shapes. In this paper we use two primitive shapes only, one is called the *worm*, the other is a circle. Deformations of these primitive shapes can be analyzed using principal component analysis (PCA) or finite element methods (modal analysis) to obtain a low-dimensional representation for the deformations.

Recognition is performed by a bottom-up/top-down control loop. The representation of the input is first calculated in a bottom-up manner, which uses only weak knowledge about animate objects, and is then matched to the prototype objects in the database. This gives a hypothesis set of candidate prototype matches. These candidate prototypes are then matched to the representation and the matching residuals are evaluated by direct comparison to the input data in a top-down verification process. The thirty-five objects in our database are successfully classified into one of seventeen categories despite variations due to viewpoint and articulation changes.

The paper is arranged as follows. In section (2) we briefly review some general shape properties of animate objects and thus motivate our choice of representation. In section (3) we introduce our method of modelling shapes properties of animate objects at an abstract level. This involves defining a low dimensional probability space for shape representation.

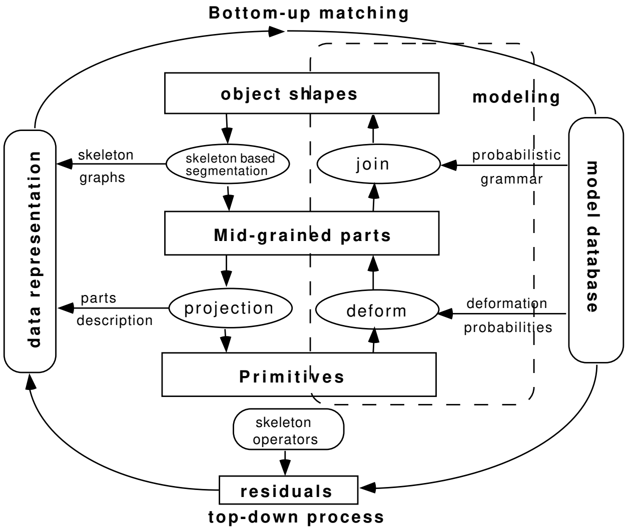


Figure 1 The architecture of our approach

In section (4), we describe a bottom-up process for calculating this representation from binary silhouette inputs. This can be thought of as the inverse of object modeling. We propose a novel skeleton algorithm, using weak assumptions about animate objects, based on deformable templates and linear prediction and error correction techniques. In section (5) we describe how recognition is achieved by a bottom-up/top-down loop. Finally, in section (6) and the appendix, we discuss the current limitations of our system and how it can be extended.

2 Motivation for the Representation

In this section we motivate our choice of representation for animate objects. We argue that basic properties of such objects will enable us to describe them by a low-dimensional representation motivated by the statistical variations of their shapes.

Although the shapes of animals and plants are very varied, their structure is not arbitrary. It is currently believed that all vertebrates evolved from fish in the Palaeozoic seas. Under the forces of climatic and geological changes, evolution generated a spectrum of animals ranging from fish to amphibia, reptiles, birds and tetrapods [1]. Elementary anatomical and physiological studies shows that all vertebrates are built upon a common plan [39], [14] and share many common properties. These properties are:

1. Skeletons. Every vertebrate has a tree-like [2] skeleton consisting of spines for the neck, torso, and tail, and bones for the limbs.[3] Typical bones for the spine and limb are shown in figure 2.

2. Parts. Each vertebrate can be decomposed into a set of parts with each edge in

[1] A tetrapod is a vertebrate (e.g. a cat, bird, or frog) with two pairs of limbs.

[2] A tree is defined to be a connected graph without loops. In other words, there are no holes in vertebrates.

[3] At this level of resolution, we can ignore the small bones in the skull.

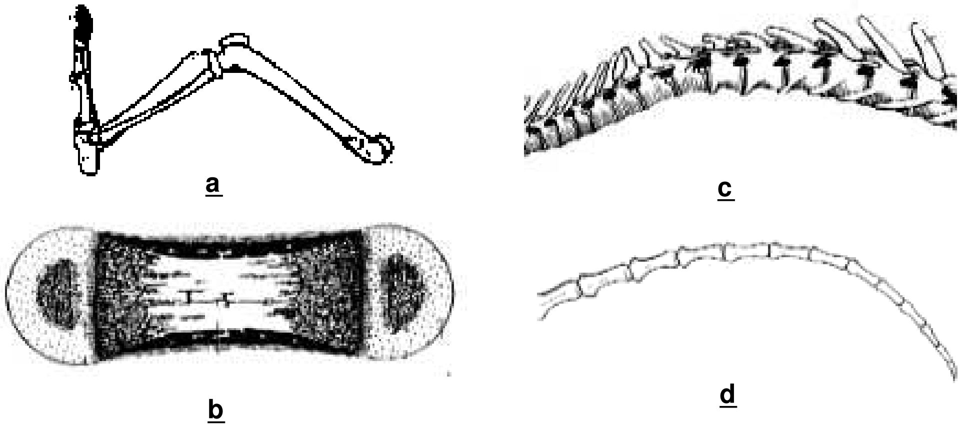


Figure 2 Typical bones for the skeletons of vertebrates. (a) is the bone for an elongated limb, which is usually decomposed into three pieces of long bones. (b) is a close view of a long bone. (c) and (d) are the backbone and the bones on the tail respectively, each segment in (c) and (d) is very short. There is free motion between those short segments, but such motion is limited and correlated by the muscles. After Hildebrand 1988.

the skeleton forming the axis of that part. The axis is then surrounded by muscles. These parts can be further classified into two classes: (i) elongated parts like the torso, tail, and long limbs, (ii) short parts like the fins and tails of fishes, the ears of horses, the heel of human feet, etc. Comparative anatomical studies found that there exists a continuous evolution from short parts to elongated parts. Figure 3 shows the process of evolution from a fin below the body of a fish to a short limbs in an amphibia and to a long limb of a tetrapod.

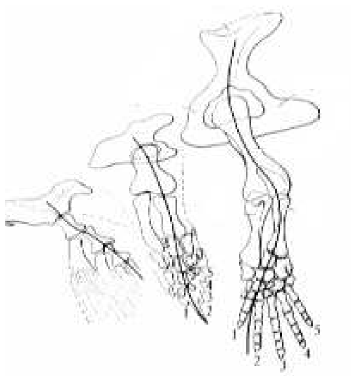


Figure 3 The continuous evolution from fin to limb. (a) is a fin of a fish, (b) is a short limb of an amphibia, and (c) is a long limb of a tetrapod. modified from Young 1981

3. Joints. There are many kinds of joints between bones, such as the serrate joint, the butt joint, the peg-and-socket joint and so on. Most of these are used to join the small bones in the skull, and only the hinge joints and ball-and-socket joints are important for general animate shapes. The hinge joint joins two long bones together, and the ball-and-socket joints are used to join the limbs to the torso at the shoulder and hip. But the projection of the ball-and-socket joint onto the image plane, is equivalent to the hinge joint. Figure 4 shows a typical hinge joint and its abstract representation.

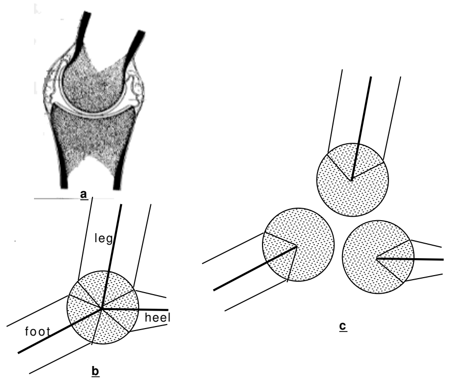


Figure 4 (a) is a hinge joint from Hildebrand 1988 and (b) is the abstract representation of a ankle joint, where the three parts–the leg, the foot, and the heel share the same joint circle in exclusive angle intervals. The ankle joint can be segmented into three parts in (c)

Although these properties are derived for vertebrate objects, they are valid for many other animate objects like leaves, and trees. For example, the stems in leaves are quite similar to the bones in vertebrates. On the other hand, there are some properties which are common to some animate objects, but are not included here. We will discuss those properties in section (6).

3 Flexible object modeling

In this section, we describe how to model the shapes of animate objects in three stages (see figure 1): (i) designing primitive shapes, (ii) characterizing their deformations, and (iii) defining a grammar for combining the deformable parts. We will describe the deformable parts using a local coordinate system. It is straightforward to add global transformations, such as translation, scaling and rotation, see section (3.4).

3.1 Designing primitive shapes

In this paper we choose only two primitive shapes. The first is the *worm*, shown in figure 5(a). It is a rectangle with joint circles **attached** at both ends. As the worm deforms, see figure 5(c), the scale of the circles may change but they will not deform in any other way. The second primitive is the *circle*, see figure 5(b). An angular section of this circle can deform, see figure 5(d). [4]. We now describe how we model the deformations of these

[4]We have also considered a worm-type primitive where the circles are also allowed to deform, but this primitive is not used in this paper. The worm and circle can be considered as special cases of this primitive. The relationship between the worm primitive and the circular primitive reflects the evolution process described in figure 3. More mundanely, one can think of the worm and the circle as corresponding to smoothed local symmetries [6] and local rotational symmetries [11] respectively.

primitives.

The shape of the worm can be characterized by a vector whose dimensions will depend on the precision required by the application. More precisely, this vector consists of the following components:

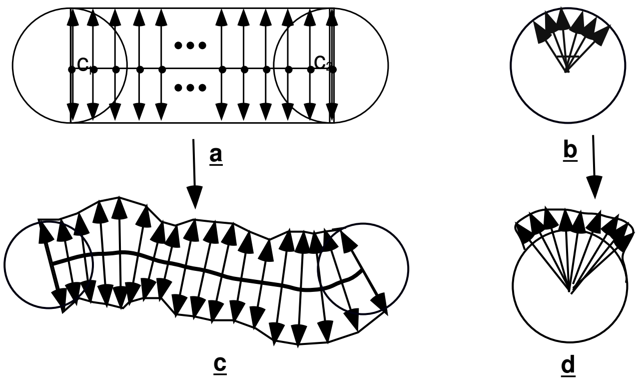


Figure 5 The Worm, the circle and their deformations. The worm primitive is shown in (a) and the circle primitive is shown in (b). Deformations of the worm and circle are shown in (c) and (d).

- The axis: we uniformly sample n points on the central axis C_1C_2 of the rectangle, which we assume has length ℓ. The stretching and bending of the axis will be represented by the arc length ℓ and the position of the points, $\vec{X}=(x_1, y_1, ..., x_n, y_n)$. For the applications in this paper we set $n = 36$.
- The rib: for each point on the axis, we measure the width perpendicular to the axis. For simplicity, we assume that the two sides are symmetric so the shape of the local deformations of the rectangle is represented by the vector $\vec{R}=(r_1, r_2, ..., r_n)$.

The circle primitive is specified by:

- The angle β specifying the angular region within which the deformations occur. Within this region m rays are sent out, at angular intervals of $\beta/(m-1)$, to measure the distances d_i from the center to the boundary. The deformed circle is represented by the vector $\vec{C}=(d_1, d_2, ..., d_m)$, the radius R and the angles β. We choose m=18 here.

By referring back to the anatomic analysis of vertebrate shapes in section (2), we see that the axis of the worm models the shape of bones, while the rib vector models the deformations of the muscles. As we will see in later sections, the circles at the ends of the worms are used as hinges to join deformed parts together (see also figure 4(b,c)). The circle primitives will model short parts like the fin of fish etc, and the circle itself will also be used as the joint hinge.

3.2 Designing deformation modes

The deformations of the primitives correspond to the variations in the axis-vector $\{x_i, y_i\}$ and the rib $\{r_i\}$, for the worm, and the radials $\{d_i\}$ for the circle. We denote these variables by the vector $\vec{S}$, which will be used to describe either deformed worms or circles.

A deformed primitive $\vec{S}$, see figures 5(c,d), can be represented as the sum of a basic, average, shape $\vec{S}_0$ and deformations $z \cdot \vec{\phi}$, where z is a scaling constant. For the worm primitive we would set $\vec{\phi} = \frac{\Delta \vec{S}}{z} = (\Delta x_1/z, \Delta y_1/z, \Delta r_1/z, ...)^{\tau}$ [5]. Similarly, for the circle primitive, $\vec{\phi} = (\Delta d_1/z, \Delta d_2/z, ...)^{\tau}$. *The range of each element in the deformation vector* $\vec{\phi}$ *is bounded, because large deformations will cause changes of the object structure, which we will model by grammatical rules, see section (3.4).* Therefore the possible shapes obtained by deforming a primitive are included in a hyper-box centered on the average shape $\vec{S}_0$. Each point in the box is called a mid-grained shape.

The regularity of animate objects in our environment makes the components of the deformation vectors $\vec{\phi}$ highly correlated, and hence enables drastic dimensional reduction. For each primitive we can find a set of orthogonal basis vectors $\vec{\Psi}_1, \vec{\Psi}_2, ...\vec{\Psi}_k \quad (k << n \text{ or } m)$ which span a subspace within the hyper-box of mid-grained shapes, so that all meaningful mid-grained shapes can be approximated, to the degree of accuracy required by our application, by their projections $(\alpha_1, \alpha_2, ..., \alpha_k)^{\tau}$ onto this subspace, where α_i is the projected length of the vector $\vec{\phi}$ onto the basis vectors $\vec{\Psi}_i$. We call the $\vec{\Psi}_i \;\; i = 1, 2, ..., k$ the deformation modes, because each basis vector $\vec{\Psi}_i$ deforms the primitive in a fixed way. The intuition is of an artist using a toolkit to deform his clay.

More specifically, we will describe all mid-grained shapes deformed from the worm as $worm(\ell, \alpha_1, \alpha_2, ..., \alpha_k)$. Suppose $k = 4$, and we only allow ten discrete values for each α_i. Then we can represent 10^4 mid-grained shapes in this way. Similarly we can defined the deformed circles by $circle(\beta, \alpha_1, \alpha_2, ...\alpha_l)$.

There are three possible approaches, the principal component analysis, the mechanic modal analysis (finite element method), and Fourier theory for calculating the basis vectors $\vec{\Psi}_1, \vec{\Psi}_2, ..., \vec{\Psi}_k$. We will discuss the first two in turn[6]. Relations between these three approaches are described in section (3.3).

Principal component analysis (PCA) is a data-driven method. To generate our data for PCA, we first partitioned a representative set of animate objects into 210 deformed worm parts and 20 deformed circle parts. Some parts segmented for a dog and a fish are shown in figures 23,24. How these parts are segmented from the object shapes automatically by the computer will be addressed in section (4).

(I). For the deformed worm parts, we applied principal component analysis to the rib deformations only,[7] and we found that only the first four eigenvectors, see figure 6, are required to describe 98% of the mid-grained parts in our animate object database with relative errors of less than 10 percent, the average relative error for all parts is 6.2%. The

[5] The primitive is scaled so that the radius of its largest end circle is equal to 1. The arc length ℓ is also scaled by z. At this stage we ignore rotation and translation.

[6] Fourier theory is too well known to require discussion here.

[7] For animate objects, the changes of the axis are typically rigid but the rib variations are usually nonrigid. Thus we treat the descriptions of the rib and the axis separately.

relative error is measured as $\frac{z\|\phi-\sum_{i=1}^{k}(\phi\cdot\Psi_i)\Psi_i\|}{\|S_0+z\phi\|}$. For example, in figure 7, we show two worm parts for a person: figure 7(a) is the torso and 7(b) is the head. Figures 7(c),(d) are respectively the reconstructed parts with 4 eigenvectors. Thus all deformed worm parts in FORMS are represented by five parameters $(\ell, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$.

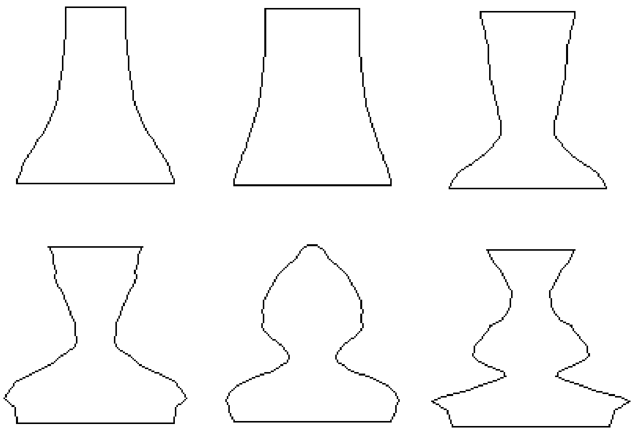


Figure 6 *The shape in the top left corner is the mean and the shapes from left to right and top to bottom are the first five eigenvectors (plus the mean) of the covariance matrix sorted in decreasing order of their corresponding eigenvalues.*

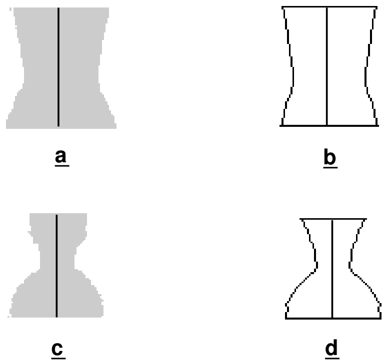


Figure 7 *(a) and (c) are respectively the torso and head when the inputs are people, and (b), (d) are the reconstructed shapes with the first 4 eigenvectors. Here the joint circles attached at the ends of these parts are not shown for reasons of clarity. Observe that the two ears are just visible in (c), but are lost in the reconstruction (d).*

(II). Figure 9 shows the 20 circular parts segmented from the objects. They are the fins of fish, ears of horses, heads of butterflies, heels of feet, etc. The deformations occur in a angular region of size β within which m=18 rays were sent to measure the variations of the radius. PCA was then applied to the 20 m-dimensional vectors. Figure 8 shows the mean and first 4 eigenvectors for a constant angle $\beta = \pi/2$. We found that using these

4 eigenvectors, all 20 circular parts can be reconstructed with relative error of less than 2%. The reconstructed shapes are also shown in figure 9 by the dark lines. Hence, each deformed circular part will be represented by five parameters $(\beta, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$.

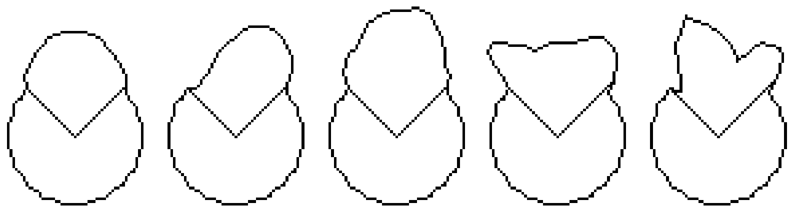


Figure 8 *The shape on the left is the mean and the four shapes to its right are the first 4 eigenvectors (plus the mean) of the covariance matrix sorted in the decreasing order of their corresponding eigenvalues.*

The second approach for dimensional reduction is mechanical modal analysis by the finite element method (FEM). By contrast with PCA this is a model-based approach which assumes a physical model for the deformations. By analogy with mechanical engineering, we think of the primitive as a *uniform* plastic plate obeying the appropriate physical equations.

The free vibration equation of the plastic plate in FEM can be written in terms of the deformation vector $\vec{\phi}$ yielding:

$$\mathcal{M}\ddot{\vec{\phi}} + \mathcal{K}\vec{\phi} = 0 \tag{1}$$

where $\mathcal{M}, \mathcal{K}$ are the mass and stiffness matrix respectively. Assume that the solution is of form $\vec{\phi} = \vec{\Psi} e^{i\omega t}$, where $\vec{\Psi}$ is the amplitude vector and ω is the frequency of vibration. Then we have:

$$\omega^2 \mathcal{M}\vec{\Psi} = \mathcal{K}\vec{\Psi} \tag{2}$$

This equation can be solved to obtain the eigenvalues $\omega_1 < \omega_2 < ... < \omega_k$ and their corresponding eigenvectors $\vec{\Psi}_1, \vec{\Psi}_2, ... \vec{\Psi}_k$. The latter are orthogonal to each other with respect to the mass matrix $\mathcal{M}$.

We tried to calculate the deformation modes for both the axis vector $\vec{X}$ and the rib vector $\vec{R}$ simultaneously.[8] To model the vibration of the rectangle accurately, we partitioned the rectangle into 36×36 regions, and acquire 35×35 rectangular finite elements. This number is much larger than those previously reported for vision modelling [30]. Then we calculated the modes of the free vibration of the plastic rectangle. The resulting modes are shown in figure 10.

It is argued in [30] that modal analysis is efficient for describing natural objects because it essentially ignores high frequency properties of the objects. But by contrast, we

[8]For the worm primitive the joint circles are attached to the rectangle, so they are not considered when we calculate the modes.

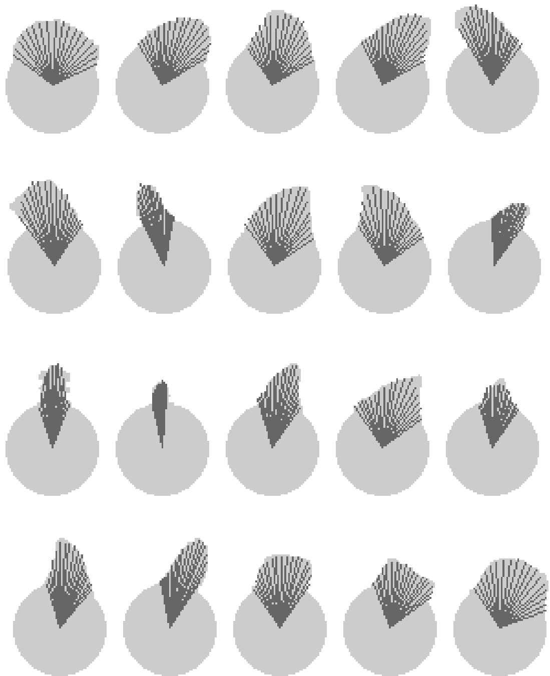


Figure 9 *20 typical circular parts with deformations shown as the shadowed areas. The reconstruction with the first 4 eigenvectors are shown by m dark lines. Reconstruction errors only occurs near the peaks of each deformation.*

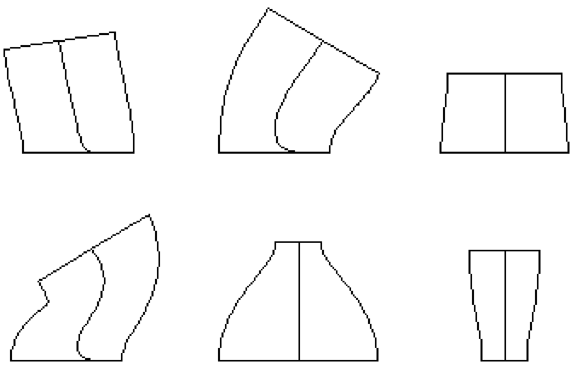


Figure 10 *The first six modes of free vibration are shown left to right and top to bottom sorted in increasing order of their corresponding vibration frequency*

argue that principal component analysis may often be more effective than modal analysis provided the necessary data is available. Firstly, modal analysis assumes a specific model for describing the material, see equation (1), which is problematic for animate objects. A more realistic model would have to take into account the material properties of bones and muscles, which currently seems impractical. Secondly, most of the variations of shape in animate object come not from elastic deformations but from statistical variations. Thirdly, in PCA the eigenvalues corresponding to the deformation modes give a measure of the amount of variation of these modes. Assuming that the data is approximately Gaussian then the distribution of the coefficients $(\alpha_1, ..., \alpha_k)$ will be a product of 1-D Gaussians with zero mean and variance proportional to the eigenvalues. This gives a distribution $P(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \ell(or \beta))$ for the parameters. In FORMS we therefore selected PCA instead of modal analysis to characterize the deformations of mid-grained shapes.

It should be stressed that these representations are intended to describe the plane projections of 3D objects. Therefore they will change with the viewpoints. In appendix A, we briefly discuss the influence of the viewpoint, the tilt and slant angles, on the parameters for simply shapes. The conclusion is that if the tilt and slant angle are small, say less than $\pi/6$, then the changes of parameters will be negligible (less then $1/24$). However, to describe an object in 3D from all viewpoints, we would need a set of characteristic views. This will be investigated in future work.

3.3 The Continuous Case

As the number of samples, n, tends to infinity we will obtain a continuous model $\vec{S}(t)$ for the deformations where $0 \le t \le 1$. This can be expressed as:

$$S(t) = S_0(t) + \sum_{i=1}^{K} \alpha_i \Psi_i(t). \tag{3}$$

where $S_0(t)$ is the basic, undeformed, shape (i.e. the mean in the PCA case) and $\Psi_i(t)$ is i-th deformation mode. Thus the higher order deformations $\Psi_i(t),\ i > K$ are all treated as noise.

This arises from taking the continuum limit as the number of sample points tends to infinity. In this case we replace the Kahunen-Loeve matrix, $\mathbf{K} = (1/N_s) \sum_\mu \vec{\phi}_\mu \vec{\phi}_\mu^T$ where $\mu = 1, ..., N_s$ labels the data samples, by a function $\mathbf{K}(t, \hat{t}) = \int \phi(t, \mu) \phi(\hat{t}, \mu) \rho(\mu) d\mu$, where ρ is a distribution over samples labelled by μ. Then the $\Psi_i(t)$ will be solutions of the equation

$$\int d\hat{t} \mathbf{K}(t, \hat{t}) \Psi(\hat{t}) = \lambda \Psi(t). \tag{4}$$

Observe if $\mathbf{K}(t, \hat{t})$ is shift-invariant, i.e. it is a function of $t - \hat{t}$ only, then the solutions will be sinusoid functions and equation (3) reduces to Fourier series expansion. It is easy to see that equation (2) is also a special case of equation (4). Therefore both Fourier series and modal analysis correspond to choices of $\mathbf{K}$ and hence, include prior assumptions about the statistics of shape variations. Thus they are likely to be less efficient than PCA which determines $\mathbf{K}$ from data samples.

3.4 Designing the shape grammar

Some shapes like hammers, hearts, and potatoes, can be described simply as mid-grained shapes. More complex objects, such as humans, fish, and leaves consist of a set of connected mid-grained parts, whose axes form the skeleton graphs. Such graphs can be described in a formal grammatical manner. Grammatical models of this type have been reported by Mandelbrot [20], Lindenmayer [18] and many others in computer graphics [36] [9]. They use simple line segments as primitives, and define grammatical rules to replace all line segments in the graph simultaneously to generate more complex graphs. In Mandelbrot's approach these rules will apply recursively to generate self-similar patterns. the segments of the tree have fixed lengths. Neither of these approaches is directly applicable to our problem.

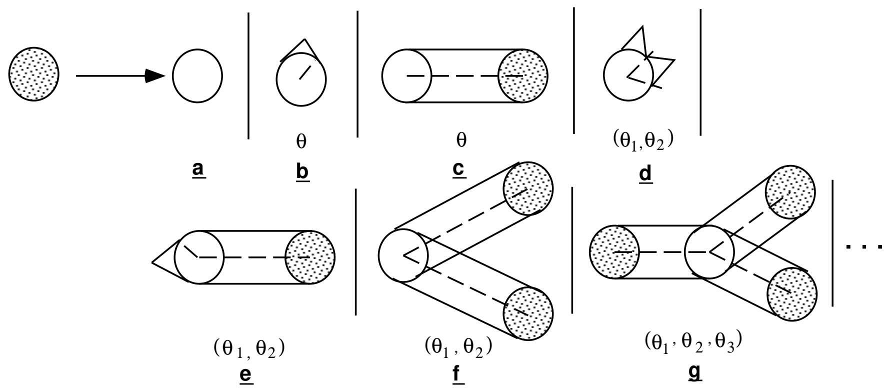


Figure 11 Grammatical rules. The dashed line in each primitive shows the direction of bifurcation. Observe that parts may overlap with each other.

The rules which we use are shown in figure 11, The shape starts from a *live cell*–the shaded circle on the left of 11(a). The live cell is able to split to generate any of the structures on the right of the arrow: (a) an empty circle, i.e. a barren cell, (b) a circle primitive with its deformed section starting at angle θ, (c) a worm part with its axis in the direction θ, (d), (e),(f) two joined parts with directions θ_1, θ_2, (g) three joined parts, etc. This process repeats on all live cells until there is no shaded circle remaining. For example, figure 12 shows how the model for a person can be generated using the above method. There is also a global rigid transformation which can be easily specified: (i) the translation and scaling of the whole shape is determined by the center and radius of the first live circle, (ii) each of the grammatical rules is parametrized by the bifurcation angles θ's, which determine the orientations for each part,

More formally: each animate model M will have several prototypical skeleton graphs $\{S_{(i)} \mid i = 1, 2, ...p\}$ due to changes of viewpoint and articulation. Each $S_{(i)}$ has an associated conditional probability $P(S_{(i)} \mid M)$ for how often $S_{(i)}$ occurs in our application domain. For each $S_{(i)}$, there are a set of grammatical rules $\{R_{(ij)} \mid j = 1, ..q\}$ for generating its specific structure. For some objects these rules are deterministic while for others they are probabilistic of form $P(\{part_{ij1}(\theta_1), part_{ij2}(\theta_2), ...\} \mid R_{(ij)})$. Finally, deformations

[9] See also work by Mjolsness on visual grammars [22].

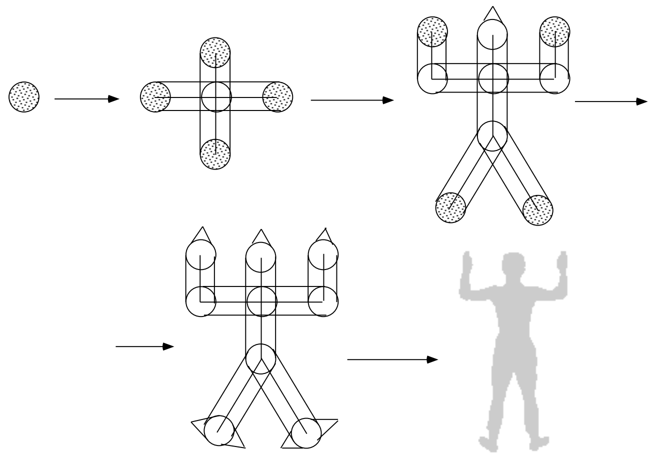


Figure 12 *Generating a human figure.*

of $part_{(ijl)}$ in rule $R_{(ij)}$ are given by the probabilities $P(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \ell(or\beta) \mid part_{(ijl)})$.

4 Obtaining the data representation—the bottom-up process

The previous section has described a probabilistic representation for animate objects. But how can we compute this representation from an input silhouette and match it to a database of model objects? One possibility would be to take advantage of the probabilistic model in section (3.4), in the spirit of Grenander's pattern theory [12], and to apply Bayes' theorem to directly obtain an optimization principle for estimating the data representation. Attempting to directly solve this optimization problem, however, would be difficult involving simultaneously determining the grammatical structure (i.e. the skeleton for animate objects), the number of mid-grained parts, the way they are joined, and the parameter values of the mid-grained parts. The grammatical model, moreover, would have to be general enough to generate all possible animate models and their possible articulations.

Instead we propose a step by step approach, see figure 1, which first calculates an initial representation for the input in a bottom-up process which only assumes weak knowledge about animate objects. This initial representation will be, to some extent, invariant to the scale and orientation of the input shapes[10], and it will include only small number of parametrized parts. But the representation obtained in the bottom-up process may not be perfect so a top-down process will be used to fix the matching residuals using the information in the database of models. Thus recognition, in FORMS, is a tightly coupled bottom-up/top-down process.

We leave the top-down matching process to section (5). The rest of this section is dedicated to describing the bottom-up process. We first propose a novel algorithm

[10] As we will discuss later, the representation is invariant to orientation but will slightly change with scale.

for calculating the skeleton – recovering the abstract grammatical structure –, then we partition the objects into parts according to its skeleton – recovering the mid-grained shapes –, then reduce the mid-grained shapes into primitives – recovering the primitives.

4.1 Obtaining the skeleton structure

4.1.1 Skeleton Basics

One way to recover the skeleton of an object is to calculate the *medial axis* [4] [5], which is also known as the SAT (symmetric axis transformation). There are several variations such as SLS [6] and PISA [17]. For our basic shape primitives in figure 5 the medial axis coincides with the central axis. But if the primitive undergoes local deformations and noise perturbations then the medial axis will become problematic. Local deformations and noise may not only cause additional branch axes but also distort the shape of the principal axis. Figure 13(b,c) shows how a small amount of noise on the boundary of the rectangle results in large changes in the skeleton of figure 13(a). Although there are a lot of algorithms [5] [6] [31] [10] for improving the medial axis by smoothing the boundary or using multiresolution schemes, as pointed out in [31] [28], the smoothing approaches may cause other problems. For example, they may not preserve the topology or the shape of the original skeleton.

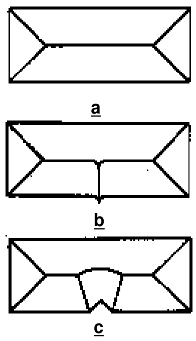


Figure 13 *The sensitivity of the medial axis to noise. (a) shows the ideal symmetry axes for a rectangle. (b) and (c) shows the distortions of these axes caused by small protrusions and indentations on the boundary.*

Another problem for the medial axis which is less often noticed is that the axis is ill-defined around bifurcation points, and thus is unreliable. In fact in a two dimensional image any bifurcation point of degree larger than three will be unstable. Figure 14(a) shows the medial axis for the two orthogonal strips, but when one strip slants a little (see figure 14(b)), one joint point will bifurcate into two. Similarly figure 14(c)(d) show how small smooth deformations on the boundary can perturb one joint point into three.

Our representation scheme in section (3) motivates a modular approach. The first module employs deformable templates and linear estimation to determine the skeleton of each mid-grained part, see section (4.1.2). The second module detects the bifurcation points, and deal with how mid-grained parts are joined together, see section (4.1.3).

In the following two subsections we describe these modules in depth. We will return

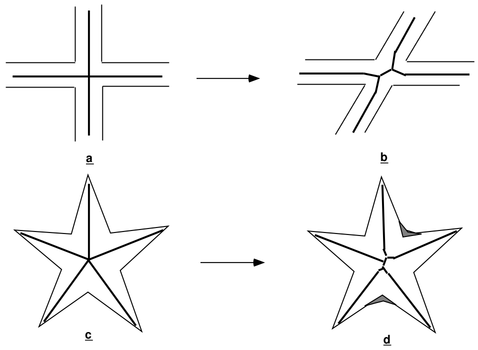


Figure 14 *The unreliability of the axis near bifurcation points*

to discuss the problems of the medial axis in later sections after describing our algorithm. Those readers who are not interested in thedetails of our algorithm are advised to skip to section (4.1.4).

4.1.2 Module 1: Estimating skeletons of mid-grained parts

First of all, we suppose that the image is preprocessed and the silhouette extracted, see section (6) for a discussion of how this might be done. The silhouette can be represented by a binary image $I(x,y)$ such that $I(x,y) = 0$ (black) if the pixel (x,y) is inside the shape and $I(x,y) = 1$ (white) otherwise.

Suppose (x,y) is a point on the axis of a mid-grained part and r is the rib at (x,y). Then (x,y,r) defines the circular domain $\mathcal{D}$, centered on (x,y) and radius r, shown in figure 15(a). To localize the point (x,y) and find its rib r, we design a deformable circle called the free traveling circle (FTC) whose behavior is determined by the following energy function:

$$\mathcal{E}^F_{(x,y,r)} = \int\int_{\mathcal{D}} (r - \sqrt{u^2+v^2})\mathcal{I}(x+u, y+v)\,du\,dv - \frac{\alpha}{a} r^a \tag{5}$$

The dynamics of the FTC center. By gradient descent with respect to x and y, we have the motion equation for the center of the FTC:

$$\begin{pmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{pmatrix} = -\begin{pmatrix} \frac{\partial \mathcal{E}^F}{\partial x} \\ \frac{\partial \mathcal{E}^F}{\partial y} \end{pmatrix} = \int_{\Gamma \cap \mathcal{D}} (r - \sqrt{u^2+v^2})(-\nabla \mathcal{I})\,ds \tag{6}$$

where Γ is the boundary of the shape and $-\nabla\mathcal{I}$ is the negative gradient direction. In the binary case, $-\nabla\mathcal{I}$ is nonzero only on the boundary Γ. It is always perpendicular to the boundary and points into the shape as shown in figure 15(b)[11] We can consider $-\nabla\mathcal{I}$ as a force which is weighted by $r - \sqrt{u^2+v^2}$, so the closer the pixel to the center (x,y), the larger the force. Thus according to equation (6), the motion of the center (x,y) is

[11] Strictly speaking $\nabla\mathcal{I}$ becomes infinite at the boundary and can only be defined there using distribution theory. This will become irrelevant when we discretize the theory for implementation.

the result of the overall force integrated along the boundary inside the circle, as shown in figure 15(c,d). The center will converge when forces from all directions are balanced.

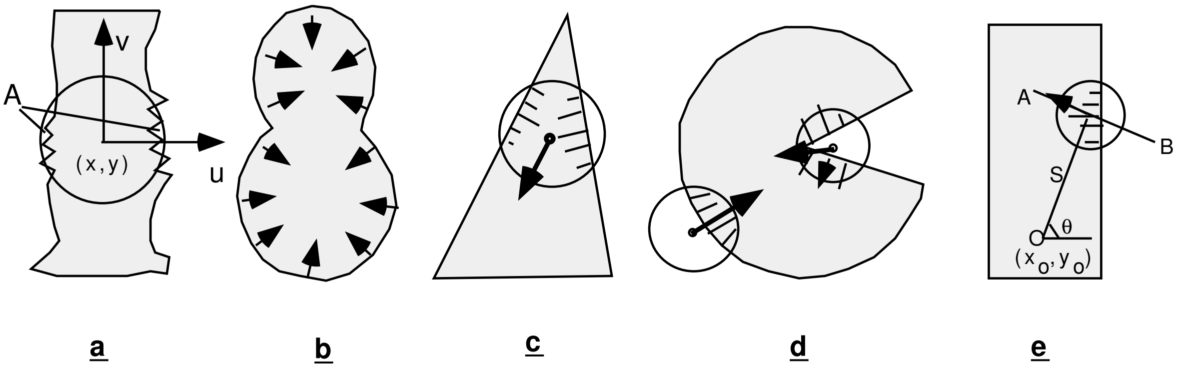


Figure 15 *the motion and convergency of the deformable circles.*

The dynamics of the FTC radius. By steepest descent with respect to r, we have the motion equation for the radius of FTC:

$$\frac{dr}{dt} = -\frac{\partial \mathcal{E}^F}{\partial r} = -\int\int_{\mathcal{D}} \mathcal{I}(x+u, y+v)dudv + \alpha r^{a-1} \tag{7}$$

In equation (7), the first term is the total area of "white" pixels included in the circle domain D, see region A in figure 15(a). It represents a force which always makes the circle shrink whenever the circle domain D contains pixels outside the shape. The second term in equation (7) represents a stretching force to enlarge the circle. The circle will converge when the area of A is αr^{a-1}. This allows the circle to tolerate some noise on the boundary [12]. How much noise the circle will tolerate depends on the choice of α and a.

(i). If $a = 1$ then $A = \alpha$. Thus, after convergence, all circles will includes the same amount of noise (i.e. the same number of white pixels). This is undesirable, since α may be too large for a small circle, and too small for a large one. If $a = 1$, and $\alpha \to 0$, then the deformable circle degenerates to the maximal circle defined by the medial axis transformation.

(ii). If $a < 3$, we are able to guarantee the convergence of the FTC equation (7). Because after the circle gets larger than the shape region, the area A (or the shrinking force) will grow at proportional to r^2, while the stretching force grows only as $\alpha r^{a-1} < r^2$.

(iii). The empirical choice used in this paper is $a = 1.5$. A drawback is that the radius r, after convergence, will not be invariant to the scale of the image. However, we tested the program for calculating the FTC on circles with radii 8,16,32,64 and obtained convergence results of 7.9, 15.9, 31.2, 63.2 respectively. Therefore the FTC is almost invariant to the scale of the images.

Once the first point on the skeleton is localized, estimating the skeleton for each mid-grained part proceeds as a stepwise tracking process. As shown in figure 16(a), let

[12]This smooths the boundary adaptively. The larger the scale, the more the smoothing.

$P_n = (x_n, y_n)$ be a point on the skeleton estimated at time step n. We first predict the next point as $\hat{P}_{n+1} = (x_n, y_n) + \hat{S}_n(cos\hat{\theta}_n, sin\hat{\theta}_n)$ (see the dashed line) with $\hat{\theta}_n$ being the estimated tracking direction, and $\hat{S}_n$ the estimated step size . The prediction $\hat{P}_{n+1}$ is then refined to point P_{n+1} on the dark solid line by the deformable circle. In this manner, we are, in fact, approximating the skeleton of each mid-grained part with a chain of short line segments. Such approximation is justifiable by referring to the anatomical analysis for the bones of vertebrates in figure 2.

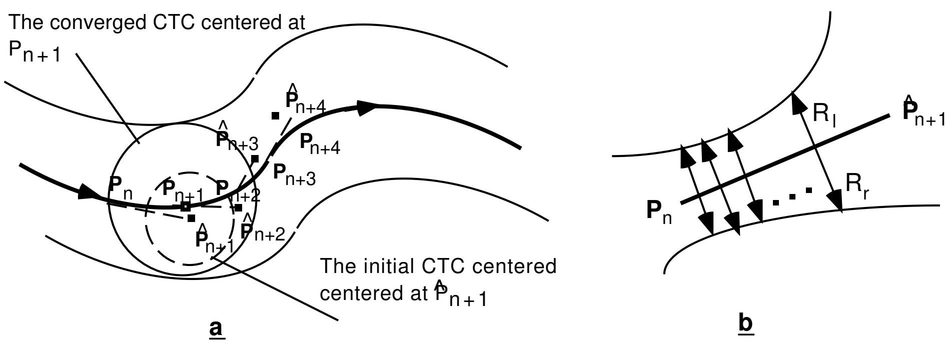


Figure 16 *Tracking the skeleton by linear prediction and error correction*

Figure 16(b) shows how $\hat{\theta}_n$ is calculated. Let $R_r(\xi, \eta)$, and $R_l(\xi, \eta)$ be the perpendicular distances from point (ξ, η) on $P_n\hat{P}_{n+1}$ to the right and left boundaries. $\hat{\theta}$ is calculated easily by minimizing the sum of square errors:

$$Error(\hat{\theta}_n) = \int_{P_n\hat{P}_{n+1}} (R_l(\xi,\eta) - R_r(\xi,\eta))^2 d\iota$$

Because R_l, R_r are not analytical functions, in practice, we calculate $Error(\hat{\theta}_n)$ around the old tracking direction θ_{n-1}, and pick up the best $\hat{\theta}_n$. We also need to measure σ_n which shows how accurate the estimated $\hat{\theta}_n$ is. For simplicity, once $\hat{\theta}_n$ is calculated, for each point (ξ, η) on $P_n\hat{P}_{n+1}$, we define $\Delta\theta(\xi, \eta) = \| \hat{\theta}_n - \arctan(\frac{\|R_r(\xi,\eta) - R_l(\xi,\eta)\|}{\sqrt{(\xi - x_n)^2 + (\eta - y_n)^2}}) \|$, then σ_n is the variance for the set $\{\Delta\theta(\xi, \eta) \mid \forall(\xi, \eta) \in P_n\hat{P}_{n+1}\}$.

The tracking step size $\hat{S}_n$ can be estimated based on $Error(\hat{\theta}_n)$. For example, the smaller $Error(\hat{\theta}_n)$, the larger $\hat{S}_n$. But such estimation is computationally expensive and moreover, the step size should be constrained by some global conditions, i.e. if $\hat{S}_n$ is too large, the skeleton tracking step may pass by a bifurcating point on the skeleton. In our implementation, we choose $S = 4$.[13].

Then based on these initial estimations $\hat{\theta}_n$ and σ_n, we localized the next point P_{n+1} by moving a deformable circle using $\hat{P}_{n+1}$ as the initial point for the center of the circle. Since we fix the tracking step size S, the deformable circle now has only two degrees of freedom–θ_n and r_{n+1}, We call it the constrained traveling circle (CTC). The CTC center

[13]The images scale we used are around 128×128

is $(x_{n+1}, y_{n+1}) = (x_n + Scos\theta_n, y_n + Ssin\theta_n)$, and the energy for CTC is:

$$\mathcal{E}^C_{(\theta_n, r_{n+1})} = \mathcal{E}^F_{(x_{n+1}, y_{n+1}, r_{n+1})} + \lambda \frac{(\theta_n - \hat{\theta}_n)^2}{\sigma_n^2} \tag{8}$$

The additional term is to constrain the motion of the center to a limited angle centered on the prediction.

By gradient descent, we have:

$$\frac{dr_{n+1}}{dt} = -\frac{\partial \mathcal{E}^C}{\partial r_{n+1}} = -\frac{\partial \mathcal{E}^F}{\partial r_{n+1}} \tag{9}$$

$$\frac{d\theta_n}{dt} = -\frac{\partial \mathcal{E}^F}{\partial \theta_n} - 2\lambda \frac{(\theta_n - \hat{\theta}_n)}{\sigma_n^2} \tag{10}$$

$$= -S \begin{pmatrix} \frac{\partial \mathcal{E}^F}{\partial x_{n+1}} \\ \frac{\partial \mathcal{E}^F}{\partial y_{n+1}} \end{pmatrix} \cdot \begin{pmatrix} -sin\theta_n \\ cos\theta_n \end{pmatrix} - 2\lambda \frac{(\theta_n - \hat{\theta}_n)}{\sigma_n^2} \tag{11}$$

The motion equation for r is the same as for the FTC's. The motion of the center is determined by two forces, the first term stands for the force which is the projection of the overall force in the FTC onto the direction perpendicular to θ_n, see line AB in figure 15(e). Because S is fixed, the center moves along an arc. The second term constrains the change of θ_n. In the implementation, we choose $\alpha = 4.5, \lambda = 10.0$, and if $\sigma < 2.5^o$, i.e. the estimation from the least square error is very reliable, we do not need to update θ_n in the CTC.

4.1.3 Module 2: Bifurcation analysis

The second module is implemented to detect whether the skeleton should bifurcate. This task is performed by calculating the *range-angle function*. This is similar to techniques used in sonar analysis, see figure (17).

Imagine that the circle sends a set of rays R_{θ_i} from its center c at angles $\{\theta_i \in [\theta_{low}, \theta_{low} + \Delta\theta, \theta_{low} + 2\Delta\theta, ..., \theta_{high}],\}$ where θ_i is the angle between the ray and the horizontal axis, and we choose $\Delta\theta = \frac{\pi}{36}$. Each ray R_{θ_i} reports the distance $r(\theta_i)$ from c to the boundary and we define $\{(\theta_i, r(\theta_i)\}$ to be the range angle function.

To detect the object boundary in binary image is easy. Each ray grows from the center along angle θ_i until it meet a "white" pixel. Due to the imprecision of the discrete image, we use a $b \times b$ kernel centered at the tip of the ray to investigate the pixels near the tip. If there are more than d white pixels within the $b \times b$ kernel, then the ray stops growing. We use $d = 2, b = 5$ for images of size of order 128×128.

The range-angle function is usually noisy. A typical profile has small local peaks imposed on large global peaks in a hierarchical structure. Each peak represents a possible branch in the shape's skeleton. First of all, we pick up a set of angles $\beta_1, \beta_2, ..., \beta_n, (\beta_i < \beta_{i+1}$ for i=1,2,...,n-1) with each β_i satisfying: (I) $r(\beta_i)$ is a minimum in the range-angle function, (II) $r(\beta_i) \leq r_n$, where r_n is the radius of the current traveling circle, i.e. we

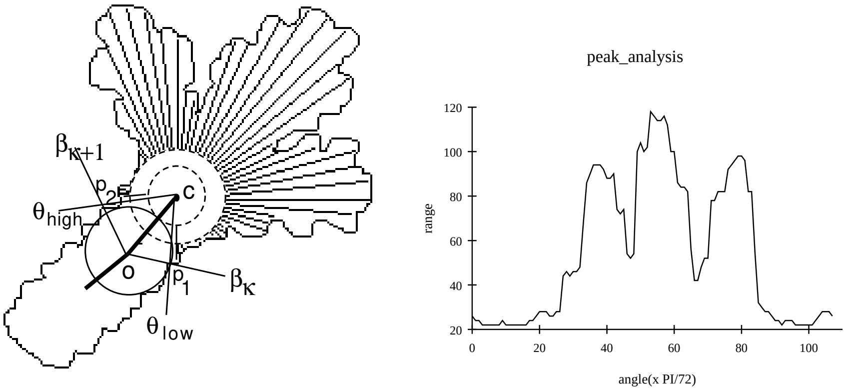


Figure 17 Sonar rays and range angle function

ignore all the possible bifurcation far away. [14]

So for each $0 < i < n-1, [\beta_i, \beta_{i+1}]$ corresponds to a peak in the profile. It may result from a branch, a local deformation, or noise, depending on how salient it is. By observation we noted two important cues for determining the salience of peaks: (i) a wide broad peak and (ii) a thin deep peak. Thus the saliency of a peak is measured by two properties: (1) the maximal range R_{max} within the peak $[\beta_i, \beta_{i+1}]$, (2) the area A within the peak – $A = \frac{1}{2}\int_{\beta_i}^{\beta_{i+1}} r^2 d\theta$.

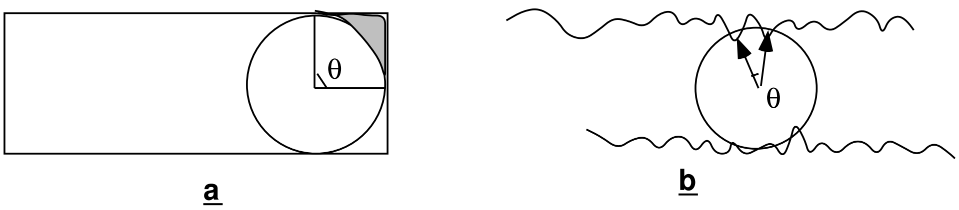


Figure 18 (a) see text. (b) the saliency measure must be chosen so that noise fluctuations on the boundaries do not cause bifurcations of the skeleton.

The precise form of $S[\beta_i, \beta_{i+1}]$ is set by evaluating it for a reference shape, in this case a rectangle, see figure 18(a). The rectangle's corner has $R_{max} = \sqrt{2} \cdot r_n$, $Area = r_n^2$, where r_n is the radius of the current CTC. We set the saliency of this corner to be 0.75 which serves as a reference point. The saliency of the peak $[\beta_i, \beta_{i+1}]$ is then defined by:

$$S[\beta_i, \beta_{i+1}] = 0.75 \cdot max(\frac{R_{max} - r_n}{(\sqrt{2}-1)R}, \frac{A}{R^2})$$

[14] It is possible, in this situation, to fit a bifurcation model if we know *a priori* that the current node is a joint node of M mid-grained parts. Then we could fit M peaks to the range-angle function and partially relieve the problems discussed earlier in figure 14. But such prior knowledge is not available until the shape has been recognized, see next section

This is actually an adaptive threshold, i.e. the saliency of a peak depends on the scale of the current deformable circle. Moreover, we set $R = max(r_n, 10) \geq 10$ to ignore the peaks when the radius r_n is very small.

Because the saliency measure is continuous it is not always easy to pick a single fixed threshold to distinguish between peaks due to genuine bifurcations of the skeleton and those caused by local deformations and noise. Hence, as for edge detection agorithms [8], we choose two threshold values, T_{low}, T_{high} (see figure 19).

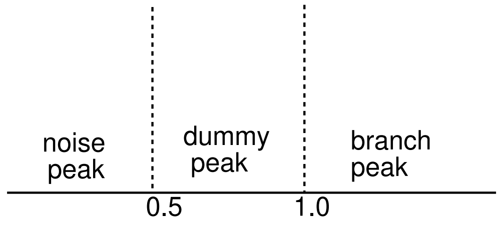


Figure 19 Bi-threshold classification of bifurcation points.

Suppose that the saliencies are measured and N branch peaks are found. There are three cases:

1. If $N = 0$, then the traveling circle has come to the end of a branch. The center of the traveling circle is defined to be a **E-node**.
2. If $N = 1$, i.e. there is only one branch peak $[\beta_k, \beta_{k+1}]$, then the traveling circle is in the middle of a branch.
3. If $N > 1$, then the center of the current CTC is a **B-node**–i.e. a bifurcation point of the skeleton.

Associated with each peak k are the angles β_k, β_{k+1}, which define two point p_1, p_2 on the boundary (see the angles shown around point o in figure 17). When we obtain the next point c in figure 17, the scan angles $\theta_{low}, \theta_{high}$ are calculated so that the first and last sonar ray pass through p_1, p_2 respectively.

Before ending the skeleton algorithm, two further points need to be addressed.

First, the measurement of peak saliency will be problematic when a peak is almost invisible from the center c. This situation occurs at the tails of animals in our application (see figure 20(a)). There are many ways to deal with this special case. For example, track along the boundary of the shape clockwise from A to B, whenever a peak is classified as non-branch peak, to see whether some part is missing. In FORMS, we simply use some points (named "scout") along the arc of the deformable circle between A and B (see the dark dots), from where sonar rays are sent to detect the boundary as shown in figure 20. It may still fail if the tail is very "jagged", but such case never happen in our animate shapes displayed later, because the bones of animals cannot be too "jagged".

Second, suppose the circle at point O in figure 20(b) is a converged traveling circle, it detects a branch at angle interval $[\theta_{low}, \theta_{high}]$ shown by the shadow region. This situation

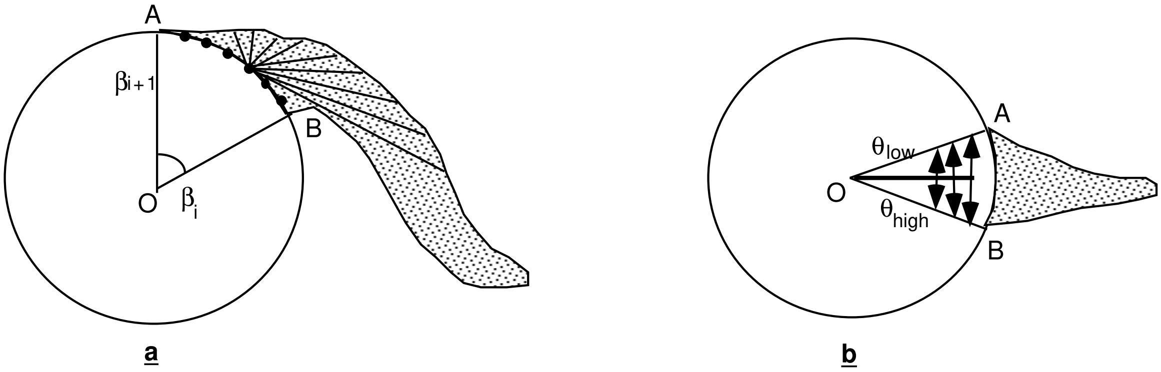


Figure 20 Two further points for the skeleton calculation. (a) is the special case for peak detection. (b) is the detailed point in estimating the tracking theta. The circles are the converged deformable circle.

often happens when O is a bifurcation point on the skeleton. To estimate the tracking direction $\hat{\theta}$ discussed in module I, we need to measure the perpendicular distances $(R_l, R_r)'s$. But since the boundary around point O outside the angle interval $[\theta_{low}, \theta_{high}]$ is almost irrelevant to the tracking direction θ of the current branch, we instead use the line segments OA OB as the boundary shown in figure 20(b) while measuring $(R_l, R_r)'s$.

4.1.4 The results of the skeleton algorithm

The resulting skeletons for some typical objects are shown in figure 21. This figure also illustrates some important phenomena where the bifurcations seem unintuitive as we discussed in figure 14: (i) the *bifurcation delay phenomenon* (**BDP**), see the bifurcation of the rear legs of the dog, which usually results from self occlusion; (ii). the *bifurcation splitting phenomenon* (**BSP**), see the bifurcation of the front legs of the dog. Both BDP and BSP are not the fault of the skeleton algorithm. Without using a prior model for a dog there is no reason to require that the two bifurcation points at the front and rear parts of the dog be coincident. Such "errors" will be corrected, see figure 28, in the top-down stage where we directly match models of hypothesized objects [15].

To test how well the algorithm work on noisy images, and on images in other domains, we ran the program on a noisy leave and an image from an MRI medical image of the brain. The results are shown in figure 22. All the skeletons in figures 21,22 are calculated using the same set of parameters and thresholds.

4.2 Recovering mid-grained shapes and the primitives

After recovering the skeleton it is still a nontrivial problem to partition the object into a set of mid-grained parts. Bifurcations in the skeleton correspond to joining together mid-grained parts, but how exactly should these parts be joined? Usually people [26],

[15] The skeleton of the object will depend weakly on the initial starting point of the algorithm. To remove this slight ambiguity we choose a starting point determined by global properties of the object, such as the center of mass.

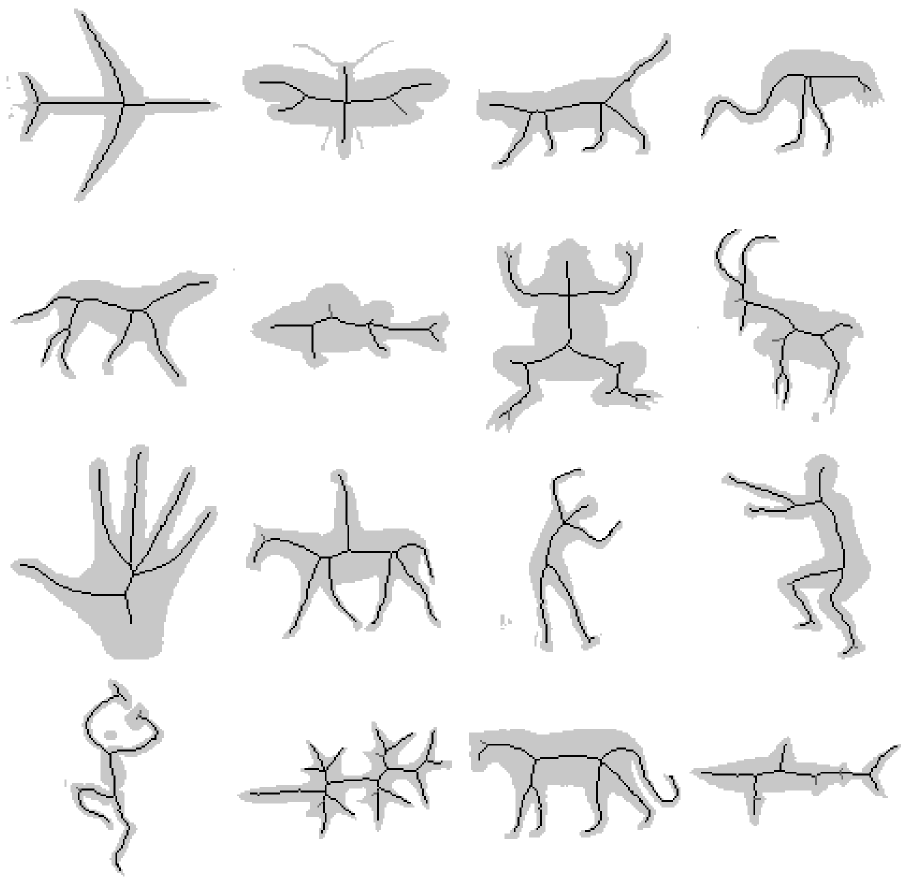


Figure 21 Skeletons of typical objects.

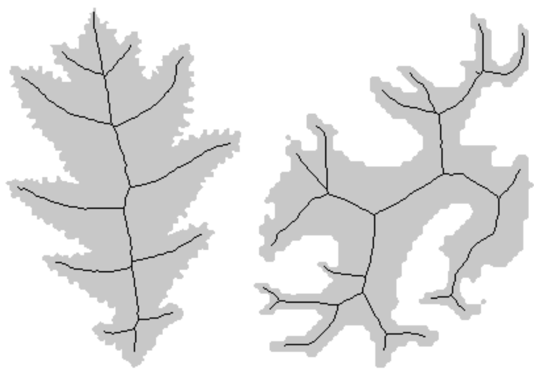


Figure 22 The skeletons calculated for a noisy leaf and an MRI image of a cross section of the brain.

[6] cut parts at minimum curvature points. But such a partition may be bad because: (i) curvature minima may not correspond to important features for flexible objects and may be unstable, (ii) after cutting away all the parts there remains a joint polygon, see figure 24(a), and (iii) sometimes the cutting of one part affects the shape of the rest. For example, in figure 24(b), a desirable partition for the dorsal fin of the fish is along a curve instead of a straight line.

In FORMS we join mid-grained parts together by overlapping their corresponding joint circles. The intuition is that these joint circles form a "hinge" about which the parts can rotate. Each part has an angular segment $[\beta_i, \beta_{i+1}]$, which is exclusive to it. As shown earlier in figure 4(c), to partition a part, we simply remove one-layer of joint circle and the connected area within the angle section $[\beta_i, \beta_{i+1}]$. Intuitively, we do not saw off the trunk but instead unearth the "root".

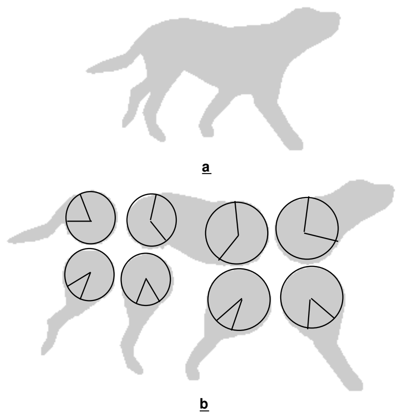


Figure 23 The dog and its segmentation into parts.

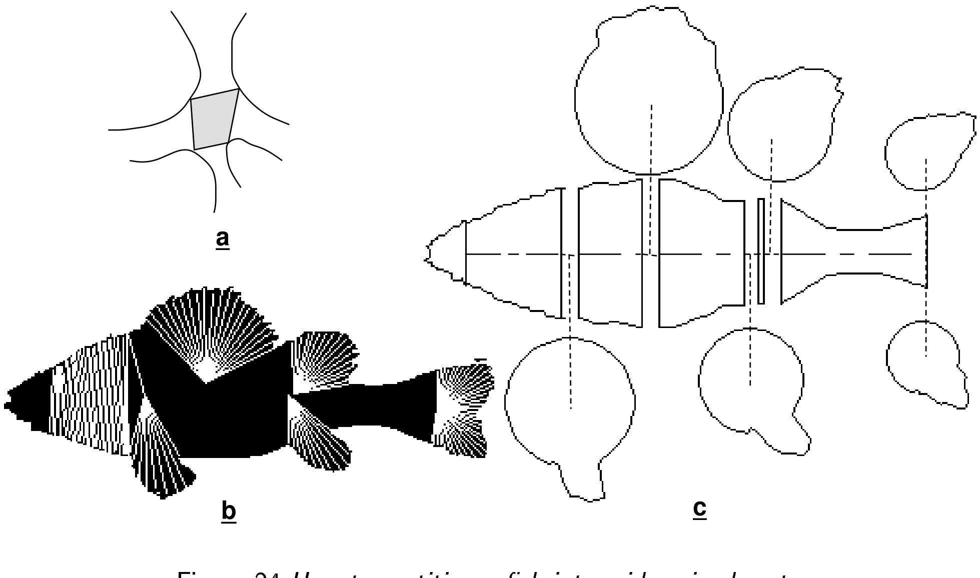


Figure 24 *How to partition a fish into mid-grained parts*

A part whose axis is between two B-nodes is classified as a deformed worm part. A part with axis between a B-node and a E-node will be considered as worm part if the length of its axis is larger than t times the radius of both deformable circles at the B-node and E-node, otherwise it is temporarily classified as a deformed circular part. We choose $t = 1.5$ here. The classification will be finally determined by the model.

The partition of animals seems intuitive, see figure 23, while the segmentations for shapes like fish are relatively hard. As an example, figure 24 shows the details of how FORMS segments a fish whose skeleton has been shown in figure (21). There are a total of five bifurcation points (B-nodes) along the axis. Virtual lines in figure 24(c) are drawn to connect parts which share the same joint circle at each B-node. There are seven end points (E-nodes) on the skeleton, one of which is at the top of the head, and the rest are on the fins and tails.

In figure 24, since the axes for the fins and tails are short compared with the radii of their corresponding joint circles the six parts are considered as circular parts (see figure 5(b)) for which no rectangle description is needed. The white lines in figure 24(b) are either: (i) radial lines stemming from B-nodes, see the fins and tails, for detecting the variations on the boundary within the angle section $[\beta_i, \beta_{i+1}]$, or (ii) lines perpendicular to the axis to measure the ribs, see the head.

After the mid-grained parts have been extracted, we compute their characteristic vectors and then project them onto the parameterized deformable primitives described in sections (3.1) and (3.2). For most animate objects the axes for these parts are almost straight lines, and the deformations of these axes are usually rigid transformations. Thus for the purpose of recognition, we ignore the detailed shape of axes for the worm parts. This is why we did not apply the principal component analysis on the axis vectors in section (3.2).

5 Matching and recognition

5.1 Data structures for representation

The process described in the previous section yields a representation of the input object in the form of a skeleton graph with each edge in the graph representing a parameterized mid-grained part. The models for each object in the database are represented in the same way. Figure 25 is the model for a person. The two short parts on the feet model the heel and the toe respectively.

We organize the models of all the objects into two databases. One collects all the mid-grained parts of the animate objects as shown in figure 26, and is named the *butcher's shop*. It can be considered as a content addressable memory. Each cell hung in the butcher's shop contains two items: (i) the deformation parameter vector $(\ell, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$ or $(\beta, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$, and other parameters such as the area, the radius of the joint circles of this part. (ii) the label indicating what objects it belongs to, for example, the *head* of the *dog*. The other database contains the skeleton graphs of all objects, with each object having several skeleton graphs due to changes of viewpoint and articulations.

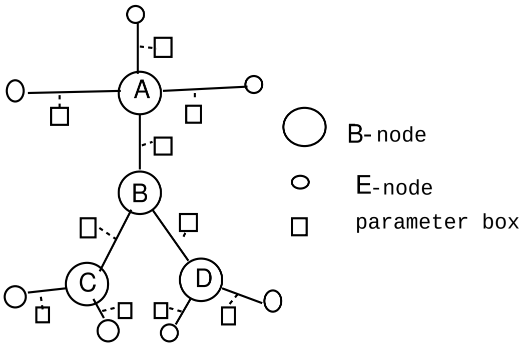


Figure 25 The model for the human body. The skeleton graph.

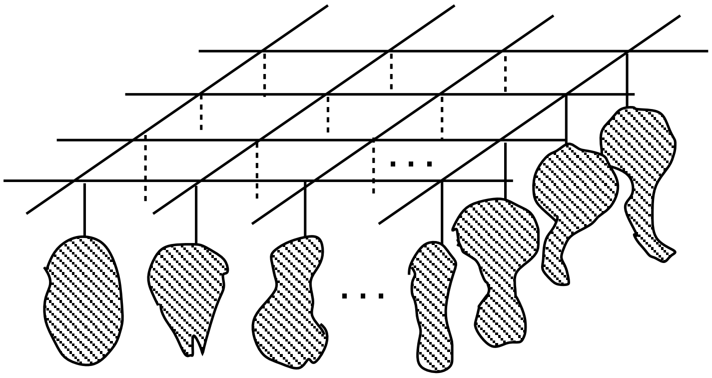


Figure 26 **The butcher's shop.**

5.2 The measurement of similarity

How to measure the similarity between objects is poorly understood both in mathematics [27] and in psychology [23]. It seems that humans often use subjective criteria in their assessment of how similar shapes are. Human perception of the similarity of shapes is determined not only by geometric properties but also by semantic content, and the latter is, in turn, influenced by the observer's cultural and social background [27]. Moreover, people have argued that perceptual similarity between shapes is not symmetric and hence cannot be described by a metric [23]. Similarly, Leyton has argued that shape representation proceeds by a process of ordered deformations [17] (i.e. an ellipse is considered to be a deformed circle, but a circle is not considered to be an undeformed ellipse.).

Our shape modeling framework discussed in section (3) also organizes shapes into a partially ordered space based on deformations of primitives. These deformation processes might be related to Leyton's results on human perception [17].

We argue that only objects which can be put in the same structured grammatical framework can be meaningfully compared. We consider geometrical similarity between shapes and will define similarity in a statistical sense.

First we need to define the similarity between two mid-grained parts. Suppose that $m = worm(\ell_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$ is a mid-grained part for model shape $\mathcal{M}$. Because the deformation modes are obtained by the use of principal component analysis, which can be thought of as computing the eigenvectors of the covariance matrix of a multi-dimensional Gaussian, it is consistent to assume that the variations of mid-grained parts are subject to the Gaussian distribution. Therefore, if $d = worm(\ell, \beta_1, \beta_2, \beta_3, \beta_4)$ is a part in the input shape $\mathcal{D}$, we define the similarity between parts m and d as the probability:

$$P_{match}[m,d] = \frac{1}{Z} exp^{-(\sum_{i=0}^{4} \frac{(\alpha_i - \beta_i)^2}{2\sigma_i^2} + \frac{(\ell - \ell_0)^2}{2\sigma_0^2})} \tag{12}$$

The σ's are the variances of the deformation parameters for model part m. [16] The similarity between circular parts is defined in the same way as equation (12).

Now, let $\mathcal{M} = [m_1, m_2, ..m_n]$ be the model which is matched against the input object $\mathcal{D} = [d_1, d_2, ..d_n]$. Suppose that the match Φ between the large-grained shapes $\mathcal{M}$ and $\mathcal{D}$ corresponds to matching the mid-grained parts so that $\Phi(m_i) = d_i$ for i=1,2,...n. The Φ must, of course, match the grammar structures described in the planar tree-like skeleton graphs of both the model and the input shape. It is reasonable to assume that the presence or absence of one part in the object is independent of the matching of the others. Thus we define the similarity between $\mathcal{M}$ and $\mathcal{D}$ under the match Φ as the probability[17]:

$$P_\Phi[\mathcal{M}, \mathcal{D}] = probability\{m_i \ \ matches \ \ d_i \mid i = 1, 2, ..., n\} \tag{13}$$

[16]Theoretically, the σ's could be estimated by statistics on the distribution of parameters for model part m. Because we only have a few sample shapes for each model M, not enough for statistical analysis, the σ's are choose to be constants for all models. For worm parts we choose $\sigma_i = 2.18$ and for circular parts we choose $\sigma_i = 2.88$ i=0,1..4.

[17]The relative orientation between mid-grained parts joined at the same joint B-node (i.e. θ's described in section (3.4)) will be certainly useful information for similarity, but for simplicity, it is ignored here

$$= \prod_{i=0}^{n} P_{match}[m_i, d_i] \quad (14)$$

If the match Φ is incomplete, in the sense that some parts of the model or the object are unmatched, then we define the similarity between $\mathcal{M}$ and $\mathcal{D}$ as the probability:

$$P_\Phi[\mathcal{M}, \mathcal{D}] = \prod_{\Phi(m_i) \neq \emptyset} P_{match}[m, \Phi(m)] \prod_{\Phi(m) = \emptyset} P_{missing}[m] \prod_{\Phi^{-1}(d) = \emptyset} P_{extra}[d] \quad (15)$$

where $P_{missing}[m]$ is the probability for part m in the model $\mathcal{M}$ to be missing from the observed input shape, and $P_{extra}[d]$ is the probability for part d in the input shape $\mathcal{D}$ to be a redundant part in the input, resulting from segmentation errors or because the input has an additional parts (for example, if the input is a human holding an umbrella in his hand). We could also incorporate some subjective semantics into each part of the model. For example, for a less important part m, we can define $P_{missing}[m]$ be close to 1.0, while for important parts $P_{missing}[m]$ should be close to 0.0. The latter means that if an important part of $\mathcal{M}$ is not observed in the input shape $\mathcal{D}$, then the probability of $\mathcal{D}$ being object $\mathcal{M}$ is very low. For example, the torso of a person is much more important than the feet for several reasons: (i) it is far larger, (ii) is a central part of the object rather than a peripheral one, and (iii) it is visible in far more viewing situations.

For simplicity, we define $P_{missing}$ and P_{extra} in the same way as followings:

$$P_{missing}[m] = \frac{1}{Z_1} exp^{-\lambda \frac{A(m)}{\frac{1}{n}\sum_{i=1}^{n} A(m_i)}} \quad (16)$$

$$P_{extra}[d] = \frac{1}{Z_2} exp^{-\mu \frac{A(d)}{\frac{1}{n}\sum_{i=1}^{n} A(d_i)}} \quad (17)$$

where λ and μ are scaling constants, and $A(m)$ is the geometric measurement determining the relative importance of part m in the model $\mathcal{M}$. In our implementations we choose $A(m)$ to be the area of part m, and similarly for $A(d)$, and we set $\lambda = \mu = 2.8$.

5.3 Matching and recognition

Let $D = [d_1, d_2, ..., d_n]$ be the input object, where the d_i are the mid-grained parts obtained using the methods described in section (4). Then the recognition proceeds in two steps:

First, for each mid-grained part d_i $(i = 1, 2, ...n)$, we hash the butchers shop to find the most similar mid-grained parts $m_1, m_2, ..., m_k$ so that $P_{match}[m_1, d_i] > P_{match}[m_2, d_i] > ... > P_{match}[m_k, d_i]$. Then the object models indicated by the labels of the m's receive credits $c_1 > c_2 > ... > c_k$ respectively. After performing this search and credit assignment for all d_i $(i = 1, 2, ...n)$, we select the m models whose credits are the highest. The accuracy of the match at this stage will depend on how accurately the skeleton is calculated by the bottom-up process. The first step is only needed when the database is really big [18]

[18] In our experiments, we used $m = k = 3$, and $c_1 = 10, c_2 = 5, c_3 = 3$ and found that in all but two cases out of thirty five the model with the highest credit is in the top three. In both these cases the number of parts is ambiguous.

Second, for each model $\mathcal{M}$ recommended by the first step, we need to find the best match between all skeleton graphs (due to the changes of viewpoint and articulation) of the model $\mathcal{M}$ and the input shape $\mathcal{D}$ using the similarity criterion defined in the previous section.

The matching proceeds basically as the branch-and-bound algorithm. It searchs over all possible matches between the skeleton graphs of the model and input shape on an and-or tree, [19] and trims those branches in the and-or tree whose costs are too large. If the data representation found in the bottom-up process is perfect then this would simply correspond to a weighted subgraph matching. But as we discussed in section (4) it is impossible to extract the perfect skeleton without using model specific information. Therefore, integrated into the searching algorithm is the top-down verification process.

Two classes of problems need to be fixed in this top-down process. First, the skeleton structure may be wrong, as discussed in section (4.1.1). For example, a B-node may split into several B-nodes due to slight deformations on the boundary. Also a circular part may be miss-interpreted as noise. Secondly, the primitives derived from the skeleton may be wrong, i.e. we may get confused between circular parts and elongated worm parts.

To treat the unreliability of the skeleton structure resulting from the bottom-up process, we employed a group of skeleton operators each of which can transform the skeleton graph into a new one. By applying these operators in sequence we can get a large number of possible skeleton graphs for the input shape which can be matched against the model. As shown in figure 27, we mainly used four skeleton operators: cut, merge, shift, and concatenate (see the caption for details). These operators are applied whenever matching residuals with the model are detected.

Theoretically these four skeleton operators are enough to adjust for the possible errors occurred in the skeleton calculation step. Only two kinds of error arise. The first is the presence of an extra branch or even an extra sub-graph due to noise or real extra objects, like a man on the horse. The opposite case when a real branch, or sub-graph, is absent can be treated similarly because it means that the corresponding branch, or sub-graph, in the model is extra. If the extra branches happen to appear at the B-node, then the cut operator in figure 27 can cut the extra branches correctly. Otherwise, if the extra branches appear between two nodes, like d_3 in figure 27(c), then the concatenate operator will cut the extra part and join the separated two part together. The second kinds of possible errors in the skeleton are the B-node splitting cases discussed in figure 14 – due to small changes in the boundary a B-node may split into several B-nodes. Similarly the opposite case when several B-nodes coincide by accident, which rarely happens, can be treated as a node splitting case in the model. The merge operator and shift operator are designed to deal with bifurcations. The difference between these two operators is that they treat different B-nodes as the "true" B-node.

When adjusting the skeleton, the new skeleton segments are calculated by interpolating the maximal circles for the worm parts and re-estimating the radials for circle parts. When a new part is generated, we need to represent it as a deformed primitive shape, by projecting it onto the deformation modes, and measuring the parameters. We include

[19] Each node of the and-or tree is a state which records the matching or partial matching with cost measurements.

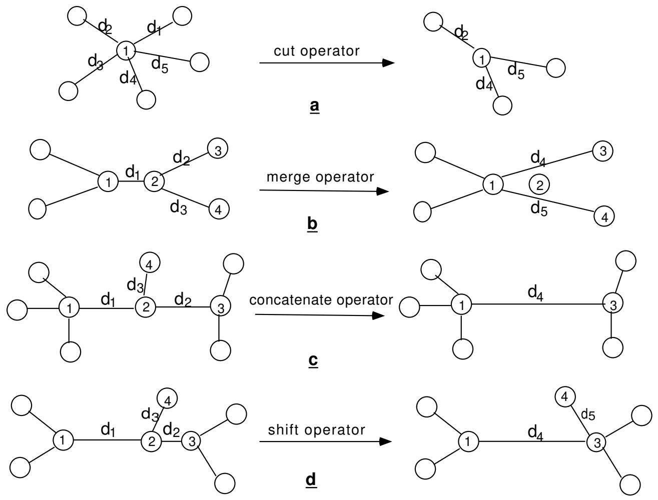


Figure 27 **the skeleton operators**. This figure shows the skeleton operators which are used to fix the matching residuals. a). if node 1 is matched to a node with degree=3 in the model, then two out of five branches should be cut off. There are ten possible combinations. b). if node 1 is matched to a node with degree=4 in the model, then node 1 tries to find another branch by merging node 2 with itself. c).when node 1 is matched, then each branch connected to node 1 should be matched with the corresponding branch in the model. The adjustment is to concatenate d_1 and d_2 and to consider branch d_3 as noise or an extra part. d). in contrast to case c). we shift d_3 to join node 3.

costs for applying the skeleton operators. For example, for the cut and join operators may pay the cost $P_{extra}[d]$ for a discarded part d, such as d_3 in figure 27(c).

The ambiguity between noise blobs and circular parts is represented by dummy branches in the skeleton graph. When a B-node of degree d is matched to a B-node of degree m in the model, if $m > d$, then the algorithm needs to find the $m - d$ missing branches. One way to do this is to apply the merge and shift operators discussed above, the other way is to re-interpret dummy branches at the current B-node as circular parts. Another place the dummy branch appears is during the skeleton operators. If the ignored branch (see d_3 in figure 27(c)) is a dummy branch then no cost will be paid.

Whether a part is a circular part or an elongated worm part will be finally up to the model. Therefore the algorithm must actively switch a part between the circular and the worm representations.

5.4 Matching Results

Figure 28 shows some of the adjusted skeletons obtained using the skeleton operators after matching some of the skeletons shown in figure 21 with their corresponding models. The BDP and BSP observed in section (4.1.4) are fixed. Some dummy peaks are eliminated, such as the small peak on the leg of the human and on the tail of the crane. Conversely, some dummy peaks are judged to be real branches, such as the ear of the lion. The skeletons in figure 28 satisfy our subjective perception about the "true" skeletons of those objects. Based on these skeletons, the objects can be segmented and then a more precise

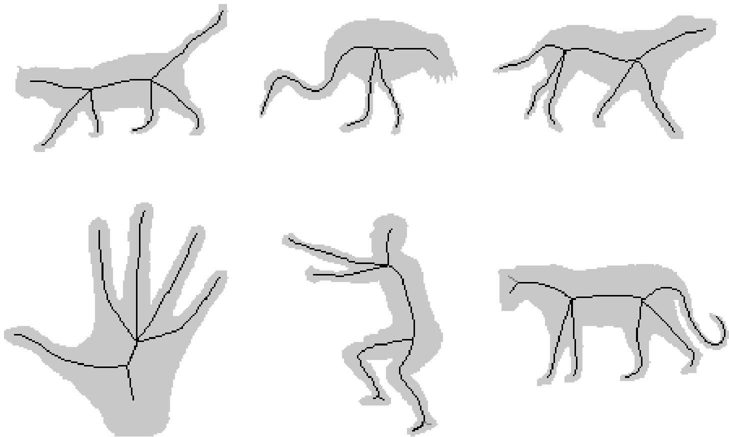


Figure 28 **Skeletons matched with models.**

parametric representation is available. Figure 35 shows the dataflow of the FORMS.

To test the performations of the FORMS, we collected a small database which contains 35 objects including people, hands, animals, fish, insects, and leaves. These objects can be classified into 17 categories.

The goodness of fit measure is defined to be $e^{(\log P/N)}$, where N is the number of parts. We use this because the probability P defined in equation (15) is typically the product of many probabilities factors and is thus very small.

Some typical matching results between objects from difference categories are shown in figures 29,30,31,32. In every case the two objects are first matched against a model (hand, horse, human, giraffe respectively), then the matched skeletons are drawn in the figures. For each pair of parts which are matched to the same part in the model, the program draws a line to connect the corresponding points. These figures show the robustness of the matching under scale, rotation, and flip transformations and with missing or additional parts. We also selected a group of objects of roughly similar form for intensive comparison. Figure 33 shows the differences in their similarities.

We note that the PCA was performed for the first 25 objects only. The remaining 10 objects were then added.

We tested the similarities between shapes in 16 categories, see figure 34 [20]. On the top of each column is the input shape. The three rows below shows the closest categories and the similarity measurements. The table is non-symmetric because the goodness of fit between a dog input and the cat model has no direct relationship to the fit between a cat input and a dog model.

Note for some categories, like the leaf, we have only one example. The model therefore is simply the example and thus the similarity measurement is close to 1.

[20] The remaining 17^{th} category is the hand.

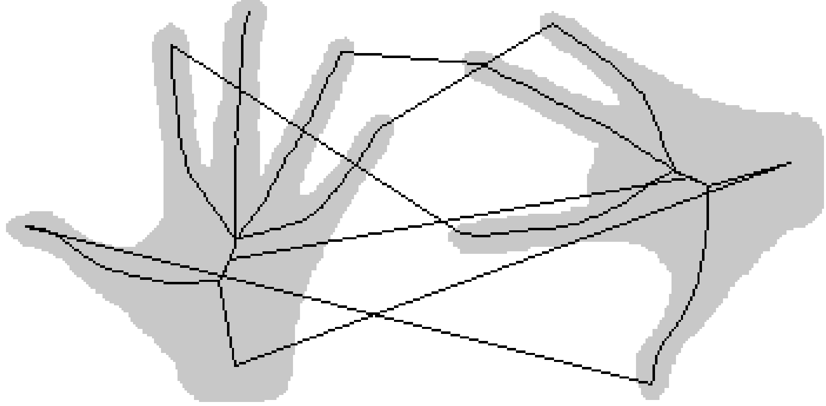


Figure 29 *One part is missing and the hand is rotated. The goodness of fit is 0.811.*

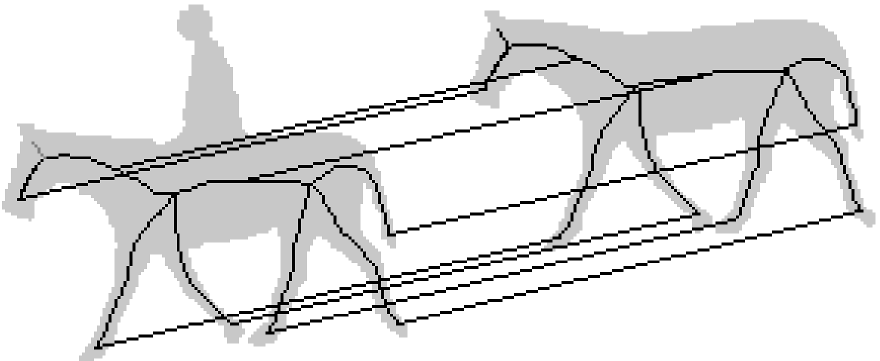


Figure 30 The man on the horse is redundant, and is therefore ignored. The goodness of fit is 0.695.

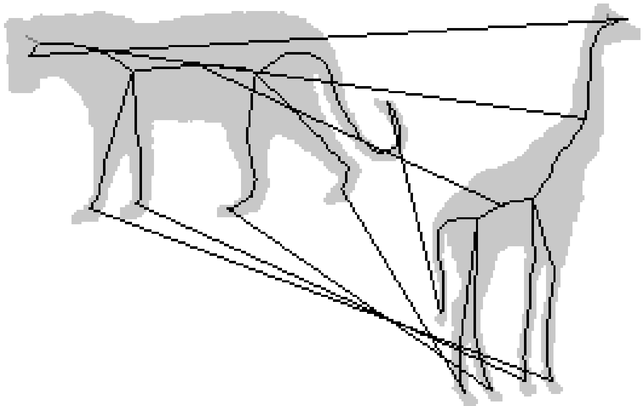


Figure 31 The corresponding parts of the lioness and the giraffe are matched, despite a flip transformation between the two animals and considerable differences in size of corresponding parts. The overall fit is only 0.542, which suggests that the two objects are different.

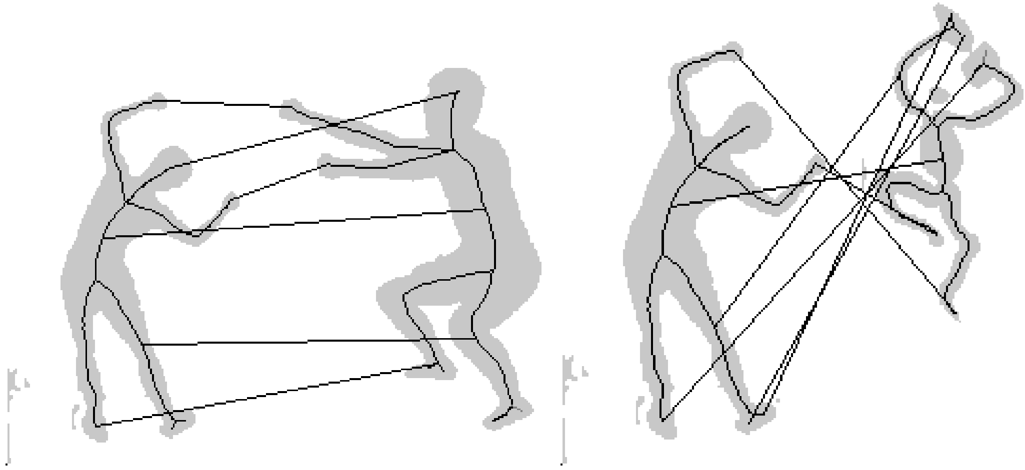


Figure 32 In the left figure the shapes undergo severe gesture and viewpoint deformations but the correct matching is attained with goodness of fit 0.761. The two shapes in the left figure, in fact, are matched to the two different skeleton graph of a person model, and the correspondence between them will be impossible in the early stages of vision. In the right figure we attempt to match our model to the figure in Picasso's Rites of Spring. Our algorithm identifies Picasso's figure as a human upside down! This occurs because the head is not connected to the torso and the hands appear like feet because they are holding mandolins. Our algorithm could be easily adapted to solve this problem correctly. The goodness of fit is only 0.113.

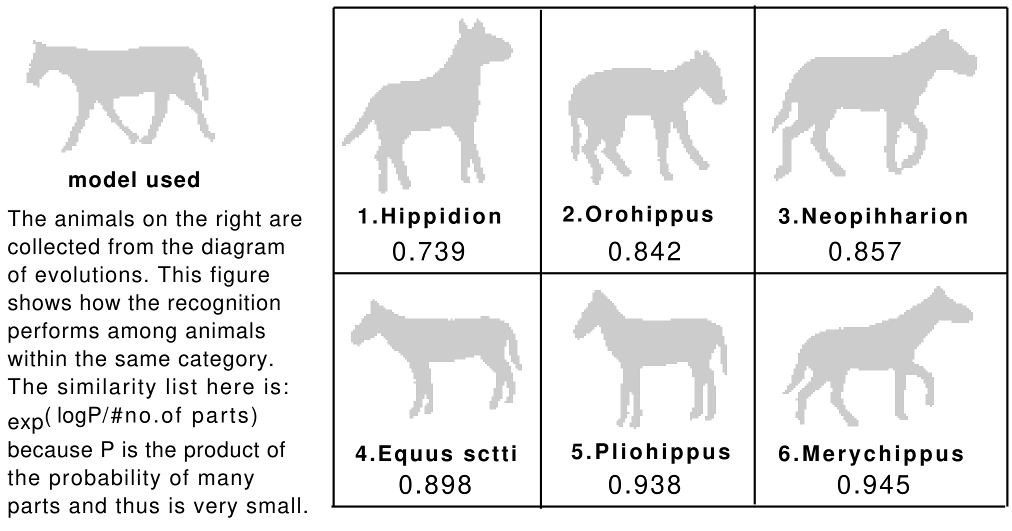
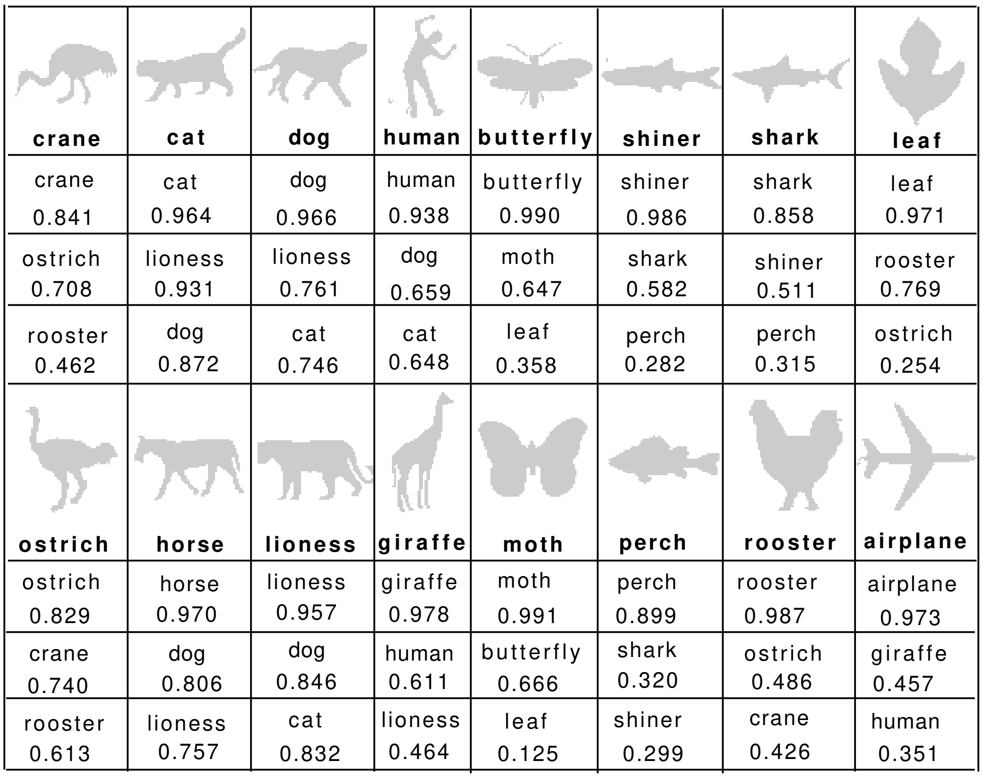


Figure 33 Similarities within the same category.



crane	cat	dog	human	butterfly	shiner	shark	leaf
crane 0.841	cat 0.964	dog 0.966	human 0.938	butterfly 0.990	shiner 0.986	shark 0.858	leaf 0.971
ostrich 0.708	lioness 0.931	lioness 0.761	dog 0.659	moth 0.647	shark 0.582	shiner 0.511	rooster 0.769
rooster 0.462	dog 0.872	cat 0.746	cat 0.648	leaf 0.358	perch 0.282	perch 0.315	ostrich 0.254

ostrich	horse	lioness	giraffe	moth	perch	rooster	airplane
ostrich 0.829	horse 0.970	lioness 0.957	giraffe 0.978	moth 0.991	perch 0.899	rooster 0.987	airplane 0.973
crane 0.740	dog 0.806	dog 0.846	human 0.611	butterfly 0.666	shark 0.320	ostrich 0.486	giraffe 0.457
rooster 0.613	lioness 0.757	cat 0.832	lioness 0.464	leaf 0.125	shiner 0.299	crane 0.426	human 0.351

Figure 34 Similarities between shape categories.

6 Discussion

In this paper, we first proposed a general model for how to generate the shapes of animate objects, such as fish, leaves, trees and insects. Then we formulated the recovery of their structures as an inverse process. We employed a bottom-up/top-down approach while matching the input shapes to the models stored in the database. The overall dataflow for the FORMS is shown in figure 35. Two more aspects need to be addressed below:

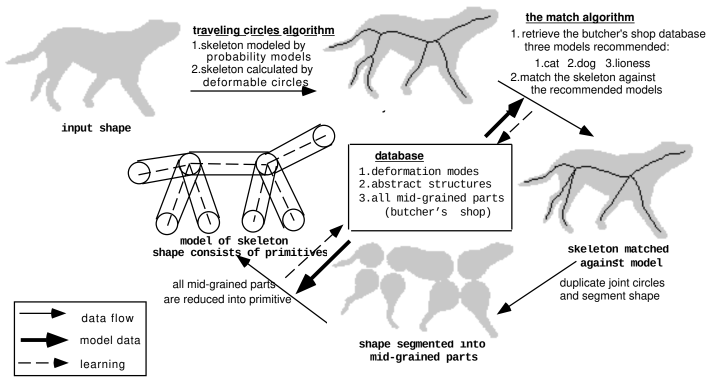


Figure 35 **dataflow of FORM**

I. Learning. Figure 35 also shows the structure of dataflow for learning. Even though learning such simple shapes as parallelograms was claimed to be a hard problem within Valiant's PAC-learning framework [1] [35], it seems a trivial task for FORMS to learn flexible objects. As shown in figure 35 the total knowledge base in FORMS is organized into three parts: (i) the deformation modes, (ii) the butcher's shop, and (iii) the skeleton graphs. Therefore learning in FORMS means using examples to adapt this knowledge in the following ways:

1. If the input shape is matched to a model and the error between the mid-grained parts and their projection onto the deformation modes is large, then it means that these parts should be considered outliers to the principal component analysis. We can then recalculate the principal components by including the new mid-grained part into the covariance matrix.

2. If the input shape is identified as a certain object in the model. We can use it to re-estimate the means and variances, (i.e. the parameters description) for each part of that model. Thus we can adapt the butcher's shop database.

3. If there is no model in the database which can be well matched to the input shape well then we can identify the input shape as a novel object. We can use its skeleton graph, as well as descriptions of its mid-grained parts, to start building a new model in the knowledge base.

II. The limits and extension of our method. FORMS will work well only in situations where our shape model is applicable. The model was created to deal with animate objects and would have to be completely modified to deal with man-made objects like houses and industrial parts.

Moreover,even for animate objects, our model is not complete. At least three factors are not taken into account: **1.** clothes may drastically change the shape of a person, such shapes cannot be modeled by elastic deformations, **2.** fine-scale structure, for example, details on the heads of animals, are ignored so the recognition of a face silhouette will be imprecise. Shapes, like the wings of birds and some wide fins of fish, see figure 36, are not well modeled. **3.** Our technique for calculating the representation is limited to those shapes which can be well represented by their 2D silhouettes. For example, the silhouette in figure 37 is insufficient to determine the object. But if internal edges are added then it is possible to identify the object as a sleeping cat. So the input representation must contain internal edges as well as the silhouette and a more complex recovery strategy should be investigated. The extraction of silhouettes, and internal edges, may be difficult from a single image but we expect that this task will be significantly easier from motion sequences, when motion occlusion will give powerful segmentation cues. These issues are under investigation.

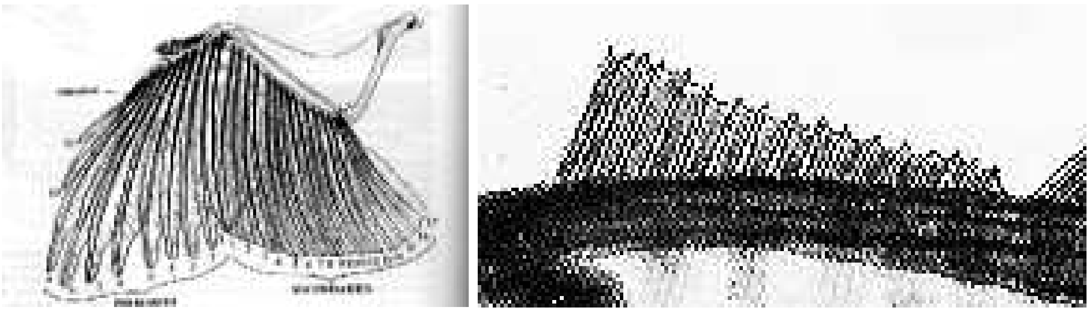


Figure 36 *The folding structure in the wing of birds (left) and some wide fins of fish (right) are not modeled by the model in section (3).*

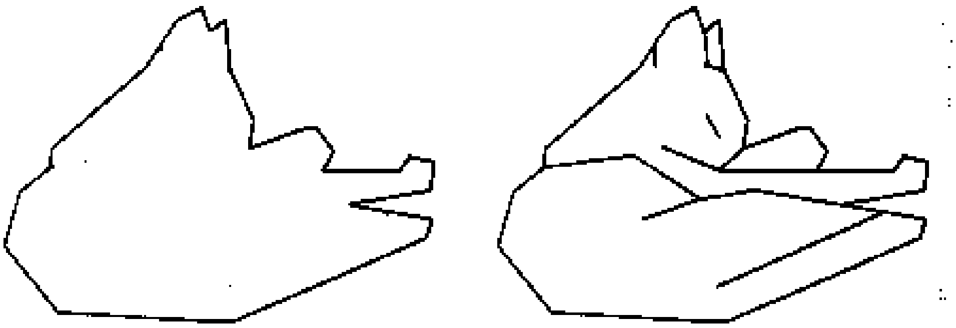


Figure 37 *The silhouette alone is not sufficient to identify this object. But when internal edges are added it becomes straightforward to recognize it as a sleeping cat.*

Appendix A. The sensitivity of the part descriptions to the affine transformation

In this appendix, we discuss how the parameters describing the mid-grained parts in the 2D plane will be influenced by 3D articulated motion and viewpoint changes. First of all, we assume weak projective projection from 3D objects to 2D silhouettes. Since the skeleton calculation is invariant to planar rotation and translation due to the isotropy of deformable circles, and furthermore since the vector for the ribs in the worm parts and the vector for the radials in the circular parts are divided by the radius of the corresponding maximal circles, we only need to consider the influences of the slant angle α and tilt angle τ under orthogonal projection as shown in figure 38. let $\vec{\theta} = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \ell)$ and $\vec{\theta} = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \omega)$ be the original parameter descriptions for the worm and circular parts respectively when $\alpha = 0, \tau = 0$.

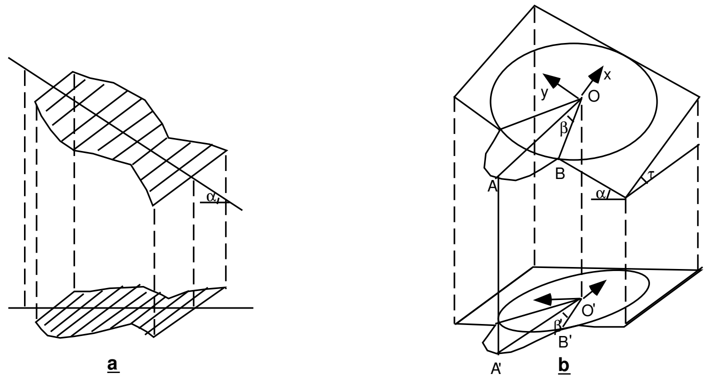


Figure 38 *the othorgonal projection of objects on planes.*

i). For most animate objects, we assume that an elongated worm part has a straight axis and is rotationally symmetrical, then only the slant angle α can influences the parameters. As shown in figure 38(a), the new parameters under α, τ is $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \ell cos\alpha)$.

ii). Let a circular part be in the up plane shown in figure 38(b). Let OA, OB be two of radials for the peak, OA is perpendicular to the y-axis in the projection plane, and β be the angle between them. Their projections are $O'A', O'B'$, and β' respectively. Since all radial lengths are normalized, we need only to consider the change of the relative size $\frac{|O'B'|}{|O'A'|} = \frac{|OB|}{|OA|}\Delta$, i.e. to see how Δ is related to α, τ, and β.

If $\tau = 0$, then $\Delta = \sqrt{1 + \sin^2\beta \tan^2\alpha} \simeq 1 + \frac{1}{2}\sin^2\beta\tan^2\alpha$, when $\alpha = \beta = \frac{PI}{6}, \Delta \simeq 1 + \frac{1}{24}$. The relationship between the angle β and its projection β' is given by: $\tan\beta' = \tan\beta / \cos\alpha$.

If $\alpha = 0$, then $\Delta = \sqrt{\cos^2\beta + \sin^2\beta\cos^2\tau}$. when $\alpha = \beta = \frac{PI}{6}, \Delta \simeq 1 - \frac{1}{32}$. $\tan\beta' = \tan\beta\cos\tau$.

Therefore, if τ and α are within 30^o, the changes of $\alpha_i' sin\vec{\theta}$ will be negligible. But the ω and ℓ change with α, τ.

In summary, the parameter descriptions $\alpha_i's$ derived in this paper are ratherly reliable if the view point is near the orthogonal directions. Otherwise if looking an animal in front of it, the description is unreliable. The parameter ℓ and ω include information for recovering the 3D orientations, but the calculation of 3D post is beyond this paper.

Acknowledgements

It is a pleasure to acknowledge many suggestions and helpful discussions with David Mumford. Peter Belhumeur, Roger Brockett, Peter Hallinan, Tai Sing Lee and Yibing Yang also provided useful feedback and ideas. James Coughlan, Russell Epstein, Sandy Pentland and Gang Xu provided useful comments on the manuscript. Cedric Xia gave help on modal analysis and Yingnian Wu gave helpful discussion on statistics. This research was supported in part by the Brown/Harvard/MIT Center for Intelligent Control Systems with U.S. Army Research Office grant number DAAL03-86-K-0171. The authors would also like to thank ARPA for an Air Force contract F49620-92-J-0466.

References

[1] M. Anthorny and N. Biggs **Computational Learning Theory**. Cambridge University Press, 1992.

[2] I. Biederman, "Recognition-by-components: A theory of Human Image Understanding", Psychological Review, Vol.94, No.2. 115-147, 1987.

[3] T.O. Binford. "Visual perception by computer". Presented at the *IEEE Syst. Sci. Cybern. Conf.* Miami, Florida. Invited paper. Dec. 1971.

[4] H.Blum, "Biological shape and visual science". *J.of Theoretical Biology.* 1973 33 PP205-287.

[5] H. Blum and R.N. Nagel, "Shape description using weighted symmetric axis features". *Patt. Recognition,* Vol.10 (167-180, 1978.

[6] J. Brady and H. Asada, "Smooth Local Symmetries and Their Implementations". *Int. J. of Robotics Reg.* 3(3) 1984.

[7] R. Brooks. "Model-Based Three-Dimensional Interpretations of Two-Dimensional Images". *IEEE Trans. on Pattern Analysis and Machine Intelligence.* Vol. PAMI-5, No. 2. March 1983.

[8] J.F. Canny. "A computational approach to edge detection". *IEEE Trans. Patt. Anal. Mach. Intell..* PAMI-8(6):679-698. 1986.

[9] J.H. Connell. **Learning Shape Descriptions**. MIT Artificial Intelligence Laboratory Technical Report 853. 1985.

[10] J.L.Crowley. "A representation for shape based on peaks and ridges in the difference of low-pass transform". *IEEE Trans. Patt. Anal. Mach. Intell..* PAMI-6(2). March 1984.

[11] M.M. Fleck. **Local Rotational Symmetries**. Masters' Thesis. MIT Artificial Intelligence Laboratory. 1985.

[12] U. Grenander, Y. Chow and D.M. Keegan. **HANDS**. Springer-Verlag. New York. 1991.

[13] W.E.L. Grimson. **Object Recognition by Computer**. MIT Press. Cambridge, Mass. 1990.

[14] M. Hildebrand, *Analysis of vertebrate structure* 3rd edition, John Wilet and Sons, Inc. 1988.

[15] A. Hill, C.J. Taylor and T. Cootes. "Object Recognition by Flexible Template Matching using Genetic Algorithms". *Proc. ECCV-2.* Genoa, Italy. 1992.

[16] D.P. Huttenlocher and S. Ullman. "Object recognition using alignment". In *Proc. First Int. Conf. Comput. Vision.* London, UK. pp 102-11. 1987.

[17] M. Leyton, **Symmetry, Causality, Mind**. MIT Press. Cambridge, Mass. 1992.

[18] A. Lindenmayer, "Mathematical models for cellular interactions in development, part I,II", *Journal of Theoretical Biology* **18**, pp.280-315.

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

[20] B.B. Mandelbrot, *The Fractal Geometry of Nature Freeman*, S.F., CA 1982.

[21] D. Marr **Vision**. W.H. Freeman and Co. 1982.

[22] E. Mjolsness. "Bayesian Inference on Visual Grammars by Neural Nets that Optimize". Research Report YALEU/DCS/TR-854. 1991.

[23] D.M. Mumford, "Geometric methods in computer vision". *Proc. SPIE-the Int. Soc. Optical Eng.* San Diego, 1991.

[24] D.M. Mumford, "Pattern theory". 1993.

[25] J. Mundy and A. Zisserman. **Geometric Invariants in Computer Vision**. MIT Press. Cambridge, Mass. 1992.

[26] R. Navatia & T.O. Binford, "Description and recognition of curved objects". *A.I.* 8. 77-98. 1977.

[27] P. J.Van Otterloo, **A contour-oriented approach to shape analysis.**. Prentice Hall International Ltd 1991.

[28] R.L.Ogniewicz, **Discrete Voronoi skeleton**. Hartung-Gorre,1993.

[29] A. P. Pentland, "Perceptual Organization and the Presentation of Natural Form". *A.I.* 28(1986) 293-331

[30] A.P. Pentland and S. Sclaroff. "Closed-form solutions for physically based shape modeling and recognition". *IEEE Trans. Pattern Analysis and Machine Intelligence*, **13**, (7), pp 715-729. 1991.

[31] S.M. Pizer, W.R. Oliver and S.H. Bloomberg, "Hierarchical shape description via the multi-resolution symmetric axis transform". *IEEE Trans PAMI-9* No.4, July 1987.

[32] T. Poggio and S. Edelman. *A Network that learns to recognize 3D objects.* Nature, 343, pp 263-266. 1990.

[33] E. Saund, "Representation and dimensions of shape deformation". *Proceedings of the Third International Conference on Computer Vision*, Osaka, Japan, December 4-7, pp 684-689. 1990.

[34] S. Sclaroff and A.Pentland, "Modal matching for correspondence and recognition" MIT Media Lab. TR No.201 May,1993.

[35] H.Shvaytser, "Learnable and nonlearnable visual concepts". *IEEE Trans. PAMI*, 12(5):459-466. May 1990.

[36] A.R.Smith, "Plants,fractals,and formal languages", **Computer Graphics** vol.**18**,No.3, July 1984.

[37] D. Terzopolous, A. Witkin, and M. Kass. "Symmetry-seeking models and 3D object recovery". *Int. J. Comput. Vision.* 1, 211-221. 1987.

[38] S. Ullman and R. Basri, "Recognition by linear combinations of models". *IEEE. Trans. Pattern Analysis and Machine Intelligence.* 13, No. 10. 1991.

[39] J.Z.Young, *The life of vertebrates* 3rd edition. Oxford Univ. press 1981.

[40] A.L.Yuille, "Deformable templates for face recognition". *J.of Cognitive Neuroscience.*. Vol.3.No.1 1991.