Appears in *Pattern Recognition* **30(4), April 1997, special issue on image databases.**

Deformable Prototypes for Encoding Shape Categories in Image Databases

Stan Sclaroff
Computer Science Department
Boston University
111 Cummington St.
Boston MA 02215

Abstract

We describe a method for shape-based image database search that uses deformable prototypes to represent categories. Rather than directly comparing a candidate shape with all shape entries in the database, shapes are compared in terms of the types of nonrigid deformations (differences) that relate them to a small subset of representative prototypes. To solve the shape correspondence and alignment problem, we employ the technique of modal matching, *an information-preserving shape decomposition for matching, describing, and comparing shapes despite sensor variations and nonrigid deformations. In modal matching, shape is decomposed into an ordered basis of orthogonal principal components. We demonstrate the utility of this approach for shape comparison in 2-D image databases.*

Keywords: Deformable models, deformable templates, combinations of models, shape matching, modal matching.

1 Introduction

Shape categories can be represented as deformations from a subset of standard or prototypical shapes; it is thought that this is one plausible mechanism for human perception [4; 18; 30; 33; 40; 45]. This basic premise is appealing for its descriptive parsimony, and has served as inspiration for many of the prototype-based representations for machine vision, robotics, and simulation.

In the work described in this paper, our aim is to represent shape categories for interactive, image database search. Rather than directly comparing a candidate shape with all shapes in the database, we propose a method that first indexes shapes in terms of their relationship to a few shape prototypes. To do this, we will employ *modal matching*, a deformable shape decomposition that allows users to specify a few example shapes and has the computer efficiently sort the set of objects based on the similarity of their shape. If desired, shapes can be more closely compared in terms of the types of nonrigid deformations (differences) that relate them to a few prototype shapes.

Our approach is related to *morphing*, a computer graphics technique that has become quite popular in advertisements. Morphing is accomplished by an artist identifying a large number of corresponding control points in two images, and then incrementally deforming the geometry of the first image so that its control points eventually lie atop the control points of the second image. Using this technique, in-between or novel views can be generated as warps between example views. This suggests an important way to obtain a low-dimensional, parametric description of shape: interpolate between known, prototype views. For instance, given views of the extremes of a motion (*e.g.*, systole and diastole, or left-leg forward and right-leg forward) we can describe the intermediate views as a smooth combination of the extremal views.

All that is required to determine this view-based parameterization of a new shape are: the prototype views, point correspondences between the new shape and the prototype views, and a method of measuring the amount of (nonrigid) deformation that has occurred between the new shape and each prototype view. The prototypes define a polytope in the space of the (unknown) underlying physical system's parameters. By measuring the amount of deformation between the new shape and extremal views, we locate the new shape in the coordinate system defined by the polytope. This coordinate in *prototype space* can be used for database indexing and fast search.

This general approach is related in spirit to the linear-combinations-of-views paradigm, where any object view can be synthesized as a combination of linearly-warped example views of Ullman and Basri [56] and Poggio, *et al.* [44]. However, it differs from their proposals in two important ways. First, we are interested not only in recognizing shapes, but also in describing the types of deformations that relate them. We want to derive a low-dimensional parametric representation of the shape that can be used to recognize and compare shapes, in the manner of Darrell and Pentland [12]. Second, we cannot be restricted to a linear framework. Nonrigid motions are inherently nonlinear, although they *are* often "physically smooth." Therefore, to employ a combination-of-views approach we must be able to determine point correspondences and measure similarities between views in a way that takes into account at least qualitative physics of nonrigid shape deformation. In computer graphics it is the job of the artist to enforce the constraint of physical smoothness; in machine vision, we need to be able to do the same automatically.

To achieve this, we will employ modal matching, a method for (1) determining point correspondences using a energy-based model, (2) warping or morphing one shape into another using

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
1995		**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Deformable Prototypes for Encoding Shape Categories in Image Databases	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Office of Naval Research,One Liberty Center,875 North Randolph Street Suite 1425,Arlington,VA,22203-1995**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**14**	

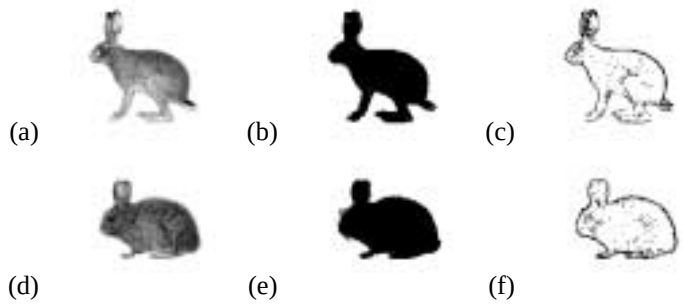

Figure 1: The data needed to build two deformable prototype shape models. Support maps are shown in (b,e) and edge maps in (c,f). The two prototype shape models depict (a) a European Hare, and (d) a Desert Cottontail. Their associated The original color images were digitized from [1].

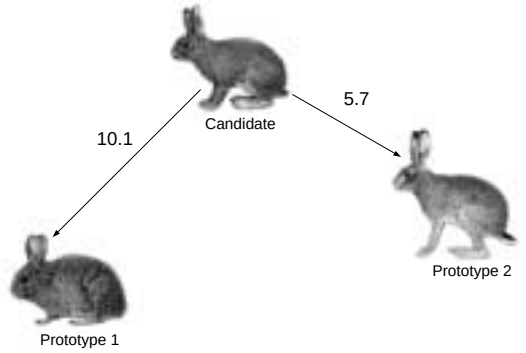

Figure 2: When a new shape is encountered, it is parameterized in terms of the energy needed to nonrigidly deform the prototype shape models into alignment with the new shape. The distance to prototypes is expressed as the square root of strain. The resulting tuple $(10.1, 5.7)$ is used to represent the shape in a space defined in terms of distance to prototypes.

energy-based interpolants, and (3) measuring the amount of deformation between an object's shape and prototype views[48; 50].

Figure 1 shows the information required to build modal shape prototypes for two rabbit shape prototypes employed in our image database experiments. In our system a shape is defined by: a cloud of feature locations (*i.e.*, edges, corners, high-curvature points) and a region of support that tells us where the shape is. Given this input, deformable prototype models are built directly from feature data, using a finite element formulation that is based on Gaussian interpolants [50]. For efficiency, we can select a subset of the feature data as nodes for a lower-resolution finite element model and then use the resulting eigenmodes in finding the higher-resolution feature correspondences as described in [50]. This subset can be a set of particularly salient features (*i.e.*, corners, T-junctions, and edge mid-points) or a randomly selected subset of (roughly) uniformly-spaced features.

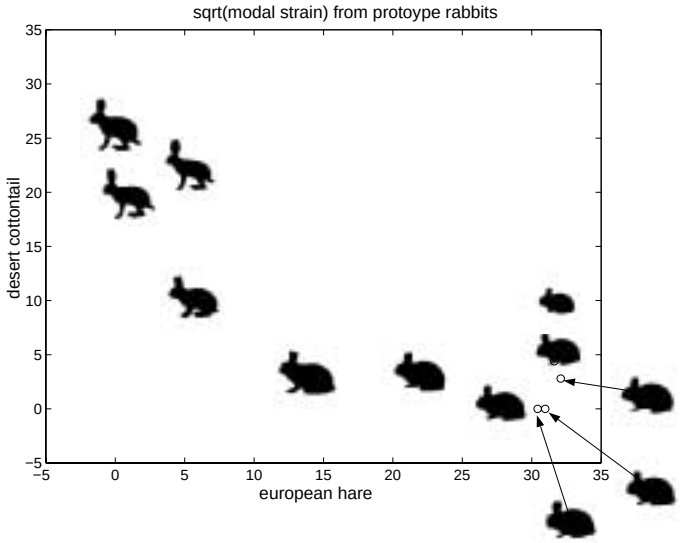

Figure 3: Scatter plot of square-root modal strain energy for rabbit prototypes used in the image database experiment. Each axis depicts the square-root of strain energy needed to align a shape with a rabbit prototype. Thus each rabbit shape has a coordinate in this space. The rabbits are clustered in terms of their 2-D shape appearance: long-legged, standing rabbits cluster at the top-left of the graph, while short-legged, seated rabbits cluster at the bottom right. There are two rabbits that map between clusters, showing the smooth ordering from long-legged, to medium-legged, to short-legged rabbits in this view-space.

When a new shape is encountered, it is parameterized in terms of the energy needed to nonrigidly deform the prototype shape models into alignment with the new shape. Similarity is thus computed in terms of the amount of strain energy needed to deform each prototype to match it to the candidate shape, as illustrated in Figure 2. The amounts of deformation are measured in terms of strain energy and stored as a n-tuple, where n is the number of prototype shapes employed. In this case, the resulting tuple is $(10.1, 5.7)$. The result is a low-dimensional parametric representation that can be used for efficient shape-based image database search. Rather than directly comparing a candidate shape with all shape entries in a database, we instead compute similarity in a *distance to prototypes space*. Using this method, we compactly represent a category of shapes in terms of a few prototype views.

Fig. 3 shows a scatter plot of the two-dimensional "rabbit space" spanned by two rabbit shape prototypes. The graph's x-axis depicts the square-root of strain energy needed to align the European Hare prototype with each rabbit shape, while the y-axis shows the energy needed to align the Desert Cottontail prototype with each rabbit shape. Each or the 12 rabbit shapes has a coordinate in this strain-energy-from-prototype subspace. As can be seen, the rabbits are clustered in terms of their 2-D shape appearance: long-legged, standing rabbits cluster at the top-left of the graph, while short-legged, seated rabbits cluster at the bottom right. There are two rabbits that map between clusters, showing the smooth ordering from long-legged, to medium-legged, to short-legged rabbits in this view-space.

We will demonstrate the utility of this approach for comparing shapes in 2-D image databases digitized from children's field guides and images of hand tools. Deformable shape models will be built and compared using support and silhouette data. The methods described in this paper are also useful for recognizing or classifying motions [49], fusing data from different sensors, and for comparing data acquired at different times or under different conditions [50].

1.1 Segmentation

The work reported in this paper addresses issues of shape categorization, even when shapes within categories can undergo both rigid and nonrigid motion. Throughout this paper, it was assumed that figure/ground segmentation information can be provided as input to the modal shape comparison modules. Since segmentation is not the topic of this paper, our current databases contain images of unoccluded objects on uniform backgrounds. Under these circumstances, a c-Means clustering and thresholding technique can be used for foreground/background separation [31]. However, for very general query by shape, foreground/background modules will be needed as a front-end to the system. The first solution would be to use motion and color to pull out foreground objects. Such figure-ground segmentation can be done reliably by use of clustering in conjunction with optical flow [5; 13; 54; 57; 58] and/or color information [8; 24; 29; 32; 37; 31].

2 Background and Notation

In the last few years researchers have made some progress toward automatic shape indexing for image databases. The general approach has been to calculate some approximately invariant statistic like shape moments, and use these to stratify the image database [9; 26; 27; 34; 36; 47].

One problem with this general approach is that it discards significant perceptual and semantic information. While indexing methods provide a means to quickly narrow a search to a more manageable subset, they often do not provide a method for closer, direct comparison of *how* they are related. Rather than discarding useful similarity information by employing only invariants, we believe that one should use a decomposition that preserves as much semantically meaningful and perceptually important information as is possible, while still providing an efficient encoding of the original signal [42].

Another important problem with these approaches is that most are only robust for rigid shapes. Although many things move rigidly, in many cases this rigid-body model is inadequate. For instance, most biological objects are flexible and articulated. To describe these deformations, therefore, it is reasonable to model the physics by which real objects deform. This rationale led to the physical modeling paradigm of active contours or *snakes*[28] and deformable templates [52; 59]. A snake has a predefined structure which incorporates knowledge about the shape and its resistance to deformation. By allowing the user to specify forces that are a function of sensor measurements, the intrinsic dynamic behavior of a physical model can be used to solve fitting, interpolation, or correspondence problems.

While snakes enforced constraints on smoothness and the amount of deformation, they could not in their original form be used to constrain the *types* of deformation valid for a particular problem domain or object class. This led to the development of algorithms which include *a priori* constraints on the types of allowable deformations for motion tracking [6; 7; 10; 16].

Cootes *et al.*[11; 3] use trainable snakes for capturing the invariant properties of a class of shapes, by finding the principle variations of a snake via the Karhunen-Loeve transform. Unfortunately, this method relies on the consistent sampling and labeling of point features across the entire training set and cannot handle large rotations. If different feature points are present in different views, or if there are very different sampling densities, then the resulting models will differ even if the object's pose and shape are identical.

Keeping these issues in mind, we use the Finite Element Method to alleviate problems with sampling, and modal analysis to provide a principled way to select the types of nonrigid deformations needed for flexibly describing shape. In the rest of this section we provide a brief review of our representation. In addition, we review our new method of building FEM models without imposing an *a priori* parameterization, and how to use the modes of this model to find point correspondences, to align objects, and to compare their shape. This initial work was applied in the area of finding corresponding features in static imagery [50] and serves as the foundation for our new representation for shape categories.

2.1 Finite Element Method

The major advantage of the finite element method is that it uses the Galerkin method of surface interpolation. This provides an analytic characterization of shape and elastic properties over the whole surface, and thereby alleviates problems caused by irregular sampling of feature points. In Galerkin's method, we set up a system of polynomial shape functions that relate the displacement of a single point to the relative displacements of all the other nodes of an object:

$$\mathbf{u}(\mathbf{x}) = \mathbf{H}(\mathbf{x})\mathbf{U} \tag{1}$$

where $\mathbf{H}$ is the interpolation matrix, $\mathbf{x}$ is the local coordinate of a point in the element where we want to know the displacement, and $\mathbf{U}$ denotes a vector of displacement components at each element node. By using these functions, we can calculate the deformations which spread uniformly over the body as a function of its constitutive parameters.

Solution to the problem of deforming an elastic body to match

the set of feature points then requires solving the *dynamic equilibrium equation*:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{D}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R}, \tag{2}$$

where $\mathbf{R}$ is the load vector whose entries are the spring forces between each feature point and the body surface, and where $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ are the element mass, damping, and stiffness matrices, respectively [2; 43].

2.2 Modal Representation

The FEM governing equations can be decoupled by posing the equations in a basis defined by the $\mathbf{M}$-orthogonalized eigenvectors of $\mathbf{K}$. These eigenvectors and values are the solution to the generalized eigenvalue problem:

$$\mathbf{K}\phi_{\mathrm{i}} = \omega_{\mathrm{i}}^2 \mathbf{M}\phi_{\mathrm{i}}. \tag{3}$$

The vector ϕ_{i} is called the ith *mode shape vector* and ω_{i} is the corresponding frequency of vibration. Each mode shape vector describes how each node is displaced by the i^{th} vibration mode. The mode shape vectors are $\mathbf{M}$-orthonormal; this means that $\mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi} = \mathbf{\Omega}^2$ and $\mathbf{\Phi}^T\mathbf{M}\mathbf{\Phi} = \mathbf{I}$. The ϕ_{i} form columns in the transform $\mathbf{\Phi}$ and ω_{i}^2 are elements of the diagonal matrix $\mathbf{\Omega}^2$. We will assume Rayleigh damping (*i.e.*, $\mathbf{D} = a_0\mathbf{M} + a_1\mathbf{K}$), thus the damping matrix will also be diagonalized by this transform [2].

This generalized coordinate transform $\mathbf{\Phi}$ is then used to transform between nodal point displacements $\mathbf{U}$ and decoupled modal displacements $\tilde{\mathbf{U}}$, $\mathbf{U} = \mathbf{\Phi}\tilde{\mathbf{U}}$. We can now rewrite Eq. 2 in terms of these generalized or modal displacements, obtaining a decoupled system of equations:

$$\ddot{\tilde{\mathbf{U}}} + \tilde{\mathbf{D}}\dot{\tilde{\mathbf{U}}} + \mathbf{\Omega}^2\tilde{\mathbf{U}} = \mathbf{\Phi}^T\mathbf{R}, \tag{4}$$

allowing for closed-form solution to the equilibrium problem [43]. Given this equilibrium solution in the two images, point correspondences can be obtained directly.

By discarding high frequency modes the amount of computation required can be minimized without significantly altering correspondence accuracy. Moreover, such a set of modal amplitudes provides a robust, canonical description of shape in terms of deformations applied to the original elastic body. This allows them to be used directly for object recognition [43].

2.3 Modal Matching

Perhaps the major limitation of previous methods is the requirement that every object be described as the deformations of a *single* prototype object. For instance, in some schemes all shapes are represented as deformations from an elliptical or circular prototype [9; 20; 43]. Such approaches implicitly impose an *a priori* parameterization upon the sensor data, and therefore implicitly determine the correspondences between data and prototype.

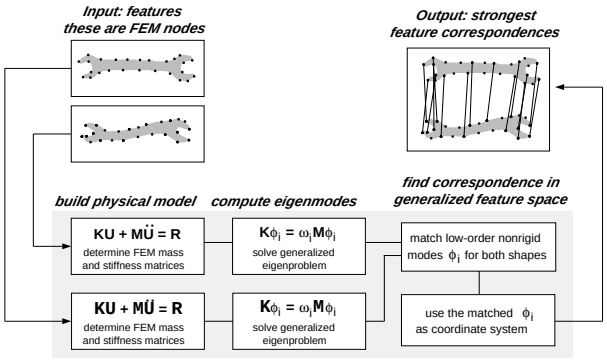

Figure 4: Modal matching system diagram (reprinted from [48]).

Furthermore, an elliptical prototype may be inadequate for many shapes, especially shapes that are not star-connected, or those that have long protrusions or deep concavities. We would like to avoid these problems as much as possible, by letting the data determine the parameterization in a natural manner. To accomplish this we use the data itself to define the deformable object, by building stiffness and mass matrices that use the positions of image feature points as the finite element nodes.

The resulting new modeling formulation is called *modal matching*, and is described in detail in [48; 50]. A flow-chart of our method is shown in Fig. 4. For each image we start with feature point locations, which are used as nodes in building a finite element model of the shape. If we are given a support function, then we can "cut" the finite element sheet into any shape. We do this by defining a support function that is zero anywhere outside the shape region, and greater than zero inside the shape region. Thus the support function can be used to define both the shape and the thickness of the elastic model. A Gaussian is then centered at each node. Together, these Gaussians form a basis for building the Galerkin interpolants of Eq. 1, and are thus used in constructing FEM mass and stiffness matrices.

When there are possibly hundreds of feature points for each shape, computing the FEM model and eigenmodes for the full feature set can become non-interactive. For efficiency, we can select a subset of the feature data to build a lower-resolution finite element model and then use the resulting eigenmodes in finding the higher-resolution feature correspondences as described in [50]. This subset can be a set of particularly salient features (*i.e.*, corners, T-junctions, and edge mid-points) or a randomly selected subset of (roughly) uniformly-spaced features.

We then compute the *modes of free vibration* $\mathbf{\Phi}$ of this model using Eq. 3. The modes of an object form an orthogonal *object-centered* coordinate system for describing feature locations. That is, each feature point location can be uniquely described in terms of *how it projects onto each eigenvector, i.e.*, how it participates in each deformation mode. The transform between Cartesian feature locations (x, y) and modal feature locations (u, v) is accom-

plished by using the eigenvectors $\mathbf{\Phi}$ as a coordinate basis:

$$\mathbf{\Phi} = [\phi_1 \mid \ldots \mid \phi_{2m}] = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{u}_m \\ \mathbf{v}_m \end{bmatrix} \tag{5}$$

where m is the number of nodes used to build the finite element model. The column vector ϕ_{i} is the i^{th} *mode shape*, and describes the modal displacement (u, v) at each feature point due to the i^{th} mode, while the row vector $\mathbf{u}_{\mathrm{i}}$ and $\mathbf{v}_{\mathrm{i}}$ are the i^{th} *generalized feature vectors*, which together describe the feature's location in the modal coordinate system.

Normally only the n lowest-order modes are used in forming this coordinate system, so that (1) we can compare objects with differing numbers of feature points, and (2) ensure that the feature point descriptions are insensitive to noise. Depending upon the demands of the application, we can also selectively ignore rigid-body modes, or low-order projective-like modes, or modes that are primarily local. Consequently, we can match, describe, and compare nonrigid objects in a very flexible and general manner.

Point correspondences can now be determined by comparing the two groups of generalized feature vectors. The important idea here is that the low-order vibration modes computed for two similar objects will be very similar — even in the presence of affine deformation, nonrigid deformation, local shape perturbation, noise, or small occlusions. The points that have the most similar and unambiguous coordinates are then matched, with the remaining correspondences determined by using the physical model as a smoothness constraint [48; 50]. Currently, the algorithm has the limitation that it cannot reliably match largely occluded or partial objects.

2.4 Recovering Modal Descriptions

Given point correspondences between two shapes, we can then determine the deformations required to align them. An important benefit of modal matching is that the eigenmodes computed for the correspondence algorithm can also be used to describe the rigid and non-rigid deformation needed to align one object with another. Once this *modal description* has been computed, we can compare shapes simply by looking at their mode amplitudes or — since the underlying model is energy-based — we can compute and compare the amount of deformation energy needed to align an object, and use this as a similarity measure. If the strain energy required to align two feature sets is relatively small, then the objects are very similar.

Our task is to recover the modal deformation parameters $\tilde{\mathbf{U}}$ that take the set of points from the first image to the corresponding points in the second. A number of different methods for recovering the modal deformation parameters are described in [48; 50]. We will only give an overview of the strain-minimizing least-squares method employed for the database experiments described in this paper.

Given that modal models have been computed for both shapes, and that correspondences have been established, we can solve for the modal displacements directly — if correspondence is known at all nodes. Unfortunately, correspondence is not usually available at all nodes, and our recovery problem becomes under-constrained. Since the modal matching algorithm computes the strength for each matched feature, we would also like to utilize these match-strengths directly in alignment. As detailed in [50], we can obtain a constrained weighted least squares solution, if we minimize alignment error that includes a modal strain energy term $\lambda\Omega_2$:

$$\tilde{\mathbf{U}} = \left[\mathbf{\Phi}^T\mathbf{W}^2\mathbf{\Phi} + \lambda\mathbf{\Omega}^2\right]^{-1}\mathbf{\Phi}^T\mathbf{W}^2\mathbf{U} \tag{6}$$

where entries of the diagonal weighting matrix $\mathbf{W}$ are inversely proportional to the affinity measure for each feature match. The entries for unmatched features are set to zero. The strain term $\lambda\Omega_2$ directly parallels the smoothness functional employed in regularization [53]. This measure allows us to incorporate some prior knowledge about how "stretchy" the shape is, how much it resists compression, *etc*. Unmatched nodes to move in a manner consistent with the material properties and the forces at the matched nodes.

3 Encoding Modal Shape Categories

We will now describe how to use modal models to encode shape categories. One key advantage in using such a prototype-based approach is that of *data reduction*: given the multitude of possible viewpoints and configurations for an object, we need to reduce this multitude down to a more efficient representation that requires only a few *characteristic views*. Shapes are compared in terms of their relative distances to prototypes, rather than directly compared with one another.

3.1 Distance Measures

Once the mode deformation parameters $\tilde{\mathbf{U}}$ have been recovered, we can compute the strain energy incurred by these deformations, and use this as a similarity metric. In general, we will want to compare the strain only in a subset of modes $\mathcal{S}$ that has been deemed important in measuring similarity:

$$\delta(A, B) = \frac{1}{2}\sum_{\mathrm{i}\in\mathcal{S}} \tilde{u}_{\mathrm{i}}^2 {\omega_{\mathrm{i}}}^2, \tag{7}$$

where the modal displacements $\tilde{u}_{\mathrm{i}}$ describe the deformation needed to align shape A with shape B. It may be desirable to make object comparisons rotation and/or position independent. To do this, we ignore displacements in the rigid body modes, thereby disregarding differences in position and orientation. In addition, we can make our comparisons robust to noise and local shape variations by discarding higher-order modes. This modal selection technique is also useful for its compactness, since we

can describe deviation from a prototype in terms of relatively few modes.

If a metric distance function is desired, then this simple energy measure needs to be modified: strain does not satisfy one of the three axioms for a metric space[55]. These three axioms are:

1. minimality: $\delta(A,B) \geq \delta(A,A) = 0$,
2. symmetry: $\delta(A,B) = \delta(B,A)$, and
3. triangle inequality: $\delta(A,B) + \delta(B,C) \geq \delta(A,C)$.

While the strain energy measure satisfies minimality and the triangle inequality, it does not satisfy symmetry. The strain energy is not symmetric for shapes of differing sizes; *i.e.,* if the scales of two objects A and B differ, then the strain energy needed to align A with B may differ from that needed to align B with A. The difference in strain will be inversely proportional to the difference in square of the object scales. Therefore, when comparing objects of differing scales we divide strain energy by the shape's area. When a support map is available, this area can be computed directly. In the infinite-support case, the area can be approximated by computing the minimum bounding circle, or the moments, for the data.

There is an additional property that proves useful in defining a metric space, segmental additivity: $\delta(A,B) + \delta(B,C) = \delta(A,C)$, if B is on the line between A and C. To satisfy segmental additivity, we can take the square root of the strain energy:

$$\delta(A,B) = \left(\frac{1}{2a} \sum_{\mathrm{i} \in \mathcal{S}} \tilde{u}_{\mathrm{i}}^2 {\omega_{\mathrm{i}}}^2 \right)^{\frac{1}{2}}, \tag{8}$$

where a is the shape's area. This results in a weighted distance metric not unlike the Mahalanobis distance: the modal amplitudes are decoupled, each having a "variance" that is inversely proportional to the mode's eigenvalue. As a result, this formulation could be used as part of a regularized learning scheme in which the initial covariance matrix, $\mathbf{\Omega}$ is iteratively updated to incorporate the observed modal parameter covariances along the lines of [19; 17; 44; 38; 39].

3.2 Modal Shape Prototypes

Instead of looking at the strain energy needed to align the two shapes, we wish to the compare mode amplitudes needed to align a third, prototype object C with each of the two objects. In this case, we first compute two modal descriptions $\tilde{\mathbf{U}}_{\mathrm{a}}$ and $\tilde{\mathbf{U}}_{\mathrm{b}}$ that align the prototype with each candidate object. We then utilize our strain-energy distance metric to order the objects based on their similarity to that prototype.

We can use distance to prototypes to define a low-dimensional space for efficient shape comparison. In such a scenario, a few prototypes are selected to span the variation of shape within each category. Every shape in the database is then aligned with each of the prototypes using modal matching, and the resulting modal strain energy is stored as an n-tuple Υ, where n is the number of prototypes. Each shape in the database now has a coordinate in this "strain-energy-from-prototypes" space; shapes can be compared simply in terms of their Euclidean distance in this space.

We have used strain energy for most of our object comparison experiments, since it has a convenient physical meaning; however, we suspect that it may sometimes be necessary to weigh higher-frequency modes less heavily, since these modes typically only describe high-frequency shape variations and are more susceptible to noise. For instance, we could directly measure distances between modal descriptions, $\tilde{\mathbf{U}}$. Our preliminary experiments in prototype-based shape description have shown that this metric yields comparable performance to the strain energy metric.

3.3 Spanning Categories with Prototypes

In our current image database system, a human operator selects a few example shapes that approximately span each category. Our system performance is therefore dependent on the user's ability to select an adequately diverse and sufficient set of prototypes. It may be desirable to have a system that could automatically select prototypes in an unsupervised fashion. An unsupervised learning or clustering (e.g., k-means, hiearchical clustering, iterative optimization, Bayes classifiers) could be adapted for automatically selecting the prototype shapes based on modal matching and modal strain. Using such methods introduces a tradeoff, because for many pattern classification and learning schemes it is critical that training data sets be large and diverse enough to characterize the variations within a particular shape class [14]. This shifts the pressure from a human selecting adequate prototypes to a human providing sufficient diverse and large training data set (and providing the number of categories present). Finding these clusters without prototypes would (in general) require matching all shapes to all other shapes before optimal clusters could be obtained. In either case, qualities missing from either the training data or the prototypes may be ignored or misinterpreted.

Another issue is orthogonality. It is unlikely that the selected shape prototypes will describe orthogonal axes in some idealized category space. To ensure orthogonality we have employed a method based on finding the principal components. Given a set of prototypes, we compute the strain-to-prototypes feature vector Υ and its covariance matrix for a randomly selected subset of shapes in the database. The eigenvectors Ψ of the covariance matrix are used to transform all Υ into new coordinates in an orthogonalized parameter space:

$$\Upsilon' = \Lambda^{-\frac{1}{2}} \Psi \Upsilon, \tag{9}$$

where Λ is a diagonal matrix containing the eigenvalues. Computing distances in this new space is equivalent to computing the Mahalonobis distance in the original strain-to-prototypes space. As before, variation orthogonal to the space spanned by the training set will not be represented. This may at first seem like a limitation; however, this property can be exploited to constrain the

allowable deformations to only those that are statistically most likely. Furthermore, principal components with eigenvalues less than a threshold can be discarded to gain a lower-dimensional parameter space as well as better robustness to noise [21].

The transform to the orthogonalized parameter space is done as a precomputation (prior to repeated database search). The method has been tested in experiments with a database of hand-tool images, as will be detailed in Section 4.

3.4 Comparison Without Direct Correspondence Computation

In an alternative method, we can measure the distance between modes and determine how similar two shape's modes are without actually computing feature correspondences. This can be accomplished by measuring the Hausdorff distance between the low-order mode vectors of the new shape and the low-order modes of a prototype.

Given two mode vectors ϕ_{a} and ϕ_{b}, one from the first shape A, and another from the second shape B, we can define the Hausdorff distance between these mode vectors as

$$H(\phi_{\mathrm{a}}, \phi_{\mathrm{b}}) = \max(h(\phi_{\mathrm{a}}, \phi_{\mathrm{b}}), h(\phi_{\mathrm{b}}, \phi_{\mathrm{a}})) \tag{10}$$

where $h(\phi_{\mathrm{a}}, \phi_{\mathrm{b}})$ is the directed Hausdorff distance from ϕ_{a} to ϕ_{b}:

$$h(\phi_{\mathrm{a}}, \phi_{\mathrm{b}}) = \max_{\phi_{\mathrm{a,i}} \in \phi_{\mathrm{a}}} \min_{\phi_{\mathrm{b,i}} \in \phi_{\mathrm{b}}} \| \phi_{\mathrm{a,i}} - \phi_{\mathrm{b,i}} \| . \tag{11}$$

The distance norm is taken between generalized features of Eq. 5: $(u_{\mathrm{a,i}}, v_{\mathrm{a,i}})$ and $(u_{\mathrm{b,i}}, v_{\mathrm{b,i}})$. In our experiments, we have used a Euclidean norm. This measure requires no specific correspondence between points on the two objects.

In this formulation, we match and compare modes; consequently, for each shape in the database we tally the number of modes that match each prototype's modes. Typically, mode distances are computed for only the lowest-order 25% or fewer of the nonrigid modes. These tallies can be stored as coordinates in an n-dimensional similarity space; thus, shape similarity is proportional to the Euclidean distance in this space.

Finally, if two shapes have no modes falling within the reasonable tolerance for similarity, then the shapes will be flagged as "no similar modes." This computation can precede direct point correspondence or alignment computation. The lack of modal similarity is a strong clue that the shapes are probably from different categories, and therefore, attempting correspondence and alignment would be unreasonable. This method is used in the experiments with the hand tool image database described in the next section.

4 Experiments in Interactive Search

In the first set of experiments, our method is used to structure an image database of fish. The images in this experimental database were digitized from a children's field guide [22]. Currently, there are 74 images of tropical fish in the database. Each image depicts a fish from the canonical viewpoint (side view), though orientation, position, and scale vary. Each fish is unoccluded and appears on a uniform background. Images for this and other experiments are available for anonymous FTP from *cs-pub.bu.edu* in the compressed tar file *sclaroff/pictures.tar.Z*.

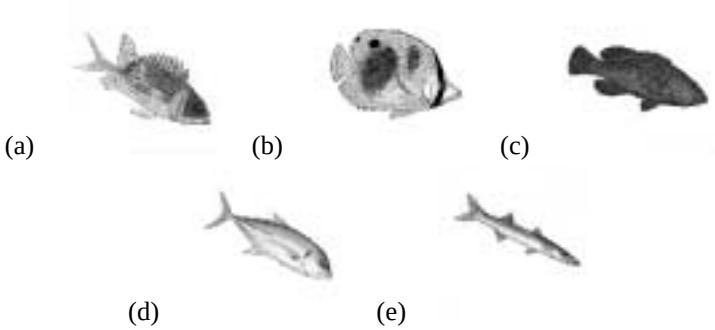

Figure 5: The five prototype shapes used in the image database experiment: (a) Squirrel Fish, (b) Spot Fin Butterflyfish, (c) Coney, (d) Horse Eye Jack, and (e) Southern Sennet.

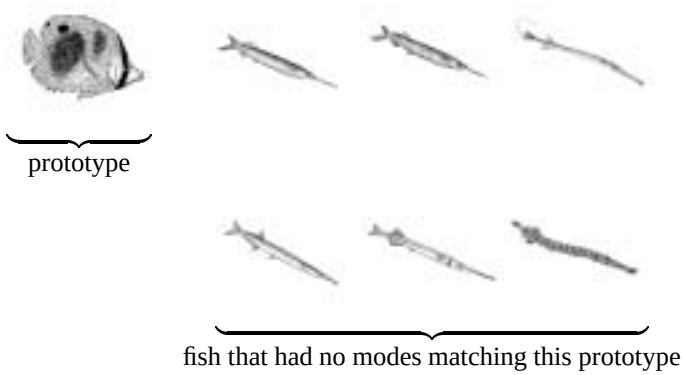

Figure 6: Six fish had no modes that came within tolerance of matching modes for the Butterfly Fish prototype in Figure 5(b), and are clearly not in the Butterfly Fish category.

We used the prototype-based shape description method formulated in Sec. 3.2, where each shape's strain-energy distance to the prototypes was precomputed and stored for interactive search later. First, for each image, a support map and edge image was computed, a finite-support shape model was built, and then the eigenmodes were determined. For the shapes in this experiment, approximately 60-70 finite element nodes were chosen so as to be roughly-regularly spaced across the support region.

Each shape in the database is then modal matched to a set of prototype images. There were five fish prototypes as shown in Fig. 5. These prototype images were selected by a human operator so as to span the range of shapes in the database. For fish prototypes, we chose prototypes that span the range from skinny fish (Fig. 5(e)), to fat fish (5(b)), and from smooth fish (5(c)) to prickly or pointy-tailed fish (5(a,d)).

Not all shapes in the database have similar modes (similarity

is measured to within a threshold). This information was quickly deterined by using the Hausdorff distance measure described in Section 3.4. Sometimes, as is shown in Fig. 6, even shapes within the same category do not have similar modes. In this particular case, the modes of the wide-bodied, Butterfly Fish prototype of Fig. 5 did not match well with the modes of the most narrow-bodied fish. Using the more efficient Hausdorff distance, we can quickly determine when modes are nowhere near being similar, and no attempt at alignment and strain energy computation is made. Such shapes are simply flagged as being "not at all similar" to a particular shape prototype, as described in Section 3.4.

The resulting modal strain energy was then used as a similarity metric in Photobook, an image database management system developed at the MIT Media Lab [42]. Using Photobook, the user selected the image at the upper left, and the system retrieved the remaining images sorted by strain energy (shape similarity) from left to right, top to bottom. The similarity measure is shown below each image.

The database searches in Figs. 7 through 10 were conducted using distance in prototype-space. In Fig. 7, a Banded Butterflyfish was selected. The matches are shown in order, starting with the most similar. Based on mode-similarity-distance, the system retrieved the animal shapes that were closest to the Banded Butterfly Fish shape (other Butterfly Fish, and other fat-bodied fish). In the second search, shown in Fig. 8, a Trumpet Fish was selected. In this case, the system retrieved similar long and skinny fish.

In both searches, the fish judged "most similar" by the system appeared on the same page in the field guide. This type of similarity judgment performance is an encouraging result, since fish appearing under the same heading are nearly always in the same taxonomic category, *e.g.*, Groupers, Jacks, Snappers, Porgies, Squirrelfishes, Butterflyfishes, Hamlets, or Damselfishes. In the cases where fish listed under the same heading are not in the same taxonomic category it is because they were grouped together due to some shape similarity, *e.g.*, "Slim-bodied fishes" is the heading under which the Trumetfish, Bluespotted Cornetfish, Balao, Needlefish, Ballyhoo, and Houndfish appear.

Fig. 9 continues this example, this time searching for shapes most similar to a Crevalle Jack and a Dog Snapper. Again, the matches are shown in order, starting with the most similar. The shapes most similar to a Crevalle Jack are other fish with similar body and tail shapes. In this case, the system rates Jolt Head Porgy over a closer relative (Yellow Jack). This is fairly reasonable, since all are closely-related, open water fish.

In the final example, Fig. 10, the user selected a Dog Snapper. Again the system rated fish from the same pages in the field guide as "most similar." In each example, search and display took less than a second on an HP 735.

Database queries were performed for each of the 72 fish images in the database for which there were other fish under same heading in the field guide. Overall, another fish under the same heading in the field guide was judged as most similar 71% of the time. To gain enhanced performance in capturing animal taxonomies, we suspect that modal matching would need to be part of a combined system that includes local feature and color information.

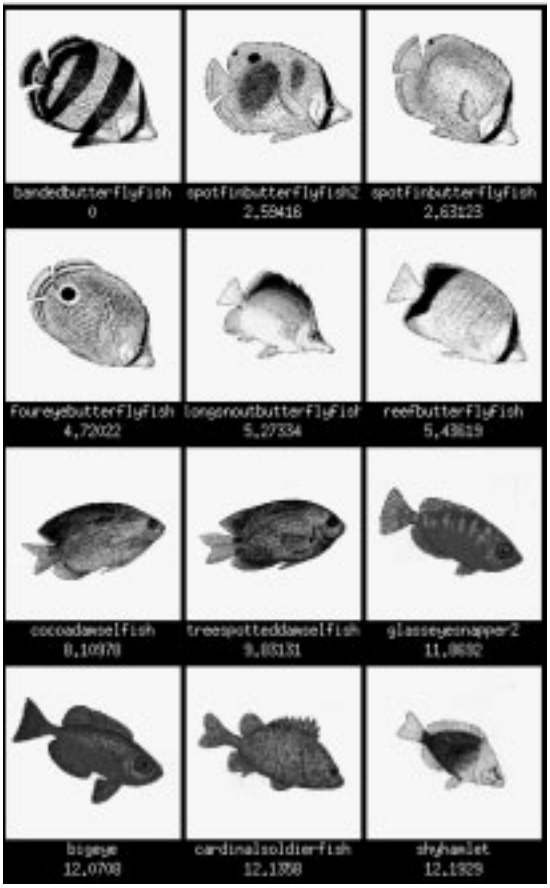

Figure 7: Searching an image database for similarly-shaped fish. In this example, distance in mode-similarity-space was used as a shape similarity metric. The figure shows the first of four examples of the ordering that resulted in searches for similar fish: a Banded Butterfly Fish. The matches are shown in order, starting with the most similar. Based on mode-similarity-distance, the system retrieved the animal shapes that were closest to the Banded Butterfly Fish shape (other Butterfly Fish, and other fat-bodied fish). The fish judged "most similar" by the system appeared on the same page in the original field guide book, and in the same taxonomic class.

For comparison, the same 72 queries were performed using moment invariants based on second- and third-order moments [15]. To gain better performance, the covariance matrix for the seven-dimensional feature vectors was computed and shapes where ordered in terms of their Mahalonobis distances to the selected shape. In this case another fish under the same heading in the field guide was judged as most similar 57% of the time.

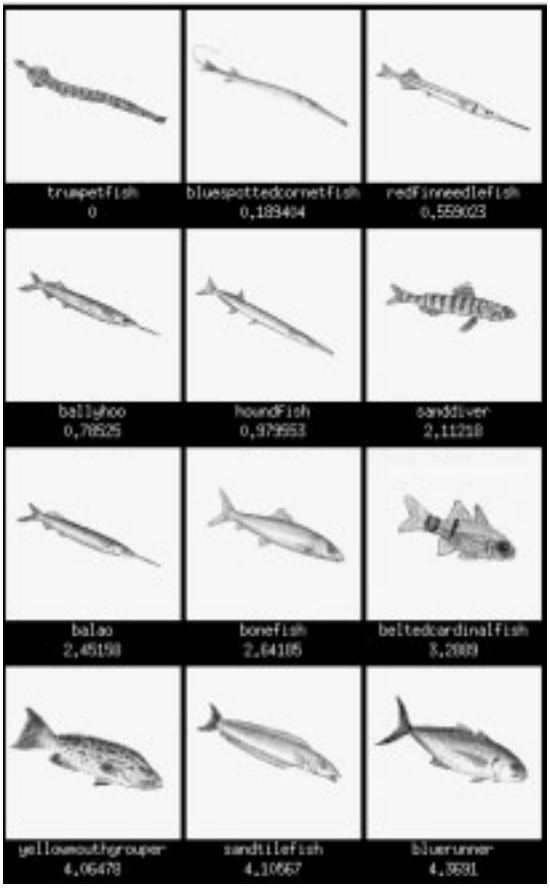

Figure 8: Searching an image database for similarly-shaped fish (continued): Trumpet Fish. The matches are shown in order, starting with the most similar. In this second search the system retrieved similar long and skinny fish. that are on the same page in the original field guide.

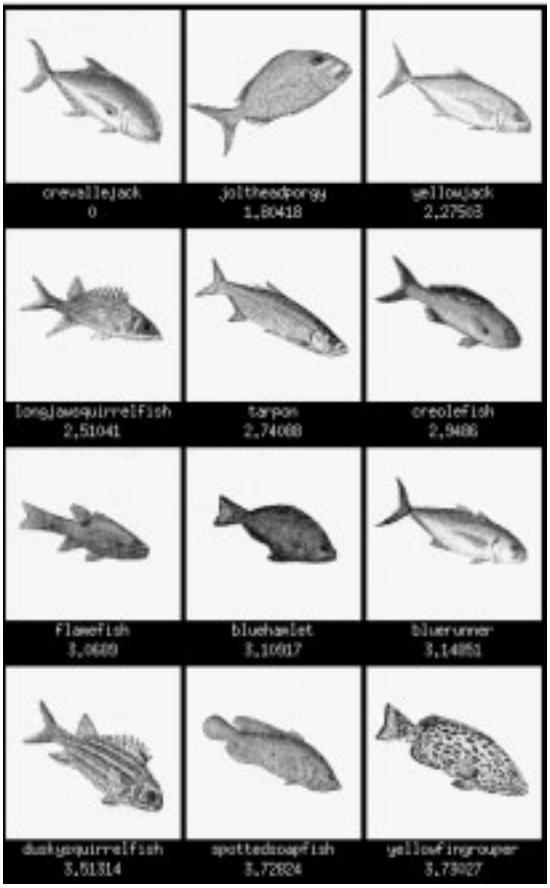

Figure 9: Ordering that resulted in searches for similar fish (continued): a Crevalle Jack. The shapes most similar to a Crevalle Jack are other fish with similar body shapes and pointed tails (other open water fish).

4.1 Evaluating Retrieval Accuracy using AVRR

Thus far retrieval performance has been measured in terms of the percentage of times that a shape in the same category is retrieved as "most similar" over a number of trials. However, the Photobook system, and other query by example (QBE) image database systems (IBM, Virage, Jacob) provide a list of possible matches ordered in terms of their similarity distance from the example image. This is in contrast to retrieval systems based on "exact match."

In "exact match" systems the standard measures of precision and recall can be employed [46]. However, as noted by Faloutsos, et al. [20], systems that offer a list of items sorted by similarity do not fall under the rubric of exact matching. We need a performance measure that embodies the positions in which target items appear in the retrieval. Ideally, if there were a total of n items of the same category in the database, then these n items would appear in the first n positions for a similarity-based retrieval.

To evaluate retrieval performance we will employ the *normalized recall* metric developed to evaluate IBM's QBIC [20]. Assume that the number of categories, shapes per category, and category membership for each shape are known. We can measure the average rank of all relevant items (AVRR) for a particular retrieval and then compare this with the ideal average rank (IAVRR) when all n images from a particular shape category appear in the first n positions. For a database that contains n shapes in each category IAVRR $= \frac{n}{2}$. In general, the equation for ideal average rank is

$$\text{IAVRR} = \frac{1}{m} \sum_{i=1}^{c} \frac{{n_i}^2}{2}, \tag{12}$$

where m is the total number of shapes in the database, c is the number of categories, and n_i is the number of shapes in the i^{th} category. The AVRR is computed based on the actual ordered ranking of shapes for each database retrieval. Thus the ratio of AVRR to IAVRR can be used to give a measure of average retrieval accuracy over a number of experimental trials.

Using this measure, the retrieval accuracy was evaluated for the previously described experiments with the fish image database in Photobook. The IAVRR for this database was 3.4 and the AVRR was 8.9. This means that on average the relevant image appears in the ninth position. The ratio of AVRR/IAVRR = 2.6. In con-

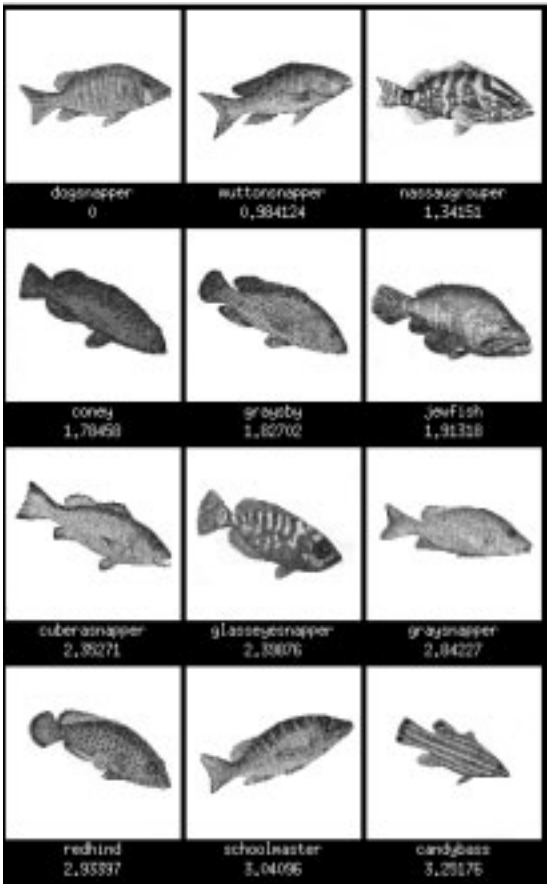

Figure 10: Ordering that resulted in search for fish similar to a Dog Snapper. The system rated fish from the same page in the field guide as "most similar." In each example, search and display took less than a second on an HP 735.

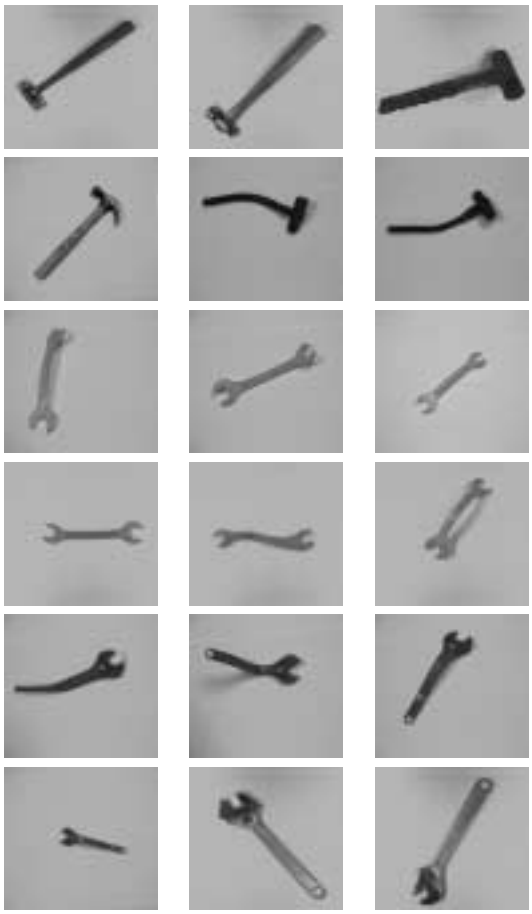

Figure 11: Some example images from the hand tools experimental image database. There are 63 images of children's toy tools and adult tools in the database, 21 each of category hammer, single-ended wrench, and double-ended wrench. Because the toy tools were made of plastic, they could be bent in various ways. Further, tools appeared in a number of orientations and/or scales, with varying lighting.

trast, the AVRR was 17.7 and AVRR/IAVRR = 5.2 when moment invariants were used.

4.2 Tool Image Database

In a second set of image database experiments we used a database of 63 grayscale images of real and toy hand tools. There were 21 images from each of three tool categories: wrenches, hammers, and crescent wrenches. Figure 11 shows example images taken from this database. Note that because the toy tools were made of plastic, they could be bent in various ways. Further, tools appeared in a number of orientations and/or scales, with varying lighting. The tools were placed on a uniform background so that a simple fuzzy c-Means clustering technique could be used for foreground/background separation [31].

For the shapes in this experiment, approximately 70-80 finite element nodes were chosen so as to be roughly-regularly spaced across the support region. Mode amplitudes for the first 32 modes were recovered and used to warp each prototype onto the other tools. As in the fish database experiments, the Hausdorff distance method was used to cull cases where no modes matched. The comparisons were made translation and rotation invariant by ignoring displacements in the rigid body modes. Comparisons were made scale invariant by recovering the scale factor before non-rigidly warping the shape to each prototype [25]. Total CPU time for database precomputation (match, align, and store n-tuple) averaged 3 seconds per prototype on an SGI Indigo2 workstation.

Matching experiments were then conducted using the coordinates produced via the orthogonalization procedure in Section 3.2. Database queries were performed for each of the 63 tool images in the database. Overall, another tool from the same category was judged as most similar 94% of the time, compared with 86% for the moments-based method.

For orthogonalized strain-from-prototypes, the AVRR was 18.2; this means that the average relevant image appears in roughly in the fifteenth position. The IAVRR for this database is 10.5. Thus the ratio of AVRR/IAVRR $= 1.7$. The moments-based method produced AVRR $= 23.1$ and AVRR/IAVRR $= 2.2$. As another point of comparison, performance for shape-based

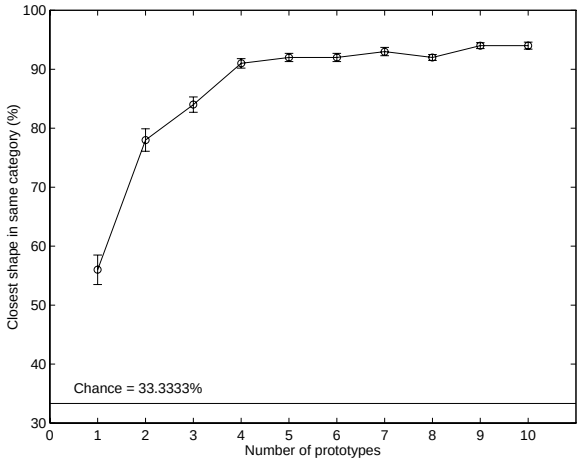

Figure 12: Graph showing how the number of prototypes affects the average performance for retrieval given a database of 63 handtools, 21 tools from each of three categories. For each trial, n prototypes were chosen at random from the database, and searches conducted in orthogonalized strain-to-prototypes space. For each n there were up to 1000 trials. Database queries were performed for each of the 63 tool images in the database. With only one prototype, another tool from the same category was judged as most similar 56% of the time. As the number of prototypes reached four, performance began to level off.

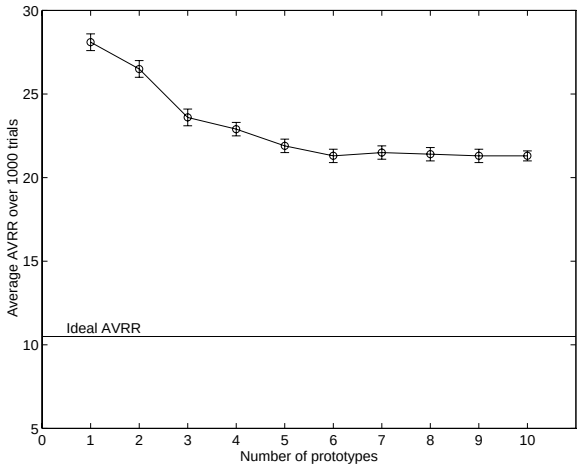

Figure 13: Graph showing how the number of prototypes affects the average rank (AVRR) for retrieval of the 21 handtools in the same tool category. For each trial, n prototypes were chosen at random from the database, and searches conducted in orthogonalized strain-to-prototypes space. For each n there were up to 1000 trials. With only one prototype, the AVRR averaged 28.1. The average performance leveled out at 5 prototypes where AVRR $= 21.9$. The ideal AVRR would be 10.5.

search in QBIC was reported to have a AVRR/IAVRR ratio of 1.8 in [20] for a database of 777 airplane silhouettes coarsely categorized by viewpoint and overall shape properties.

4.3 Number of Prototypes and Retrieval Accuracy

Using the tool image database, an experiment was conducted to evaluate retrieval accuracy as a function of the number of prototypes used. Multiple trials were conducted using between one and ten prototypes. These n prototypes were selected at random (uniformly distributed), in 1000 trials for each n. Average matching perfmance was evaluated using the coordinates produced via the orthogonalization procedure in Section 3.2. In each trial, database queries were performed for each of the 63 tool images in the database.

Figures 13 and 12 show the resulting performance curves. The graph in Figure 12 shows how the number of prototypes affects the average performance for database queries performed for each of the 63 tool images in the database. With only one prototype, another tool from the same category was judged as most similar 56% of the time. As the number of prototypes reached four, performance began to level off at approximately 90%.

The graph in Figure 13 shows how the number of prototypes affects the average AVRR for retrieval of the 21 handtools in the same tool category. With only one prototype, the AVRR averaged 28.1. The average performance leveled out at 5 prototypes where AVRR $= 21.9$. The ideal AVRR would be 10.5. The ratio AVRR/IAVRR is greater than two.

5 Discussion

One of the main motivations for this research was to provide improved shape representations for query by image content. While the shape comparison algorithms developed in the machine vision and pattern recognition communities can serve as a good starting point for developing shape-based image database search methods, retrieval by shape is still considered to be one of the most difficult aspects of content-based image search [20].

IBM's Query By Image Content system (QBIC) [20; 36] is perhaps the most advanced image database system to date; it is available as a commercial product. QBIC can perform searches that combine information about shape, color, and texture. As input, the system assumes non-occluded, planar shapes that are represented as a binary image. Shape-based search in QBIC cannot deal well with nonrigid deformation. Algebraic moment invariants [51] were intended for modeling rigid objects only. In addition, the higher moments are dominated by points that are farthest from the centroid; therefore, they are highly susceptible to outliers. Similar moments do not necessarily guarantee perceptually similar shapes.

Other shape indexing schemes have been based on local boundary features [34; 23], and are therefore not very robust to noise, scale, and sampling. Another system, proposed by Chen [9] identified 2-D aircraft shapes using elliptic Fourier descriptors. Because it is Fourier descriptor-based, Chen's system suffers from problems with sampling and parameterization. Jagadish introduced a multidimensional indexing scheme that offered the advantage that it could index images much faster than previous techniques [27]. However, the system had limited descriptive power, because the shape similarity measure was too simple (the

area difference between two shapes) and the underlying shape representation was polyhedral (representing shapes in terms of K-d-b trees of overlapping minimum bounding rectangles).

In contrast to previous formulations, the FEM integrals used in the modal model formulation provide greater robustness to sampling, outliers, and missing data. Furthermore, modal models provide quasi-invariance to different types of nonrigid deformation, while also providing an ordered, orthogonal, *encoding* of the nonrigid deformation that relates a candidate shape to a shape prototype or shape category.

5.1 Matching Human Similarity Judgments

For a image database search to be useful, it is critical that the shape similarity metric be able to match human judgments of similarity. This is *not* to say that the computation must somehow mimic the human visual system; but rather that computer and human judgments of similarity must be generally correlated. Without this, the images the computer finds will not be those desired by the human user.

For human shape similarity judgments, sometimes scale and rotation invariance are important, other times not [35]; it is therefore desirable to duplicate this performance in our image database search algorithms. In QBIC, a weighted metric allows for subset selection, and thus it provides selective invariance to size and orientation [20]. Modal matching also provides this invariance to size and orientation, but unlike any of the shape representations used in QBIC, modal representations can also be made invariant to affine deformations, and thus selectively invariant to changes in camera viewpoint. More importantly, the modal representation provides deformation "control knobs" that correspond qualitatively with human's notions of perceptual shape similarity [4; 41]. Shape is thought of in terms of an ordered set of deformations from an initial shape: starting with bends, tapers, shears, and moving up towards higher-frequency shape variations.

5.2 Speed of Image Database Search

Another concern in image database search is the computation speed. Shape-based image database search must be efficient enough to be interactive. A search that requires minutes per image is simply not useful in a database with millions of images. Furthermore, interactive search speed makes it possible for users to recursively refine a search by selecting examples from the currently retrieved images and using these to initiate a new select-sort-display cycle. Thus users can iterate a search to quickly "zero in on" what they are looking for.

As demonstrated in our image database experiments, searches on databases over one hundred images take less than a second (including image display) on an HP 735 workstation. In addition, search time in our system scales linearly on the number of shapes in the database. Finally, it is possible that the notion of building up prototype-based modal categories could be exploited to structure databases into taxonomic trees, thereby improving the computational complexity of image database search.

6 Conclusion

A new image database search method has been described. The method uses strain energy from deformable prototypes to encode shape categories. Retrieval accuracy of this approach has been demonstrated in a series of experiments with image databases of animals scanned from children's field guides and of deformable hand tools digitized via a video camera. In these experiments, the method performed consistently better than search on moment invariants. Experiments were also conducted to evaluate retrieval accuracy as a function of the number of prototypes used. Relatively few prototypes were needed to produce stable performance when a new orthogonalization scheme was employed.

7 Acknowledgment

This work began while the author was at the MIT Media Laboratory and is now supported in part by the Department of the Navy, Office of Naval Research Young Investigator Award N00014-96-1-0661.

References

[1] P. Alden and R. Grossenheider. *Peterson First Guide to Mammals of North America*. Houghton Mifflin Co., Boston, 1987.

[2] K. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, 1982.

[3] A. Baumberg and D. Hogg. Learning flexible models from image sequences. In *Proc. European Conference on Computer Vision*, pages 299–308, Stockholm, Sweden, May 1994.

[4] I. Biederman. Human Image Understanding: Recent Research and a Theory. *Computer Vision, Graphics and Image Processing*, 32(1):29–73, 1985.

[5] M. Black. Recursive non-linear estimation of discontinuous flow fields. In *Proc. European Conference on Computer Vision*, May 1994.

[6] A. Blake, R. Curwen, and A. Zisserman. A framework for spatiotemporal control in the tracking of visual contours. *International Journal of Computer Vision*, 11(2):127–146, 1993.

[7] F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. on Pattern*

Analysis and Machine Intelligence, 11(6):567–585, June 1989.

[8] M. Celenk. A color clustering technique for image segmentation. *Computer Graphics, Vision, and Image Processing*, 52:145–170, 1990.

[9] Z. Chen and S. Y. Ho. Computer vision for robust 3D aircraft recognition with fast library search. *Pattern Recognition*, 24(5):375–390, 1991.

[10] I. Cohen, N. Ayache, and P. Sulger. Tracking points on deformable objects. In *Proc. European Conference on Computer Vision*, Santa Margherita Ligure, Italy, May 1992.

[11] T. Cootes, D. Cooper, C. Taylor, and J. Graham. Trainable method of parametric shape description. *Image and Vision Computing*, 10(5):289–294, June 1992.

[12] T. Darrell and A. Pentland. Space-time gestures. In *Proc. CVPR*, pages 335–340, June 1993.

[13] T. Darrell and A. P. Pentland. Robust estimation of multiple models using support maps. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(5), May 1995.

[14] R. O. Duda and P. E. Hart. *Pattern Recognition and Scene Analysis*. John Wiley, New York, 1973.

[15] S. Dudani, K. Breeding, and R. McGhee. Aircraft identification by moment invariants. *IEEE Trans. Computers*, 26(1):39–47, 1977.

[16] J. Duncan, R. Owen, L. Staib, and P. Anandan. Measurement of non-rigid motion using contour shape descriptors. In *Proc. CVPR*, pages 318–324, 1991.

[17] S. Edelman. On learning to recognize 3-D objects from examples. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(8):833–837, 1993.

[18] S. Edelman. Representation, similarity, and chorus of prototypes. *Minds and Machines*, 5:45–68, 1995.

[19] S. Edelman and D. Weinshall. A self-orgainizing multiple-view representation for 3-D objects. *Biological Cybernetics*, 64:209–219, 1991.

[20] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262, 1994.

[21] K. Fukunaga and W. Koontz. Application of the karhunen-loeve expansion to feature selection and ordering. *IEEE Trans. Communications*, 19(4), 1970.

[22] I. Greenberg. *Guide to Corals and Fishes of Florida, th Bahamas and the Caribbean*. Seahawk Press, Miami, Florida, 1977.

[23] W. Grosky, P. Neo, and R. Mehrotra. A pictorial index mechanism for model-based matching. *Data and Knowledge Engineering*, 8:309–327, 1992.

[24] G. Healey. Segmenting images using normalized color. *IEEE Trans. on Systems, Man, and Cybernetics*, 22:64–73, 1992.

[25] B. Horn. Closed-form solution of absolute orientation using unit quarternions. *JOSA-A*, 4:629–642, 1987.

[26] M. A. Ireton and C. S. Xydeas. Classification of shape for content retrieval of images in a multimedia database. In *Proc. Sixth International Conference on Digital Processing of Signals in Communications*, pages 111–116, Loughborough, UK, September 1990.

[27] H. V. Jagadish. A retrieval technique for similar shapes. In *Proc. International Conference on Management of Data, ACM SIGMOD 91*, pages 208–217, Denver, CO, May 1991.

[28] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.

[29] G. J. Klinker, S.A. Shafer, and T. Kanade. A physical approach to color image understanding. *International Journal of Computer Vision*, 4:7–38, 1990.

[30] M. Leyton. Perceptual organization as nested control. *Biological Cybernetics*, 51:141–153, 1984.

[31] Young Won Lim and Sang Uk Lee. On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. *Pattern Recognition*, 23(9):935–952, 1990.

[32] J.Q. Liu and Y.H. Yang. Multiresolution color image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:689–700, 1994.

[33] D. Marr and K. Nishihara. Representation and Recognition of the Spatial Organization of Three-dimensional Shapes. In *Proc. of the Royal Society - London B*, volume 200, 1978.

[34] R. Mehrotra and W. I. Grosky. Shape matching utilizing indexed hypotheses generation and testing. *IEEE Transactions of Robotics and Automation*, 5(1):70–77, 1989.

[35] D. Mumford. Mathematical theories of shape: Do they model perception? In *Proc. SPIE Conf. on Geometric Methods in Computer Vision*, volume 1570, 1991.

[36] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, and P. Yanker. The QBIC project: Querying images by content using color, texture, and shape. In *Proc. SPIE Conf. on Storage and Retrieval of Image and Video Databases*, volume 1908, February 1993.

[37] Y. Ohta, T. Kanade, and T. Sakai. Color information for region segmentation. *Computer Graphics, Vision, and Image Processing*, 13:222–241, 1980.

[38] E. Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, 5:927–935, 1992.

[39] E. Oja and J. Karhunen. Nonlinear PCA: Algorithms and Applications. Technical Report A18, Helsinki University of Technology, Laboratory of Computer and Information Sciences, SF-02150 Espoo, Finland, 1993.

[40] A. Pentland. Perceptual organization and representation of natural form. *Artificial Intelligence*, 28(3):293–331, 1986.

[41] A. Pentland. Automatic extraction of deformable part models. *International Journal of Computer Vision*, 4(2):107–126, March 1990.

[42] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. *International Journal of Computer Vision*, to appear fall 1995.

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

[44] T. Poggio and F. Girosi. A theory of networks for approximation and learning. Technical Report A.I. Memo No. 1140, Artificial Intelligence Lab, MIT, Cambridge, MA, July 1989.

[45] E. Rosch. Cognitive representations of semantic categories. *Journal of Experimental Psychology: General*, 104:193–233, 1975.

[46] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[47] B. Scassellati, S. Alexopoulos, and M. Flickner. Retrieving images by 2D shape: comparison of computation methods with human perceptual judgements. In *Proc. SPIE Conf. on Storage and Retrieval of Image and Video Databases II*, San Jose, CA, February 1994.

[48] S. Sclaroff. *Modal Matching: A Method for Describing, Comparing, and Manipulating Digital Signals*. PhD thesis, MIT Media Lab, January 1995.

[49] S. Sclaroff and A. Pentland. Physically-based combinations of views: Representing rigid and nonrigid motion. In *Proc. IEEE Workshop on Nonrigid and Articulate Motion*, Austin, TX, Nov 1994.

[50] S. Sclaroff and A. Pentland. Modal Matching for Correspondence and Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(6):545–561, 1995.

[51] G. Taubin and D. Cooper. Recognition and positioning of rigid objects. In *Proc. SPIE Conf. on Geometric Methods in Computer Vision*, volume 1570, 1991.

[52] D. Terzopoulos. *Topical Meeting on Machine Vision*, volume 12 of *Technical Digest Series*, chapter On matching deformable models to images: Direct and iterative solutions, pages 160–167. Optical Society of America, Washington, DC, 1987.

[53] D. Terzopoulos. The Computation of Visible Surface Representations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(4):417–438, July 1988.

[54] W. B. Thompson. Combining motion and constrast for segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2:543–549, 1980.

[55] A. Tversky. Features of Similarity. *Psychological Review*, 84(4):327–352, 1977.

[56] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(10):992–1006, 1991.

[57] J.Y.A. Wang and E. H. Adelson. Layered representation for motion analysis. In *Proc. CVPR*, 1993.

[58] J. Weber and J. Malik. Robust computation of optical flow in a multi-scale differential framework. In *Proc. International Conference on Computer Vision*, pages 12–20, 1993.

[59] A. Yuille, D. Cohen, and P. Hallinan. Feature extraction from faces using deformable templates. In *Proc. CVPR*, pages 104–109, San Diego, 1989.