1995-03

Modal Matching for Correspondence and Recognition

Boston University Computer Science Dept. TR95-008. Appears in IEEE TPAMI, June 1995.

Modal Matching for Correspondence and Recognition

S. Sclaroff and A. Pentland

Abstract— **Modal matching is a new method for establishing correspondences and computing canonical descriptions. The method is based on the idea of describing objects in terms of generalized symmetries, as defined by each object's eigenmodes. The resulting modal description is used for object recognition and categorization, where shape similarities are expressed as the amounts of modal deformation energy needed to align the two objects. In general, modes provide a global-to-local ordering of shape deformation and thus allow for selecting which types of deformations are used in object alignment and comparison. In contrast to previous techniques, which required correspondence to be computed with an initial or prototype shape, modal matching utilizes a new type of finite element formulation that allows for an object's eigenmodes to be computed directly from available image information. This improved formulation provides greater generality and accuracy, and is applicable to data of any dimensionality. Correspondence results with 2-D contour and point feature data are shown, and recognition experiments with 2-D images of hand tools and airplanes are described.**

Keywords— **Correspondence, shape description, shape invariants, object recognition, deformation, finite element methods, modal analysis, vibration modes, eigenmodes.**

I. Introduction

A key problem in machine vision is how to describe features, contours, surfaces, and volumes so that they can be recognized and matched from view to view. The primary difficulties are that object descriptions are sensitive to noise, that an object can be nonrigid, and that an object's appearance deforms as the viewing geometry changes. These problems have motivated the use of deformable models [6;7;9;14;17;22;34;36;37], to interpolate, smooth, and warp raw data.

Deformable models do not by themselves provide a method of computing canonical descriptions for recognition, or of establishing correspondence between sets of data. To address the recognition problem we proposed a method of representing shapes as canonical deformations from some prototype object [18;22]. By describing object shape terms of the eigenvectors of the prototype object's stiffness matrix, it was possible to obtain a robust, frequency-ordered shape description. Moreover, these eigenvectors or *modes* provide an intuitive method for shape description because they correspond to the object's generalized axes of symmetry. By representing objects in terms of modal deformations we developed robust methods for 3-D shape modeling, object recognition, and 3-D tracking utilizing point, contour, 3-D, and optical flow data [18;20;22].

However this method still did not address the problem of determining correspondence betweeen sets of data, or between data and models. This was because every object had to be described as deformations from a *single* prototype object. This implicitly imposed an *a priori* parameterization upon the sensor data, and therefore implicitly determined the correspondences between data and the prototype.

In this paper we generalize our earlier method by obtaining the modal shape invariants directly from the sensor data. This will allow us to compute robust, canonical descriptions for recognition *and* to solve correspondence problems for data of any dimensionality. For the purposes of illustration, we will give a detailed mathematical formulation for 2D problems, and demonstrate it on gray-scale image and point feature data. The extension to data of other dimensionality is described in a technical report [28]. To illustrate the use of this method for object recognition and category classification, we will present an example of recognizing and categorizing images of hand tools.

II. The Basic Idea

Imagine that we are given two sets of image feature points, and that our goal is to determine if they are from two similar objects. The most common approach to this problem is to try to find distinctive local features that can be matched reliably; this fails because there is insufficient local information, and because viewpoint and deformation changes can radically alter local feature appearance.

An alternate approach is to first determine a body-centered coordinate frame for each object, and then attempt to match up the feature points. Once we have the points described in intrinsic or *body-centered* coordinates rather than Cartesian coordinates, it is easy to match up the bottom-right, top-left, etc. points between the two objects.

Many methods for finding a body-centered frame have been suggested, including moment-of-inertia methods, symmetry finders, and polar Fourier descriptors (for a review see [1]). These methods generally suffer from three difficulties: sampling error, parameterization error, and non-uniqueness. The main contribution of this paper is a new method for computation of a local coordinate frame that largely avoids these three difficulties.

Manuscript submitted August 1, 1993. Revised October 3, 1994. This research was done at the MIT Media Lab and was funded by British Telecom.

S. Sclaroff is with the Computer Science Department, Boston University, 111 Cummington St., Boston, MA 02215.
A. P. Pentland is with the Media Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139.

Sampling error is the best understood of the three. Everyone in vision knows that which features you see and their location can change drastically from view to view. The most common solution to this problem is to only use global statistics such as moments-of-inertia; however, such methods offer a weak and partial solution at best.

Parameterization error is more subtle. The problem is that when (for instance) fitting a deformable sphere to 3-D measurements one implicitly imposes a radial coordinate system on the data rather than letting the data determine the correct coordinate system. Consequently, the resulting description is strongly affected by, for instance, the compressive and shearing distortions typical of perspective. The number of papers on the topic of skew symmetry is indicative of the seriousness of this problem.

Non-uniqueness is an obvious problem for recognition and matching, but one that is all too often ignored in the rush to get *some* sort of stable description. Virtually all spline, thin-plate, and polynomial methods suffer from this inability to obtain canonical descriptions; this problem is due to fact that in general, the parameters for these surfaces can be arbitrarily defined, and are therefore not invariant to changes in viewpoint, occlusion, or nonrigid deformations.

Our solution to these problems has three parts:

1. We compute a shape description that is robust with respect to sampling by using Galerkin interpolation, which is the mathematical underpinning of the finite element method (FEM).
2. We introduce a new type of Galerkin interpolant based on Gaussians that allows us to efficiently derive our shape parameterization directly from the data.
3. We then use the eigenmodes of this shape description to obtain a canonical, frequency-ordered orthogonal coordinate system. This coordinate system may be thought of as the shape's *generalized symmetry axes.*

By describing feature point locations in this body-centered coordinate system, it is easy to match corresponding points, and to measure the similarity of different objects. This allows us to recognize objects, and to determine if different objects are related by simple physical transformations.

A flow-chart of our method is shown in Figure 1. For each image we start with feature point locations $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_m]$ and use these as nodes in building a finite element model of the shape. We can think of this as constructing a model of the shape by covering each feature point with a Gaussian blob of rubbery material; if we have segmentation information, then we can fill in interior areas and trim away material that extends outside of the shape.

We then compute the eigenmodes (eigenvectors) ϕ_i of the finite element model. The eigenmodes provide an orthogonal frequency-ordered description of the shape and its natural deformations. They are sometimes referred to as *mode shape vectors* since they describe how each mode deforms the shape by displacing the original feature locations, *i.e.*,

$$\mathbf{X}_{deformed} = \mathbf{X} + a\phi_i, \tag{1}$$

where a is a scalar.

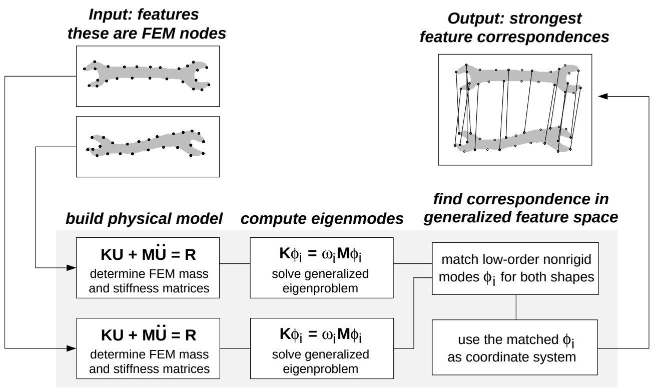


Fig. 1. System diagram.

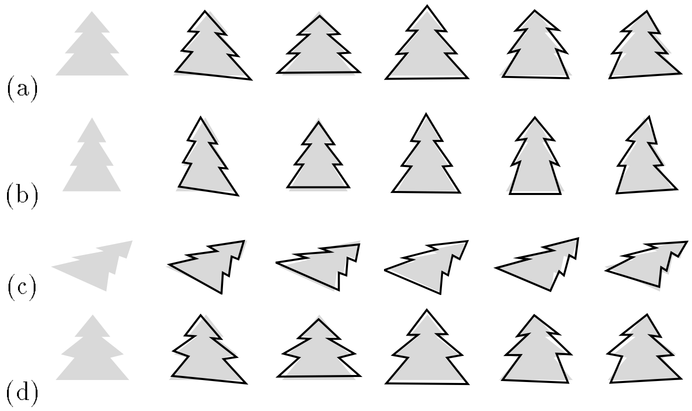


Fig. 2. Similar shapes have similar low order modes. This figure shows the first five low-order eigenmodes for similar tree shapes: (a) prototypical, (b) stretched, (c) tilted, and (d) two middle branches stretched.

The first three eigenmodes are the rigid body modes of translation and rotation, and the rest are nonrigid modes. The nonrigid modes are ordered by increasing frequency of vibration; in general, low-frequency modes describe global deformations, while higher-frequency modes describe more localized shape deformations. This global-to-local ordering of shape deformation will prove very useful for shape matching and comparison.

The eigenmodes also 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 moves within each eigenmode.* The transform between Cartesian feature locations and modal feature locations is accomplished by using the FEM eigenvectors as a coordinate basis. In our technique, two groups of features are compared in this eigenspace. The important idea here is that the low-order modes computed for two similar objects will be very similar — even in the presence of affine deformation, nonrigid deformation, local shape perturbation, or noise.

To demonstrate this, Figure 2 shows a few of the low-order nonrigid modes computed for four related tree

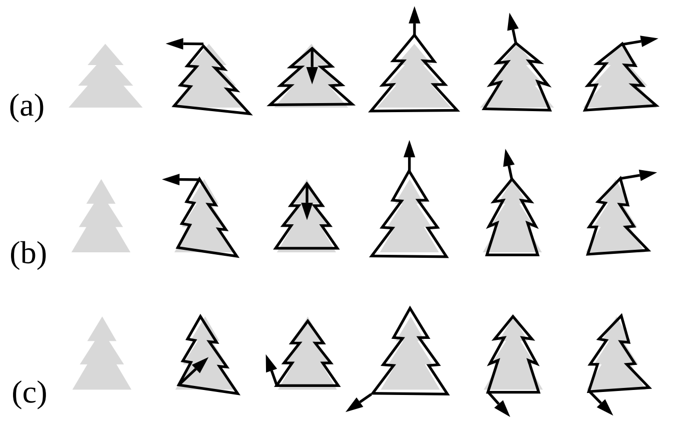


Fig. 3. Computing correspondences in modal signature space. Given two similar shapes, correspondences are found by comparing the direction of displacement at each node (shown by vectors in figure). For instance, the top points on the two trees (a, b) have very similar displacement signatures, while the bottom point (shown in c) has a very different displacement signature. Using this property, we can reliably compute correspondence affinities in this modal signature space.

shapes: (a) upright, (b) stretched, (c) tilted, and (d) two middle branches stretched. Each row in the figure shows the original shape in gray, and its low-order mode shapes are overlaid in black outline. By looking down a column of this figure, we can see how a particular low-order eigenmode corresponds nicely for the related shapes. This eigenmode similarity allows us to match the feature locations on one object with those of another despite sometimes large differences in shape.

Using this property, feature correspondences are found via *modal matching.* The concept of modal matching is demonstrated on the two similar tree shapes in Figure 3. Correspondences are found by comparing the direction of displacement at each node. The direction of displacement is shown by vectors in figure. For instance, the top points on the two trees in Figure 2(a, b) have very similar displacements across a number of low-order modes, while the bottom point (shown in Figure 2(c)) has a very different displacement signature. Good matches have similar displacement signatures, and so the system matches the top points on the two trees.

Point correspondences between two shapes can be reliably determined by comparing their trajectories in this modal space. In the implementation described in this paper, points that have the most similar unambiguous coordinates are matched via modal matching, with the remaining correspondences determined by using the physical model as a smoothness constraint. Currently, the algorithm has the limitation that it cannot reliably match largely occluded or partial objects.

Finally, given correspondences between many of the feature points on two objects, we can measure their difference in shape. Because the modal framework decomposes deformations into an orthogonal set, we can selectively measure rigid-body differences, or low-order projective-like deformations, or deformations that are primarily local. Consequently, we can recognize objects in a very flexible and general manner.

Alternatively, given correspondences we can align or warp one shape into another. Such alignment is useful for fusing data from different sensors, or for comparing data acquired at different times or under different conditions. It is also useful in computer graphics, where the warping of one shape to another is known as "morphing." In current computer graphics applications the correspondences are typically determined by hand [4;31;44].

III. Background and Notation

A. Eigen-Representations

In the last few years there has been a revival of interest in pattern recognition methods, due to the surprisingly good results that have been obtained by combining these methods with modern machine vision representations. Using these approaches researchers have built systems that perform stable, interactive-time recognition of faces [39], cars [16], and biological structures [6;19], and allowed interactive-time tracking of complex and deformable objects [5;8;20;38].

Typically, these methods employ eigen-decompositions like the modal decomposition or any of a family of methods descended from the Karhunen-Loève transform. Some are feature-based *eigenshapes* [3;8;26;27;30;32], others are physically-based *eigensnakes* [5;6;19;22;27], and still others are based on (preprocessed) image intensity information, *eigenpictures* [11;15;16;21;38;39].

In these methods, image or shape information is decomposed into an ordered basis of orthogonal principal components. As a result, the less critical and often noisy high-order components can be discarded in order to obtain overconstrained, canonical descriptions. This allows for the selection of only the most important components to be used for efficient data reduction, real-time recognition and navigation, and robust reconstruction. Most importantly, the orthogonality of eigen-representations ensures that the recovered descriptions will be unique, thus making recognition problems tractable.

Modal matching, the new method described in this paper, utilizes the eigenvectors of a physically-based shape representation, and is therefore most closely related to eigenshapes and eigensnakes. At the core of all of these techniques is a positive definite matrix that describes the connectedness between features. By finding the eigenvectors of this matrix, we can obtain a new, generalized coordinate system for describing the location of feature points.

One such matrix, the *proximity matrix*, is closely related to classic potential theory and describes Gaussian-weighted distances between point data. Scott and Longuet-Higgins [30] showed that the eigenvectors of this matrix can be used to determine correspondences between two sets of points. This coordinate system is invariant to rotation, and somewhat robust to small deformations and noise. A substantially improved version of this approach was developed by Shapiro and Brady [32;33]. Similar methods have been applied to the problem of weighted graph matching by Umeyama [41], and for Gestalt-like clustering of dot stimuli by van Oeffelen and Vos [42]. Unfortunately, proximity methods are not information preserving, and therefore cannot be used to interpolate intermediate deformations or to obtain canonical descriptions for recognition.

In a different approach, Samal and Iyengar [26] enhanced the generalized Hough transform (GHT) by computing the Karhunen-Loève transform for a set of binary edge images for a general population of shapes in the same family. The family of shapes is then represented by its significant eigenshapes, and a reference table is built and used for a Hough-like shape detection algorithm. This makes it possible for the GHT to represent a somewhat wider variation (deformation) in shapes, but as with the GHT, their technique cannot deal very well with rotations, and it has the disadvantage that it computes the eigenshapes from binary edge data.

Cootes, *et al.* [3;8] introduced a chord-based method for capturing the invariant properties of a class of shapes, based on the idea of finding the principal variations of a snake. Their *point distribution model* (PDM) relies on representing objects as sets of labeled points, and examines the statistics of the variation over the training set. A covariance matrix is built that describes the displacement of model points along chords from the prototype's centroid. The eigenvectors are computed for this covariance matrix, and then a few of the most significant components are used as deformation control knobs for the snake. Unfortunately, this method relies on the consistent sampling and hand-labeling of point features across the entire training set and cannot handle large rotations.

Each of these previous approaches is based directly on the sampled feature points. When different feature points are present in different views, or if there are different sampling densities in different views, then the shape matrix for the two views will differ even if the object's pose and shape are identical. In addition, these methods cannot incorporate information about feature connectivity or distinctiveness; data are treated as clouds of identical points. Most importantly, none of these approaches can handle large deformations unless feature correspondences are given.

To get around these problems, we propose a formulation that uses the finite element technique of Galerkin surface approximation to avoid sampling problems and to incorporate outside information such as feature connectivity and distinctiveness. The eigenvectors of the resulting matrices can be used both for describing deformations and for finding feature correspondences. The previous work in physically-based correspondence is described briefly in the next section.

B. Physically-Based Correspondence and Shape Comparison

Correspondence has previously been formulated as an equilibrium problem, which has the attractive feature of allowing integration of physical constraints [18;22;20;37;36]. To accomplish this, we first imagine that the collection of feature points in one image is attached by springs to an elastic body. Under the load exerted by these springs, the elastic body will deform to match the shape outlined by the set of feature points. If we repeat this procedure in each image, we can obtain a feature-to-feature correspondence by noting which points project to corresponding locations on the two elastic bodies.

If we formulate this equilibrium problem in terms of the eigenvectors of the elastic body's stiffness matrix, then closed-form solutions are available [18]. In addition, high-frequency eigenvectors can be discarded to obtain overconstrained, canonical descriptions of the equilibrium solution. These descriptions have proven useful for object recognition [22] and tracking [20].

The most common numerical approach for solving equilibrium problems of this sort is the *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, rather than just at the nodes [2] (nodes are typically the spring attachment points). The ability to integrate material properties over the whole surface alleviates problems caused by irregular sampling of feature points. It also allows variation of the elastic body's properties in order to weigh reliable features more than noisy ones, or to express *a priori* constraints on size, orientation, smoothness, *etc.* The following section will describe this approach in some detail.

C. Finite Element Method

Using Galerkin's method for finite element discretization, we can set up a system of shape functions that relate the displacement of a single point to the relative displacements of all the other nodes of an object. This set of shape functions describes an *isoparametric finite element.* By using these functions, we can calculate the deformations that spread uniformly over the body as a function of its constitutive parameters.

In general, the polynomial shape function for each element is written in vector form as:

$$\mathbf{u}(\mathbf{x}) = \mathbf{H}(\mathbf{x})\mathbf{U} \tag{2}$$

where **H** is the interpolation matrix, **x** is the local coordinate of a point in the element where we want to know the displacement, and **U** denotes a vector of displacement components at each element node.

For most applications it is necessary to calculate the strain due to deformation. Strain ϵ is defined as the ratio of displacement to the actual length, or simply the ratio of the change in length. The polynomial shape functions can be used to calculate the strains (ϵ) over the body provided the displacements at the node points are known. Using this fact we can now obtain the corresponding element strains:

$$\epsilon(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{U} \tag{3}$$

where **B** is the strain displacement matrix. The rows of **B** are obtained by appropriately differentiating and combining rows of the element interpolation matrix **H**.

As mentioned earlier, we need to solve the problem of deforming an elastic body to match the set of feature points. This requires solving the *dynamic equilibrium equation*:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{D}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R}, \tag{4}$$

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.

Both the mass and stiffness matrices are computed directly:

$$\mathbf{M} = \textstyle\int_V \rho \mathbf{H}^T \mathbf{H} dV \quad \text{and} \quad \mathbf{K} = \textstyle\int_V \mathbf{B}^T \mathbf{C} \mathbf{B} dV, \tag{5}$$

where ρ is the mass density, and $\mathbf{C}$ is the *material matrix* that expresses the material's particular stress-strain law.

If we assume Rayleigh damping, then the damping matrix is simply a linear combination of the mass and stiffness matrices:

$$\mathbf{D} = \alpha \mathbf{M} + \beta \mathbf{K}, \tag{6}$$

where α and β are constants determined by the desired critical damping [2].

D. Mode Superposition Analysis

This system of equations can be decoupled by posing the equations in a basis defined by the $\mathbf{M}$-orthonormalized eigenvectors of $\mathbf{M}^{-1}\mathbf{K}$. These eigenvectors and values are the solution $(\boldsymbol{\phi}_i, \omega_i^2)$ to the following generalized eigenvalue problem:

$$\mathbf{K}\boldsymbol{\phi}_i = \omega_i^2 \mathbf{M} \boldsymbol{\phi}_i. \tag{7}$$

The vector $\boldsymbol{\phi}_i$ is called the ith *mode shape vector* and ω_i is the corresponding frequency of vibration.

The mode shapes can be thought of as describing the object's generalized (nonlinear) axes of symmetry. We can write Equation 7 as

$$\mathbf{K}\mathbf{\Phi} = \mathbf{M}\mathbf{\Phi}\mathbf{\Omega}^2 \tag{8}$$

where

$$\mathbf{\Phi} = [\phi_1 \mid \ldots \mid \phi_m] \quad \text{and} \quad \mathbf{\Omega}^2 = \begin{bmatrix} \omega_1^2 & & \\ & \ddots & \\ & & \omega_m^2 \end{bmatrix}. \tag{9}$$

As mentioned earlier, each mode shape vector ϕ_i is $\mathbf{M}$-orthonormal, this means that

$$\mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi} = \mathbf{\Omega}^2 \quad \text{and} \quad \mathbf{\Phi}^T \mathbf{M} \mathbf{\Phi} = \mathbf{I}. \tag{10}$$

This generalized coordinate transform $\mathbf{\Phi}$ is then used to transform between nodal point displacements $\mathbf{U}$ and decoupled modal displacements $\tilde{\mathbf{U}}$:

$$\mathbf{U} = \mathbf{\Phi}\tilde{\mathbf{U}} \tag{11}$$

We can now rewrite Equation 4 in terms of these generalized or modal displacements, obtaining a decoupled system of equations:

$$\ddot{\tilde{\mathbf{U}}}_t + \tilde{\mathbf{D}}\dot{\tilde{\mathbf{U}}} + \mathbf{\Omega}_t^2 \tilde{\mathbf{U}}_t = \mathbf{\Phi}_t^T \mathbf{R}, \tag{12}$$

where $\tilde{\mathbf{D}}$ is the diagonal modal damping matrix.

By decoupling these equations, we allow for closed-form solution to the equilibrium problem [22]. Given this equilibrium solution in the two images, point correspondences can be obtained directly.

By discarding high frequency eigenmodes 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 [22].

IV. A New Formulation

Perhaps the major limitation of previous methods is that the procedure of attaching virtual springs between data points and the surface of the deformable object implicitly imposes a standard parameterization on the data. We would like to avoid this as much as is possible, by letting the data determine the parameterization in a natural manner.

To accomplish this we will 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. We will first develop a finite element formulation using Gaussian basis functions as Galerkin interpolants, and then use these interpolants to obtain generalized mass and stiffness matrices.

Intuitively, the interpolation functions provide us with a smoothed version of the feature points, in which areas between close-by feature points are filled in with a virtual material that has mass and elastic properties. The filling-in or smoothing of the cloud of feature points provides resistance to feature noise and missing features. The interpolation functions also allow us to place greater importance on distinctive or important features, and to discount unreliable or unimportant features. This sort of emphasis/de-emphasis is accomplished by varying the "material properties" of the virtual material between feature points.

A. Gaussian Interpolants

Given a collection of m sample points $\mathbf{x}_i$ from an image, we need to build appropriate stiffness and mass matrices. The first step towards this goal is to choose a set of interpolation functions from which we can derive $\mathbf{H}$ and $\mathbf{B}$ matrices. We require a set of continuous interpolation functions h_i such that:

1. their value is unity at node i and zero at all other nodes
2. $\sum_{i=1}^{m} h_i = 1.0$ at any point on the object

In a typical finite element solution for engineering, Hermite or Lagrange polynomial interpolation functions are used [2]. Stiffness and mass matrices $\mathbf{K}$ and $\mathbf{M}$ are precomputed for a simple, rectangular isoparametric element, and then this simple element is repeatedly warped and copied to tessellate the region of interest. This *assemblage* technique has the advantage that simple stiffness and mass matrices can be precomputed and easily assembled into large matrices that model topologically complex shapes.

Our problem is different in that we want to examine the eigenmodes of a cloud of feature points. It is akin to the problem found in interpolation networks: we have a fixed number of scattered measurements and we want to find a set of basis functions that allows for easy insertion and movement of data points. Moreover, since the position of nodal points will coincide with feature and/or sample points from our image, stiffness and mass matrices will need to be built on a per-feature-group basis. Gaussian basis functions are ideal candidates for this type of interpolation problem [23;24]:

$$g_i(\mathbf{x}) = e^{-\|\mathbf{x}-\mathbf{x}_i\|^2/2\sigma^2} \tag{13}$$

where $\mathbf{x}_i$ is the function's n-dimensional center, and σ its standard deviation.

We will build our interpolation functions h_i as the sum of m basis functions, one per data point $\mathbf{x}_i$:

$$h_i(\mathbf{x}) = \sum_{k=1}^{m} a_{ik} g_k(\mathbf{x}) \tag{14}$$

where a_{ik} are coefficients that satisfy the requirements outlined above. The matrix of interpolation coefficients can be solved for by inverting a matrix of the form:

$$\mathbf{G} = \begin{bmatrix} g_1(\mathbf{x}_1) & \ldots & g_1(\mathbf{x}_m) \\ \vdots & & \vdots \\ g_m(\mathbf{x}_1) & \ldots & g_m(\mathbf{x}_m) \end{bmatrix}. \tag{15}$$

By using these Gaussian interpolants as our shape functions for Galerkin approximation, we can easily formulate finite elements for any dimension. A very useful aspect of Gaussians is that they are factorizable: multidimensional interpolants can be assembled out of lower dimensional Gaussians. This not only reduces computational cost, it also has useful implications for VLSI hardware and neural-network implementations [23].

Note that these sum-of-Gaussians interpolants are nonconforming, i.e., they do not satisfy condition (2) above. As a consequence the interpolation of stress and strain between nodes is not energy conserving. Normally this is of no consequence for a vision application; indeed, most of the finite element formulations used in vision research are similarly nonconforming [37]. If a conforming element is desired, this can be obtained by including a normalization term in h_i in Equation 14,

$$h_i(\mathbf{x}) = \frac{\displaystyle\sum_{k=1}^{m} a_{ik} g_k(\mathbf{x})}{\displaystyle\sum_{j=1}^{m}\sum_{k=1}^{m} a_{jk} g_k(\mathbf{x})} . \tag{16}$$

In this paper we will use the simpler, non-conforming interpolants, primarily because the integrals for mass and stiffness can be computed analytically. The differences between conforming and nonconforming interpolants do not affect the results reported in this paper.

B. Formulating a 2-D Mass Matrix

For the sake of illustration we will now give the mathematical details for a two dimensional implementation. We begin by assembling a 2-D interpolation matrix from the shape functions developed above:

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} h_1 & \ldots & h_m & 0 & \ldots & 0 \\ 0 & \ldots & 0 & h_1 & \ldots & h_m \end{bmatrix}. \tag{17}$$

Substituting into Equation 5 and multiplying out we obtain a mass matrix for the feature data:

$$\mathbf{M} = \int_A \rho \mathbf{H}^T \mathbf{H} dA = \begin{bmatrix} \mathbf{M}_{aa} & 0 \\ 0 & \mathbf{M}_{bb} \end{bmatrix}, \tag{18}$$

where the m by m submatrices $\mathbf{M}_{aa}$ and $\mathbf{M}_{bb}$ are positive definite symmetric, and $\mathbf{M}_{aa} = \mathbf{M}_{bb}$. The elements of $\mathbf{M}_{aa}$ have the form:

$$m_{aa_{ij}} = \rho \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sum_{k,l} a_{ik}\, a_{jl}\, g_k(\mathbf{x})\, g_l(\mathbf{x})\, dx\, dy. \tag{19}$$

We then integrate and regroup terms:

$$m_{aa_{ij}} = \rho\,\pi\,\sigma^2 \sum_{k,l} a_{ik}\, a_{jl} \sqrt{g_{kl}} \tag{20}$$

where $g_{kl} = g_k(\mathbf{x}_l)$ is an element of the $\mathbf{G}$ matrix in Equation 15.

This can be rewritten in matrix form:

$$\mathbf{M}_{aa} = \mathbf{M}_{bb} = \rho\,\pi\,\sigma^2 \mathbf{A}^T \mathcal{G} \mathbf{A} = \rho\,\pi\,\sigma^2 \mathbf{G}^{-1} \mathcal{G} \mathbf{G}^{-1}, \tag{21}$$

where the elements of $\mathcal{G}$ are the square roots of the elements of the $\mathbf{G}$ matrix in Equation 15.

C. Formulating a 2-D Stiffness Matrix

To obtain a 2-D stiffness matrix $\mathbf{K}$ we need to compute a stress-strain interpolation matrix $\mathbf{B}$ and material matrix $\mathbf{C}$. For our two dimensional problem, $\mathbf{B}$ is a $(3 \times 2m)$ matrix:

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x} h_1 & \ldots & \frac{\partial}{\partial x} h_m & 0 & \ldots & 0 \\ 0 & \ldots & 0 & \frac{\partial}{\partial y} h_1 & \ldots & \frac{\partial}{\partial y} h_m \\ \frac{\partial}{\partial y} h_1 & \ldots & \frac{\partial}{\partial y} h_m & \frac{\partial}{\partial x} h_1 & \ldots & \frac{\partial}{\partial x} h_m \end{bmatrix}, \tag{22}$$

and the general form for the material matrix $\mathbf{C}$ for a plane strain element is:

$$\mathbf{C} = \beta \begin{bmatrix} 1 & \alpha & 0 \\ \alpha & 1 & 0 \\ 0 & 0 & \xi \end{bmatrix}. \tag{23}$$

This matrix embodies an isotropic material, where the constants α, β, and ξ are a function of the material's modulus of elasticity E and Poisson ratio ν:

$$\alpha = \frac{\nu}{1-\nu}, \qquad \beta = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}, \quad \text{and} \quad \xi = \frac{1-2\nu}{2(1-\nu)}. \tag{24}$$

Substituting into Equation 5 and multiplying out we obtain a stiffness matrix for the 2-D feature data:

$$\mathbf{K} = \int_A \mathbf{B}^T \mathbf{C} \mathbf{B} dA = \left[\begin{array}{cc} \mathbf{K}_{aa} & \mathbf{K}_{ab} \\ \mathbf{K}_{ba} & \mathbf{K}_{bb} \end{array} \right] \tag{25}$$

where each m by m submatrix is positive semi-definite symmetric, and $\mathbf{K}_{ab} = \mathbf{K}_{ba}$. The elements of $\mathbf{K}_{aa}$ have the form:

$$k_{aa_{ij}} = \beta \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sum_{k,l} a_{ik}\, a_{jl} \left[\frac{\partial g_k}{\partial x} \frac{\partial g_l}{\partial x} + \xi \frac{\partial g_k}{\partial y} \frac{\partial g_l}{\partial y} \right] dx\, dy. \tag{26}$$

Integrate and regroup terms:

$$k_{aa_{ij}} = \pi \beta \sum_{k,l} a_{ik}\, a_{jl} \left[\frac{1+\xi}{2} - \frac{\left(\hat{x}_{kl}^2 + \xi \hat{y}_{kl}^2\right)}{4\sigma^2} \right] \sqrt{g_{kl}}, \tag{27}$$

where $\hat{x}_{kl} = (x_k - x_l)$ and $\hat{y}_{kl} = (y_k - y_l)$. Similarly, the elements of $\mathbf{K}_{bb}$ have the form:

$$k_{bb_{ij}} = \pi \beta \sum_{k,l} a_{ik}\, a_{jl} \left[\frac{1+\xi}{2} - \frac{\left(\hat{y}_{kl}^2 + \xi \hat{x}_{kl}^2\right)}{4\sigma^2} \right] \sqrt{g_{kl}}. \tag{28}$$

Finally, the elements of $\mathbf{K}_{ab}$ have the form:

$$k_{ab_{ij}} = \beta \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sum_{k,l} a_{ik}\, a_{jl} \left[\alpha \frac{\partial g_k}{\partial x} \frac{\partial g_l}{\partial y} + \xi \frac{\partial g_k}{\partial y} \frac{\partial g_l}{\partial x} \right] dx\, dy. \tag{29}$$

When integrated this becomes:

$$k_{ab_{ij}} = -\frac{\pi \beta (\alpha + \xi)}{4\sigma^2} \sum_{k,l} a_{ik}\, a_{jl}\, \hat{x}_{kl}\, \hat{y}_{kl} \sqrt{g_{kl}}. \tag{30}$$

V. Determining Correspondences

To determine correspondences, we first compute mass and stiffness matrices for both feature sets. These matrices are then decomposed into eigenvectors ϕ_i and eigenvalues λ_i as described in Section III.D. The resulting eigenvectors are ordered by increasing eigenvalue, and form the columns of the modal matrix $\mathbf{\Phi}$:

$$\mathbf{\Phi} = [\phi_1 \mid \ldots \mid \phi_{2m}] = \left[\begin{array}{c} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_m^T \\ \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_m^T \end{array} \right] \tag{31}$$

where m is the number of nodes used to build the finite element model. The column vector ϕ_i is called 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 vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ are called the i^{th} *generalized feature vectors*, and together describe the feature's location in the modal coordinate system.

Modal matrices $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$ are built for both images. Correspondences can now be computed by comparing mode shape vectors for the two sets of features; we will characterize each nodal point by its relative participation in several eigenmodes. Before actually describing how this matching is performed, it is important to consider which and how many of these eigenmodes should be incorporated into our feature comparisons.

A. Modal Truncation

For various reasons, we must select a subset of mode shape vectors (column vectors ϕ_i) before computing correspondences. The most obvious reason for this is that the number of eigenvectors and eigenvalues computed for the source and target images will probably not be the same. This is because the number of feature points in each image will almost always differ. To make the dimensionalities of the two generalized feature spaces the same, we will need to truncate the number of columns at a given dimensionality.

Typically, we retain only the lowest-frequency 25% of the columns of each mode matrix, in part because the higher-frequency modes are the ones most sensitive to noise. Another reason for discarding higher-frequency modes is to make our shape comparisons less sensitive to local shape variations.

We will also want to discard columns associated with the rigid-body modes. Recall that the columns of the modal matrix are ordered in terms of increasing eigenvalue. For a two-dimensional problem, the first three eigenmodes will represent the rigid body modes of two translations and a rotation. These first three columns of each modal matrix are therefore discarded to make the correspondence computation invariant to differences in rotation and translation.

In summary, this truncation breaks the generalized eigenspace into three groups of feature vectors:

$$\begin{gathered} \mathbf{\Phi}_1 = [\underbrace{\phi_{1,1} \mid \phi_{1,2} \mid \phi_{1,3}}_{\substack{\text{rigid body} \\ \text{modes}}} \mid \underbrace{\phi_{1,4} \mid \ldots \mid \phi_{1,p}}_{\substack{\text{intermediate} \\ \text{modes}}} \mid \underbrace{\phi_{1,p+1} \mid \ldots \mid \phi_{1,2m}}_{\substack{\text{high-order} \\ \text{modes}}}] \\ \mathbf{\Phi}_2 = [\overbrace{\phi_{2,1} \mid \phi_{2,2} \mid \phi_{2,3}} \mid \overbrace{\phi_{2,4} \mid \ldots \mid \phi_{2,p}} \mid \overbrace{\phi_{2,p+1} \mid \ldots \mid \phi_{2,2n}}] \end{gathered} \tag{32}$$

where m and n are the number of features in each image. We keep only those columns that represent the intermediate eigenmodes; thus, the truncated generalized feature space will be of dimension $2(p-3)$ for a 2D problem.

We now have a set of mode-truncated feature vectors:

$$\bar{\mathbf{\Phi}} = [\phi_4 \mid \ldots \mid \phi_p] = \left[\begin{array}{c} \bar{\mathbf{u}}_1^T \\ \vdots \\ \bar{\mathbf{u}}_m^T \\ \bar{\mathbf{v}}_1^T \\ \vdots \\ \bar{\mathbf{v}}_m^T \end{array} \right], \tag{33}$$

where the two row vectors $\bar{\mathbf{u}}_i$ and $\bar{\mathbf{v}}_i$ store the displacement signature for the i^{th} node point, in truncated mode space. The vector $\bar{\mathbf{u}}_i$ contains the x, and $\bar{\mathbf{v}}_i$ contains the y, displacements associated with each of the $p-3$ modes.

It is sometimes the case that a couple of eigenmodes have nearly equal eigenvalues. This is especially true for the low-order eigenmodes of symmetric shapes and shapes whose aspect ratio is nearly equal to one. In our current system, such eigenmodes are excluded from the correspondence computation because they would require the matching of eigenmode subspaces.

B. Computing Correspondence Affinities

Using a modified version of an algorithm described by Shapiro and Brady [33], we now compute what are referred to as the affinities z_{ij} between the two sets of generalized feature vectors. These are stored in an *affinity matrix* $\mathbf{Z}$, where:

$$\mathbf{z}_{ij} = \| \bar{\mathbf{u}}_{1,i} - \bar{\mathbf{u}}_{2,j} \|^2 + \| \bar{\mathbf{v}}_{1,i} - \bar{\mathbf{v}}_{2,j} \|^2. \qquad (34)$$

The affinity measure for the i^{th} and j^{th} points, z_{ij}, will be zero for a perfect match and will increase as the match worsens. Using these affinity measures, we can easily identify which features correspond to each other in the two images by looking for the minimum entry in each column or row of $\mathbf{Z}$. Shapiro and Brady noted that the symmetry of an eigendecomposition requires an intermediate sign correction step for the eigenvectors ϕ_i. This is due to the fact that the direction (sign) of eigenvectors can be assigned arbitrarily. Readers are referred to [32] for more details about this.

To obtain accurate correspondences the Shapiro and Brady method requires three simple, but important, modifications. First, only the generalized features that match with the greatest certainty are used to determine the deformation; the remainder of the correspondences are determined by the deformation itself as in our previous method. By discarding affinities greater than a certain threshold, we allow for tokens that have no strong match. Second, as described earlier, only the low-order twenty-five percent of the eigenvectors are employed, as the higher-order modes are known to be noise-sensitive and thus unstable [2]. Lastly, because of the reduced basis matching, similarity of the generalized features is required in both directions, instead of one direction only. In other words, a match between the i^{th} feature in the first image and the j^{th} feature in the second image can only be valid if z_{ij} is the minimum value for its row, and z_{ji} the minimum for its column. Image points for which there was no correspondence found are flagged accordingly.

In cases with low sampling densities or with large deformations, the mode ordering can vary slightly. Such cases require an extra step in which neighborhoods of similarly-valued modes are compared to find the best match.

C. Coping with Large Rotations

As described so far, our affinity matrix computation method works best when there is little difference in the orientation between images. This is due to the fact that the modal displacements are described as vectors (u, v) in image space. When the aligning rotation for two sets of features is potentially large, the affinity calculation can be made rotation invariant by transforming the mode shape vectors into a polar coordinate system. In two dimensions, each mode shape vector takes the form

$$\phi_i = [u_1 \ldots u_m, v_1 \ldots v_m]^T \qquad (35)$$

where the modal displacement at the i^{th} node is simply (u_i, v_i). To obtain rotation invariance, we must transform each (u, v) component into a coordinate in (r, θ) space as shown in Figure 4. The angle θ is computed relative to the vector from the object's centroid to the nodal point $\mathbf{x}$. The radius r is simply the magnitude of the displacement vector $\mathbf{u}$.

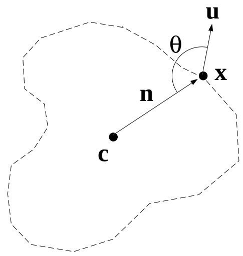


Fig. 4. Transforming a modal displacement vector $\mathbf{u} = (u, v)$ into (θ, r). The angle θ is computed relative to the vector $\mathbf{n}$ from the object's centroid $\mathbf{c}$ to the nodal point $\mathbf{x}$. The radius r is simply the length of $\mathbf{u}$.

Once each mode shape vector has been transformed into this polar coordinate system, we can compute feature affinities as was described in the previous section. In our experiments, however, we have found that it is often more effective to compute affinities using either just the r components or just the θ components, *i.e.*:

$$\mathbf{z}_{ij} = \| \bar{\theta}_{1,i} - \bar{\theta}_{2,j} \|^2. \qquad (36)$$

In general, the r components are scaled uniformly based on the ratio between the object's overall scale versus the Gaussian basis function radius σ. The θ components, on the other hand, are immune to differences in scale, and therefore a distance metric based on θ offers the advantage of scale invariance.

D. Multiresolution Models

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. The procedure for this is as follows.

First, a subset of m feature points is selected to be finite element nodes. 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. As before, a FEM model is built for each shape, eigenmodes are obtained, and modal truncation is performed as described in Section V.A. The resulting eigenmodes are then matched and sign-corrected using the lower-resolution models' affinity matrix.

With modes matched for the feature subsets, we now proceed to finding the correspondences for the full sets of features. To do this, we utilize interpolated modal matrices which describe each mode's shape for the full set of features:

$$\hat{\mathbf{\Phi}} = \hat{\mathbf{H}}\mathbf{\Phi}. \qquad (37)$$

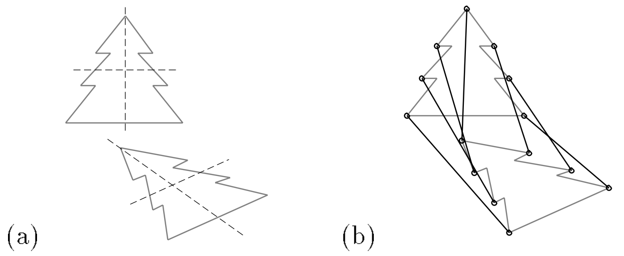


Fig. 5. Two flat tree shapes, one upright and one lying flat (a), together with the obtained correspondence (b). The 18 low-order modes were computed for each tree and then correspondences were determined using the algorithm described in the text.

The interpolation matrix $\hat{\mathbf{H}}$ relates the displacement at the nodes (low-resolution features) to displacements at the higher-resolution feature locations $\mathbf{x}_i$:

$$\hat{\mathbf{H}} = \begin{bmatrix} \mathbf{H}(\mathbf{x}_1) \\ \vdots \\ \mathbf{H}(\mathbf{x}_n) \end{bmatrix}, \tag{38}$$

where each submatrix $\mathbf{H}(\mathbf{x}_i)$ is a $2 \times 2m$ interpolation matrix as in Eq. 17.

Finally, an affinity matrix for the full feature set is computed using the interpolated modal matrices, and correspondences are determined as described in the previous sections.

VI. Correspondence Experiments

In this section we will first illustrate the method on a few classic problems, and then demonstrate its performance on real imagery. In each example the feature points are treated independently; no connectivity or distinctiveness information was employed. Thus the input to the algorithm is a cloud of feature points, not a contour or 2-D form. The mass and stiffness matrices were then computed, and the **M**-orthonormalized eigenvectors determined. Finally, correspondences were obtained as described above.

The left-hand side of Figure 5(a) shows two views of a flat, tree-like shape, an example illustrating the idea of skewed symmetry adapted from [13]. The first 18 modes were computed for both trees, and were compared to obtain the correspondences shown in Figure 5(b). The fact that the two figures have similar low-order symmetries (eigenvectors) allows us to recognize that two shapes are closely related, and to easily establish the point correspondences.

Figure 6 shows another classic example [25]. Here we have pear shapes with various sorts of bumps and spikes. Roughly 300 points were sampled regularly along the contour of each pear's silhouette. Correspondences were then computed using the first 32 modes. Because of the large number of data points, only two percent of the correspondences are shown. As can be seen from the figure, reasonable correspondences were found.

Figure 7(a) illustrates a more complex correspondence example, using real image data. Despite the differences between these two hands, the low-order descriptions are quite similar and consequently a very good correspondence

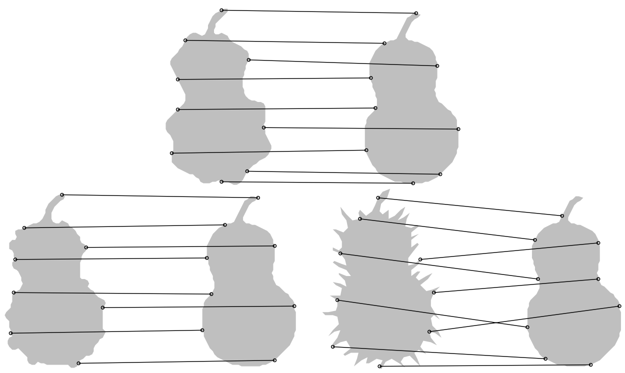


Fig. 6. Correspondence obtained for bumpy, warty, and prickly pears. Roughly 300 silhouette points were matched from each pear. Because of the large number of data points, only two percent of the correspondences are shown in this figure.

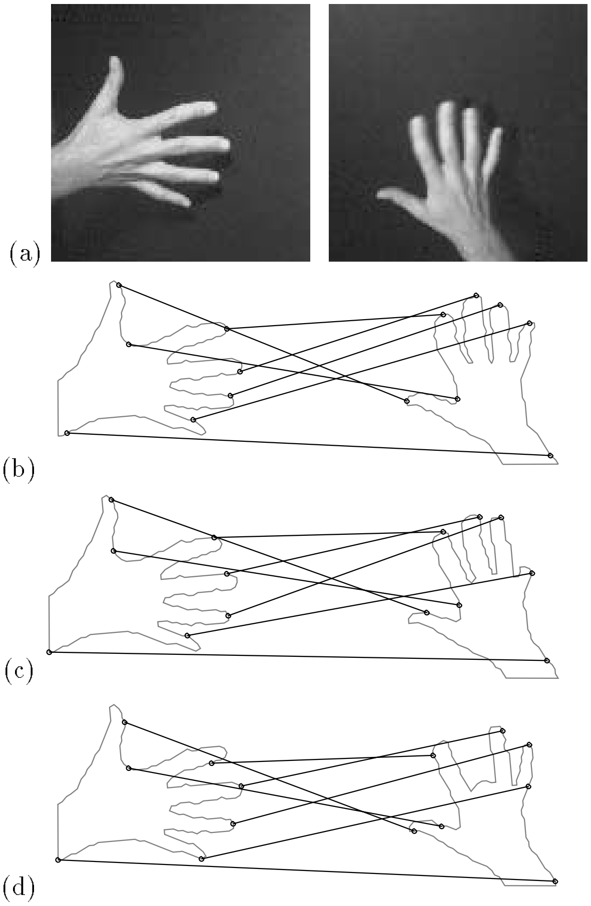


Fig. 7. (a) Two hand images, (b) correspondences between silhouette points, (c),(d) correspondences after digital surgery. Roughly 400 points were sampled from each hand silhouette. Correspondences were computed for all points using the first 32 modes. For clarity, only correspondences for key points are shown in this figure.

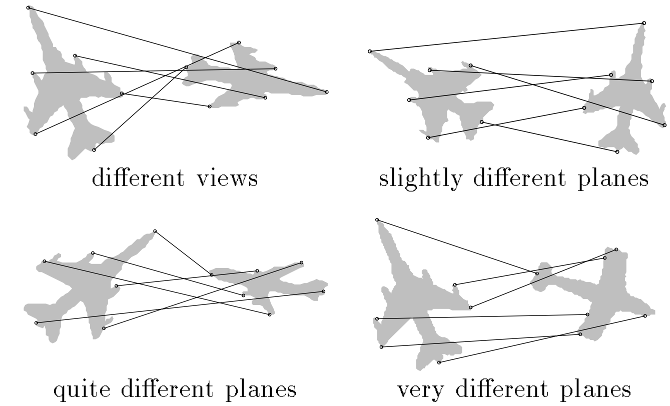


Fig. 8. Correspondence obtained for outlines of different types of airplanes. The first example shows the correspondences found for different (rotated in 3D) views of the same fighter plane. The others show matches between increasingly different airplanes. In the final case, the wing position of the two planes is quite different. As a consequence, the best-matching correspondence has the Piper Cub flipped end-to-end, so that the two planes have more similar before-wing and after-wing fuselage lengths. Despite this overall symmetry error, the remainder of the correspondence appears quite accurate. Roughly 150 silhouette points were matched from each plane. Because of the large number of data points, only critical correspondences are shown in this figure.

is obtained, as shown in Figure 7(b). Roughly 400 points were sampled from each hand silhouette. Correspondences were computed for all points using the first 32 modes. As in the previous example, only two percent of the correspondences are shown.

Figures 7(c) and (d) show the same hand data after digital surgery. In Figure 7(c), the little finger was almost completely removed; despite this, a nearly perfect correspondence was maintained. In Figure 7(d), the second finger was removed. In this case a good correspondence was still obtained, but not the most natural given our knowledge of human bone structure.

The next example, Figure 8, uses outlines of three different types of airplanes as seen from a variety of different viewpoints (adapted from [45]). In the first three cases the descriptions generated are quite similar, and as a consequence a very good correspondence is obtained. Again, only two percent of the correspondences are shown.

In the last pair, the wing position of the two planes is quite different. As a result, the best-matching correspondence has the Piper Cub flipped end-to-end, so that the two planes have more similar before-wing and after-wing fuselage lengths. Despite this overall symmetry error, the remainder of the correspondence appears quite accurate.

Our final example is adapted from [40] and utilizes multi-resolution modal matching to efficiently find correspondences for a large number of feature points. Figure 9 shows the edges extracted from images of two different cars taken from varying viewpoints. Figure 9(a) depicts a view of a Volkswagen Beetle (rotated 15^o from side view) and Figure 9(b) depicts two different views of a Saab (rotated 15^o and 45^o). If we take each edge pixel to be a feature, then each car has well over 1000 feature points.

As described in Section V.D, when there are a large number of feature points, modal models are first built from a roughly uniform sub-sampling of the features. Figures 9(c) and 9(d) show the subsets of between 30 and 40 features that were used in building the finite element models. Both silhouette and interior points were used in building the model.

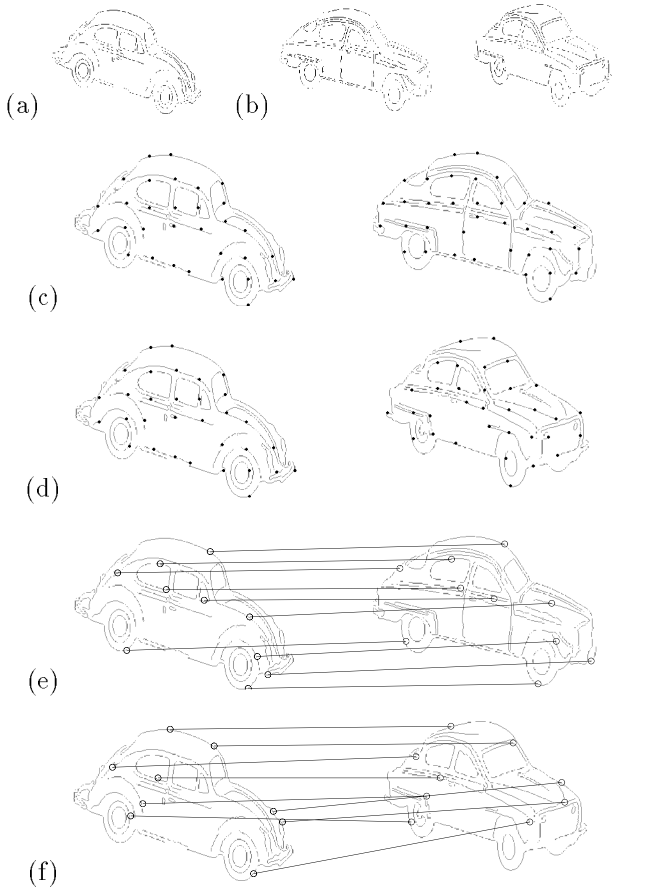


Fig. 9. Finding correspondence for one view of a Volkswagen (a) and a two views of a Saab (b) taken from [40]. Each car has well over 1000 edge points. Note that both silhouette and interior points can be used in building the model. As described in the text, when there are a large number of feature points, modal models are first built from a uniform sub-sampling of the features as is shown in (c,d). In this example, roughly 35 points were used in building the finite element models. Given the modes computed for this lower-resolution model, we can use modal matching to compute feature matches for the higher-resolution. Correspondences between similar viewpoints of the VW and Saab are shown in (e), while in (f) a different viewpoint is matched (the viewpoints differ by 30^o). Because of the large number of data points, only a few of the correspondences are shown in this figure.

The modes computed for the lower-resolution models were then used as input to an interpolated modal matching which paired off the corresponding higher-resolution features. Some of the strongest corresponding features for two similar views of the VW and Saab are shown in 9(e). The resulting correspondences are reasonable despite moderate differences in the overall shape of the cars. Due to the large number of feature points, only a few of the strongest correspondences are shown in this figure.

In Figure 9(f), the viewpoints differ by 30^o. Overall, the resulting correspondences are still quite reasonable, but this example begins to push the limits of the matching algorithm. There are one or two spurious matches; *e.g.*, a headlight is matched to a sidewall. We expect that performance could be improved if information about intensity, color, or feature distinctiveness were included in our model.

VII. Object Alignment, Comparison and Description

An important benefit of our technique 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 a physical one — we can compute and compare the amount of deformation energy needed to align an object, and use this as a similarity measure. If the modal displacements or strain energy required to align two feature sets is relatively small, then the objects are very similar.

Recall that for a two-dimensional problem, the first three modes are the rigid body modes of translation and rotation, and the rest are nonrigid modes. The nonrigid modes are ordered by increasing frequency of vibration; in general, low-frequency modes describe global deformations, while higher-frequency modes describe more localized shape deformations. Such a global-to-local ordering of shape deformation allows us to select which types of deformations are to be compared.

For instance, it may be desirable to make object comparisons rotation, position, and/or scale independent. To do this, we ignore displacements in the low-order or rigid body modes, thereby disregarding differences in position, orientation, and scale. In addition, we can make our comparisons robust to noise and local shape variations by discarding higher-order modes. As will be seen later, this modal selection technique is also useful for its compactness, since we can describe deviation from a prototype in terms of relatively few modes.

But before we can actually compare two sets of features, we first need to recover the modal deformations $\tilde{\mathbf{U}}$ that deform the matched points on one object to their corresponding positions on a prototype object. A number of different methods for recovering the modal deformation parameters are described in the next section.

A. Recovering Deformations

We want to describe the deformation parameters $\tilde{\mathbf{U}}$ that take the set of points from the first image to the corresponding points in the second. Given that $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$ have been computed, and that correspondences have been established, then we can solve for the modal displacements directly. This is done by noting that the nodal displacements $\mathbf{U}$ that align corresponding features on both shapes can be written:

$$\mathbf{u}_i = \mathbf{x}_{1,i} - \mathbf{x}_{2,i}, \qquad (39)$$

where $\mathbf{x}_{1,i}$ is the i^{th} node on the first shape and $\mathbf{x}_{2,i}$ is its matching node on the second shape.

Recalling that $\mathbf{U} = \mathbf{\Phi}\tilde{\mathbf{U}}$, and using the identity of Equation 10, we find:

$$\tilde{\mathbf{U}} = \mathbf{\Phi}^{-1}\mathbf{U} = \mathbf{\Phi}^T\mathbf{M}\mathbf{U}. \qquad (40)$$

Normally there is not one-to-one correspondence between the features. In the more typical case where the recovery is underconstrained, we would like unmatched nodes to move in a manner consistent with the material properties and the forces at the matched nodes. This type of solution can be obtained in a number of ways.

In the first approach, we are given the nodal displacements $\mathbf{u}_i$ at the matched nodes, and we set the loads $\mathbf{r}_i$ at unmatched nodes to zero. We can then solve the equilibrium equation, $\mathbf{K}\mathbf{U} = \mathbf{R}$, where we have as many knowns as unknowns. Modal displacements are then obtained via Eq. 40. This approach yields a closed-form solution, but we have assumed that forces at the unmatched nodes are zero.

By adding a strain-energy minimization constraint, we can avoid this assumption. The strain energy can be measured directly in terms of modal displacements, and enforces a penalty that is proportional to the squared vibration frequency associated with each mode:

$$E_I = \frac{1}{2}\tilde{\mathbf{U}}^T\mathbf{\Omega}^2\tilde{\mathbf{U}}. \qquad (41)$$

Since rigid body modes ideally introduce no strain, it is logical that their $\omega_i \approx 0$.

We can now formulate a constrained least squares solution, where we minimize alignment error that includes this modal strain energy term:

$$E = \underbrace{\left[\mathbf{U} - \mathbf{\Phi}\tilde{\mathbf{U}}\right]^T\left[\mathbf{U} - \mathbf{\Phi}\tilde{\mathbf{U}}\right]}_{\text{squared fitting error}} + \underbrace{\lambda\tilde{\mathbf{U}}^T\mathbf{\Omega}^2\tilde{\mathbf{U}}.}_{\text{strain energy}} \qquad (42)$$

This strain term directly parallels the smoothness functional employed in regularization [35].

Differentiating with respect to the modal parameter vector yields the strain-minimizing least squares equation:

$$\tilde{\mathbf{U}} = \left[\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{\Omega}_2\right]^{-1}\mathbf{\Phi}^T\mathbf{U}. \qquad (43)$$

Thus we can exploit the underlying physical model to enforce certain geometric constraints in a least squares solution. The strain energy measure allows us to incorporate some prior knowledge about how stretchy the shape is, how much it resists compression, *etc.* Using this extra knowledge, we can infer what "reasonable" displacements would be at unmatched feature points.

Since the modal matching algorithm computes the strength for each matched feature, we would also like to utilize these match-strengths directly in alignment. This is achieved by including a diagonal weighting matrix:

$$\tilde{\mathbf{U}} = \left[\mathbf{\Phi}^T\mathbf{W}^2\mathbf{\Phi} + \lambda\mathbf{\Omega}^2\right]^{-1}\mathbf{\Phi}^T\mathbf{W}^2\mathbf{U} \qquad (44)$$

The diagonal entries of $\mathbf{W}$ are inversely proportional to the affinity measure for each feature match. The entries for unmatched features are set to zero.

B. Dynamic Solution: Morphing

So far, we have described methods for finding the modal displacements that directly deform and align two feature sets. It is also possible to solve the alignment problem by physical simulation, in which the finite element equations are integrated over time until equilibrium is achieved. In this case, we solve for the deformations at each time step via the *dynamic equation* (Eq. 12). In so doing, we compute the intermediate deformations in a manner consistent with the material properties that were built into the finite element model. The intermediate deformations can also be used for physically-based morphing.

When solving the dynamic equation, we use features of one image to exert forces that pull on the features of the other image. The dynamic loads $\mathbf{R}(t)$ at the finite element nodes are therefore proportional to the distance between matched features:

$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + k(\mathbf{x}_{1,i} + \mathbf{u}_i(t) - \mathbf{x}_{2,i}), \qquad (45)$$

where k is an overall stiffness constant and $\mathbf{u}_i(t)$ is the nodal displacement at the previous time step. These loads simulate "ratchet springs," which are successively tightened until the surface matches the data [10].

The modal dynamic equilibrium equation can be written as a system of $2m$ independent equations of the form:

$$\ddot{\tilde{u}}_i(t) + \tilde{d}_i \dot{\tilde{u}}_i(t) + \omega_i^2 \tilde{u}_i(t) = \tilde{r}_i(t), \qquad (46)$$

where the $\tilde{r}_i(t)$ are components of the transformed load vector $\tilde{\mathbf{R}}(t) = \mathbf{\Phi}^T \mathbf{R}(t)$. These independent equilibrium equations can be solved via an iterative numerical integration procedure (*e.g.,* Newmark method [2]). The system is integrated forward in time until the change in load energy goes below a threshold. The loads $\mathbf{r}_i(t)$ are updated at each time step by evaluating Equation 45.

C. Coping with Large Rotations

If the rotation needed to align the two sets of points is potentially large, then it is necessary to perform an initial alignment step before recovering the modal deformations. Orientation, position, and (if desired) scale can be recovered in closed-form via quaternion-based algorithms described by Horn [12] or by Wang and Jepson [43].[1]

Using only a few of the strongest feature correspondences (recall that strong matches have relatively small values in the affinity matrix $\mathbf{Z}$) the rigid body modes can be solved for directly. The resulting additional alignment parameters are:

$\mathbf{p}_0$	position vector
$\mathbf{q}$	unit quaternion defining orientation
s	scale factor
$\mathbf{c}_1$ and $\mathbf{c}_2$	centroids for the two objects

Since this initial orientation calculation is based on only the strongest matches, these are usually a very good estimate of the rigid body parameters.

The objects can now be further aligned by recovering the modal deformations $\tilde{\mathbf{U}}$ as described previously. As before, we compute virtual loads that deform the features in the first image towards their corresponding positions in the second image. Since we have introduced an additional rotation, translation, and scale, Equation 39 will be modified so as to measure distances between features in the correct coordinate frame:

$$u_i = (\frac{1}{s}\mathcal{R}^T\left[\mathbf{x}_{2,i} - \mathbf{p}_0 - \mathbf{c}_1\right] + \mathbf{c}_1 - \mathbf{x}_{1,i}), \qquad (47)$$

where $\mathcal{R}$ is a rotation matrix computed from the unit quaternion $\mathbf{q}$.

Through the initial alignment step, we have essentially reduced virtual forces between corresponding points; the spring equation accounts for this force reduction by inverse transforming the matched points $\mathbf{x}_{2,i}$ into the finite element's local coordinate frame. The modal amplitudes $\tilde{\mathbf{U}}$ are then solved for via a matrix multiply (Eq. 40) or by solving the dynamic system (Eq. 12).

D. Comparing Objects

Once the mode amplitudes have been recovered, we can compute the strain energy incurred by these deformations by plugging into Equation 41. This strain energy can then be used as a similarity metric. As will be seen in the next section, we may also want to compare the strain in a subset of modes deemed important in measuring similarity, or the strain for each mode separately. The strain associated with the i^{th} mode is simply:

$$E_{mode_i} = \frac{1}{2}\tilde{u}_i^2 \omega_i{}^2. \qquad (48)$$

Since each mode's strain energy is scaled by its frequency of vibration, there is an inherent penalty for deformations that occur in the higher-frequency modes. In our experiments, we have used strain energy for most of our object comparisons, since it has a convenient physical meaning; however, we suspect that (in general) it will 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.

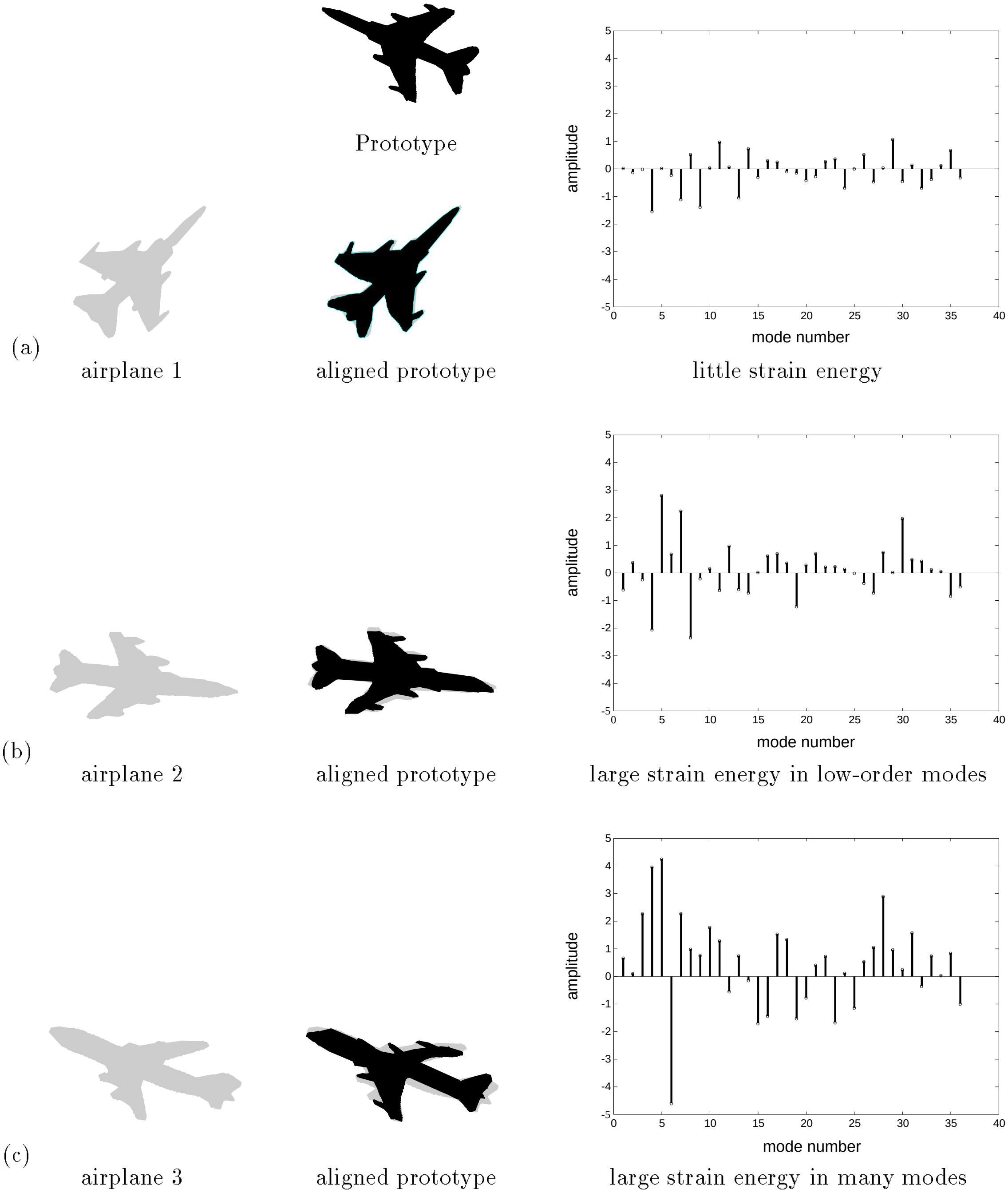
Instead of looking at the strain energy needed to align the two shapes, it may be desirable to directly compare mode amplitudes needed to align a third, prototype object with each of the two objects. In this case, we first compute two modal descriptions $\tilde{\mathbf{U}}_1$ and $\tilde{\mathbf{U}}_2$, and then utilize our favorite distance metric for measuring the distance between the two modal descriptions.

VIII. Recognition Experiments

A. Alignment and Description

Figure 10 demonstrates how we can align a prototype shape with other shapes, and how to use this computed strain energy as a similarity metric. As input, we are given the correspondences computed for the various airplane silhouettes shown in Figure 8. Our task is to align and describe the three different target airplanes (shown in

[1] While all the examples reported here are two-dimensional, it was decided that for generality, a 3-D orientation recovery method would be employed. For 2-D orientation recovery problems, simply set z coordinates to zero.



Prototype

(a) airplane 1 aligned prototype little strain energy

(b) airplane 2 aligned prototype large strain energy in low-order modes

(c) airplane 3 aligned prototype large strain energy in many modes

Fig. 10. Describing planes in terms of a prototype. The graphs show the 36 mode amplitudes used to align the prototype with each target shape. (a) shows that similar shapes can be aligned with little deformation; (b) shows that viewpoint changes produce mostly low-frequency deformations, and (c) shows that to align different shapes requires both low and high frequency deformations.

gray) in terms of modal deformations of a prototype airplane (shown in black). In each case, there were approximately 150 contour points used, and correspondences were computed using the first 36 eigenmodes. On the order of 50 strongest corresponding features were used as input to Equation 43. The modal strain energy was computed using Equation 41.

The graphs in Figure 10 show the values for the 36 recovered modal amplitudes needed to align or warp the prototype airplane with each of the target airplanes. These mode amplitudes are essentially a recipe for how to build each of the three target airplanes in terms of deformations from the prototype.

Figure 10(a) shows an airplane that is similar to the prototype and is viewed from a viewpoint that results in a similar image geometry. As a consequence the two planes can be accurately aligned with little deformation, as indicated by the graph of mode amplitudes required to warp the prototype to the target shape.

Figure 10(b) depicts an airplane that is from the same class of airplanes as the prototype, but viewed from a very different angle. In this case, the graph of mode amplitudes shows a sizable strain in the first few modes. This makes sense, since generally the first six to nine deformation modes account for affine-like deformations that are similar to the deformations produced by changes in viewpoint.

The final example, Figure 10(c), is very different from the prototype airplane, and is viewed from a different viewpoint. In this case, the recovered mode deformations are large in both the low and higher-frequency modes.

This figure illustrates how the distribution of strain energy in the various modes can be used judge the similarity of different shapes, and to determine if differences are likely due primarily to changes in viewpoint. Figure 10(a) shows that similar shapes can be aligned with little deformation; (b) shows that viewpoint changes produce mostly low-frequency deformations, and (c) shows that to align different shapes generally requires deformations of both low and high frequency.

B. Determining Relationships Between Objects

By looking more closely at the mode strains, we can pin-point which modes are predominant in describing an object. Figure 11 shows what we mean by this. As before, we can describe one object's silhouette features in terms of deformations from a prototype. In this case, we want to compare different hand tools. The prototype is a wrench, and the two target objects are a bent wrench and hammer. Silhouettes were extracted from the images, and thinned down to approximately 80 points per contour. Using the strongest matched contour points, we then recovered the first 22 modal deformations that warp the prototype onto the other tools. A rotation, translation, and scale invariant alignment stage was employed as detailed in Section V.C.

The strain energy attributed to each modal deformation is shown in the graph at the bottom of the figure. As can be seen from the graph, the energy needed to align the prototype with a similar object (the bent wrench) was mostly isolated in two modes: modes 6 and 8. In contrast, the strain energy needed to align the wrench with the hammer is much greater and spread across the graph.

Figure 12 shows the result of aligning the prototype with the two other tools using only the two most dominant modes. The top row shows alignment with the bent wrench using just the sixth mode (a shear), and then just the eighth mode (a simple bend). Taken together, these two modes do a very good job of describing the deformation needed to align the two wrenches. In contrast, aligning the wrench with the hammer (bottom row of Figure 12) cannot be described simply in terms of a few deformations of the wrench.

By observing that there is a simple physical deformation that aligns the prototype wrench and the bent wrench, we can conclude that they are probably closely related in category and functionality. In contrast, the fact that there is no simple physical relationship between the hammer and the wrench indicates that they are likely to be different types of object, and may have quite different functionality.

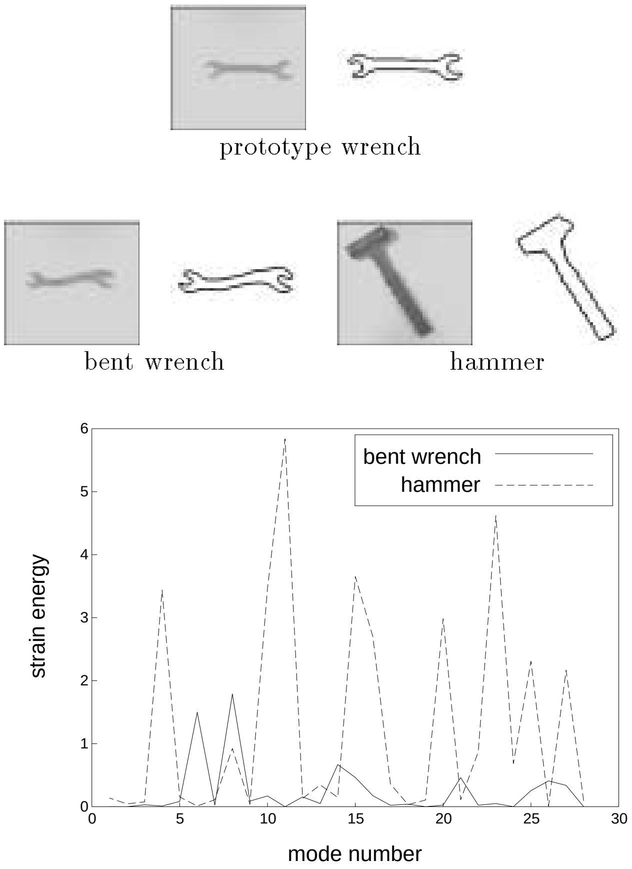


Fig. 11. Describing a bent wrench and a hammer in terms of modal deformations from a prototype wrench. Silhouettes were extracted from the images, and then the strongest corresponding contour points were found. Using these matched contour points, the first 28 modal deformations that warp the prototype's contour points onto the other tools were then recovered and the resulting strain energy computed. A graph of the *modal strain* attributed to each modal deformation is shown at the bottom of the figure.

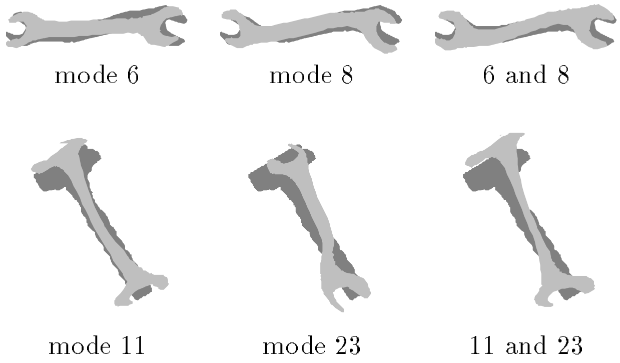


Fig. 12. Using the two modes with largest strain energy to deform the prototype wrench to two other tools. The figures demonstrates how the top two highest-strain modal deformations contribute to the alignment of a prototype wrench to the bent wrench and a hammer of Figure 11.

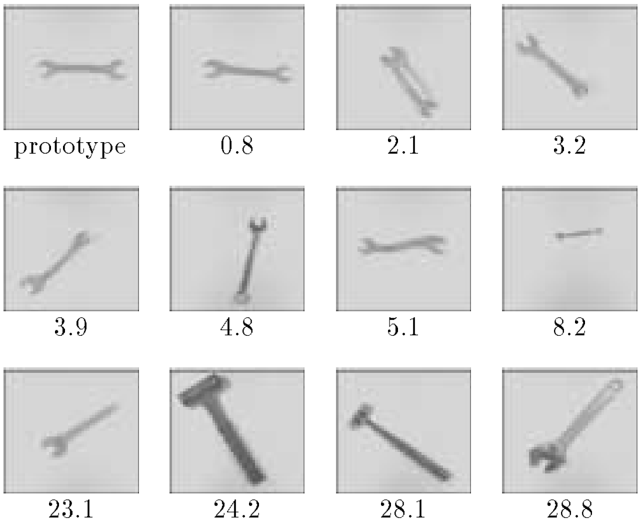


Fig. 13. Using modal strain energy to compare a prototype wrench with different hand tools. As in Figure 11, silhouettes were first extracted from each tool image, and then the strongest corresponding contour points were found. Mode amplitudes for the first 22 modes were recovered and used to warp the prototype onto the other tools. The modal strain energy that results from deforming the prototype to each tool is shown below each image in this figure. As can be seen, strain energy provides an good measure for similarity.

C. Recognition of Objects and Categories

In the next example (Figures 13 and 14) we will use modal strain energy to compare three different prototype tools: a wrench, hammer, and crescent wrench. As before, silhouettes were first extracted and thinned from each tool image, and then the strongest corresponding contour points were found. Mode amplitudes for the first 22 modes were recovered and used to warp each prototype onto the other tools. The modal strain energy that results from deforming the prototype to each tool is shown below each image. Total CPU time per trial (match, align, and compare) averaged 11 seconds on an HP 735 workstation.

Figure 13 depicts the use of modal strain energy in comparing a prototype wrench with thirteen other hand tools. As this figure shows, the shapes most similar to the wrench prototype are those other two-ended wrenches with approximately straight handles. Next most similar are closed-ended and bent wrenches, and most dissimilar are hammers and single-ended wrenches. Note that the matching is orientation and scale invariant (modulo limits imposed by pixel resolution).

Figure 14 continues this example using as prototypes the hammer and a single-ended wrench. Again, the modal strain energy that results from deforming the prototype to each tool is shown below each image.

When the hammer prototype is used, the most similar shapes found are three other images of the same hammer, taken with different viewpoints and illumination. The next most similar shapes are a variety of other hammers. The least similar shapes are a set of wrenches.

For the single-ended wrench prototype, the most similar shapes are a series of single-ended wrenches. The next most similar is a straight-handled double-ended wrench, and the least similar are a series of hammers and a bent, double-ended wrench.

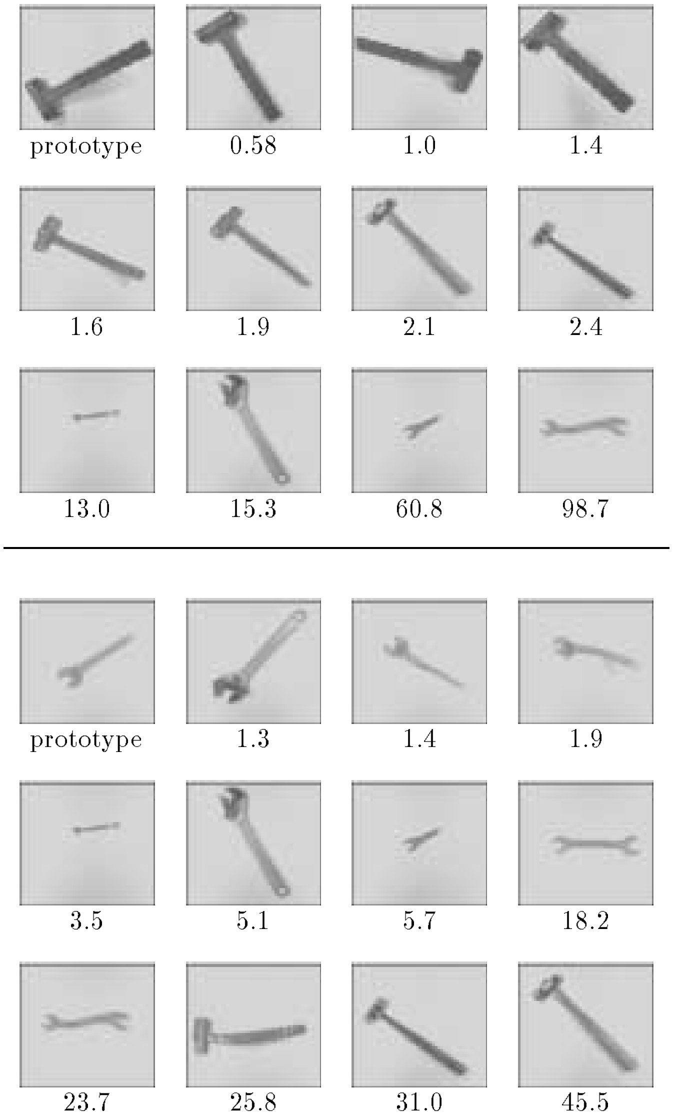


Fig. 14. Using modal strain energy to compare a crescent wrench with different hand tools, and a prototype hammer with different hand tools. Strain energies were computed as in Figure 13. The modal strain energy that results from deforming the prototype to each tool is shown below each image.

The fact that the similarity measure produced by the system corresponds to functionally-similar shapes is important. It allows us to recognize the most similar wrench or hammer from among a group of tools, even if there is no tool that is an exact match. Moreover, if for some reason the most-similar tool can't be used, we can then find the next-most-similar tool, and the next, and so on. We can find (in order of similarity) all the tools that are likely to be from the same *category*.

IX. Conclusion

The advantages afforded by our method stem from the use of the finite element technique of Galerkin surface approximation to avoid sampling problems and to incorporate outside information such as feature connectivity and distinctiveness. This formulation has allowed us to develop an information-preserving shape matrix that models the distribution of "virtual mass" within the data. This shape matrix is closely related to the proximity matrix formulation [30;32;33] and preserves its desirable properties, *e.g.*, rotation invariance. In addition, the combination of finite element techniques and a mass matrix formulation have allowed us to avoid setting initial parameters, and to handle much larger deformations.

Moreover, it is important to emphasize that the transformation to modal space not only allows for automatically establishing correspondence between clouds of feature points; the same modes (and the underlying FEM model) can then be used to describe the deformations that take the features from one position to the other. The amount of deformation required to align the two feature clouds can be used for shape comparison and description, and to warp the original images for alignment and sensor fusion. The power of this method lies primarily in it's ability to unify the correspondence and comparison tasks within one representation.

Finally, we note that the descriptions computed are canonical, and vary smoothly even for very large deformations. This allows them to be used directly for object recognition as illustrated by the airplane and hand-tool examples in the previous section. Because the deformation comparisons are physically-based, we can determine whether or not two shapes are related by a simple physical deformation. This has allowed us to identify shapes that appear to be members of the same category.

Acknowledgment

Thanks are given to Joe Born, Irfan Essa, and John Martin for their help and encouragement, and to Ronen Basri for providing the edge images of Saabs and Volkswagens.

References

[1] D. Ballard and C. Brown. *Computer Vision*, chapter 8. Prentice-Hall, Inc., 1982.

[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] T. Beier and S. Neeley. Feature based image metamorphosis. *Computer Graphics*, 26(2):35–42, July 1992.

[5] 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.

[6] F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.

[7] 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.

[8] 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.

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

[10] A. Gupta and C.-C. Liang. 3-D model-data correspondence and nonrigid deformation. In *Proc. Computer Vision and Pattern Recognition*, pages 756–757, 1993.

[11] P. W. Hallinan. A Low-Dimensional Representation of Human Faces For Arbitrary Lighting Conditions. Technical Report 93-6, Harvard Robotics Lab, Cambridge, MA, December 1993.

[12] B. Horn. Closed-form solution of absolute orientation using unit quarternions. *Journal of the Optical Society of America A*, 4:629–642, 1987.

[13] T. Kanade. Geometrical aspects of interpreting images as a three-dimensional scene. *Proc. of the IEEE*, 71(7):789–802, 1983.

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

[15] M. Kirby and L. Sirovich. Application of the Karhunen-Loeve procedure for the charachterization of human faces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.

[16] H. Murase and S. Nayar. Learning and Recognition of 3D Objects from Appearance. In *Proc. of IEEE Workshop on Qualitative Vision*, pages 39–50, New York, NY, June 1993.

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

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

[19] A. Pentland. Computational Complexity Versus Virtual Worlds. *Computer Graphics*, 24(2):185–192, 1990.

[20] A. Pentland and B. Horowitz. Recovery of non-rigid motion and structure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):730–742, July 1991.

[21] A. Pentland, B. Moghaddam, T. Starner, O. Oliyide, and M. Turk. View-based and modular eigenspaces for face recognition. In *Proc. Computer Vision and Pattern Recognition*, pages 84–91, 1994.

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

[23] 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.

[24] M. Powell. Radial basis functions for multivariate

interpolation: a review. Technical Report DAMPT 1985/NA12, Cambridge, University, 1985.

[25] W. Richards and D. Hoffman. Codon constraints on closed 2D shapes. *Computer Vision, Graphics, and Image Processing*, 31:265–281, 1985.

[26] A. Samal and P. Iyengar. Natural shape detection based on principle components analysis. *SPIE Journal of Electronic Imaging*, 2(3):253–263, July 1993.

[27] S. Sclaroff and A. Pentland. A modal framework for correspondence and recognition. In *Proc. Fourth International Conference on Computer Vision*, pages 308–313, May 1993.

[28] S. Sclaroff and A. Pentland. Modal Matching for Correspondence and Recognition. Technical Report Vision and Modeling TR-201, MIT, May 1993.

[29] S. Sclaroff and A. Pentland. Object recognition and categorization using modal matching. In *Proc. Second CAD-Based Vision Workshop*, pages 258–265, Feb 1994.

[30] G. Scott and H. Longuet-Higgins. An algorithm for associating the features of two images. In *Proc. Royal Society of London*, number 244 in B, pages 21–26, 1991.

[31] T. Sederberg and E. Greenwood. A physically-based approach to 2-D shape blending. *Computer Graphics*, 26(2):25–34, July 1992.

[32] L. Shapiro. Towards a vision-based motion framework. Technical report, Oxford University, 1991.

[33] L. Shapiro and J. Brady. Feature-based correspondence: an eigenvector approach. *Image and Vision Computing*, 10(5):283–288, June 1992.

[34] L. Staib and J. Duncan. Parametrically deformable contour models. In *Proc. Computer Vision and Pattern Recognition*, pages 98–103, 1989.

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

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

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

[38] C. Thorpe. Machine learning and human interface for the cmu navlab. In *Proc.. Computer Vision for Space Applications*, Juan-les-Pins, France, September 1993.

[39] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

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

[41] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(5):695–703, September 1988.

[42] M. P. van Oeffelen and P. G. Vos. An algorithm for pattern description on the level of relative proximity. *Pattern Recognition*, 16(3):341–348, 1983.

[43] Z. Wang and A. Jepson. A new closed-form solution of absolute orientation. In *Proc. Computer Vision and Pattern Recognition*, pages 129–134, 1994.

[44] G. Wolberg. *Digital Image Warping.* IEEE Computer Society Press, 1990.

[45] D. Wood. *Jane's World Aircraft Recognition Handbook.* Jane's Pub. Co., London, 1979.