M.I.T. Media Laboratory Perceptual Computing Section Technical Report No. 201, May 1993
Portions of this report initially appeared in the *Proceedings of ICCV93*.

Modal Matching for Correspondence and Recognition

Stan Sclaroff and Alex Pentland
Perceptual Computing Section, The Media Laboratory
Massachusetts Institute of Technology
20 Ames St., Cambridge MA 02139

Abstract

We describe a new method for establishing correspondence, computing canonical descriptions, and recognizing objects that is based on the idea of describing objects by their generalized symmetries, as defined by the object's vibration or deformation modes. The resulting *modal description* is useful for object classification, where object similarities are computed in terms of the amounts of modal deformation energy needed to align the two objects. In general, modes provide a global-to-local ordering of shape deformation which allows us to select which types of deformations are to be used in object alignment and comparison. In contrast to previous methods [14, 30, 17] we are able to compute the object's deformation modes directly from available image information, rather than requiring the computation of correspondence with an initial or prototype shape. This results in greater generality and accuracy, and is applicable to data of any dimensionality.

Keywords: Correspondence, shape description, shape invariants, object recognition, deformation, Finite element methods, vibration modes.

1 Introduction

A key problem in machine vision is how to describe features, contours, surfaces, and volumes so that they can be identified 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. In recent years this has lead the machine vision community to focus on the description of data using deformable models [14, 5, 6, 15, 9, 17, 27, 29, 30], as such models provide a useful method to interpolate and smooth raw data. However, deformable models do not by themselves provide a method of computing canonical descriptions, or of establishing correspondence between sets of data.

To address the problem of obtaining canonical descriptions, we have proposed a method in which all shapes are represented as modal deformations from some prototype object [15, 17]. By describing this deformation in terms of the eigenvectors of the prototype object's stiffness matrix, we were able 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*. Using this approach, we developed robust methods for 3-D shape modeling, object recognition, and 3-D tracking utilizing point, contour, 3-D, and optical flow data [15, 16, 17].

However our previous method was limited by the requirement that every object be described as the deformation of a *single* prototype object. This had the effect of implicitly imposing an *a priori* parameterization upon the sensor data, and therefore implicitly determining the correspondences between data and the prototype.

In this paper we will generalize our approach by obtaining the modal shape invariants directly from the sensor data. This will allow us to both solve correspondence problems and compute robust, canonical descriptions 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 contour data. The extension to data of other dimensionality is given in Appendix B. To illustrate the this use of this method for object recognition and category classification, we will present an example of recognizing and categorizing images of hand tools.

2 The Basic Idea

Imagine that we are given two sets of image feature points, and our goal is to determine if they are from two similar objects. One way to solve this problem is to try to subject the first set of points to all possible translations and rotations in an attempt to get the first set of points to closely overlay the second set. If, however, the points were given in body-centered coordinates rather than Cartesian coordinates, the problem would be nearly trivial. This suggests that the best way to match two sets of feature points is to first compute a body-centered coordinate frame. The stable computation of such a coordinate frame is the key contribution of this paper.

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.

Sampling error is the best understood of these 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 computing (for instance) a polar Fourier descriptor one 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 and use these locations 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 *modes of free vibration* of this model. These modes are the eigenvectors of the finite element model, and provide an orthogonal, frequency-ordered description of the shape and its natural deformations. Figure 2 shows the low-order modes for an upright tree-like shape. The first three 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 modes 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 deformation mode*. 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 vibration modes computed for two similar objects will be very similar — even in the presence of affine deformation, nonrigid deformation, local shape perturbation, or noise.

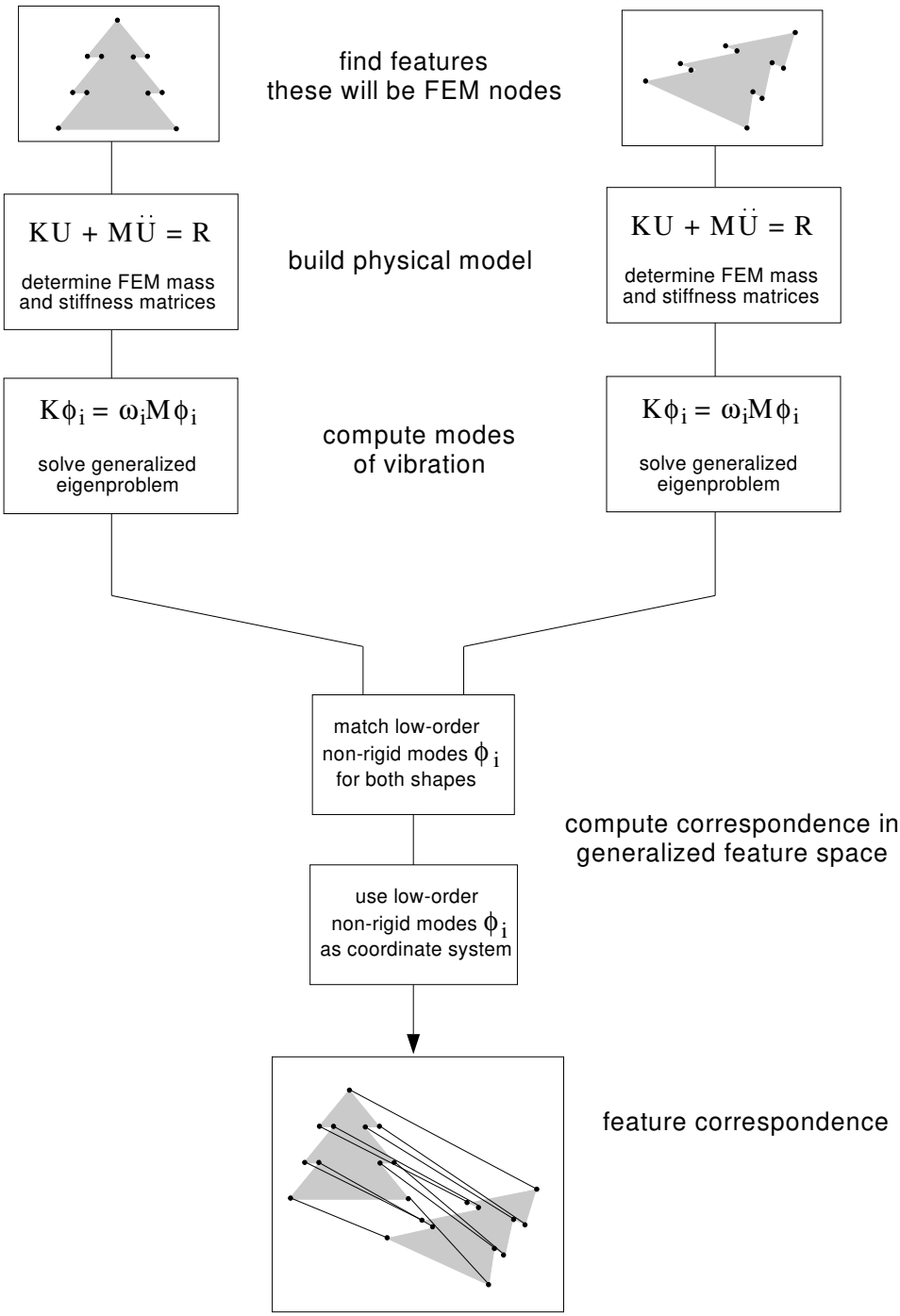


Figure 1: System diagram.

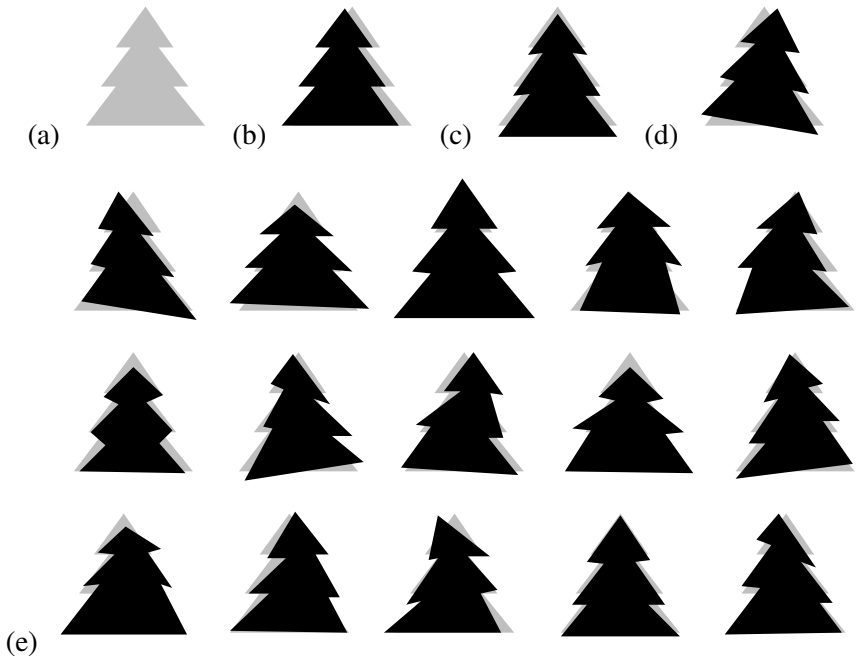


Figure 2: The low-order 18 modes computed for the upright tree shape (a). The first three modes are translation (b,c) and rotation (d). The others are nonrigid (e).

Figure 3 shows the low-order deformation modes computed for four related tree shapes. We can see how the modes of one shape correspond to the modes of another shape. This then allows us to match the feature locations in one object with those of another object despite sometimes large differences in shape.

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 determined by hand [33, 32, 23, 3].

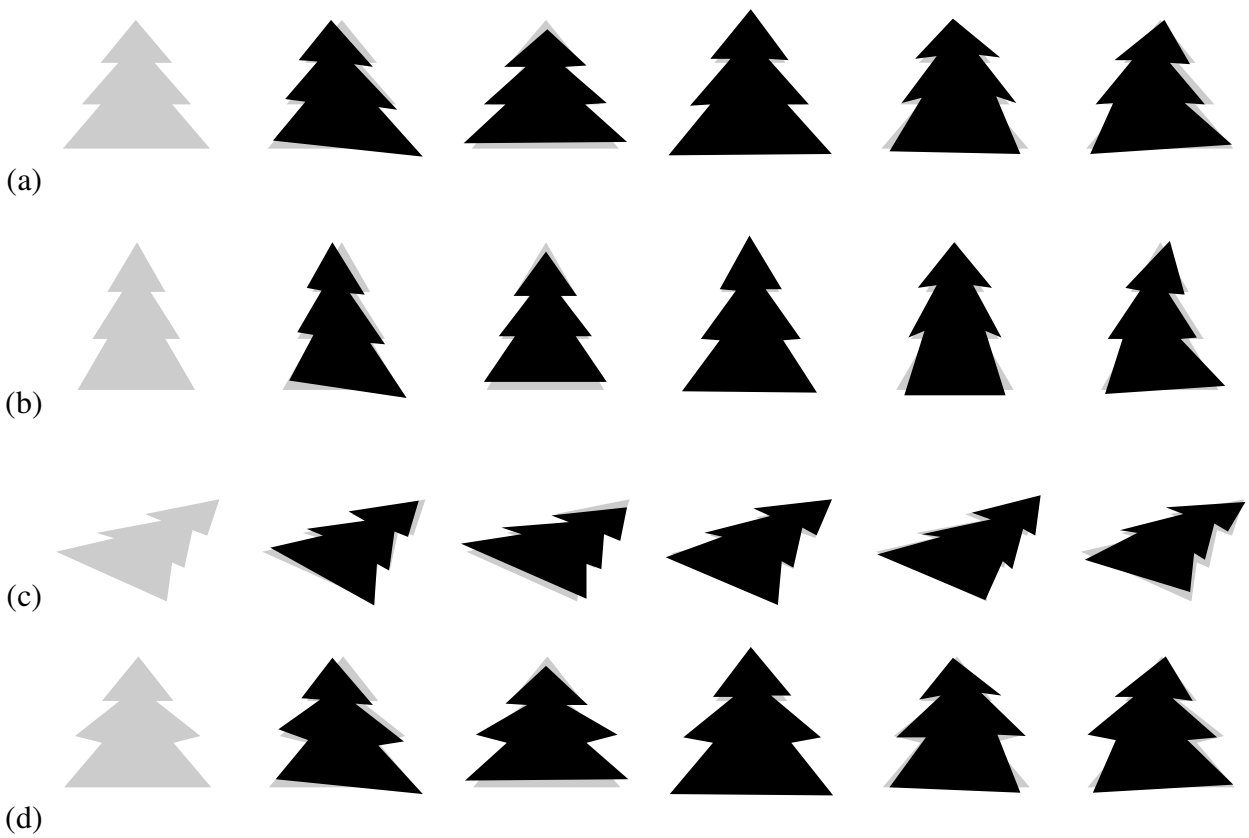


Figure 3: Similar shapes have similar low order modes. This figure shows the first five low-order vibration modes for similar tree shapes: (a) prototypical, (b) stretched, (c) tilted, and (d) two middle branches stretched.

3 Background and Notation

3.1 Correspondence and Shape Comparison

Correspondence has previously been formulated as an equilibrium problem, which has the attractive feature of allowing integration of physical constraints [30, 15, 17, 16]. 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 [15]. 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 [17] and tracking [16].

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 (nodes are typically the spring attachment points)[2]. 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.

3.2 Finite Element Method

Using Galerkin's method for finite element discretization, we can 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. This set of shape functions describes an *isoparametric finite element*. By using these functions, we can calculate the deformations which 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:

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

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.

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:

$$\boldsymbol{\epsilon}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{U} \tag{2}$$

where $\mathbf{B}$ is the strain displacement matrix. The rows of $\mathbf{B}$ are obtained by appropriately differentiating and combining rows of the element interpolation matrix $\mathbf{H}$.

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*:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{D}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R}, \tag{3}$$

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:

$$\mathbf{M} = \int_V \rho \mathbf{H}^T \mathbf{H} dV \quad \text{and} \quad \mathbf{K} = \int_V \mathbf{B}^T \mathbf{C} \mathbf{B} dV, \tag{4}$$

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

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

$$\mathbf{D} = \alpha\mathbf{M} + \beta\mathbf{K}, \tag{5}$$

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

3.3 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:

$$\mathbf{K}\boldsymbol{\phi}_i = \omega_i^2 \mathbf{M}\boldsymbol{\phi}_i. \tag{6}$$

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 6 as

$$\mathbf{K}\boldsymbol{\Phi} = \mathbf{M}\boldsymbol{\Phi}\boldsymbol{\Omega}^2 \tag{7}$$

where

$$\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \mid \boldsymbol{\phi}_2 \mid \ldots \mid \boldsymbol{\phi}_m] \quad \text{and} \quad \boldsymbol{\Omega}^2 = \begin{bmatrix} \omega_1^2 & & & \\ & \omega_2^2 & & \\ & & \ddots & \\ & & & \omega_m^2 \end{bmatrix}. \tag{8}$$

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

$$\mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi} = \mathbf{\Omega}^2 \quad \text{and} \quad \mathbf{\Phi}^T \mathbf{M} \mathbf{\Phi} = \mathbf{I}. \tag{9}$$

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

$$\mathbf{U} = \mathbf{\Phi} \tilde{\mathbf{U}} \tag{10}$$

We can now rewrite Equation 3 in terms of these generalized or modal displacements, obtaining a decoupled system of equations:

$$\ddot{\tilde{\mathbf{U}}}_t + \tilde{\mathbf{D}} \dot{\tilde{\mathbf{U}}} + \mathbf{\Omega}_t^2 \tilde{\mathbf{U}}_t = \mathbf{\Phi}_t^T \mathbf{R}, \tag{11}$$

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

By decoupling these equations, we allow for closed-form solution to the equilibrium problem [17]. 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 provide 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 [17].

4 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 which 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.

4.1 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 which model topologically complex shapes.

Our problem is different in that we want to examine the deformation modes 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 which 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 [18, 19]:

$$g_i(\mathbf{x}) = e^{-||\mathbf{x}-\mathbf{x}_i||^2/2\sigma^2} \tag{12}$$

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

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

$$h_i(\mathbf{x}) = \sum_{k=1}^{m} a_{ik} g_k(\mathbf{x}) \tag{13}$$

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

$$\mathbf{G} = \begin{bmatrix} g_1(\mathbf{x}_1) & \dots & g_1(\mathbf{x}_m) \\ \vdots & & \vdots \\ g_m(\mathbf{x}_1) & \dots & g_m(\mathbf{x}_m) \end{bmatrix}. \tag{14}$$

By using these Gaussian interpolants as our trial 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 [18].

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 [30]. If a conforming element is desired, this can be obtained by including a normalization term in h_i in Equation 13,

$$h_i(\mathbf{x}) = \frac{\sum_{k=1}^{m} a_{ik} g_k(\mathbf{x})}{\sum_{j=1}^{m} \sum_{k=1}^{m} a_{jk} g_k(\mathbf{x})} \quad . \tag{15}$$

In this paper we will use the simpler, non-conforming interpolants, primarily for the sake of simplicity. The differences between conforming and nonconforming interpolants do not affect the results reported in this paper.

4.2 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 interpolation functions developed above:

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} h_1 & \dots & h_m & 0 & \dots & 0 \\ 0 & \dots & 0 & h_1 & \dots & h_m \end{bmatrix}. \tag{16}$$

Substituting into Equation 4 and multiplying out we obtain a mass matrix for the feature data:

$$\mathbf{M} = \int_A \rho \mathbf{H}^T \mathbf{H} dA = \begin{bmatrix} \mathbf{M}_{aa} & 0 \\ 0 & \mathbf{M}_{bb} \end{bmatrix}, \tag{17}$$

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

$$m_{aa_{ij}} = \rho \int_A \sum_{k,l} a_{ik}\, a_{jl}\, g_k(\mathbf{x})\, g_l(\mathbf{x}) dA \tag{18}$$

By moving the integrals inside the sum we get

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

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 14.

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 14.

4.3 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:

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

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 4 and multiplying out we obtain a stiffness matrix for the 2-D feature data:

$$\mathbf{K} = \int_A \mathbf{B}^T \mathbf{C} \mathbf{B} dA = \begin{bmatrix} \mathbf{K}_{aa} & \mathbf{K}_{ab} \\ \mathbf{K}_{ba} & \mathbf{K}_{bb} \end{bmatrix} \tag{25}$$

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:

$$k_{aa_{ij}} = \beta \int_A \sum_{k,l} a_{ik}\, a_{jl} \left[\frac{\partial g_k}{\partial x} \frac{\partial g_l}{\partial x} + \xi \frac{\partial g_k}{\partial y} \frac{\partial g_l}{\partial y} \right] dA. \tag{26}$$

By moving the integrals inside the sum:

$$k_{aa_{ij}} = \beta \sum_{k,l} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a_{ik}\, a_{jl} \left[\frac{\partial g_k}{\partial x} \frac{\partial g_l}{\partial x} + \xi \frac{\partial g_k}{\partial y} \frac{\partial g_l}{\partial y} \right] dx\, dy \tag{27}$$

Integrate and regroup terms:

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

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:

$$k_{bb_{ij}} = \pi \beta \sum_{k,l} a_{ik}\, a_{jl} \left[\frac{1+\xi}{2} - \frac{(\hat{y}_{kl}^2 + \xi \hat{x}_{kl}^2)}{4\sigma^2} \right] \sqrt{g_{kl}}. \tag{29}$$

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

$$k_{ab_{ij}} = \beta \int_A \sum_{k,l} a_{ik}\, a_{jl} \left[\alpha \frac{\partial g_k}{\partial x}\, \frac{\partial g_l}{\partial y} + \xi \frac{\partial g_k}{\partial y}\, \frac{\partial g_l}{\partial x} \right] dA. \tag{30}$$

When integrated this becomes:

$$k_{ab_{ij}} = \beta(\alpha + \xi) \sum_{k,l} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a_{ik}\, a_{jl} \left[\frac{\partial g_k}{\partial x}\, \frac{\partial g_l}{\partial y} \right] dx\, dy \tag{31}$$

$$k_{ab_{ij}} = -\frac{\pi \beta(\alpha + \xi)}{4\sigma^2} \sum_{k,l} a_{ik}\, a_{jl}\, \hat{x}_{kl}\, \hat{y}_{kl} \sqrt{g_{kl}}. \tag{32}$$

5 Determining Correspondences

To determine correspondences, we first compute mass and stiffness matrices for both sets of features in images $\mathbf{I}_1$ and $\mathbf{I}_2$. These matrices are then decomposed into eigenvectors ϕ_i and eigenvalues λ_i as described in Section 3.3. The resulting eigenvectors are ordered by increasing eigenvalue, and form the columns of the modal matrix $\mathbf{\Phi}$:

$$\mathbf{\Phi} = [\phi_1 \mid \ldots \mid \phi_{2m}] = \begin{bmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_m^T \\ \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_m^T \end{bmatrix} \tag{33}$$

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 $\mathbf{I}_1$ and $\mathbf{I}_2$. 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 vibration modes. Before actually describing how this matching is performed, it is important to consider which and how many of these vibration modes should be incorporated into our feature comparisons.

5.1 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 modes will have near-zero eigenvalues, and will represent 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:

$$\begin{aligned} \mathbf{\Phi}_1 &= [\underbrace{\phi_{1,1} \mid \phi_{1,2} \mid \phi_{1,3}}_{\text{rigid body modes}} \mid \underbrace{\phi_{1,4} \mid \ldots \mid \phi_{1,p}}_{\text{intermediate modes}} \mid \underbrace{\phi_{1,p+1} \mid \ldots \mid \phi_{1,2m}}_{\text{high-order modes}}] \\ \mathbf{\Phi}_2 &= [\overbrace{\phi_{2,1} \mid \phi_{2,2} \mid \phi_{2,3}}^{\text{rigid body modes}} \mid \overbrace{\phi_{2,4} \mid \ldots \mid \phi_{2,p}}^{\text{intermediate modes}} \mid \overbrace{\phi_{2,p+1} \mid \ldots \mid \phi_{2,2n}}^{\text{high-order modes}}] \end{aligned} \tag{34}$$

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

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

$$\bar{\mathbf{\Phi}}_1 = [\phi_{1,4} \mid \ldots \mid \phi_{1,p}] = \begin{bmatrix} \bar{\mathbf{u}}_{1,1}^T \\ \vdots \\ \bar{\mathbf{u}}_{1,m}^T \\ \bar{\mathbf{v}}_{1,1}^T \\ \vdots \\ \bar{\mathbf{v}}_{1,2}^T \end{bmatrix} \quad \text{and} \quad \bar{\mathbf{\Phi}}_2 = [\phi_{2,4} \mid \ldots \mid \phi_{2,p}] = \begin{bmatrix} \bar{\mathbf{u}}_{2,1}^T \\ \vdots \\ \bar{\mathbf{u}}_{2,2n}^T \\ \bar{\mathbf{v}}_{2,1}^T \\ \vdots \\ \bar{\mathbf{v}}_{2,2n}^T \end{bmatrix}, \tag{35}$$

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.

5.2 Computing Correspondence Affinities

Using a modified version of an algorithm described by Shapiro and Brady [26], 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:

$$\mathbf{z}_{ij} = \| \bar{\mathbf{u}}_{1,i} - \bar{\mathbf{u}}_{2,j} \|^2 + \| \bar{\mathbf{v}}_{1,i} - \bar{\mathbf{v}}_{2,j} \|^2. \tag{36}$$

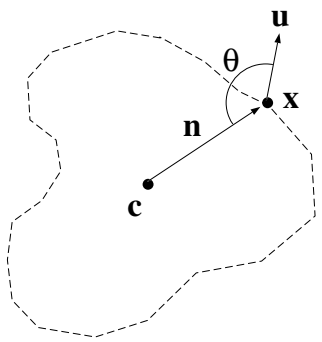


Figure 4: Transforming a modal displacement vector $\mathbf{u} = (u, v)$ into (θ, r). The angle θ is computed relative to the vector $\mathbf{n}$ from the object's centroid $\mathbf{c}$ to the nodal point $\mathbf{x}$. The radius r is simply the length of $\mathbf{u}$.

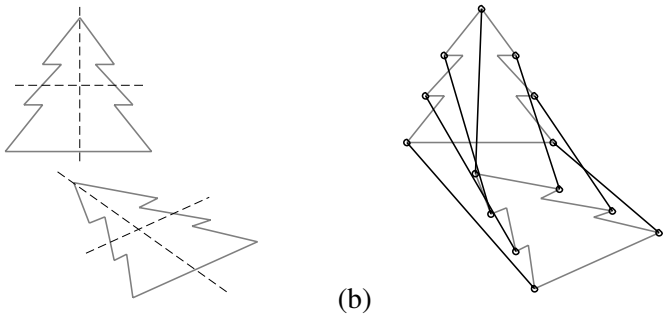
The affinity measure for the i^{th} and j^{th} points, z_{ij}, will be zero for a perfect match and will increase as the match worsens. Using these affinity measures, we can easily identify which features correspond to each other in the two images by looking for the minimum entry in each column or row of $\mathbf{Z}$.[1] The corresponding points in both $\mathbf{I}_1$ and $\mathbf{I}_2$ are represented here as $\mathbf{x}_1$ and $\mathbf{x}_2$, where $\mathbf{x}_1, i$ corresponds to $\mathbf{x}_2, i$. Image points for which there was no correspondence found are flagged accordingly.

To obtain accurate correspondences the Shapiro and Brady method requires three simple, but important, modifications. First, only the generalized features which 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 which 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]. And finally, 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 $\mathbf{I}_1$ and the j^{th} feature in $\mathbf{I}_2$ can only be valid if z_{ij} is the minimum value for its row, and z_{ji} the minimum for its column.

5.3 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

[1] Shapiro and Brady noted that the symmetry of the modal decomposition 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 [25] for more details about this.



(a) (b)

Figure 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.

into a polar coordinate system. In two dimensions, each mode shape vector takes the form

$$\phi_i = \begin{bmatrix} u_1 \\ \vdots \\ u_m \\ v_1 \\ \vdots \\ v_m \end{bmatrix} \tag{37}$$

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 $\mathbf{u}$.

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 preliminary 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.*:

$$\mathbf{z}_{ij} = \| \bar{\theta}_{1,i} - \bar{\theta}_{2,j} \|^2. \tag{38}$$

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.

6 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,

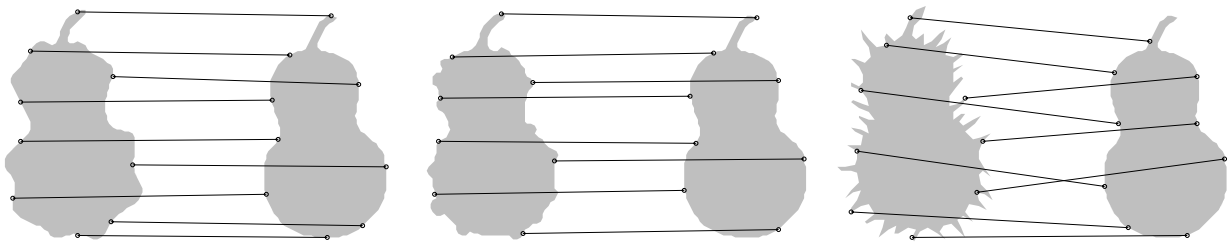


Figure 6: Correspondence obtained for bumpy, warty, and prickly pears. Roughly 300 silhouette points were matched from each pear. Because of the large number of data points, only two percent of the correspondences are shown in this figure.

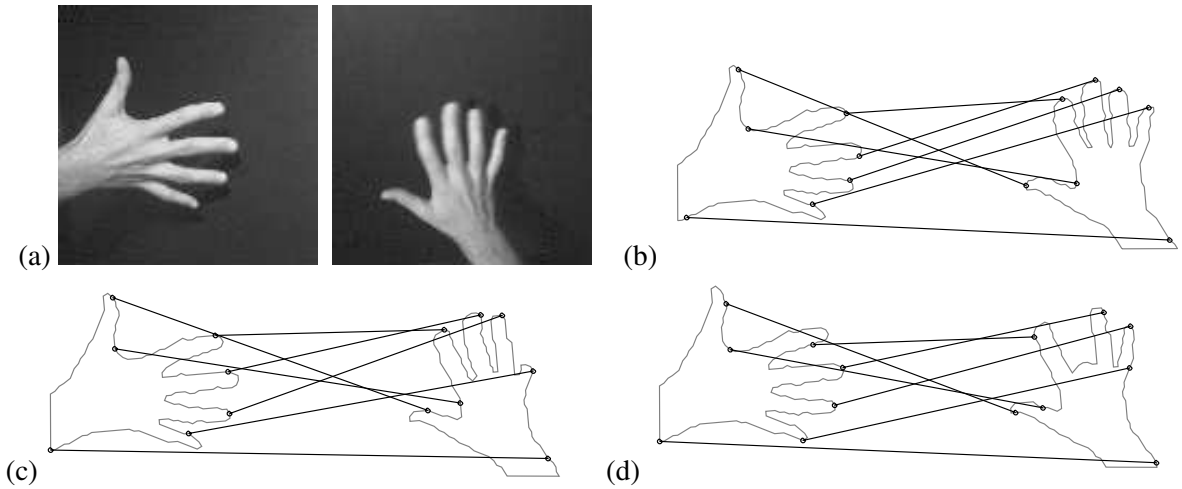


Figure 7: (a) Two hand images, (b) correspondences between silhouette points, (c),(d) correspondences after digital surgery. Roughly 400 points were sampled from each hand silhouette. Correspondences were computed for all points using the first 32 modes. For clarity, only correspondences for key points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

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 [12]. The first 22 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 [20]. 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.

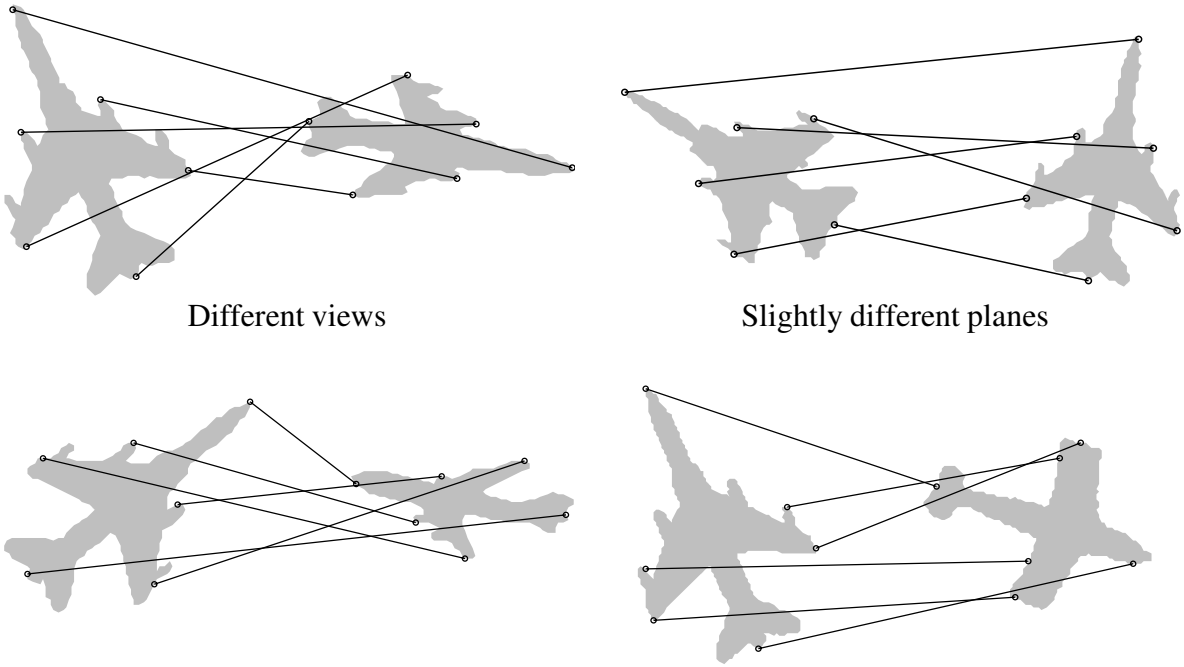


Figure 8: Correspondence obtained for outlines of different types of airplanes. The first example shows the correspondences found for different (rotated in 3D) views of the same fighter plane. The others show matches between increasingly different airplanes. In the final case, the wing position of the 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.

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 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.

In our final example, Figure 8, we use outlines of three different types of airplanes as seen from a variety of different viewpoints (adapted from [34]). 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.

7 Object Alignment, Comparison and Description

An important benefit of our technique is that the modal vibrations 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 will be 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 $\tilde{\mathbf{U}}$ which 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.

7.1 Recovering Deformations

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

$$\mathbf{u}_i = \mathbf{x}_{1,i} - \mathbf{x}_{2,i}. \tag{39}$$

Recalling that $\mathbf{U} = \mathbf{\Phi}\tilde{\mathbf{U}}$, and using the identity of Equation 9, we find:

$$\tilde{\mathbf{U}} = \mathbf{\Phi}^T\mathbf{M}\mathbf{U}. \tag{40}$$

Normally we do not have one-to-one correspondences between the points in $\mathbf{I}_1$ and $\mathbf{I}_2$. In cases where the recovery is overconstrained, we can solve via least squares. In cases where it is underconstrained, we could set the nodal displacements $\mathbf{u}_i$ to be zero for all unmatched nodes, and then solve Equation 40. This yields the $\tilde{\mathbf{U}}$ which displaces matched nodes while leaving unmatched nodes fixed.

Instead of leaving unmatched nodes fixed, it is more desirable to let them move in a manner consistent with the material properties and the loads 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 for the unknown displacements by plugging into the equilibrium equation:

$$\mathbf{K}\mathbf{U} = \mathbf{R}, \tag{41}$$

where we have as many knowns as unknowns. This approach has the advantage that the displacements are obtained in closed-form, but it has the disadvantage that the loads at unmatched nodes are assumed to be zero.

By imposing an additional constraint, it is possible to find a solution for the displacements in which we allow the loads at unmatched nodes to be nonzero. We find loads at the unmatched nodes which minimize the strain energy:

$$E_I = \frac{1}{2}\mathbf{U}^T\mathbf{K}\mathbf{U} = \frac{1}{2}\mathbf{U}^T\mathbf{R}. \tag{42}$$

The unknown nodal displacements and loads can then be solved for via gradient descent. The resulting nodal displacements are then transformed into modal displacements via the modal transform $\mathbf{\Phi}$.

Another approach solves for the modal displacements directly, by using the equation:

$$\mathbf{U} = \mathbf{\Phi}\tilde{\mathbf{U}}. \tag{43}$$

Assume that we have found correspondences for p of the m nodes. We reduce the degrees of freedom by discarding $m - p$ of the high-frequency eigenvectors (columns ϕ_i); this yields a mode-truncated transformation matrix $\bar{\mathbf{\Phi}}$. We then discard the rows of $\bar{\mathbf{\Phi}}$ which are associated with unmatched nodes. The resulting system of equations has $2p$ equations and $2p$ unknowns, so we can solve for the modal displacements directly. We again obtain a solution in closed-form, but we have assumed that the modal displacements $\tilde{\mathbf{u}}_i = 0$, for $i > p$. As before, by requiring that strain-energy be minimized, we can use an iterative technique for obtaining non-zero $\tilde{\mathbf{u}}_i$, for $i > p$.

When solving in modal space, the minimum strain energy can be measured directly in terms of modal displacements, *i.e.:*

$$E_I = \frac{1}{2}\tilde{\mathbf{U}}^T \mathbf{\Omega}^2 \tilde{\mathbf{U}}. \tag{44}$$

This strain energy equation enforces a penalty which is proportional to the squared vibration frequency associated with each mode. Since rigid body modes ideally introduce no strain, it is logical that their $\omega_i \approx 0$.

7.1.1 Dynamic Solution: Morphing

So far, we have described methods for finding the modal displacements which directly deform and align two feature sets. In some cases, we may want to solve for the intermediate deformations, a process called morphing in the computer graphics literature. One way to achieve this is by interpolating in-between frames from the initial feature positions in $\mathbf{I}_1$ and the displaced feature positions in the target image $\mathbf{I}_2$, as described by the modal deformations $\tilde{\mathbf{U}}$. This method is simple, but has little grounding in the physical world. Another way to do this would be to compute the intermediate deformations in a manner consistent with the material properties which we built into the finite element model. In this case, we solve for the deformations at each time step via the *dynamic equation*:

$$\ddot{\tilde{\mathbf{U}}}(t) + \tilde{\mathbf{D}}\dot{\tilde{\mathbf{U}}}(t) + \mathbf{\Omega}^2\tilde{\mathbf{U}}(t) = \mathbf{\Phi}^T\mathbf{R}(t), \tag{45}$$

where $\mathbf{R}(t)$ is now a dynamic load vector:

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + (\mathbf{x}_{1,i} + \mathbf{u}_i(t) - \mathbf{x}_{2,i}), \tag{46}$$

where $\mathbf{u}_i(t)$ is the nodal displacement at the previous time step. When solving the dynamic equation, we use features of one image to exert forces which 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.

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

$$\ddot{\tilde{u}}_i(t) + \tilde{d}_i\dot{\tilde{u}}_i(t) + \omega_i^2\tilde{u}_i(t) = \tilde{r}_i(t), \tag{47}$$

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. The system is integrated forward in time until the change in load energy goes below a threshold δ, *e.g.*:

$$\| \mathbf{R}(t + \Delta t) - \mathbf{R}(t) \|^2 < \delta^2. \tag{48}$$

The loads $\mathbf{r}_i(t)$ are updated at each time step by evaluating Equation 46.

7.2 Comparing Objects

Once the mode amplitudes have been recovered, we can compute the strain energy incurred by these deformations by plugging into Equation 44. This strain energy can then be used as a similarity metric. As will be seen in the examples, we may also want to compare the strain in a subset of modes only, or the strain for each mode separately. The strain associated with the i^{th} mode is simply:

$$E_{mode_i} = \frac{1}{2}\tilde{u}_i^2 {\omega_i}^2. \tag{49}$$

Since each mode's strain energy is scaled by its frequency of vibration, there is an inherent penalty for deformations which 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 local shape variations.

Instead of looking at the strain energy needed to align $\mathbf{I}_1$ with $\mathbf{I}_2$, 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.

7.3 Alignment and Description

Figure 9 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 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 vibration modes. On the order of 50 strongest corresponding features were used as input to the strain-minimizing version of the Equation 43. The modal strain energy was computed using Equation 44.

The graphs in Figure 9 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 9(a) shows an airplane that is similar to the prototype, and which 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 9(b) depicts an airplane which is from the same class of airplanes as the prototype, but viewed from a very different angle. In this case, the graph of deformation 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 which are similar to the deformations produced by changes in viewpoint.

The final example, Figure 9(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 9(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.

7.4 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 10 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 between 60 and 120 points per contour. Using the strongest matched contour points, we then recovered the first 28 modal deformations which warp the prototype onto the other tools. 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 11 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 11) cannot be described simply in terms of a few deformations of the wrench.

By observing that there is a simple physical deformation which 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.

7.5 Recognition of Objects and Categories

In the next example (Figures 12 and 13) we will use modal strain energy to compare three different prototype tools: a wrench, hammer, and crescent wrench. As before, silhouettes were first extracted

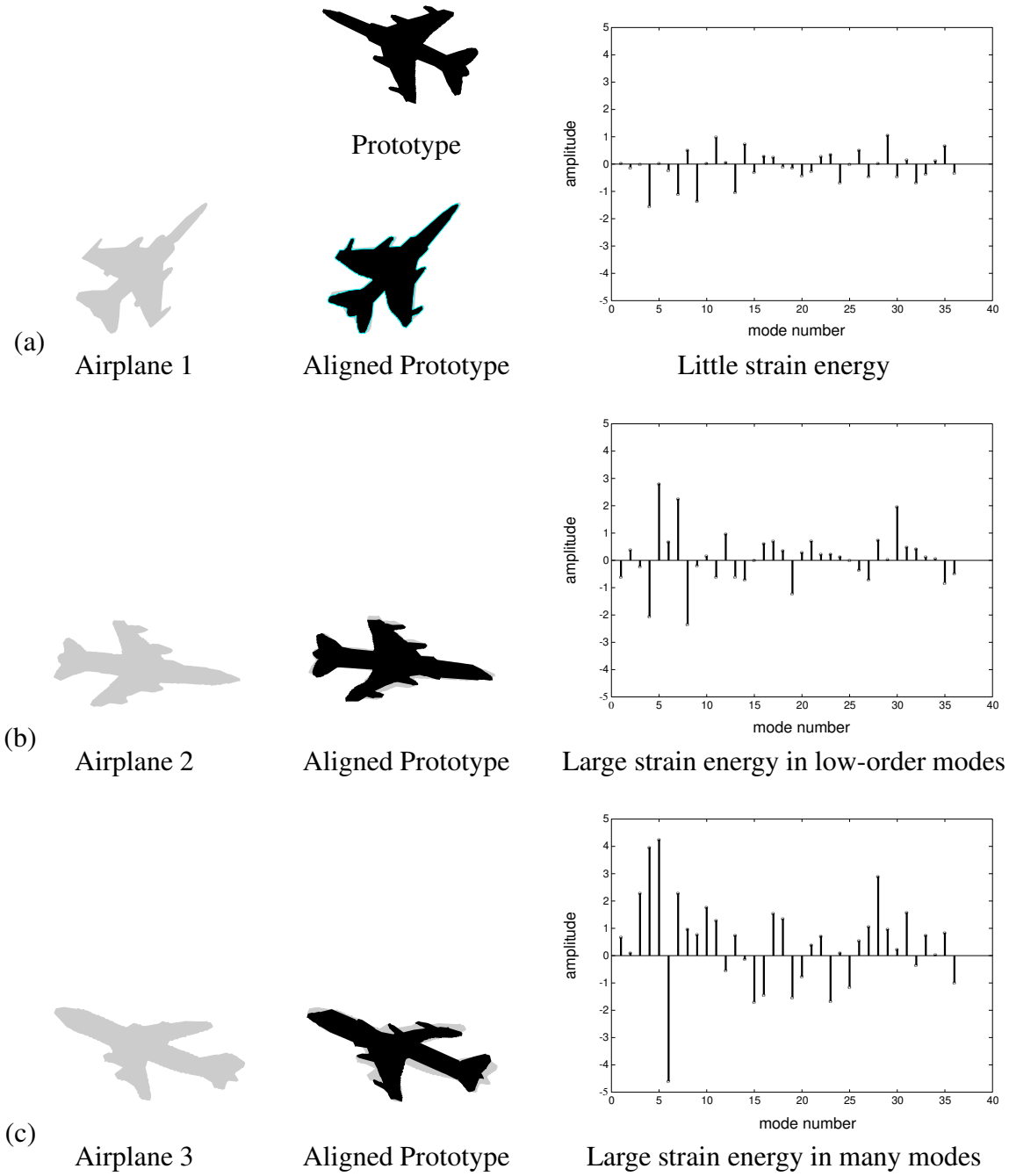


Figure 9: Describing planes in terms of a prototype. The graphs show the 36 mode amplitudes used to align the prototype with each target shape. (a) shows that similar shapes can be aligned with little deformation; (b) shows that viewpoint changes produce mostly low-frequency deformations, and (c) shows that to align different shapes requires both low and high frequency deformations.

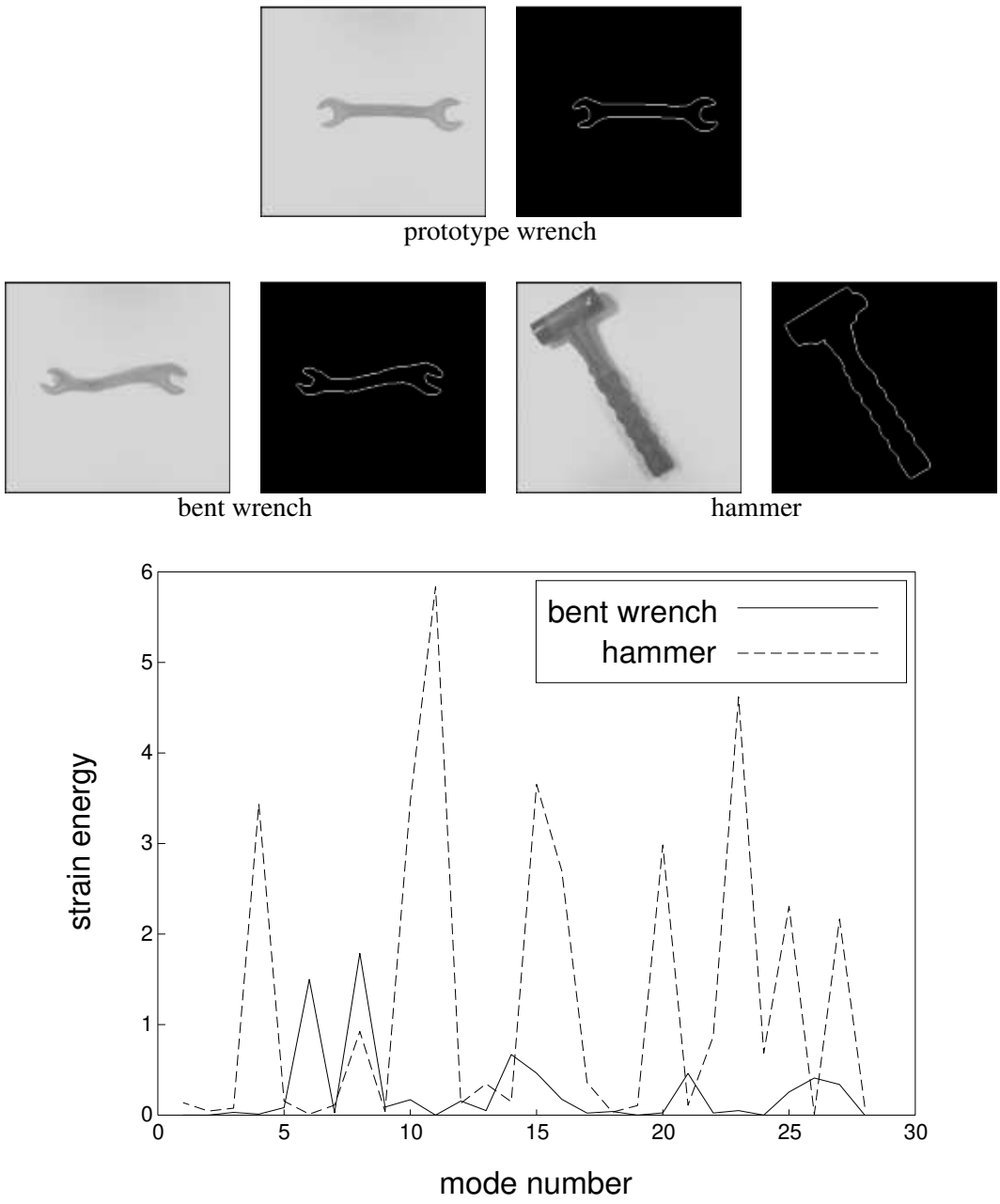


Figure 10: Describing a bent wrench and a hammer in terms of modal deformations from a prototype wrench. Silhouettes were extracted from the images, and then the strongest corresponding contour points were found. Using these matched contour points, the first 28 modal deformations which warp the prototype's 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.

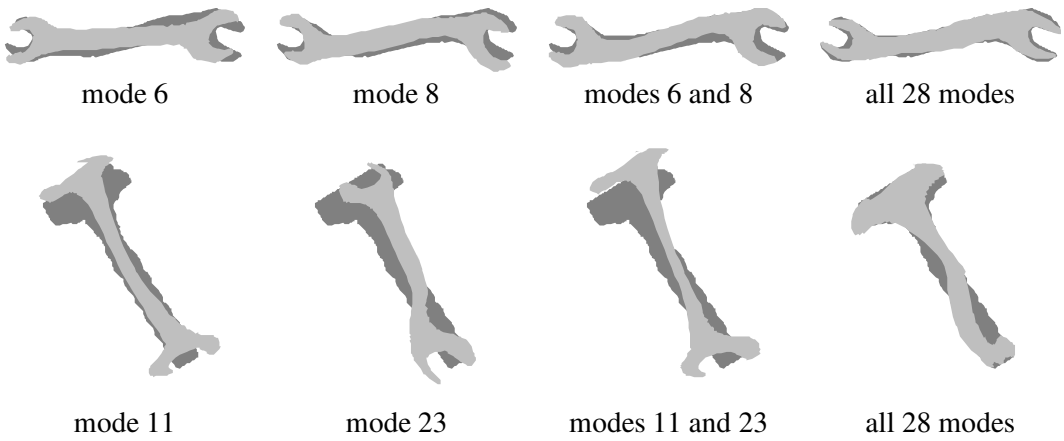


Figure 11: Using the two modes with largest strain energy to deform the prototype wrench to two other tools. The figures demonstrates how the top two highest-strain modal deformations contribute to the alignment of a prototype wrench to the bent wrench and a hammer of Figure 10.

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 which results from deforming the prototype to each tool is shown below each image.

Figure 12 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 13 continues this example using as prototypes the hammer and a single-ended wrench. Again, the modal strain energy which 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.

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*.

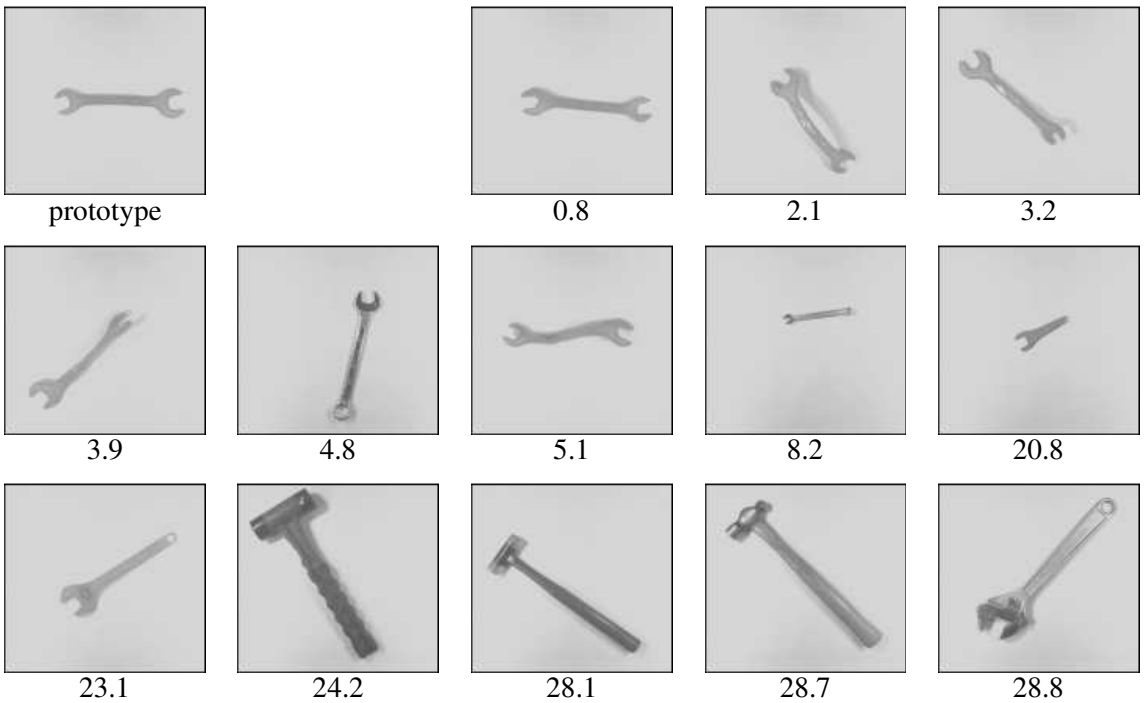
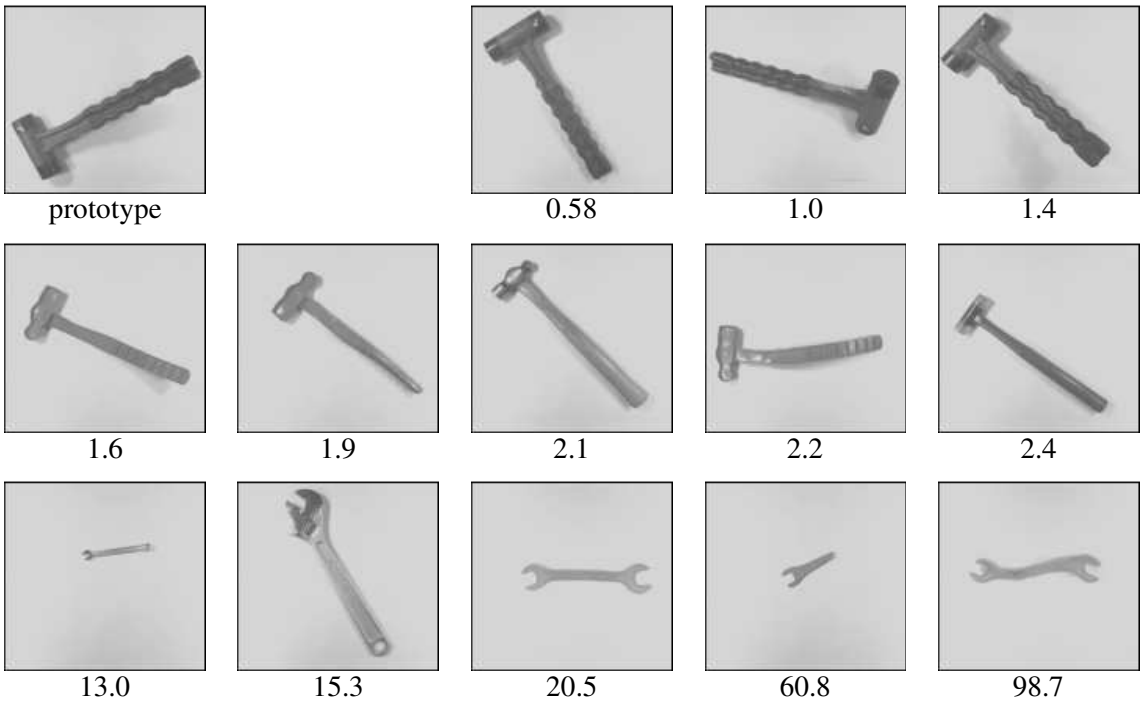


Figure 12: Using modal strain energy to compare a prototype wrench with different hand tools. As in Figure 10, silhouettes were first extracted from each tool image, and then the strongest corresponding contour points were found. Mode amplitudes for the first 22 modes were recovered and used to warp the prototype onto the other tools. The modal strain energy which 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.



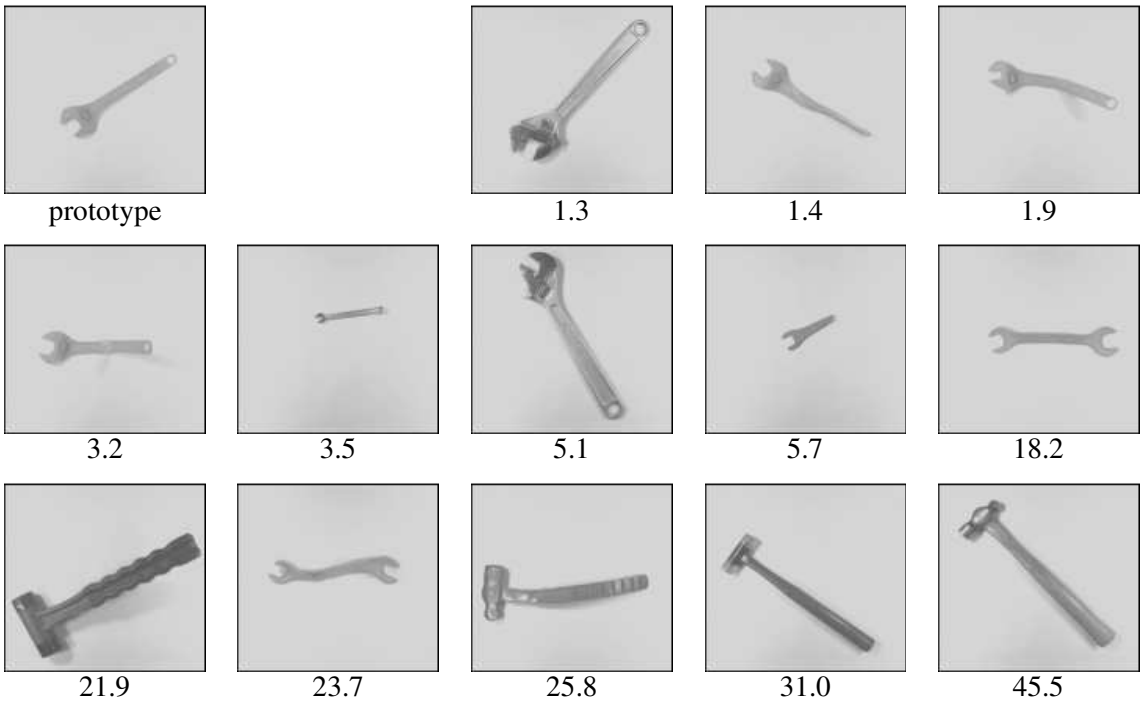


Figure 13: Using modal strain energy to compare a crescent wrench with different hand tools, and a prototype hammer with different hand tools. Strain energies were computed as in Figure 12. The modal strain energy which results from deforming the prototype to each tool is shown below each image.

8 Related Work

Recently several authors have developed similar techniques for establishing correspondence that are based on the idea of describing object deformation by the eigenvectors of an autocorrelation-like "shape matrix."

Scott and Longuet-Higgins [22] developed a *proximity matrix* description which describes Gaussian-weighted distances between point data. They then showed that the eigenvectors of this matrix could be used to determine correspondences between two sets of points. A substantially improved version of this approach was developed by Shapiro and Brady [25, 26].

Cootes, Cooper, Taylor, and Graham [8] developed a chord-length description matrix which describes the chord length distribution between point features. They then used the eigenvectors of this matrix to describe sets of feature points and their deformation.

The idea behind these approaches is similar to ours. Shape is described by a positive definite symmetric matrix measuring connectedness between data points. This shape description can be uniquely decomposed into a set of linearly-superimposed components by an eigenvector analysis of this shape matrix.

However, there are great differences in performance and range of applicability:

- The proximity methods are not information preserving, and cannot be used to interpolate intermediate deformations or to obtain canonical descriptions for recognition. Our method is information preserving, and provides a robust, canonical description that is directly useful for recognition.
- The chord-length method requires that full correspondences be given, while our method and the proximity methods only require choosing the size of Gaussian weighting functions to define the scale of analysis.
- Both the proximity and chord length descriptions are based directly on the sampled feature points. Consequently, if different feature points are present in different views, or if there are very different sampling densities, then the shape matrix for the two views will differ even if the object's pose and shape are identical. Our approach avoids most of this difficulty by embedding the samples in a virtual material.
- The proximity and chord length methods cannot incorporate information about feature connectivity or distinctiveness; data are treated as clouds of identical points. Such information can easily be expressed when simulated mass and stiffness are included in the representation.

9 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 [22, 25, 26] 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 which 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. Finally, 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.

Acknowledgments. Thanks are given to Joe Born, Irfan Essa, John Martin, and Larry Shapiro for their help and encouragement, and to Karen Donoghue and Steve Librande of the Visible Language Workshop for providing images of airplanes.

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

[4] F. Bookstein. *Morphometric Tools for Landmark Data*. Cambridge University Press, 1991.

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

[6] 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.

[7] L. Cohen. On active contour models and balloons. *Computer Vision, Graphics and Image Processing: Image Understanding*, To Appear 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] I. Essa, S. Sclaroff, and A. Pentland. *Directions in Geometric Computing*, chapter Physicall-based Modeling for Graphics and Vision. Information Geometers, U.K., 1992.

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

[12] T. Kanade and J. Kender. *Human and Machine Vision*, chapter Mapping image properties into shape constraints: skewed symmetry, affine transformable patterns, and the shape from texture paradigm, pages 237–257. Academic Press, 1983.

[13] P. Lipson, A. Yuille, D. O'Keefe, J. Cavanaugh, J. Taafe, and D. Rosenthal. Deformable templates for feature extraction from medical images. In *Proc. European Conference on Computer Vision*, pages 413–417, 1990.

[14] A. Pentland, Perceptual Organization and the Representation of Natural Form, *AI Journal*, Vol. 28, No. 2, pp. 1-38, 1986.

[15] A. Pentland. Automatic extraction of deformable part models. *International Journal of Computer Vision*, 107–126, 1990.

[16] A. Pentland and B. Horowitz. Recovery of Non-Rigid Motion and Structure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7), July 1991.

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

[18] 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, 1989.

[19] M. Powell. *Radial basis functions for multivariate interpolation: a review*. Technical Report DAMPT 1985/NA12, Cambridge, University, 1985.

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

[21] S. Sclaroff and A. Pentland. A modal framework for correspondence and description. In *Proc. Fourth Internation Conference on Computer Vision*, May 1993.

[22] G. Scott and H. Longuet-Higgins. An algorithm for associating the features of two patterns. In *Proc. Royal Society of London*, pages 21–26, 1991.

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

[24] L. Segerlind. *Applied Finite Element Analysis*. John Wiley and Sons, 1984.

[25] L. Shapiro. *Towards a vision-based motion framework*. Technical Report, Oxford University, 1991.

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

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

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

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

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

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

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

[33] G. Wolberg. Skeleton based image warping. *Visual Computer*, 5(1):95–108, 1989.

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

Appendix

A Initial Alignment Step

If the rotation needed to align the two sets of points is potentially large, then we perform an initial alignment step before recovering the modal deformations. The method is quaternion-based, and was originally introduced by Horn [11]. The following is a summary of this orientation recovery method for 3-D (to apply to 2-D orientation recovery problems, set z coordinates to zero). It is worth noting that using Horn's technique before recovering the modal deformations is unnecessary if the rotation needed to align the two point sets is relatively small.

Using only a few of the strongest feature correspondences (strong matches have relatively small values in the affinity matrix $\mathbf{Z}$) we can solve for rigid body modes directly. This initial orientation calculation is based on only the strongest matches, thus we are likely to get a very good estimate of the rigid body parameters which take features in $\mathbf{I}_2$ and line them up with their counter parts in $\mathbf{I}_1$.

Once $\mathbf{I}_1$ and $\mathbf{I}_2$ have been aligned in this way, the features may not be completely aligned. We can now solve for the modal deformations $\tilde{\mathbf{U}}$ which completely align the matched features. This two step process of alignment allows us to independently solve for orientation and deformation parameters, allowing for robustness in the face of large rotation and translations.

This separation of orientation, *etc.* is a very important detail when it comes to dealing with large changes in rotation. It is also very useful for animation, where it may be necessary to separate scripts animation/morphing for changes in position, orientation, and scale from scripts for deformation.

In addition, since only the strongest features have been matched, and $\mathbf{I}_2$ oriented and deformed to align with $\mathbf{I}_1$, we may want to go back and find correspondences for the unmatched features, or features for which the affinity measure z_{ij} was above a threshold. This can be done by finding the closest feature, or closest point on the other body.

As input, the method is given two sets of n matched points $\mathbf{x}_1$ and $\mathbf{x}_2$ in images $\mathbf{I}_1$ and $\mathbf{I}_2$, where $\mathbf{x}_{1,i}$ corresponds to $\mathbf{x}_{2,i}$. We first find the centroids for these two point sets:

$$\bar{\mathbf{x}}_1 = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_{1,i} \quad \text{and} \quad \bar{\mathbf{x}}_2 = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_{2,i}. \tag{50}$$

These centroids are then subtracted from the two point sets, so that they are now expressed in terms of their local coordinates:

$$\mathbf{x'}_{1,i} = \mathbf{x}_{1,i} - \bar{\mathbf{x}}_1 \quad \text{and} \quad \mathbf{x'}_{2,i} = \mathbf{x}_{2,i} - \bar{\mathbf{x}}_2. \tag{51}$$

We are now ready to recover the unit quaternion $\mathbf{q} = q_0 + iq_x + jq_y + kq_z$ which describes the

rotation which orients $\mathbf{x}_1$ with $\mathbf{x}_2$. This is done by first building the symmetric matrix

$$\mathrm{M} = \begin{bmatrix} (S_{xx}+S_{yy}+S_{zz}) & S_{yz}-S_{zy} & S_{zx}-S_{xz} & S_{xy}-S_{yx} \\ S_{yz}-S_{zy} & (S_{xx}-S_{yy}-S_{zz}) & S_{xy}+S_{yx} & S_{zx}+S_{xz} \\ S_{zx}-S_{xz} & S_{xy}+S_{yx} & (-S_{xx}+S_{yy}-S_{zz}) & S_{yz}+S_{zy} \\ S_{xy}-S_{yx} & S_{zx}+S_{xz} & S_{yz}+S_{zy} & (-S_{xx}-S_{yy}+S_{zz}) \end{bmatrix} \tag{52}$$

where

$$S_{xx} = \sum_{i=1}^{n} x'_{1,i}\, x'_{2,i}, \quad S_{xy} = \sum_{i=1}^{n} x'_{1,i}\, y'_{2,i}, \quad \textit{etc.} \tag{53}$$

We then find the most positive eigenvalue for the matrix M. Horn has shown that the eigenvector associated with this eigenvalue represents the unit quaternion $\mathbf{q}$ which gives the optimal rotation of $\mathbf{x}_1$ to $\mathbf{x}_2$ (optimal in that it minimizes root mean squared error).

The scale s can be found independently of the rotation:

$$s = \left(\sum_{i=1}^{n} \| \mathbf{x}'_{2,i} \|^2 \Big/ \sum_{i=1}^{n} \| \mathbf{x}'_{1,i} \|^2 \right)^{\frac{1}{2}}. \tag{54}$$

In a departure from the Horn technique, we calculate the translation $\mathbf{x}_0$ as the difference between the centroids $\bar{\mathbf{x}}_1$ and $\bar{\mathbf{x}}_2$:

$$\mathbf{x}_0 = \bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1. \tag{55}$$

This modification allows us to express the optimal rotation and scaling in terms of the local coordinate systems for each body.

The matrix which describes the aligning rotation $\mathcal{R}$ is built from the recovered quaternion:

$$\mathcal{R} = \begin{bmatrix} (q_0^2+q_x^2+q_y^2+q_z^2) & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 2(q_y q_x + q_0 q_z) & (q_0^2-q_x^2+q_y^2-q_z^2) & 2(q_y q_z - q_0 q_x) \\ 2(q_z q_x - q_0 q_y) & 2(q_z q_y + q_0 q_x) & (q_0^2-q_x^2-q_y^2+q_z^2) \end{bmatrix}. \tag{56}$$

A.1 Using Affinities as Confidence Weights

Appendix A.2 of Horn's paper describes how to include point match confidence weights w_i into the recovery formulation. In our application, these weights can be made inversely proportional to the affinity measure, *e.g.,* $w_i = 1/(1+z_{ij})$. Since weights are used, the centroids become weighted centroids:

$$\bar{\mathbf{x}}_1 = \sum_{i=1}^{n} w_i \mathbf{x}_{1,i} \Big/ \sum_{i=1}^{n} w_i \quad \text{and} \quad \bar{\mathbf{x}}_2 = \sum_{i=1}^{n} w_i \mathbf{x}_{2,i} \Big/ \sum_{i=1}^{n} w_i\,. \tag{57}$$

Computing the scale factor also includes the weights

$$s = \left(\sum_{i=1}^{n} w_i \| \mathbf{x}'_{2,i} \|^2 \Big/ \sum_{i=1}^{n} w_i \| \mathbf{x}'_{1,i} \|^2 \right)^{\frac{1}{2}}; \tag{58}$$

Finally, the entries in the matrix $\mathbf{M}$ change to:

$$S_{xx} = \sum_{i=1}^{n} w_i {x'}_{1,i} {x'}_{2,i}, \quad S_{xy} = \sum_{i=1}^{n} w_i {x'}_{1,i} {y'}_{2,i}, \quad \textit{etc.} \tag{59}$$

A.2 Recovering Deformations with Initial Orientation Step

In the modified technique, we first align the two point sets using the rotation, translation, and scale recovered using the Horn method. The points can now be further aligned by recovering the modal deformations $\tilde{\mathbf{U}}$ as described previously. As before, we compute virtual loads $\mathbf{R}$ which 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, we must modify Equation 39 so as to measure distances between features in the correct coordinate frame:

$$u_i = (\frac{1}{s}\mathcal{R}^T \left[\mathbf{x}_{2,i} - \mathbf{x}_0 - \bar{\mathbf{x}}_1\right] + \bar{\mathbf{x}}_1 - \mathbf{x}_{1,i}). \tag{60}$$

Through the initial aligment 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. 45).

The modal displacements now act in a rotated and scaled space; as a result, we must update our displacement interpolation equation:

$$\mathbf{u}(\mathbf{x}) = s\mathcal{R}\mathbf{H}\mathbf{\Phi}\tilde{\mathbf{U}} \tag{61}$$

where $\mathcal{R}$ is a rotation matrix.

This method of initial alignment and deformation is shown for the two flat tree shapes in Figure 14. This figure shows how the two shapes can be progessively aligned using rigid body modes and then modal deformations. Given only a few of the strongest correspondences between corner points on both trees, we use Horn's algorithm to first recover the rigid body modes and align the two trees. The trees are aligned further using modal deformations. The number of modes used determines how closely the trees are aligned. The bottom row of this figure shows the results using six and eighteen modes. As can be seen, six non-rigid modes are sufficient to account for most of the deformation due to shear. It has been our experience that the first six to nine non-rigid modes can adequately model the standard affine deformations.

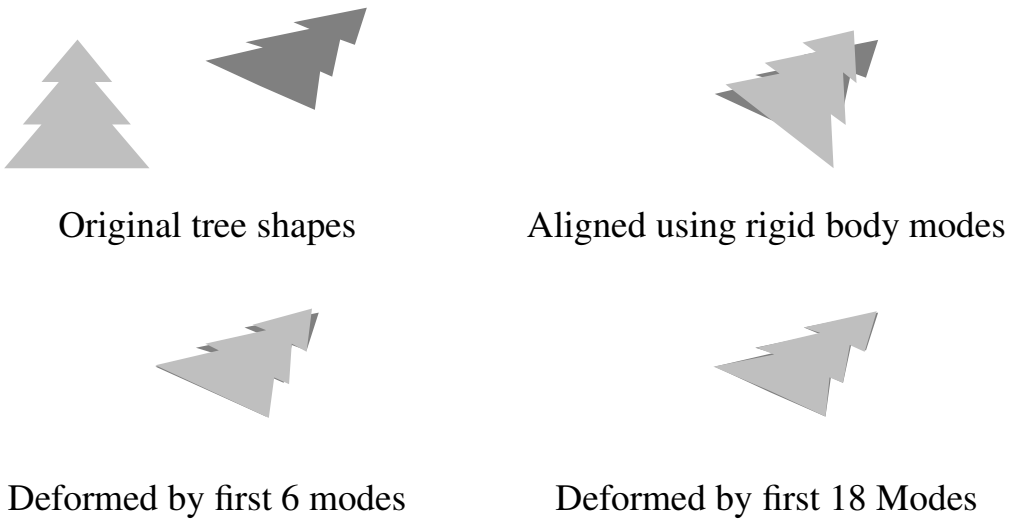


Figure 14: Using rigid body modes and then deformations to align two flat tree shapes. Given correspondences between corner points on both trees, we use Horn's algorithm to first recover the rigid body modes and align the two trees. The trees are aligned further using modal deformations which are recovered as described in the text. The number of modes used determines how closely the trees are aligned. The bottom row of this figure shows the results using six and eighteen modes.

B Building Mass and Stiffness Matrices for n-dimensional Problems

This is a description of how to extend the 2-D finite element matrix formulation to 3-D, and then how to generalize this formulation to solve n-dimensional $(n > 3)$ problems. The idea of an n-dimensional finite element may seem bizarre (and computationally impractical!) from a mechanical engineering standpoint. However, this physical analogy will prove useful in applying our method to higher dimensional feature matching and alignment problems. And, as has been noted before, we can significantly curtail the computational cost associated with computing in higher dimensions by using factorizable basis functions (Gaussians).

For ease of notation, we define $\mathcal{H}$ to be a row vector which contains the finite element interpolation functions h_i:

$$\mathcal{H} = [\ h_1 \quad h_2 \quad \ldots \quad h_m\ ]. \tag{62}$$

We define $\mathbf{x}$ to be an n-dimensional coordinate whose components are written $x_1, x_2, \ldots x_n$.

Finally, we define the partial derivative of $\mathcal{H}$ with respect to x_i to be:

$$\frac{\partial}{\partial x_i}\mathcal{H} = \left[\ \frac{\partial}{\partial x_i}h_1 \quad \frac{\partial}{\partial x_i}h_2 \quad \ldots \frac{\partial}{\partial x_i}h_m\ \right]. \tag{63}$$

B.1 Formulating the 3-D Mass Matrix

The Gaussian interpolants are assembled into $3 \times 3m$ interpolation matrix,

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} h_1 & \dots & h_m & & & & & & \\ & & & h_1 & \dots & h_m & & & \\ & & & & & & h_1 & \dots & h_m \end{bmatrix} = \begin{bmatrix} \mathcal{H} & & \\ & \mathcal{H} & \\ & & \mathcal{H} \end{bmatrix}. \tag{64}$$

where m is the number of finite element nodes.

We the substitute this into the standard equation for the mass matrix:

$$\mathbf{M} = \int_V \rho \mathbf{H}^T \mathbf{H} dV = \begin{bmatrix} \mathrm{M}_{11} & 0 & 0 \\ 0 & \mathrm{M}_{22} & 0 \\ 0 & 0 & \mathrm{M}_{33} \end{bmatrix} \tag{65}$$

where the m by m submatrices M_{11}, M_{22}, and M_{33} are positive semi-definite symmetric, and M_{11} = M_{22} = M_{33}. The elements of M_{11} have the form:

$$m_{11_{ij}} = \rho \int_V \sum_{kl} a_{ik}\, a_{jl}\, g_k(\mathbf{x})\, g_l(\mathbf{x}) dV. \tag{66}$$

where a_{ik} are the interpolation coefficients, and g_k and g_l are 3-dimensional Gaussian basis functions.

B.2 Generalizing the Mass Matrix to n Dimensions

It can be shown that the mass matrix for an n-dimensional problem will be a block diagonal matrix of the form

$$\mathbf{M} = \int_V \rho \mathbf{H}^T \mathbf{H} dV = \begin{bmatrix} \mathrm{M}_{11} & & & \\ & \mathrm{M}_{22} & & \\ & & \ddots & \\ & & & \mathrm{M}_{nn} \end{bmatrix} \tag{67}$$

where the submatrices are positive semi-definite symmetric, and M_{11} = M_{22} $\dots$ = M_{nn}. The elements $m_{11_{ij}}$ of these submatrices are computed by using n-dimensional Gaussian basis functions in Equation 66.

B.3 Formulating the 3-D Stiffness Matrix

To calculate the stiffness matrix, we will need to first formulate a strain interpolation matrix $\mathbf{B}$. Strain for at a 3-D element node is defined as $\boldsymbol{\epsilon} = [\epsilon_{xx}, \epsilon_{yy}, \epsilon_{zz}, \gamma_{xy}, \gamma_{xz}, \gamma_{yz}]^T$, where:

$$\begin{gathered} \epsilon_{xx} = \frac{\partial u}{\partial x} \qquad \epsilon_{yy} = \frac{\partial v}{\partial y} \qquad \epsilon_{zz} = \frac{\partial w}{\partial z} \\ \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \qquad \gamma_{xz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \qquad \gamma_{yz} = \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \end{gathered} \tag{68}$$

where $\mathbf{u} = (u, v, w)$ is the displacement vector at the node.

The corresponding strain displacement matrix $\mathbf{B}$ is obtained by appropriately differentiating the interpolation functions h_i. For the three dimensional problem, $\mathbf{B}$ is a $(6 \times 3m)$ matrix:

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x}\mathcal{H} & & \\ & \frac{\partial}{\partial y}\mathcal{H} & \\ & & \frac{\partial}{\partial z}\mathcal{H} \\ \frac{\partial}{\partial y}\mathcal{H} & \frac{\partial}{\partial x}\mathcal{H} & \\ \frac{\partial}{\partial z}\mathcal{H} & & \frac{\partial}{\partial x}\mathcal{H} \\ & \frac{\partial}{\partial z}\mathcal{H} & \frac{\partial}{\partial y}\mathcal{H} \end{bmatrix} \tag{69}$$

The 3-D stiffness matrix of the element is then computed directly:

$$\mathbf{K} = \int_V \mathbf{B}^T \mathbf{C}\mathbf{B} dV = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \mathbf{K}_{13} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \mathbf{K}_{23} \\ \mathbf{K}_{31} & \mathbf{K}_{32} & \mathbf{K}_{33} \end{bmatrix}. \tag{70}$$

The stiffness matrix $\mathbf{K}$ is symmetric; therefore the submatrices satisfy $\mathbf{K}_{12}{}^T = \mathbf{K}_{21}$, $\mathbf{K}_{31}^T = \mathbf{K}_{13}$, *etc.* Since we use Gaussian basis functions, its actually the case that the submatrices themselves are symmetric; therefore $\mathbf{K}_{12} = \mathbf{K}_{21}$, *etc.*

Included in the formula for the stiffness matrix is a material matrix $\mathbf{C}$ which expresses the simulated material properties:

$$\beta \begin{bmatrix} 1 & \alpha & \alpha & & & \\ \alpha & 1 & \alpha & & & \\ \alpha & \alpha & 1 & & & \\ & & & \xi & & \\ & & & & \xi & \\ & & & & & \xi \end{bmatrix}. \tag{71}$$

This particular 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:

$$\alpha = \frac{\nu}{1-\nu}, \qquad \beta = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}, \quad \text{and} \quad \xi = \frac{1-2\nu}{2(1-\nu)}. \tag{72}$$

It is possible to use other material matrices which model anistropic matrials. An anisotropic material would be useful in modeling the difference between deformations in time and space — for instance, modeling "fibers" along the time axis in a 3-D volume created by a motion picture image sequence.

B.4 Generalizing the Stiffness Matrix to n Dimensions

To determine what the components of the n-dimensional strain vector will be, we can build an upper triangular matrix of the form:

$$\begin{bmatrix} \epsilon_{x_1x_1} & \gamma_{x_1x_2} & \gamma_{x_1x_3} & \cdots & \gamma_{x_1x_{n-1}} & \gamma_{x_1x_n} \\ & \epsilon_{x_2x_2} & \gamma_{x_2x_3} & \cdots & \gamma_{x_2x_{n-1}} & \gamma_{x_2x_n} \\ & & \epsilon_{x_3x_3} & \cdots & \gamma_{x_3x_{n-1}} & \gamma_{x_3x_n} \\ & & & \ddots & \vdots & \vdots \\ & & & & \epsilon_{x_{n-1}x_{n-1}} & \gamma_{x_{n-1}x_n} \\ & & & & & \epsilon_{x_nx_n} \end{bmatrix}. \tag{73}$$

where x_i is i^{th} component of the n-dimensional vector $\mathbf{x}$.

Thus, to generalize the notion of strain to n dimensions, we note that the strain can be represented by a $\frac{n^2+n}{2}$ vector which contains the elements of this upper triangular matrix:

$$\boldsymbol{\epsilon}^T = [\ \underbrace{\epsilon_{x_1x_1}, \ldots \epsilon_{x_nx_n}}_{\text{diagonal elements}}\ ,\ \underbrace{\gamma_{x_1x_2}, \gamma_{x_1x_3}, \ldots \gamma_{x_{n-1}x_n}}_{\text{upper triangle elements}}\ ], \tag{74}$$

where

$$\epsilon_{x_ix_i} = \frac{\partial u_i}{\partial x_i} \quad \text{and} \quad \gamma_{x_ix_j} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}, \tag{75}$$

where the u_i are components of the n-dimensional displacement vector at the particular FEM node.

The n-dimensional strain displacement matrix $\mathbf{B}$ is obtained by appropriately differentiating

the interpolation functions h_i. The matrix $\mathbf{B}$ is a $(\frac{n^2+n}{2} \times nm)$ matrix of the form:

$$\mathbf{B}(\mathbf{x}) = \left[\begin{array}{c|cc|c} \frac{\partial}{\partial x_1}\mathcal{H} & & & \\ & \frac{\partial}{\partial x_2}\mathcal{H} & & \\ & & \frac{\partial}{\partial x_3}\mathcal{H} & \\ & & & \ddots \\ & & & \frac{\partial}{\partial x_n}\mathcal{H} \\ \hline \frac{\partial}{\partial x_2}\mathcal{H} & \frac{\partial}{\partial x_1}\mathcal{H} & & \\ \frac{\partial}{\partial x_3}\mathcal{H} & & \frac{\partial}{\partial x_1}\mathcal{H} & \\ \vdots & & & \ddots \\ \frac{\partial}{\partial x_n}\mathcal{H} & & & \frac{\partial}{\partial x_1}\mathcal{H} \\ \hline & \frac{\partial}{\partial x_3}\mathcal{H} & \frac{\partial}{\partial x_2}\mathcal{H} & \\ & \vdots & & \ddots \\ & \frac{\partial}{\partial x_n}\mathcal{H} & & \frac{\partial}{\partial x_2}\mathcal{H} \\ \hline & etc. & & \end{array}\right]. \tag{76}$$

To get the n-dimensional stiffness matrix, we use ageneralized version of Equation 71. Assuming an isotropic material, the n-dimensional material matrix $\mathbf{C}$ will take the form:

$$\beta \left[\begin{array}{cccc|ccc} 1 & \alpha & \dots & \alpha & & & \\ \alpha & 1 & \dots & \alpha & & & \\ \vdots & \vdots & \ddots & \vdots & & & \\ \alpha & \alpha & \dots & 1 & & & \\ \hline & & & & \xi & & \\ & & & & & \xi & \\ & & & & & & \ddots \\ & & & & & & \xi \end{array}\right] \tag{77}$$

where the upper submatrix is $n \times n$, and the lower submatrix is $\frac{n^2-n}{2} \times \frac{n^2-n}{2}$.