M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 273
Will appeared in *Proc. of the IEEE Workshop on Nonrigid and Articulate Motion*, **Austin, TX, Nov. 1994.**

Physically-Based Combinations of Views: Representing Rigid and Nonrigid Motion

Stan Sclaroff *
Computer Science Dept.
Boston University
111 Cummington St., Boston MA 02215

Alex P. Pentland
Perceptual Computing Section
The MIT Media Laboratory
20 Ames St., Cambridge MA 02139

Abstract

Nonrigid motion can be described as morphing or blending between extremal shapes, e.g., heart motion can be described as transitioning between the systole and diastole states. Using physically-based modeling techniques, shape similarity can be measured in terms of forces and strain. This provides a physically-based coordinate system in which motion is characterized in terms of physical similarity to a set of extremal shapes. Having such a low-dimensional characterization of nonrigid motion allows for the recognition and the comparison of different types of nonrigid motion.

1 Introduction

The computer graphics technique of *morphing* has become quite popular in advertisements. Morphing is accomplished by an artist identifying a large number of corresponding control points in two images, and then incrementally deforming the geometry of the first image so that its control points eventually lie atop the control points of the second image. While this deformation is occurring, the grey-level values of the two images are also interpolated. If the artist selects control point correspondences that produce a geometrically smooth deformation field *and* a smooth transition in grey level, then a visually compelling transition from the first image to the second is obtained [20].

This suggests an important way to obtain a parametric description of rigid, nonrigid, or articulated motion: interpolate between known views. Given views of the extremes of a motion (*e.g.*, systole and diastole, or left-leg forward and right-leg forward) we can describe the intermediate views as a smooth combination of the extremal views. Importantly, we can derive this parameterization *without* knowing all the details of the physical system, although such detailed knowledge would help in obtaining a more accurate, physically-meaningful parameterization.

All that is required to determine the view-based parameterization of a new image are the extremal views, point correspondences between the new image and the extremal views, and a method of measuring the amount of (nonrigid) deformation that has occurred between the new image and each extremal view. The extremal views define a polytope in the space of the (unknown) underlying physical system's parameters. By measuring the amount of deformation between the new image and extremal views, we locate the new image in the coordinate system defined by the polytope.

*This work was done while the author was at the Media Lab.

This approach to describing motion is related to the view-based shape recognition proposals of Ullman and Basri [19] and Poggio, *et al.* [13]. It entails description by interpolating among examples, rather than description by some more abstract, view-independent representation.

However, it differs from their proposals in two important ways. First, we are interested not only in recognizing shapes, but also in describing motion (including nonrigid and articulated motion). We want to derive a low-dimensional parametric representation of motion that can be used to recognize and compare motion trajectories, in the manner of Darrell and Pentland [7]. Second, we cannot be restricted to a linear framework. Nonrigid motions are inherently nonlinear, although they *are* often "physically smooth." Therefore, to employ a combination-of-views approach we must be able to determine point correspondences and measure similarities between views in a way that takes into account at least qualitative physics, and detailed physics if that information is available. In computer graphics it is the job of the artist to enforce the constraint of physical smoothness; in machine vision, we need to be able to do the same automatically.

In this paper we describe a method for (1) determining point correspondences using a physically-based model, (2) warping or morphing one shape into another using physically-based interpolants, and (3) measuring the amount of physical deformation between an object's shape and the extremal views of that object. The result is a low-dimensional parametric representation of the object's motion that is qualitatively related to the underlying physical parameters. Such physically-based parametric descriptions are useful for recognizing or classifying motions, fusing data from different sensors, and for comparing data acquired at different times or under different conditions.

2 Background

Although many things move rigidly, in many cases the rigid-body model is inadequate. For instance, most biological objects are flexible and articulated. To describe these deformations, therefore, it is reasonable to model the physics by which real objects deform. This rationale led to the physical modeling paradigm of active contours or *snakes*[11]. A snake has a predefined structure which incorporates knowledge about the shape and its resistance to deformation. By allowing the user to specify forces that are a function of sensor measurements, the intrinsic dynamic behavior of a physical model can be used to solve fitting, interpolation, or correspondence problems.

While snakes enforced constraints on smoothness and the amount of deformation, they could not in their original form be used to constrain the *types* of deformation valid for a particular problem domain or object class. This led to the development of algorithms which include *a priori* constraints on the types of allowable deformations for motion tracking [3; 4; 5; 8].

Cootes *et al.*[6; 2] use trainable snakes for capturing the invariant properties of a class of shapes, by finding the principle variations of a snake via the Karhunen-Loeve transform. Unfortunately, this method relies on the consistent sampling and labeling of point features across the entire training set and cannot handle large rotations. If different feature points are present in different views, or if there are very different sampling densities, then the resulting PDMs will differ even if the object's pose and shape are identical.

Keeping these issues in mind, we use the Finite Element Method to alleviate problems with sampling, and modal analysis to provide a principled way to select the types of nonrigid deformations needed for tracking nonrigid motion. In the rest of this section we provide a brief review of our representation. In addition, we review our new method of building FEM models without imposing an *a priori* parameterization, and how to use the modes of this model to find point correspondences, to align objects, and to compare their shape. This initial work was applied in the area of finding corresponding features in static imagery [14] and serves as the foundation for our new representation for nonrigid motion.

2.1 Finite Element Method

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

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

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

Solution to the problem of deforming an elastic body to match the set of feature points then requires solving the *dynamic equilibrium equation*:

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

where $\mathbf{R}$ is the load vector whose entries are the spring forces between each feature point and the body surface, and where $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ are the element mass, damping, and stiffness matrices, respectively [1; 12].

2.2 Modal Representation

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

$$\mathbf{K}\boldsymbol{\phi}_i = \omega_i^2 \mathbf{M}\boldsymbol{\phi}_i. \tag{3}$$

The vector $\boldsymbol{\phi}_i$ is called the ith *mode shape vector* and ω_i is the corresponding frequency of vibration. Each mode shape vector describes how each node is displaced by the i^{th} vibration mode. The mode shape vectors are $\mathbf{M}$-orthonormal; this means that $\boldsymbol{\Phi}^T\mathbf{K}\boldsymbol{\Phi} = \boldsymbol{\Omega}^2$ and $\boldsymbol{\Phi}^T\mathbf{M}\boldsymbol{\Phi} = \mathbf{I}$. The ϕ_i form columns in the transform $\boldsymbol{\Phi}$ and ω_i^2 are elements of the diagonal matrix $\boldsymbol{\Omega}^2$. We will assume Rayleigh damping (*i.e.*, $\mathbf{D} = a_0\mathbf{M} + a_1\mathbf{K}$), thus the damping matrix will also diagonalized by this transform [1].

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

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

allowing for closed-form solution to the equilibrium problem [12]. Given this equilibrium solution in the two images, point correspondences can be obtained directly.

By discarding high frequency modes the amount of computation required can be minimized without significantly altering correspondence accuracy. Moreover, such a set of modal amplitudes provides a robust, canonical description of shape in terms of deformations applied to the original elastic body. This allows them to be used directly for object recognition [12].

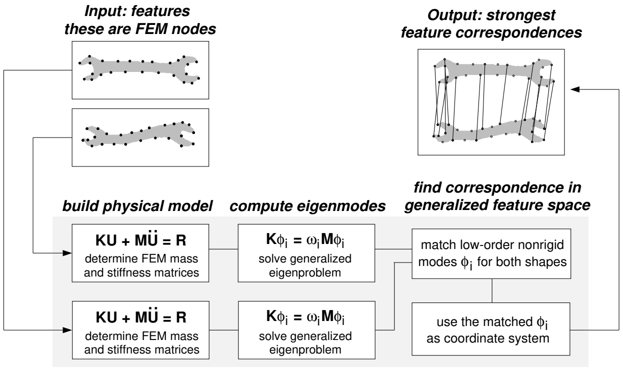


Figure 1: Modal matching system diagram.

2.3 Modal Matching

Perhaps the major limitation of previous methods is the requirement that every object be described as the deformations of a *single* prototype object. This implicitly imposes an *a priori* parameterization upon the sensor data, and therefore implicitly determines the correspondences between data and prototype. We would like to avoid this as much as possible, by letting the data determine the parameterization in a natural manner. To accomplish this we use the data itself to define the deformable object, by building stiffness and mass matrices that use the positions of image feature points as the finite element nodes.

A flow-chart of our method is shown in Fig. 1. For each image we start with feature point locations, which are used as nodes in building a finite element model of the shape. A Gaussian is centered at each node, and these Gaussians are used as Galerkin interpolants in constructing the mass and stiffness matrices [14]. The use of Galerkin interpolants reduces the effects of missing or dislocated features.

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

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

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

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

Point correspondences can now be determined by comparing the two groups of generalized feature vectors. The important idea here is that the low-order vibration modes computed for two similar objects will be very similar — even in the presence of affine deformation, nonrigid deformation, local shape perturbation, noise, or small occlusions. The points that have the most similar and unambiguous coordinates are then matched, with the remaining correspondences determined by using the physical model as a smoothness constraint [14]. 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. Typically this is accomplished by measuring the *strain energy* required to warp the feature points of one object into alignment with those of the other object. The modal alignment parameters $\tilde{\mathbf{U}}$ are found by solving the physical system in Eq. 2. The resulting strain is given by:

$$E_I = \frac{1}{2}\tilde{\mathbf{U}}^T \mathbf{\Omega}^2 \tilde{\mathbf{U}}. \tag{6}$$

In some cases, it is useful to normalize the strain energy by the number of points matched.

3 Modal Motion Analysis and Synthesis

Given the modal matching system as our starting point, we would like to describe nonrigid motion as a combination of some collection of extremal shapes. Unlike previous linear-combinations-of-views techniques, we will employ a physically-based, frequency-ordered description of shape; as a result we can analyze and decompose nonrigid shape deformation (and then synthesize shapes) in a principled way. Furthermore, the modal matching system provides a convenient framework for automatically computing corresponding points in views.

Fig. 2(a) shows a *shape space* defined by three extremal views of airplanes. As an airplane flies around, its silhouette moves, rotates, and deforms due to changes in viewing geometry. Using the algorithm described above, we can determine correspondences and similarity (strain energy) between each of the extremal views. Each edge in Fig. 2(a) is labeled with its associated strain.

Traveling along an edge in this triangle performs a physically-based blend, using the modal deformations, from one extremal view to another. Thus, each edge of the triangle describes a family of views which can be represented as combinations of the two extremal views. Similarly, we can describe an entire family of shapes by moving around inside the triangle defined by three extremal views.

Adding a fourth view to the triangle creates a pyramid, unless the new view can be exactly described as a combination of the extremal views. Fig. 2(b) shows how the fourth plane view can be synthesized from a combination of the three base models. As before, the edges connecting the new model to the pyramid's base have lengths proportional to the strain energy required to align each of the base models with the new model.

In this example the three base models cannot completely account for all of the new plane's shape (there are are missing nacelles, for instance). As a result, the fourth model does not lie in the plane defined by the three base models. The differences between the new plane and the plane synthesized from the triangle of base shapes defines the similarity between the new plane and the *class* of shapes defined by combinations of the prototype models.

Using this similarity measure, we can decide whether or not the new shape is a member of the shape space defined by the prototype models. If it is not, then we can add this fourth shape to our set of extremal views. In this way we can build up a set of extremal views even for complex, articulated motions.

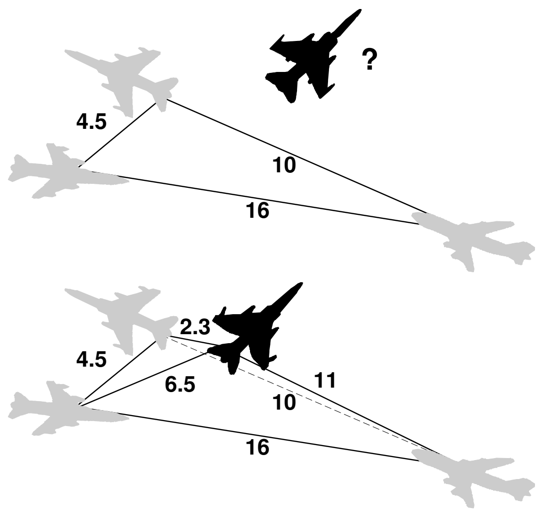
3.1 Modal Morphing

So far, we have described methods for finding the modal displacements which directly deform and align two feature sets, and how to measure their similarity using strain energy. This was sufficient to define the shape space of Fig. 2(a). However, to decide if a new shape is part of a shape space, we must be able to synthesize an interpolated shape for comparison. The dark plane in Fig. 2(b) shows such an interpolated shape; it can be directly compared with the dark shape in Fig. 2(a).

Let us now imagine that we have been given two extremal views of a nonrigid motion and want to describe the motion in between these two extremes as a combination of the extremal states.

We could use a linear interpolation of the extremal states for comparison to the new shape. This method is simple, but has little grounding in the physical world. It is used in computer graphics, but is successful only when the artist carefully selects the point correspondences.

Another way to compare the current image with the extremal states would be to compute the deformations in a



(a)

(b)

Figure 2: Given three gray models in (a), we can define a triangle with edge lengths proportional to the amount of strain needed to align each model; thus, each model is a vertex in this triangle. When we encounter a new model (shown in black), we want see how it can be synthesized from deformed versions of the original three models. As is shown in (b), by adding a fourth model we typically create a pyramid, where the edge lengths are proportional how "easy" it was (how much strain it required) to synthesize the model from each of the three known models.

manner consistent with the object's material properties, *e.g.*, using a finite element model. In this case, the modal deformations $\tilde{\mathbf{U}}$ at an intermediate location are determined by

$$\mathbf{\Omega}^2\tilde{\mathbf{U}} = \mathbf{\Phi}^T\mathbf{R}, \tag{7}$$

where the load vector $\mathbf{R}$ has elements:

$$\mathbf{r}_i = \mathbf{x}_{1,i} - \mathbf{x}_{2,i}, \tag{8}$$

The loads $\mathbf{R}$ at the finite element nodes are therefore proportional to the deformation between matched features.

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

$$\omega_i^2 \tilde{u}_i = \tilde{r}_i, \tag{9}$$

where the $\tilde{r}_i$ are components of the transformed load vector $\tilde{\mathbf{R}} = \mathbf{\Phi}^T\mathbf{R}$.

3.2 Modal Flow

The above technique serves to interpolate feature sets in a physically-based manner. However, we would also like to interpolate all the image points between features, so that we can compare the new image and interpolated image directly. To do this, we will need to generate flow fields. To do this for images, a flow field is a dense 2-D vector field showing where features or pixels move from one image to another.

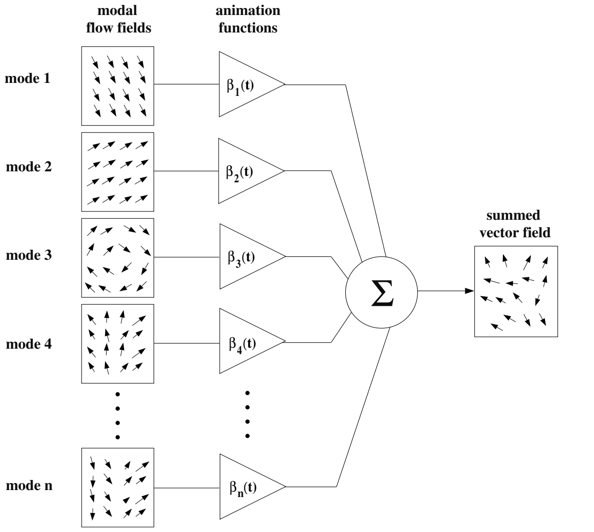


Figure 3: The total flow field for an image warp can be represented as the sum of many *modal* flow fields. Mixture of these flow fields is modulated by coefficients $\beta_i(t)$, which are the distance in modal space between the new shape and an extremal view.

We propose that the finite element interpolation functions h_i should be used for this task, since they are physically-motivated and have already been computed and are therefore readily available. Just as the nodal displacements can be represented as the linear superposition of the decoupled *modal displacements*, the image flow field can be represented as the superposition of decoupled *modal flow fields*. This is illustrated in Fig. 3.

Each of these modal flow fields corresponds with deformations whose flow at the nodes is described by the mode shape vector ϕ_i and mode amplitude $\tilde{u}_i$. For in-between shapes, these flow fields are modulated by coefficients $\beta_i(t)$, which are the distance in modal space between the new shape and an extremal view.

3.3 Computing dense modal vector fields

Recall that given nodal displacements, we can use the FEM interpolation functions $\mathbf{H}$ to compute the displacement at any image location $\mathbf{x}$. This allows us to compute the displacement vectors at the nodes, and then iterate over the image, plugging each pixel's image coordinate $\mathbf{x}$ into Eq. 1, thereby computing the vector field. As a result, the vector field for each nonrigid mode can also be easily obtained.

For each mode, the nodal displacements are proportional to the mode's shape vector ϕ_i, and are stored as a column in the modal transformation matrix $\mathbf{\Phi}$. It follows then that the equation for the i^{th} mode's vector field can be expressed in terms the FEM displacement interpolation function:

$$\mathbf{u}(\mathbf{x}) = \tilde{u}_i \mathbf{H} \phi_i. \tag{10}$$

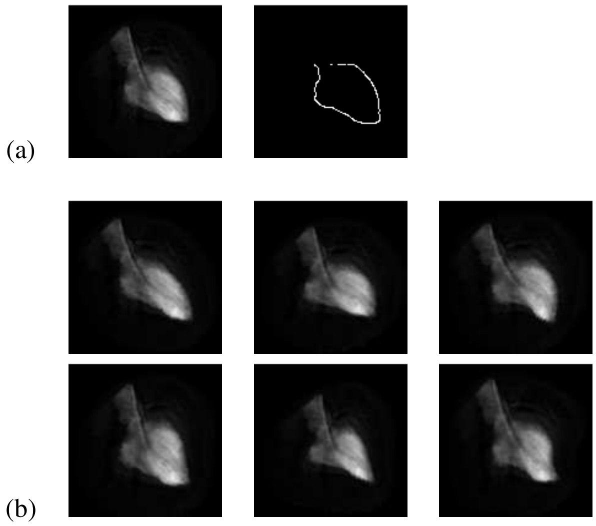


Figure 4: An original image and its extracted contour are shown in (a). The first six nonrigid modes for the heart image are shown in (b).

where $\tilde{u}_i$ is the recovered amplitude for the i^{th} mode. All vector fields are linear and can therefore be precomputed. When aligning the images requires a large rotation, then this linearization of the rotational field becomes invalid. In such cases, we must include an additional alignment step which is accomplished in closed form via Horn's quaternion-based method[9].

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. The separation of rigid body from nonrigid deformation modes makes the warping more robust to large rotations – this is due to the use of quaternions rather than linearized rotation. This separation of rigid and nonrigid flow also allows us to examine rigid and nonrigid motions separately. For more details see [15].

4 Examples

Fig. 4(a) shows an X-ray image of a heart ventricle together with its bounding contour. Fig. 4(b) shows the first several nonrigid modes of the heart, computed using a qualitative "rubber sheet" model of the heart's elasticity. Note that it is easy to incorporate more detailed information about the physical properties of the heart if this is desired.

Fig. 5(1-9) shows a series of nine frames in which the bounding contours deform as the heart beats. Frames 1 and 5 were chosen to represent extremal views of the heart's deformation, and are used to parameterize the heart's motion.

Silhouettes were first extracted and thinned to appoximately 60 contour points from each image. Point corre-

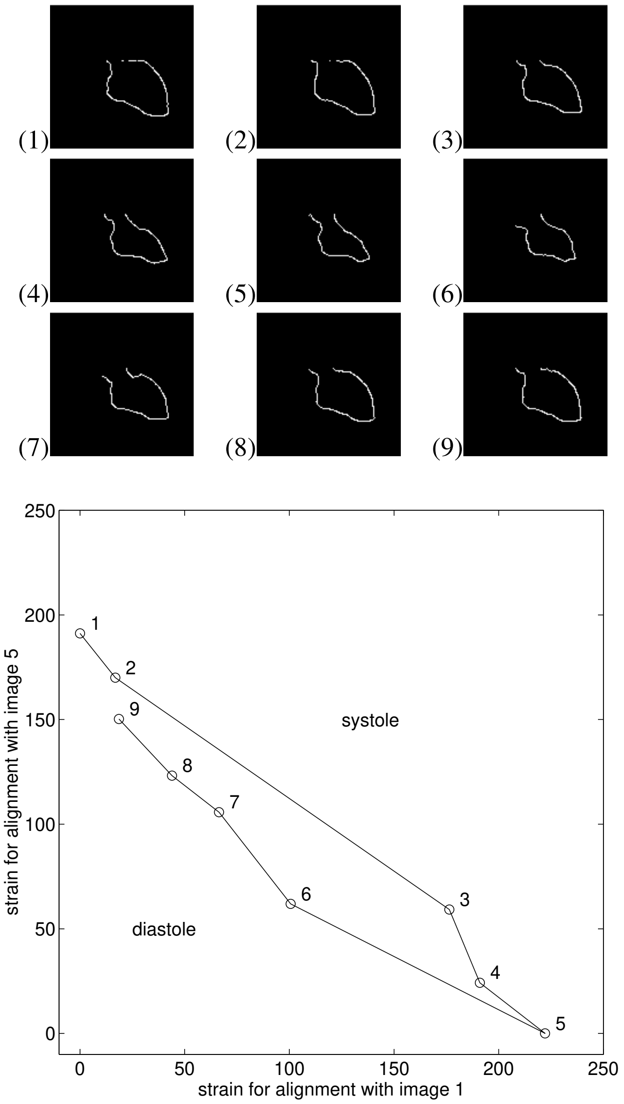


Figure 5: Representing a beating heart in terms of warps of extremal views. Given a modal model for the heart as shown in Fig. 4, we code a series of frames (1-9) in which the bounding contours deform as the heart beats. Frames 1 and 5 represent extremal views of the heart's deformation, and are used to parameterize the heart's motion. The resulting graph shows a plot of the strain energy needed to align each contour in the sequence with these extremal views. The result is a "phase portrait" in this physically-based similarity shape space.

spondence and similarity to the extremal views was then measured using the first 18 modal deformations, as described above. The computation included strain due to rotation and scaling. Total CPU time per comparison (build FEM model, match, align, and compare) was approximately 10 seconds on an HP 735 workstation. The similarities for each frame of the heart sequence are plotted in the graph shown at the bottom of Fig. 5. As can be seen, the beating of the heart forms a nice "phase portrait" in this physically-based similarity shape space. Such phase portraits can be used to analyze and recognize motion using methods described by Shavit and Jepson [18].

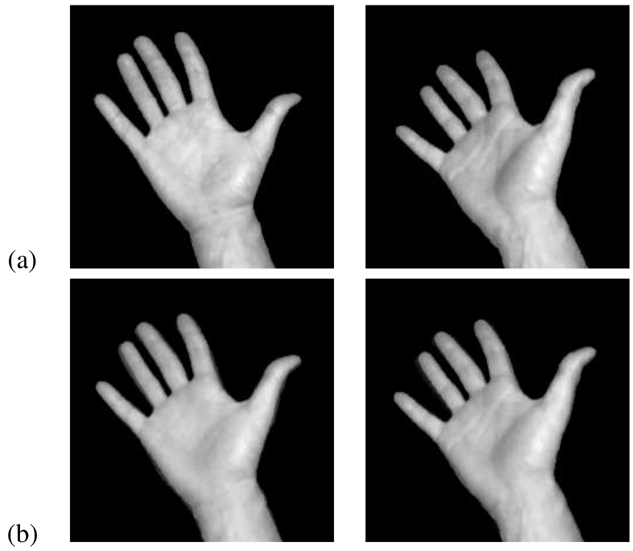


Figure 6: Given two extremal images (a) and their corresponding features, we can use modal flow fields to synthesize intermediate images (b) as linear combinations of modal-deformed versions of the extremal views.

The general methods can also be extended to model articulated motions, although for large, complex articulated motions the correspondence problem becomes too hard to solve by the method described above. Fig. 6(a) shows two extremal views of a moving, articulating hand. Correspondences between these hand images were automatically determined as described above, and intermediate images synthesized using the modal flow method. Fig. 6(b) shows two intermediate images at points between the two prototype hand shapes. These intermediate images can be directly compared to new hand images, thus allowing us to describe new hand images in terms of their similarity to these two prototype images.

Fig. 7 shows the first nine nonrigid modal warps used to deform the first extremal view in Fig. 6(a). Most of the warps seem natural (the bending of the thumb for instance); however, a few warps are inconsistent with our knowledge of human bone structure. This example pushes our current system to its representational limits: motion of articulated structures can be only roughly approximated by the deformation modes for a single isotropic sheet. However, if the hand is modeled as an articulated structure, then the resulting nonrigid modes will better capture a hand's actual modes of variation. In a future version of the system, the modal model will include articulated shapes and anisotropic materials. In addition, our system can perform physics-based active part detection along the lines of [10].

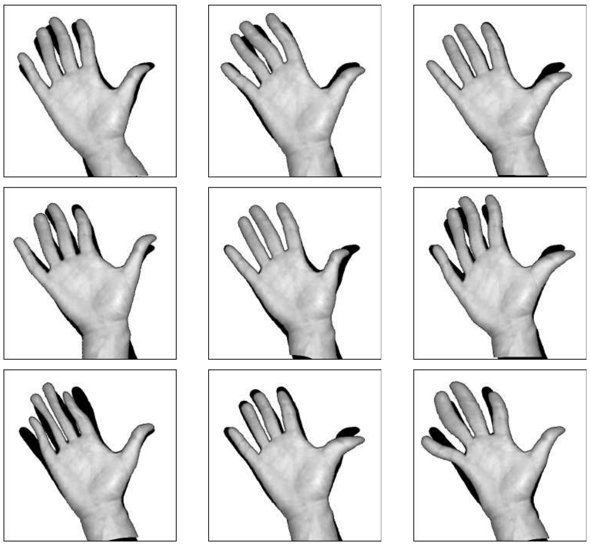


Figure 7: The first nine nonrigid modal warps for the first extremal view in Fig. 6(a). The metamorphosis between extremal views is described in terms of modal warps for an 2-D elastic hand shape using the algorithm described in the text. The warped grayscale images are drawn over the original hand image silhouette (shown in black).

5 Summary

We can obtain a parametric description of rigid, nonrigid, or articulated motion in terms of its similarity to known extremal views, thus providing us with a low-dimensional parameterization of the motion. We can derive this parameterization *without* knowing all the details of the physical system, although obviously such detailed knowledge would help in obtaining a more accurate, physically-meaningful parameterization.

Our method described starts by determining point correspondences between a new shape and known extremal views. It then interpolates between these views using a physically-based method, and finally measures the similarity between the new shape and the interpolated shape. The result is a low-dimensional parametric representation of the object's motion that is qualitatively related to the underlying physical parameters.

This approach to describing motion is related to the view-based shape recognition proposals of Ullman, *et al.* [19] and Poggio, *et al.* [13]. It is description by interpolating among examples, rather than description by some more abstract, view-independent representation.

However our method differs from theirs in that we are interested in describing motion as well as recognizing shapes, that we derive a low-dimensional parametric representation of motion that we can use to recognize and compare motion trajectories, and that we employ a physically-based framework for interpolation.

References

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

[2] A. Baumberg and D. Hogg. Learning flexible models from image sequences. In *Proc. ECCV*, May 1994.

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

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

[5] I. Cohen, N. Ayache, and P. Sulger. Tracking points on deformable objects. In *Proc. ECCV*, May 1992.

[6] 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.

[7] T. Darrell and A. Pentland. Space-time gestures. In *Proc. CVPR*, June 1993.

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

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

[10] I. Kakadiaris, D. Metaxas, R. Bajcsy. Active part-decomposition, shape and motion estimation of articulated objects: a physics-based approach. In *Proc. CVPR*, June 1994.

[11] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1:321–331, 1987.

[12] A. Pentland and S. Sclaroff. Closed-form solutions for physically-based shape modeling and recognition. *IEEE PAMI*, 13(7):715–729, July 1991.

[13] T. Poggio and F. Girosi. A theory of networks for approximation and learning. Memo No. 1140, MIT AI Lab, Cambridge, MA, July 1989.

[14] S. Sclaroff and A. Pentland. A modal framework for correspondence and recognition. In *Proc. ICCV*, May 1993.

[15] S. Sclaroff and A. Pentland. Modal Matching for Correspondence and Recognition. Vision and Modeling TR-201, MIT Media Lab, May 1993. To appear in *IEEE PAMI*.

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

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

[18] E. Shavit and A. Jepson. Motion understanding using phase portraits. In *Proc. IJCAI Looking at People Workshop*, August 1993.

[19] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE PAMI*, 13(10):992–1006, 1991.

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