Modal Matching: A Method for Describing, Comparing, and Manipulating Digital Signals

by

Stanley Edward Sclaroff

B.S., Computer Science and English Tufts University (1984)
S.M., Massachusetts Institute of Technology (1991)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1995

Author . . . .
Program in Media Arts and Sciences
January 13, 1995

Certified by . . . .
Alex P. Pentland
Associate Professor of Media Arts and Sciences
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by . . . .
Stephen A. Benton
Chairman, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Modal Matching: A Method for Describing, Comparing, and Manipulating Digital Signals

by

Stanley Edward Sclaroff

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on January 13, 1995, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis introduces *modal matching*, a physically-motivated method for establishing correspondences and computing canonical shape descriptions. The method is based on the idea of describing objects in terms of generalized symmetries, as defined by each object's eigenmodes. The resulting modal description is used for object recognition and categorization, where shape similarities are expressed as the amounts of modal deformation energy needed to align two shapes. Modal matching is also used for a physically-motivated linear-combinations-of-models paradigm, where the computer synthesizes a shape in terms of a weighted combination of modally deformed prototype shapes. In general, modes provide a global-to-local ordering of shape deformation and thus allow for selecting the types of deformations used in object alignment and comparison.

In contrast to previous techniques, which required correspondence to be computed with an initial or prototype shape, modal matching utilizes a new type of finite element formulation that allows for an object's eigenmodes to be computed directly from available shape information. This improved formulation provides greater generality and accuracy, and is applicable to data of any dimensionality. Correspondence results with 2-D contour and point feature data are shown. Recognition experiments for image databases are described, in which a user selects example images and then the computer efficiently sorts the set of images based on the similarity of their shape.

While the primary focus of this thesis is matching shapes in 2-D images, the underlying shape representation is quite general and can be applied to compare signals in other modalities or in higher dimensions, for instance in sounds or scientific measurement data.

Thesis Supervisor: Alex P. Pentland
Title: Associate Professor of Media Arts and Sciences
Program in Media Arts and Sciences

Doctoral Committee

Thesis Advisor . . . .
Alex P. Pentland
Associate Professor of Media Arts and Sciences
Program in Media Arts and Sciences

Thesis Reader . . . .
Whitman A. Richards
Professor of Psychophysics
Dept. of Brain and Cognitive Sciences

Thesis Reader . . . .
Tomaso A. Poggio
Professor of Vision Sciences and Biophysics
Dept. of Brain and Cognitive Sciences

Thesis Reader . . . .
Andrew Witkin
Professor of Computer Science
Carnegie Mellon University

Contents

List of Figures

Acknowledgments

I gratefully acknowledge my mentor and thesis advisor, Sandy Pentland, who over the last six years has been an inexhaustible source of intellectual and creative energy. I have the deepest respect for Sandy as a colleague, and have felt priviledged to share, invent, and develop research ideas with a person who has such a genuine interest and enthusiasm for collaboration. His guidance, support, vision and friendship will be a long-lasting influence in my professional and personal life.

I would like to thank the other members of my thesis committee for their guidance and feedback: Tomaso Poggio, Whitman Richards, and Andy Witkin. Their challenging questions and insightful suggestions helped to strengthen the work and helped to shape the final document.

The Vision and Modeling Group at the Media Lab has been a wonderfully diverse and dynamic research environment. I thank the faculty, Ted Adelson, Aaron Bobick, Sandy Pentland, and Roz Picard, and staff, Laureen Chapman, Judy Bornstein, and Laurie Pillsbury for cultivating such an excellent lab. Thanks also go to the graduate students (past and present) who provided a continuous supply of humor, stimulation, and technical support: Ali Azarbayejani, Lee Campbell, Bill Freeman, Martin Friedmann, Monika Gorkani, Bradley Horowitz, Fang Liu, John Maeda, John Martin, Baback Moghaddam, Chris Perry, Kris Popat, Alex Sherstinsky, Eero Simoncelli, Matt Turk, John Wang, and many others. Special thanks go to Irfan Essa for our many helpful research discussions, to Janet Cahn for her wry wit, and to Trevor Darrell for his invaluable perspective and up-to-the-minute discounted airfare information.

Heart-felt thanks to my father, my sister Sara, and my friends, Chris Bellonci, Matthew Clark, Gary Daffin, Nancy Etcoff, Chip Gidney, Brian Kelley, Steven Wilson, and the many others who gave moral support throughout my time at MIT. I would especially like to acknowledge Rolando Niella, Reid Orvedahl, and Jonathan Worth, close friends who have given me unflagging love and encouragement during this six year adventure.

This research was conducted at the MIT Media Laboratory and was sponsored in part by British Telecom and by the Office of Naval Research (ONR) under contract No. DAAL01-93-K-0115. The opinions expressed here are those of the author and do not necessarily represent those of the sponsors.

In loving memory of my grandfather
Ed Hahn, 1913 — 1990.

Chapter 1

Introduction

Multimedia databases are an increasingly important component of computer and telecommunications usage. The social, educational, and commercial usefulness of such databases could be enormous. Yet progress towards the dream of large, commercially-viable multimedia databases is stymied by the lack of a mature and reliable technology for organizing and searching these digital signals based on their content. To humans these databases contain movies, news, speech, graphics, music, medical or other scientific data, text, etc. — things that have semantic content; while, to a computer these databases contain merely bits — vast heterogeneous collections of digital signals.

For a text database, we can avoid this problem because inquiries can be based on the database elements themselves: strings of characters. Questions about database content can be answered by simply comparing sets of ASCII strings. Because this search is efficient, users can search for their answers at query time rather than having to pre-annotate everything. To accomplish the same thing for multimedia databases we need methods that allow efficient comparison of the signals themselves in order to see if they have the same (or more generally, similar) content.

For a video database, the current approach is for a human to enter textual descriptions of each image's content and then enter these annotations into a standard text database. The signals themselves are not really part of the database; they are only referenced by text strings or pointers. The problem with this approach is that in most images there are literally hundreds of objects that could be referenced, and each imaged object has a long list of attributes. Even worse, spatial relationships are important in understanding image content, and if we must also consider relations among images, then the problem quickly becomes intractable. The old saying "a picture is worth a thousand words" is an understatement.

Researchers have recently made progress in automatic indexing for image databases. For instance, they have proposed a variety of automatic image indexing methods, based on shape [26, 55, 56, 79], color [13, 54, 133], or combinations of such indices [85]. The general approach is to calculate some approximately invariant statistic, like a color histogram or invariants of shape moments, and use these to stratify the image database. Unfortunately, significant semantic information may get lost in this process. For instance, do we really want our database to think that apples, Ferrarris, and tongues are "the same" just because they have the same color histogram? Indexing gives a way to limit search space, but does not answer "looks like" questions except in constrained datasets.

1.1 Approach

One key in bridging this "communications gap" is more effective models of a human's sense of what is similar and what is not. Given digital signals, we need decompositions that preserve perceptual similarities while providing an efficient encoding of the signal. In particular, these representations should employ decompositions that preserve semantically meaningful and perceptually important information about the original digital signals [94, 95]. It can be argued that we need an arsenal of specially trained decompositions, each with appropriate "control knobs" for describing a particular type of object or context (for instance the Karhunen-Loève transform for grayscale images [144] or the Wold decomposition for textures [100]). The goal of this thesis will be to develop a new method for semantics-preserving matching, encoding, and comparing signal shape.

Shape matching is complicated by the fact that humans will report that shapes are "similar" even when the two shapes are actually deformed versions of each other. Therefore, to measure the shape similarities between two objects, we must be able to describe the deformations that relate them. The shape decomposition should provide deformation "control knobs" that roughly correspond to a human user's notions of perceptual similarity.

As part of this, the decomposition should allow for selecting weighted subsets of shape parameters that are deemed significant for a particular category or context. Thus the computer, like the human, can filter out deformations which may be inconsequential in a particular shape class or context.

For this thesis I developed a shape understanding method that allows users to select examples, and has the computer carry out a search for similar objects in the database, based on the similarity of the signals' shapes. In machine vision this is known as taking a *view-based* approach. From an implementation standpoint, this approach offers the advantage of representing 3-D objects without having to construct detailed 3-D models. From a user-interface standpoint, a view-based approach is particularly appealing because it offers an intuitive, viewer-centered paradigm for guiding a database search: users can easily pick out examples from sub-images, or construct a search based on an image composite. It is also straight-forward for users to specify analogy-based searches like "A is to B as C is to?" or employ negative examples like "find me more of things like A,B but not like C,D."

The key is being able to efficiently encode shape and deformation. In images, deformation differences between objects of the same type are sometimes due to changes in viewing geometry: e.g., foreshortening or distance change. Other times they are due to physical deformation: one object is a [stretched, bent, tapered, dented, ...] version of the other. For instance, most biological objects are flexible and articulated. More generally, real-world signal shape deformations are the result of physical processes, and so it seems logical that the underlying shape representation must somehow embody (approximately) the physics of objects in the world.

This rationale led Terzopolous, Witkin, and Kass [141], and many others in the machine vision community [18, 27, 33, 38, 89, 90, 96, 131] to focus on the description of data using physics-based deformable models. The power of these techniques stems not only from their ability to exploit knowledge about physics to disambiguate visual input, but also from their capacity to interpolate and smooth raw data, based on prior assumptions about noise, material properties, etc. Unfortunately, deformable models do not by themselves provide a method for computing reliable canonical shape descriptions, or for establishing correspondence between shapes.

To address the problem of obtaining canonical descriptions, Pentland and Sclaroff developed a method in which all shapes are represented as modal deformations from some prototype object [90, 96]. By describing deformation in terms of the eigenvectors of the prototype object's stiffness matrix, they 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, they developed robust methods for 3-D shape modeling, object recognition, and 3-D tracking utilizing point, contour, range, and optical flow data [91].

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

Modal matching, the new formulation described in this thesis, generalizes these earlier approaches by obtaining the modal shape invariants directly from the sensor data. This will allow us to compute robust, canonical descriptions for recognition *and* to solve correspondence problems for data of any dimensionality. A detailed mathematical formulation for 2-D problems is given, and method is tested on the shapes found in digital grayscale images, image contour data, and graphics. For the sake of this thesis, a shape will be defined as: a cloud of feature locations, a region of support that tells us where the shape is, and the original digital image (grayscale or color).

Modal matching is a semantics-preserving shape-encoding scheme. It decomposes shape into a description of how to build the shape from deformations, and corresponds qualitatively to shape similarity experiments on humans by Gestalt psychologists [3, 59] and others [12, 51, 68, 72]. In the modal representation shape is thought of in terms of an ordered set of deformations from an initial shape: starting with bends, tapers, shears, and moving up towards higher-frequency wiggles. The resulting mathematical representation (modes) yields a shape description that corresponds neatly with words in English; this is

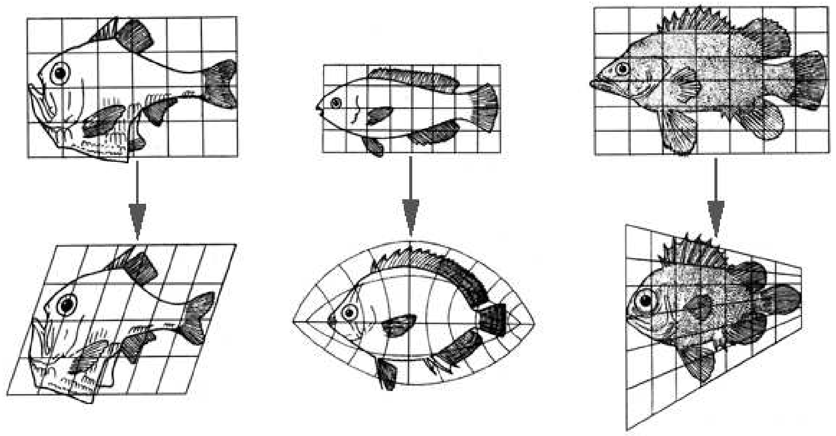


Figure 1-1: Fish morphometry: an example of biological shape deformation, adapted from D'Arcy Thompson [142]. Nonrigid deformation can be important in describing variations across species.

evidence that we are on the right track[90, 96].

Finally, the modal representation is supported by the theories put forth by biologists studying the morphometry of animal bones and shape [142, 19], in which the shape of different species is related by deformations. Figure 1-1 (adapted from [142]) demonstrates the importance of deformation in relating different species of fish.

Figure 1-2 shows some examples of the type of shapes (and deformations) the system will be expected to match and compare in 2-D imagery. The first set of shapes comes from a database of hand tool images. This database currently contains on the order of 60 views of 12 hand tools, where images were collected while varying 3-D orientation and lighting, and non-rigid deformation. Note that the last two tools are nonrigidly deformed. Figure 1-2(b) shows a second set of images, this time silhouette-based: outlines of airplanes for different plane types, under varying 3-D orientation. The third set of shapes (Figure 1-2(c)) comes from a database of children's field guide illustrations. The database contains over 100 views of different animal species, under varying orientation, scale, and allowing for small articulated motions. Given a user-selected view, the system will find similar animals, based on deformations from the example view.

While the primary focus of this research will be matching shapes in 2-D images, the underlying shape representation is quite general and can be applied to compare signals

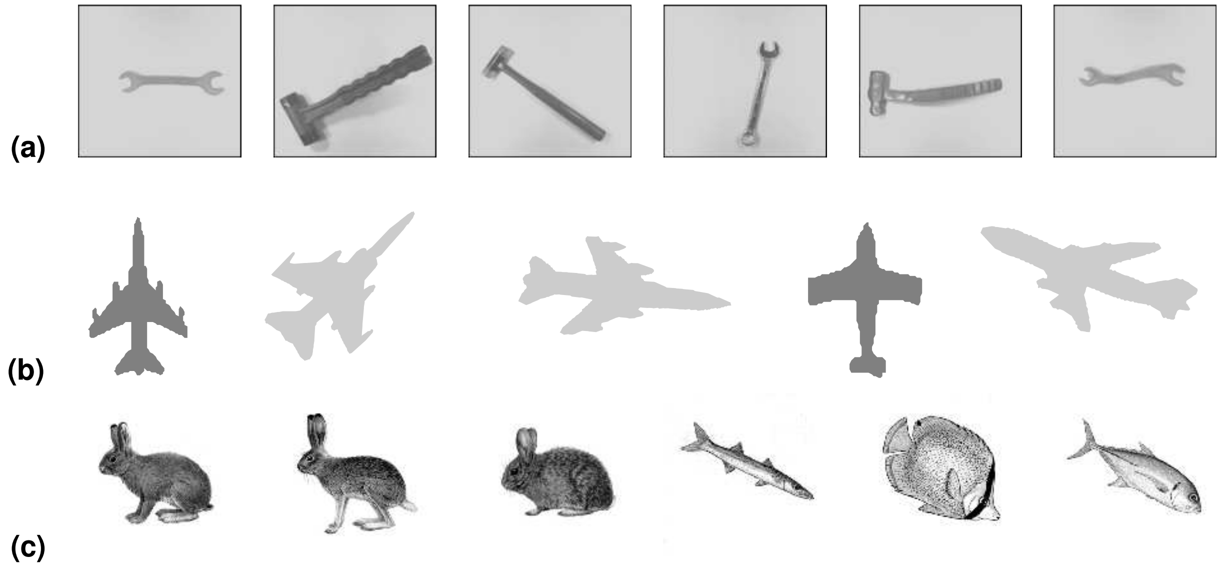


Figure 1-2: Examples of three types of deformed shapes that the system will match and compare. The images of hand tools (a) come from a database of 60 views of 12 hand tools, where images were collected while varying 3-D orientation, lighting, and non-rigid deformation (note that the last two tools are nonrigidly deformed). The airplane silhouettes (b) come from a collection of views generated for different plane types, under varying 3-D orientations. The images of animals (c) were extracted from a database of over 100 field guide illustrations. Given a user-selected view, the system will find similar animals, based on deformation from the example view.

in other modalities or in higher dimensions, including sounds or scientific measurement data. In addition, the results of this thesis have useful applications in other areas: the "sensor fusion" problems found in aligning medical image data, improved techniques for image metamorphosis, and new methods for data compression.

The modal matching technique offers three clear advantages over previous techniques. First, it can be used to automatically identify and label corresponding features on two shapes so that the alignment, comparison, and morphing can take place without human operator intervention. Second, the underlying deformation representation allows for separating out different types of deformation (i.e., rotation, scaling, bending, etc.). Lastly, the deformation parameters correspond qualitatively with those reportedly used by humans in describing shape similarity and thus provide a meaningful interface for animation and for database inquiry.

1.2 Thesis Overview

Besides this introductory chapter and a brief concluding chapter, there are seven other chapters in this thesis.

Chapter 2 gives a review and analysis of the previous work done in view-based systems for machine vision. It touches on four main areas: linear combinations of models, object-centered coordinate frames, deformable models, and eigen-representations for images and shapes. There is a summary of the state of the art given at chapter end.

Chapter 3 first gives a top-level overview of our approach, and then gives the mathematical details of our new finite element model. This new formulation utilizes Gaussian interpolants to avoid previous problems with sampling and model parameterization. Formulations for 2-D finite elements are provided.

The next three chapters give expanded formulation details for the main system components: modal matching, modal descriptions, and modal combinations of models.

Chapter 4 describes modal matching, the new method for determining corresponding features on similar shapes. Modal matching utilizes a shape-intrinsic coordinate system for robust correspondence computation. This intrinsic coordinate system is built from the eigenvectors of the new physical shape model formulated in Chapter 3, is invariant to rotation, scaling, and translation, and is robust to nonrigid deformation and noise.

The next chapter, **Chapter 5**, first formulates a number of different methods for recovering modal shape descriptions. It then details how the resulting modal descriptions can be utilized for recognition, category representation, and similarity judgment. The underlying modal model provides a global-to-local ordering of shape deformation. This makes it easy to select which types of deformations are to be compared. Finally, various energy-based measures for shape similarity are proposed.

Chapter 6 describes how the core technology of modal matching can be used to obtain a new physically-based framework for linear-combinations-of-views. Our modal method offers the advantage of employing an ordered description of shape. As a result, we will be able to analyze and decompose shape deformations (and then resynthesize shapes) in a flexible, principled manner. We then formulate algorithms for describing categories of objects in terms of linear combinations of modally-warped examples, and for characterizing non-rigid motions in terms of warps from extremal views.

Chapter 7 details and evaluates experimental results. Modal matching is first applied to some classic 2-D shape correspondence problems, and then performance is demonstrated on real image data. Experiments in shape alignment and shape comparison are then described. The modal parameters make it possible to pin-point the types of deformation needed to align shapes, thereby making it possible to determine relationships between objects and categories. These ideas are then extended to solve problems in shape-based image database search and shape categorization. Finally, the concept of physically-based linear-combinations-of-views is tested on problems in nonrigid and articulated motion description, and image metamorphosis for computer graphics.

The results of experiments described in the previous chapter reveal some of the strengths and limitations of the modal matching method. **Chapter 8** discusses these limitations, and then proposes some interesting directions for future research.

Finally, there are two appendices. The first appendix describes the extension of the finite element formulation to three dimensions and higher. The second appendix details how modal matching and morphing could be applied in another signal domain: sound.

This thesis presents some work that also appears in journals and conference proceedings. Chapters 3, 4 and 5 are based in part on [117, 118, 121]. Part of Chapter 6 appears in [119], and some of the image database results of Chapter 7 appear in [120].

Chapter 2

Background

The method developed in this thesis employs a deformable shape decomposition that allows users to specify a few example objects and have the computer efficiently sort the set of objects based on the similarity of their shape. Shapes are compared in terms of the types of nonrigid deformations (differences) that relate them to examples. This chapter will provide a review and analysis of the previous work done in example-based (view-based) systems for machine vision. Though the simplicity of an example-based representation is alluring, there are some problems in actually using such a scheme for recognizing deformable shapes. In particular, there are four main problem areas:

Data reduction View-based approaches are inefficient in terms of storage: to exhaustively collect enough views to represent an object requires too many images. This is especially true if we allow for nonrigid and articulated objects: not only do we need to account for all possible views, but also for all valid configurations/deformations. Many of these views are very similar and carry redundant information. We can take advantage of this redundancy by storing only the *characteristic views*, views in which things change significantly. To generalize this approach Poggio and Edelman [102] and Ullman and Basri [147] utilized an effective method for data reduction: a **linear combinations of models** paradigm, where any view can be synthesized as a linearly-warped combination of example views.

Correspondence To match views to images, we need to reliably extract and match features. One way to solve this problem is to take an alignment approach: we try to subject the first view to all possible scalings, translations and rotations in an attempt to get the first view to closely overlay the second. This makes shape comparison extremely inefficient, since we have to first deal with the potentially combinatoric complexity of aligning models. A second class of methods employs local feature matching; these methods can be unreliable in the presence of deformation and noise. The more promising solutions to the correspondence problem take the form of **body-centered coordinate frames**, coordinate systems that are intrinsic to the shape.

Nonrigid deformation Some inherent problems with correspondence schemes come from the fact that often nonrigid deformations can only be modeled as noise. To better deal with this nonrigidity, we can employ **deformable models**, essentially embedding the data in a sheet that can be optimally warped and stretched to match other signals. One of the first such schemes was a dynamic programming approach called *elastic matching*. Unfortunately, elastic matching has traditionally been computationally slow, problematic with regard to correspondence, and prone to noise stability problems. In light of these limitations, researchers introduced *physics-based models*, which actually simulate the physics of a rubber sheet for better deformation prediction and stability to noise.

Canonical description Because they are mesh-based, most deformable modeling techniques suffer from an inability to obtain unique descriptions. This is because (in general) the parameters for these surfaces can be arbitrarily defined, and are therefore not invariant to changes in viewpoint, occlusion, or nonrigid deformations. To obtain canonical descriptions, we can employ **eigen-representations** like the *modal representations* or any of a family methods descended from the Karhunen-Loève transform. In these representations, we can discard the high-eigenvalued components in order to obtain overconstrained, canonical descriptions of the equilibrium solution. This basis truncation is also useful for data reduction, since it can be used to optimally span a set of views.

The rest of this chapter will give a more in-depth analysis of previous work in the four areas: linear-combinations of models paradigms, object-centered coordinate frames, deformable models, and eigen-representations for shapes and images. Since the primary application area of this thesis is image and video, my main emphasis will be on previous image-based approaches and other related work in machine vision. However, it will become clear that many of these concepts are easily generalized to signals in other modalities.

2.1 Linear Combinations of Models

Representing an object in terms of its aspects or *characteristic views* is a familiar idea in the vision community [42, 71, 73, 65]. The underlying concept is powerful and simple: imagine that we walk around an object and store views of the object in our memory as we go. In our memory, an object can be represented in terms of a few critical views, and all views in between can be generated as simple interpolations. This pictorial approach is powerful because it can be used to model a three dimensional object without the use of a three dimensional model.

Interestingly, the view-based approach offers a mixed blessing. On the one hand, the approach offers the advantage that we need not have a complete model of the object (we can fill it in with more critical views as we go); on the other hand, the approach has the disadvantage that we cannot (in general) guarantee that we have a complete collection of critical views. Finally, this approach suffers from problems with alignment and nonrigidity. But despite these problems, the technique has been shown to be quite useful.

Ullman and Basri extended this idea, and suggested that a model can be represented in terms of a linear combination of 2-D images [147]. They showed that this technique works over changes in viewpoint, and claim (believably) that it can also represent nonrigid deformations, and changes in lighting. Poggio and others have used this linear combinations of views paradigm and applied it to the problem of teaching a neural net to recognize 3-D objects from 2-D views [34, 102] and to the problem of recovering projective structure [128]. The linear combinations of views method has the problem that — though feature positions can be accurately interpolated from view to view — the interpolation between points is piece-wise affine, and can therefore provide only a first order approximation in modeling the nonrigid warping of the underlying image pixels, nor can it handle large rotations.

This technique is also useful for generating novel views of the object. Given corresponding points in a sufficient number of known images (sufficient to span the space of all possible images of an object — for 3-D rotations, this is six) we can synthesize a new image from any viewpoint in terms of a weighted and warped combination of these known images. To recognize an object from a 2-D image, we first attempt to reconstruct it in terms of a linear combination, and then measure similarity using a metric that has two terms: one dependent on the amount of deformation/transformation needed to align the features, and the other dependent on the pixel-by-pixel difference between the original and synthesized grayscale or color images.

2.1.1 View interpolation for image synthesis

To interpolate between views, Poggio and others utilize a network of radial basis functions (RBFs) [102, 105]. These functions are combined to interpolate a hyper-dimensional spline surface that smoothly approximates the image-space trajectories for feature points across 2-D views [103, 101]. These point trajectories can then be used to smoothly deform the underlying images. This technique of view interpolation has been applied in the area of modeling human facial expression by Beymer *et al.* [11]. Beymer's algorithm uses dense flow to describe how to deform from one expression to another, and each expression's dense flow field is an "example" in the RBF interpolation framework. Flow fields derived from one person's expressions can be used to drive the deformation of another person's face.

View interpolation can also be useful in computer graphics where an animator wants to be able to specify a few keyframes, and have the in-between frames interpolated. Librande has successfully used the RBF interpolation framework for computing in-between images [69] for 2-D character animation. Image interpolation can be used to transform one object into an another object, this is known in the computer graphics and movie-making communities as *morphing* [10, 153, 154]. There has also been interest in view-based image coding, where interpolation can be used to generate new views of an object. A system for view interpolation for image synthesis has been developed by Chen and Williams [25]; in this system, a computer renders a few expensive, high-quality images of synthetic computer scenes, and in-between frames are interpolated cheaply based on precomputed correspondence map and depth information.

Each of these view interpolation schemes assumes that corresponding image features are known ahead of time. Most of these schemes do include a simple feature matching algorithm, but correspondence computation is not integrated with the underlying representation. Automatic correspondence computation is an important part of many other machine vision algorithms — matching stereo image pairs, computing structure from motion, and aligning models with images, for instance. Unfortunately, these matching schemes exploit some strong (and application limiting) constraints: rigidity, small deformation, small camera rotations, *etc.*

In the computer graphics community, corresponding features are most often laboriously determined by an artist — though there has been some recent work by Sederberg *et al.* [124, 123] to automate correspondence computation, these methods only work for objects that are defined in terms of closed two-dimensional contours. Thus, automatically determining correspondences remains a crucial unsolved problem in view-based approaches.

2.2 Object-Centered Coordinate Frames

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

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

Many methods for finding a body-centered frame have been suggested, including moment-of-inertia methods [7], polar Fourier descriptors [99], strip trees [6], the generalized Hough transform [5], codons [110], curvature scale space matching [66, 80, 152], and axial shape descriptors [16, 17, 21, 23, 58, 112]. These methods generally suffer from three difficulties: sampling error, parameterization error, and non-uniqueness. One of the main contributions of this thesis is a new method for computation of a local coordinate frame that largely avoids these three difficulties.

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

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

Non-uniqueness is an obvious problem for recognition and matching, but one that is all too often ignored in the rush to get *some* sort of stable description. Virtually all spline, thin-plate, and polynomial methods suffer from this inability to obtain canonical descriptions; this problem is due to the 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.

2.3 Deformable Models

As can be seen, there are inherent problems with finding corresponding points in different views of an object. These problems often stem from the fact that perspective and nonrigid deformations can only be modeled as noise. However, these types of deformation are commonplace in the real world: plants sway in the breeze, arms and legs bend, cheeks bulge, *etc*. Clearly, shape deformations are not noise; deformation is constrained by and actually conveys information about the physical nature of the objects we see. To better deal with this nonrigidity, we can employ a deformable model, essentially embedding our data in a sheet that can optimally warp and stretch to match other signals.

Over the years, there have been a number of methods suggested for matching deformed signals. Some of the first schemes were grouped into the general category of *elastic matching* [22, 24, 41, 81, 107], which is essentially an image-based version of *dynamic time warping* [114]. Elastic matching utilizes a dynamic programming approach to solve for constrained warps of the signal shape. The resulting search algorithm minimizes template distortion while maximizing the match with an object in the image.

In their seminal paper, Fischler and Eschlager [41] described a system that built up subtemplates that corresponded to significant features, and then searched for a match using a two step process: 1) find subtemplate matches, and then 2) find match configurations that satisfy relational constraints. Burr [24] and Broit [22] later used elastic matching to optimally align deformed images, and line drawings. This warping method for matching has also been used for matching stereo image pairs [81, 107]. In some cases, the warps for elastic matching can be controlled using parametric control knobs (for instance, parameters that predict the geometric distortion due to changes in camera position) as was discussed by Barrow *et al.* [8]. In this method, image matching is performed by searching over the model parameters to find the warp that best matches the data.

Elastic matching has been applied to signal matching problems in other domains. Reiner *et al.* used it in gas chromatography to identify microorganisms [109] and tissue samples [108]; they used time-warping to correct for nonlinear distortions of the time axis. Widrow [151] employed "rubber masks" to elastically match chromosome data, and to match EKG data. Elastic matching has also been applied to the problem of on-line handwriting recognition (for a survey see [135]).

Unfortunately, elastic matching has traditionally been computationally slow, has problems with correspondence, and has noise stability problems. Most of the methods were not completely automatic, and assumed that features were selected and matched as a separate, preprocessing step. While the dynamic programming model tried to capture deformation, it did not model the underlying physics of deformation. Given an object's physical properties, we would like to utilize this knowledge to parameterize our shape representation in such a way as to capture physically-allowable shape variations, and thereby model the variation over a category of shapes.

2.3.1 Physics-based models

The notion of employing physical constraints in shape modeling has been suggested by many authors [67, 51, 89]; however, the seminal paper by Terzopoulos, Witkin, and Kass[141], who obtained 3-D models by fitting simulated rubbery sheets and tubes, has focused attention on modeling methods that draw on the mathematics used for simulating the dynamics of real objects. One motivation for using such physically-based representations is that vision is often concerned with estimating changes in position, orientation, and shape, quantities that physical models accurately describe. Another motivation is that 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.

Kass, Witkin, and Terzopoulos also introduced this physical modeling paradigm in the area of contour tracking via *snakes* [61, 138]. Snakes are simulated strings of a rubbery material that can be placed near a contour in an image, and then pulled and pushed closer to the contour (or valley) via artificial forces. Typically these forces are computed directly from the image intensity. Once settled into place, snakes can be used to establish correspondence between contours in different images, they can also be used to track deforming shapes over a moving image sequence.

This initial concept of snakes has lead to the development of specialized deformable templates for finding and recognizing faces [70, 156], for tracking hand gestures [14], and for elastically matching medical data [4, 33, 131]. The template for a face consists of an interlocked set of snakes that look for particular facial features: eyes, mouth, nose, *etc.* Others have sought more general ways to incorporate prior knowledge about the particular object being searched for, and have derived the physics of their templates directly from a training set [2, 30, 74, 75, 122, 127]. I will review some of these statistical techniques in more depth in Section 2.4.

While this initial family of physically-based shape representations enabled vision researchers to better model nonrigid shape deformation, there was the drawback that these representations relied on the mathematical technique of *finite differences* for discretizing and numerically integrating the underlying physical equations. The use of finite differences leads to severe problems with sampling, since the method cannot allow for spring attachment points other than at the discretization nodes. This sampling problem can be alleviated somewhat by increasing the number of discretization points, but the computation time needed is multiplied by the requirement for a regularly sampled discretization grid, and by the need for proportionally smaller time steps in time integration. Terzopoulos suggested the use of multigrid methods [137] as a way to make the computation more tractable at increased resolutions, but this did not alter the requirement that springs have to be attached at the discretization nodes.

2.3.2 Finite element methods

To better deal with sampling and scale problems, we can utilize the finite element method (FEM) for physical modeling. This approach was employed in modeling deformable solid shapes (superquadrics) by Pentland and Sclaroff [90, 96] and later by Terzopoulos and Metaxas [140]. Cohen and Cohen have worked out a finite element formulation for snakes [28]. This lead to modeling shape in terms of 3-D deformable balloons that can inflate to fit medical and other three-dimensional data sets [27, 78], and to rubber blob models for tracking moving objects in 2-D images [60].

In the FEM, interpolation functions are developed that allow continuous material properties, such as mass and stiffness, to be integrated across the region of interest. Note that this is quite different from the finite difference schemes just described — as is explained in [9] and [96] — although the resulting equations are quite similar. One important difference between the FEM and the finite difference schemes is that the FEM provides an analytic characterization of the surface between nodes or pixels, whereas finite difference methods do not. Another important difference is that the FEM can guarantee convergence on a solution [9, 86], whereas the finite difference methods cannot.

Unfortunately — whether we use finite differences or finite elements — these physically-motivated representations cannot be directly used for comparing objects. Virtually all spline, thin-plate, and polynomial methods suffer from an inability to obtain canonical descriptions; this problem is due to the 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. For any mesh-based representation, the only general method for determining if two surfaces are equivalent is to generate a number of sample points at corresponding positions on the two surfaces, and observe the distances between the two sets of sample points. Not only is this a clumsy and costly way to determine if two surfaces are equivalent, but when the two surfaces have very different parameterizations it can also be quite difficult to generate sample points at "corresponding locations" on the two surfaces.

2.3.3 Modal decomposition

Pentland and Sclaroff addressed this problem of non-uniqueness by adopting an approach based on *modal analysis* [96]. In modal analysis, the standard FEM computations are simplified by posing the dynamic equations in terms of the equations' eigenvectors. These eigenvectors are known as the object's *deformation modes*, and together form a frequency-ordered orthonormal basis set for shape. Pentland and his colleagues subsequently applied modal analysis to problems in shape recovery and recognition [97, 90, 96], and for nonrigid motion tracking [92].

Bookstein also employed a similar eigenmethod to model the biological shape changes found in medical landmark data[18]. He computed a set of *principal warps* that served as an orthogonal basis transform for regularization — this is essentially the modal representation for thin-plate splines. Bookstein's method has the problem that it assumes that corresponding landmarks are located beforehand.

In general, the modal representation has the advantage that it decouples the degrees of freedom within the non-rigid dynamic system. Decoupling the degrees of freedom yields substantial advantages — the most important of these being that the fitting problem has a simple, efficient, closed-form solution. The modal transform does not by itself reduce the total number of degrees of freedom, and thus a complete modal representation suffers from the same non-uniqueness problems as all of the other representations.

The solution to the problem of non-uniqueness is to discard enough of the high-frequency modes that we can obtain an overconstrained estimate of shape. Use of a reduced-basis modal representation also results in a *unique* representation of shape because the modes (eigenvectors) form a frequency-ordered orthonormal basis set similar to a 3-D Fourier decomposition. Just as with the Fourier decomposition, reducing the number of sample points does not change the low-frequency components, assuming regular subsampling. Similarly, local sampling and measurement noise primarily affect the high-frequency modes rather than the low-frequency modes. The result is a parametric deformable model, where the parameters are physically meaningful, and correspond to how much bending, tapering, pinching, etc., is needed to represent an object. Thus the solution is well-suited for recognition and database tasks.

The modes used for describing object deformation are computed by solving an eigenvalue problem on a large matrix; this means that modal analysis can have the disadvantage that the basis can be prohibitively expensive to compute on the fly. It has been noted that the modes for a particular class of similar shapes can be precomputed and generalized [90, 96]. In some cases — topological tubes and spheres — Nastar has shown that the deformation modes can be found analytically [84].

Employing the modal representation provides a computationally convenient way to get a parametric finite element model, while also solving the problems of sampling and non-uniqueness; however, another nettlesome problem remains unsolved. In each of these physics-based methods, attaching a virtual spring between a data measurement and a deformable object implicitly specifies a correspondence between some of the model's nodes and the sensor measurements. In most situations this correspondence is not given, and so must be determined in parallel with the equilibrium solution. Sometimes, for the method to be reliable, the user must hand place an initial deformable model in the image at the appropriate size and orientation. The attachment problem is the most problematic aspect of the physically-based modeling paradigm. This is not surprising, however, as the attachment problem is similar to the correspondence problems found in many other vision applications.

2.4 Eigen-representations

There is a class of eigenmethods that derive their parameterization directly from the shape data, and thus try to avoid this spring attachment problem. Some of these techniques also attempt to explicitly and automatically compute correspondences between sets of feature points, while others avoid specifying feature correspondences at all, matching images using a more global or *Gestalt* approach. Like modal analysis, each of these eigenmethods decomposes shape deformation into an ordered, orthogonal basis. These methods fall roughly into three categories: *eigenshapes*, *eigenwarps*, and *eigenpictures*.

2.4.1 Eigenshapes

The idea behind eigenshapes is simple: 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, and the resulting eigenvectors can be used to describe the most significant deformations for that class of objects. In some ways, these principle deformations are akin to an object's generalized (nonlinear) axes of symmetry, in that they describe the principle axes of deformation over a training set of shapes.

One such matrix, the *proximity matrix*, is closely related to classic potential theory and describes Gaussian-weighted distances between point data. Scott and Longuet-Higgins [122] showed that the eigenvectors of this matrix can be used to determine correspondences between two sets of points. This coordinate system is invariant to rotation, and somewhat robust to small deformations and noise. A substantially improved version of this approach was developed by Shapiro and Brady [126, 127]. Similar methods have been applied to the problem of weighted graph matching by Umeyama [148], and for Gestalt-like clustering of dot stimuli by van Oeffelen and Vos [149]. Unfortunately, proximity methods are not information preserving, and therefore cannot be used to interpolate intermediate deformations or to obtain canonical descriptions for recognition.

In a different approach, Samal and Iyengar [113] enhanced the generalized Hough transform (GHT) by computing the Karhunen-Loève transform for a set of binary edge images for a general population of shapes in the same family. The family of shapes is then represented by its significant eigenshapes, and a reference table is built and used for a Hough-like shape detection algorithm. This makes it possible for the GHT to represent a somewhat wider variation (deformation) in shapes, but as with the GHT, their technique cannot deal very well with rotations, and it has the disadvantage that it computes the eigenshapes from binary edge data.

2.4.2 Eigenwarps

Cootes, Taylor, *et al.* [30, 31] introduced a chord-based method for capturing the invariant properties of a class of shapes, based on the idea of finding the principal variations of a contour model (snake). Their method relies on representing objects as sets of labeled points, and examines the statistics of the variation over the training set. A covariance matrix is built that describes the displacement of model points along chords from the prototype's centroid. A principal components analysis is computed for this covariance matrix, and then a few of the most significant components are used as deformation control knobs for the snake.

This technique has the advantage that it can be trained to capture information along the most important axes of variation for the training set (and thus for the expected variation in new models it encounters within the same object class). Furthermore, the eigenvalues give a principled way to put bounds on the amount of variation allowed (Cootes and Taylor limit this to one standard of deviation). Unfortunately, this method has three disadvantages: it relies on the consistent sampling and labeling of point features across the entire training set, it is edge-based, and it cannot handle large rotations.

This technique is based directly on the sampled feature points. When different feature points are present in different views, or if there are different sampling densities in different views, then the shape matrix for the two views will differ even if the object's pose and shape are identical. In addition, this method cannot incorporate information about feature connectivity or distinctiveness; data are treated as clouds of identical points. Most importantly, this approach cannot handle large deformations unless feature correspondences are given.

2.4.3 Eigenpictures

Another group of eigen-description methods is known as *eigenpictures*, because these methods compute the principal components directly on the grayscale images themselves. Using these approaches researchers have built systems that perform stable, interactive-time recognition of faces [145], cars [83], and allowed interactive-time tracking of cars and roads [143].

One such eigenpicture technique is a particularly familiar one in the Media Lab: eigenfaces [64, 144, 145]. The basic idea is simple: by finding the principal variations over a large training set of grayscale face images, we can use a small number of these orthogonal *eigenfaces* to describe novel faces. To recognize a new face, we first recover a recipe that tells us how to synthesize this face as a weighted combination eigenfaces. We can then recognize the novel face by comparing the new face's recipe against recipes for known faces.

In the limit, the eigenface decomposition can contain infinitely many base-faces, but by truncating to only the first 10-20 eigenfaces, we can usually account for over 99 percent of the variance over the training set. Thus, the description of faces can be achieved with only a few parameters. Furthermore, this transform into the truncated eigenspace describes a subspace called *face space*. This fact can be used for face detection: if a subimage does not project correctly into this face image subspace, then it is not considered to be a face [145]. Another strength of the eigenpicture method is its robustness to occlusion and noise.

As originally proposed, the technique has problems in dealing with changes in scale, orientation, and lighting. Problems in modeling lighting changes for faces are being carefully addressed by Hallinan in his Ph.D. work at Harvard [50]. In addition, Pentland *et al.* have extended the eigenface method to incorporate *modular eigenspaces* that can incorporate information about change in viewpoint, and *eigentemplates* to better model salient facial features like the eyes, nose and mouth [93]. Murase and Nayar [83] have applied the eigenpicture technique to the more general problem of representing aspect graphs of any object. The aspect graphs can be used to capture both changes in viewpoint around an object and changes in lighting. However, despite these improvements, eigenpictures still cannot adequately cope with scaling, rotation, nor deformation.

2.5 Summary

All shapes can be represented as deformations from a standard or prototypical shape; this basic premise has served as inspiration for many of the view-based shape representations for machine vision. One key issue in using a view-based approach is that of *data reduction*: given the multitude of possible viewpoints and configurations for an object, we need to reduce this multitude down to a more efficient representation that requires only a few *characteristic views*. This general approach is embodied in the linear-combinations-of-views paradigm, where any object view can be synthesized as a combination of linearly-warped example views.

A key issue in this framework is that of *correspondence*: how to describe features, contours, and surfaces so that they can be recognized and matched from view to view. One promising family of correspondence methods first computes a shape-intrinsic coordinate frame for each object; correspondences are then computed in this alternate *object-centered* coordinate frame. Unfortunately, most object-centered methods are sensitive to parameterization, sampling, and noise. These problems are further complicated because real-world objects appear differently depending upon sensor placement and type; also, real objects can deform and move.

In order to better model how objects appear in sensor data we must account, at least qualitatively, for their physics. This has motivated the use of physically-motivated, deformable models to exploit *a priori* knowledge about physics to interpolate and smooth raw data, based on prior assumptions about noise, material properties, *etc.* Using a physically-based shape representation allows us to better model nonrigid shape deformation, but the initial formulation had the problem that it utilized a *finite differences*. This lead to problems with sampling. To gain better robustness to sampling, we can employ the finite element method for building our physical models.

Physically-motivated representations are quite good for modeling deformation; however, they cannot be used for comparing objects. In general, the recovered parameters for these physical models can be arbitrarily defined, and are therefore not invariant to changes in viewpoint, occlusion, or nonrigid deformations. Thus, if our goal is to recover descriptions for recognition and comparison, the physical modeling paradigm as such is flawed.

This problem of non-uniqueness can be cured if we adopt an eigendecomposition approach. In recent years there has been a revival of interest in eigendecomposition-based pattern recognition methods, due to the surprisingly good results that have been obtained by combining these methods with modern machine vision representations. In these methods, image or shape information is decomposed into an ordered basis of orthogonal principal components. As a result, the less critical and often noisy high-order components can be discarded in order to obtain overconstrained, canonical descriptions. This allows for the selection of only the most important components to be used for efficient data reduction, real-time recognition and navigation, and robust reconstruction.

The eigenvectors of a finite element model are known as *eigenmodes*. In some ways, the eigenmodes are akin to an object's generalized (nonlinear) axes of symmetry, in that they describe the principal axes of deformation over a training set of shapes. This generalized coordinate system provides a robust representation for establishing correspondences between similar shapes. Most importantly, the orthogonality of the eigen-representation ensures that the recovered descriptions will be unique, thus making recognition problems tractable.

Chapter 3

Approach

Our approach will combine many of the good features of the previous techniques. We will develop a new view-based elastic matching framework that makes use of the eigen-decomposition to ensure canonical descriptions and robustness.

The approach has three basic modules:

Modal Matching

At the core of this shape alignment and comparison framework is the idea of *modal matching* for computing feature correspondences and model similarity. Modal matching is 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.

Modal Descriptions

The resulting *modal description* will be used for object comparison, 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 that allows us to select which types of deformations are to be used in object alignment and comparison. In contrast to previous methods, this method will allow us 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.

Modal Combinations of Models

We will also introduce a physically-motivated linear-combinations-of-models paradigm, where the computer synthesizes an image of the object in terms of a weighted combination of modally deformed prototype images. This method is different from previous linear combinations of models techniques, in that it employs a frequency-ordered description of shape. As a result, we will be able to analyze and decompose shape deformations (and then resynthesize shapes) in a flexible, principled manner. This will have application in image and shape metamorphosis, since it will allow us to synthesize novel images and shapes based on examples.

We will now give a brief overview of each of these components. The mathematical details of the finite element formulation are provided in the last section of this chapter. We introduce a new type of Galerkin interpolant based on Gaussians that allows us to efficiently derive our shape parameterization directly from the data.

Though the primary application focus is matching shapes in 2-D images, the formulation can be generalized to both solve correspondence problems and compute robust, canonical descriptions for signals of any dimensionality (extension to N-dimensional problems is provided in Appendix A).

3.1 Modal Matching

Modal matching uses the eigenmodes of a physical shape model 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 3-1. For each image we start with feature point locations $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_m]$ and use these as nodes in building a finite element model of the shape. Gaussian interpolants are centered at each feature, and mass and stiffness are then integrated over the shape. We can think of this as constructing a model of the

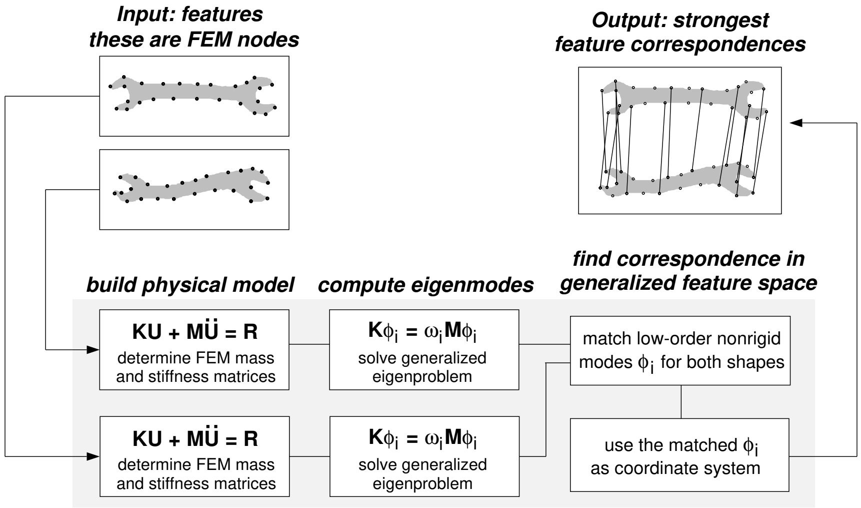


Figure 3-1: Modal matching system diagram.

shape by covering each feature point with a Gaussian blob of rubbery material; if we have segmentation information, then we can fill in interior areas and trim away material that extends outside of the shape.

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

$$\mathbf{X}_{deformed} = \mathbf{X} + a\phi_i, \tag{3.1}$$

where a is a scalar.

Figure 3-2 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 alignment and comparison, as will be described in Chapter 5.

The eigenmodes also form an orthogonal *object-centered* coordinate system for describing feature locations. That is, each feature point location can be uniquely described

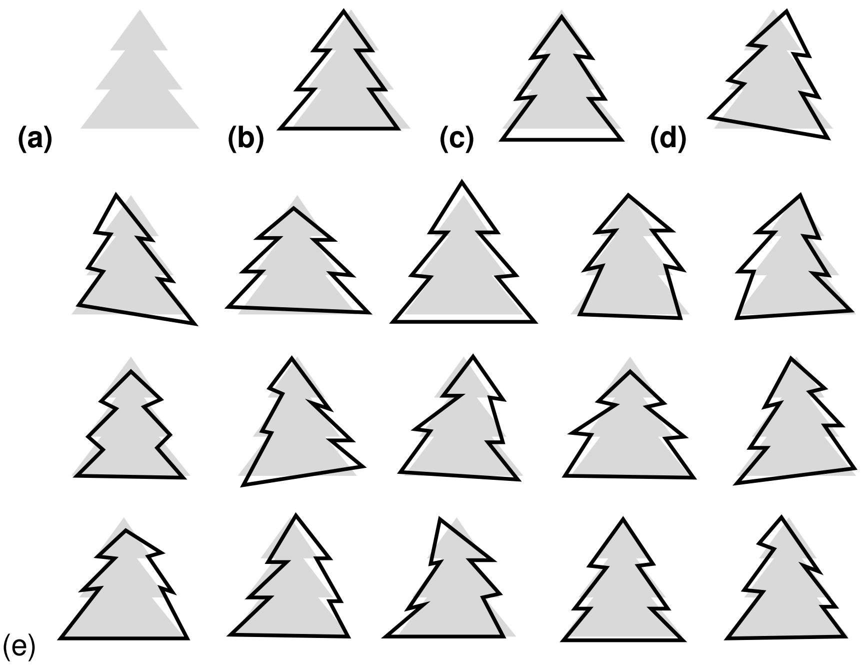


Figure 3-2: The low-order 18 modes computed for the upright tree shape (a). The first three modes are translation (b,c) and rotation (d). The others are nonrigid (e).

in terms of *how it moves within each deformation mode*, i.e., how it participates in each deformation mode. The transform between Cartesian feature locations and modal feature locations is accomplished by using the eigenvectors as a coordinate basis. In our technique, two groups of features are compared in the modal 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 3-3 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.

Using this property, we can reliably compute correspondence in this modal space. The concept of modal matching is demonstrated on the two similar tree shapes in Figure 3-4. Correspondences are found by comparing the direction of displacement at each node. The direction of displacement is shown by vectors in figure. For instance, the top points

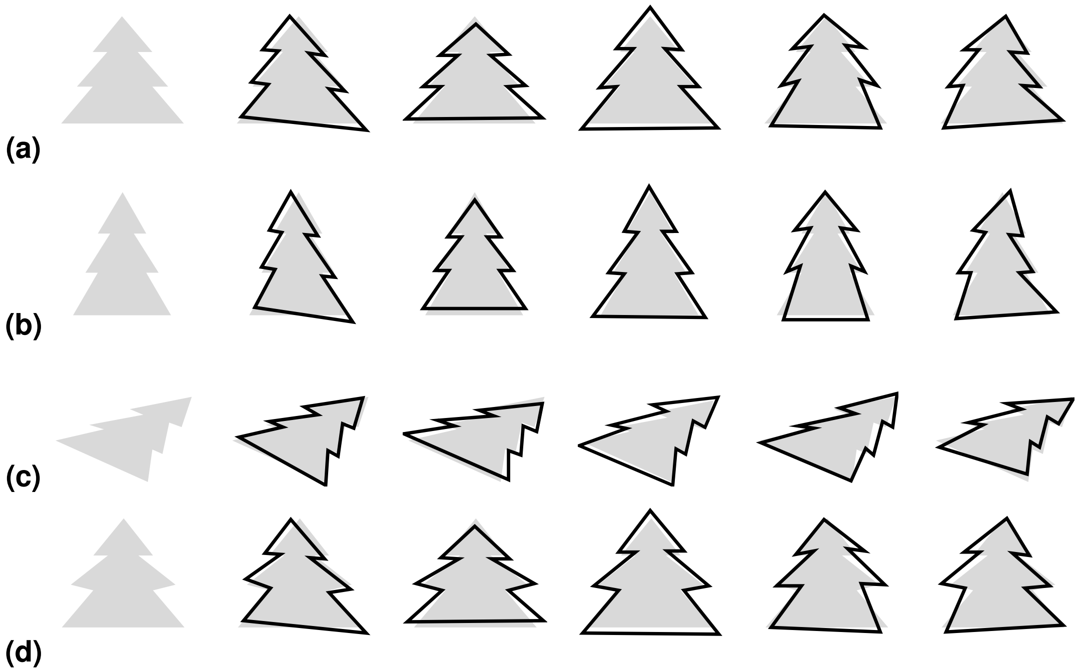


Figure 3-3: Similar shapes have similar low order modes. This figure shows the first five low-order vibration modes for similar tree shapes: (a) prototypical, (b) stretched, (c) tilted, and (d) two middle branches stretched.

on the two trees in Figure 3-4(a, b) have very similar displacements across a number of low-order modes, while the bottom point (shown in Figure 3-4(c)) has a very different displacement signature. Good matches have similar displacement signatures, and so the system matches the top points on the two trees.

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

Normally only the first 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) 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

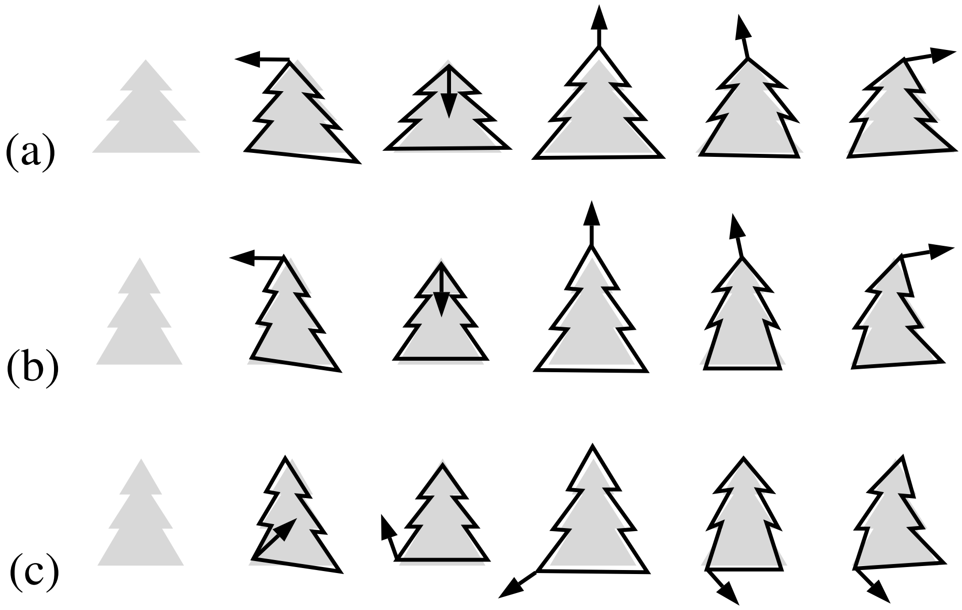


Figure 3-4: Correspondences are found by comparing the direction of displacement at each node (shown by vectors in figure). For instance, the top points on the two trees (a, b) have very similar displacement signatures, while the bottom point (shown in c) has a very different displacement signature. Using this property, we can reliably compute correspondence affinities in this modal signature space.

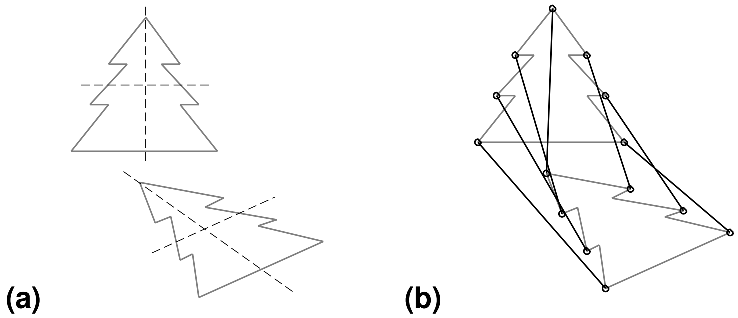


Figure 3-5: Two flat tree shapes, one upright and one lying flat (a), together with the obtained correspondence (b). The 18 low-order modes were computed for each tree and then correspondences were determined using the algorithm described in the text.

modes, or modes that are primarily local. Consequently, we can describe, track, and compare nonrigid objects in a very flexible and general manner.

Example output of the modal matching is shown in Figure 3-5. The left-hand side of the figure shows two views of a flat, tree-like shape, an example illustrating the idea of skewed symmetry adapted from [58]. The first 18 modes were computed for both trees, and were compared to obtain the correspondences shown in Figure 3-5(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.

In summary, the correspondence algorithm is based on matching up each point's trajectory signature in this modal space. Point correspondences are determined by comparing the two groups of features in the eigenmode coordinate system. That is, the set of points for one object is compared with the set of points for the second. The points that have the most similar and unambiguous modal signatures are then matched. The remaining correspondences are determined by using the physical model as a smoothness constraint [118].

3.2 Modal Descriptions

Given correspondences we can align or warp one shape into another. 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 for alignment. These deformations are measured in terms of mode amplitudes and are essentially a recipe for how to build a shape in terms of deformations from a prototype shape. 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 (strain) 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. 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 and position independent. To do this, we ignore displacements in the low-order or rigid body modes, thereby disregarding differences in position and orientation. In addition, we can make our comparisons robust to noise and local shape variations by discarding higher-order modes. This modal selection technique is also useful for its compactness, since we can describe deviation from a prototype in terms of relatively few modes.

3.3 Modal Combinations of Models

Using methods similar to those employed by Ullman and Basri[147] and by Poggio, *et al.* [11, 69, 102, 103], we would also like to describe objects as linear combinations of some collection of base models. The difference here is that we have a frequency-ordered description of shape; as a result we can analyze and decompose nonrigid shape deformation (and then synthesize shapes) in a principled way. One important advantage of our framework is that the modal representation can be used to reliably compute the necessary corresponding points in example views, despite relatively large deformations, differences in sampling, or noise.

Our method starts by determining point correspondences between a new shape and known extremal views. Given these correspondences we could use a standard technique for solving the image interpolation problem; however, modal image interpolation combines three important advantages. First, the physical model can be used to enforce elastic and smoothness constraints in warping pixels that lie between features. Second, the modal representation gives separate parameters; this allows for selecting and scheduling the types of deformations that can occur. Lastly, the modal representation gives a frequency-ordered description of the deformations needed to align the two feature sets. By discarding highest-frequency modes, the image warp can be made more robust with respect to noise.

Thus the core technology of modal matching will yield an improved physically-based framework for linear-combinations-of-views. The system interpolates between example views using modal image warps, and uses strain-energy to measure the similarity between the new shape and the interpolated shape. Thus we can describe entire categories of objects in terms of a few modally-warped examples.

Using this framework, we can also 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. We thereby obtain a low-dimensional parametric representation of motion that we can use to recognize and compare motion trajectories.

3.4 Mathematical Formulation

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

To accomplish this we will use the data itself to define the deformable object, by building stiffness and mass matrices that use the positions of image feature points as the finite element nodes. We will first develop a finite element formulation using Gaussian basis functions as Galerkin interpolants, and then use these interpolants to obtain generalized mass and stiffness matrices.

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

3.4.1 Brief Review of the Finite Element Method

A shape's modal representation is based on the eigenvectors of its physical model. The mathematical formulation of this physical model is based on the finite element method (FEM), the standard engineering technique for simulating the dynamic behavior of an object. This section provides a brief overview of the standard finite formulation. For an in-depth treatment of the formulation and convergence its properties, readers are directed to [9, 125].

In the FEM, interpolation functions are developed that allow continuous material properties, such as mass and stiffness, to be integrated across the region of interest. Using Galerkin's method for finite element discretization, we set up a system of shape functions that relate the displacement of a single point to the relative displacements of all the other nodes of an object. This set of shape functions describes an *isoparametric finite element*. By using these functions, we can calculate the deformations that spread uniformly over the body as a function of its constitutive parameters.

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

$$\mathbf{u}(\mathbf{x}) = \mathbf{H}(\mathbf{x})\mathbf{U} \tag{3.2}$$

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:

$$\epsilon(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{U} \tag{3.3}$$

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*:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{D}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R}, \tag{3.4}$$

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

Both the mass and stiffness matrices are computed directly:

$$\mathbf{M} = \int_V \rho \mathbf{H}^T \mathbf{H} dV \quad \text{and} \quad \mathbf{K} = \int_V \mathbf{B}^T \mathbf{C} \mathbf{B} dV, \tag{3.5}$$

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

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

$$\mathbf{D} = \alpha \mathbf{M} + \beta \mathbf{K}, \tag{3.6}$$

where α and β are constants determined by the desired critical damping [9].

3.4.2 Modal 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:

$$\mathbf{K}\boldsymbol{\phi}_i = \omega_i^2 \mathbf{M} \boldsymbol{\phi}_i. \tag{3.7}$$

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

The mode shapes can be thought of as describing the object's generalized (nonlinear) axes of symmetry. We can write Equation 3.7 as

$$\mathbf{K}\boldsymbol{\Phi} = \mathbf{M}\boldsymbol{\Phi}\boldsymbol{\Omega}^2 \tag{3.8}$$

where

$$\mathbf{\Phi} = [\phi_1 \mid \ldots \mid \phi_m] \quad \text{and} \quad \mathbf{\Omega}^2 = \begin{bmatrix} \omega_1^2 & & \\ & \ddots & \\ & & \omega_m^2 \end{bmatrix}. \tag{3.9}$$

As mentioned earlier, each mode shape vector ϕ_i is **M**-orthonormal, this means that

$$\mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi} = \mathbf{\Omega}^2 \quad \text{and} \quad \mathbf{\Phi}^T \mathbf{M} \mathbf{\Phi} = \mathbf{I}. \tag{3.10}$$

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

$$\mathbf{U} = \mathbf{\Phi} \tilde{\mathbf{U}} \tag{3.11}$$

We can now rewrite Equation 3.4 in terms of these generalized or modal displacements, obtaining a decoupled system of equations:

$$\ddot{\tilde{\mathbf{U}}}_t + \tilde{\mathbf{D}} \dot{\tilde{\mathbf{U}}} + \mathbf{\Omega}_t^2 \tilde{\mathbf{U}}_t = \mathbf{\Phi}_t^T \mathbf{R}, \tag{3.12}$$

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

$$\tilde{\mathbf{D}} = \mathbf{\Phi}^T \mathbf{D} \mathbf{\Phi} = \alpha \mathbf{I} + \beta \mathbf{\Omega}^2. \tag{3.13}$$

By decoupling these equations, we allow for closed-form solution to the equilibrium problem [96]. Given this equilibrium solution in the two images, point correspondences can be obtained directly.

By discarding high frequency eigenmodes the amount of computation required can be minimized without significantly altering correspondence accuracy. Moreover, such a set of modal amplitudes provides a robust, canonical description of shape in terms of deformations applied to the original elastic body. This allows them to be used directly for object recognition [96].

3.4.3 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 [9]. Stiffness and mass matrices $\mathbf{K}$ and $\mathbf{M}$ are precomputed for a simple, rectangular isoparametric element, and then this simple element is repeatedly warped and copied to tessellate the region of interest. This *assemblage* technique has the advantage that simple stiffness and mass matrices can be precomputed and easily assembled into large matrices that model topologically complex shapes.

Our problem is different in that we want to examine the eigenmodes of a cloud of feature points. It is akin to the problem found in interpolation networks: we have a fixed number of scattered measurements and we want to find a set of basis functions that allows for easy insertion and movement of data points. Moreover, since the position of nodal points will coincide with feature and/or sample points from our image, stiffness and mass matrices will need to be built on a per-feature-group basis. Gaussian basis functions are ideal candidates for this type of interpolation problem [103, 105]:

$$g_i(\mathbf{x}) = e^{-\|\mathbf{x}-\mathbf{x}_i\|^2/2\sigma^2} \tag{3.14}$$

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

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

$$h_i(\mathbf{x}) = \sum_{k=1}^{m} a_{ik} g_k(\mathbf{x}) \tag{3.15}$$

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

$$\mathbf{G} = \begin{bmatrix} g_1(\mathbf{x}_1) & \dots & g_1(\mathbf{x}_m) \\ \vdots & & \vdots \\ g_m(\mathbf{x}_1) & \dots & g_m(\mathbf{x}_m) \end{bmatrix}. \tag{3.16}$$

For the formulation described here, the basis function widths σ_i are kept the same at all nodes. Assuming a roughly uniform sampling of features, the widths should be equal to the average distance between feature points. If inter-feature distances vary greatly, then it may become necessary to use variable-width Gaussian interpolants to avoid an ill-conditioned finite element model.

To help visualize the resulting interpolants, Figure 3-6 shows graphs of three interpolants used in building a one-dimensional finite element. The element had twenty-five evenly spaced nodes and the basis function width σ was set equal to the spacing between nodes. The resulting interpolant for an internal node is shown in (a), the interpolant for an endpoint is show in (b), and near-border node in (c). Note that the nodes near the boundary have slightly different interpolant profiles. For evenly-spaced nodes, the internal node's interpolants will generally look like windowed sinc functions. When the element nodes are unevenly-spaced, these interpolants will stretch and shrink.

By using these Gaussian interpolants as our shape functions for Galerkin approximation, we can easily formulate finite elements for any dimension. A very useful aspect of Gaussians is that they are factorizable: multidimensional interpolants can be assembled out of lower dimensional Gaussians. This not only reduces computational cost, it also has useful implications for VLSI hardware and neural-network implementations [103].

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 [141]. If a conforming element is desired, this can be obtained by

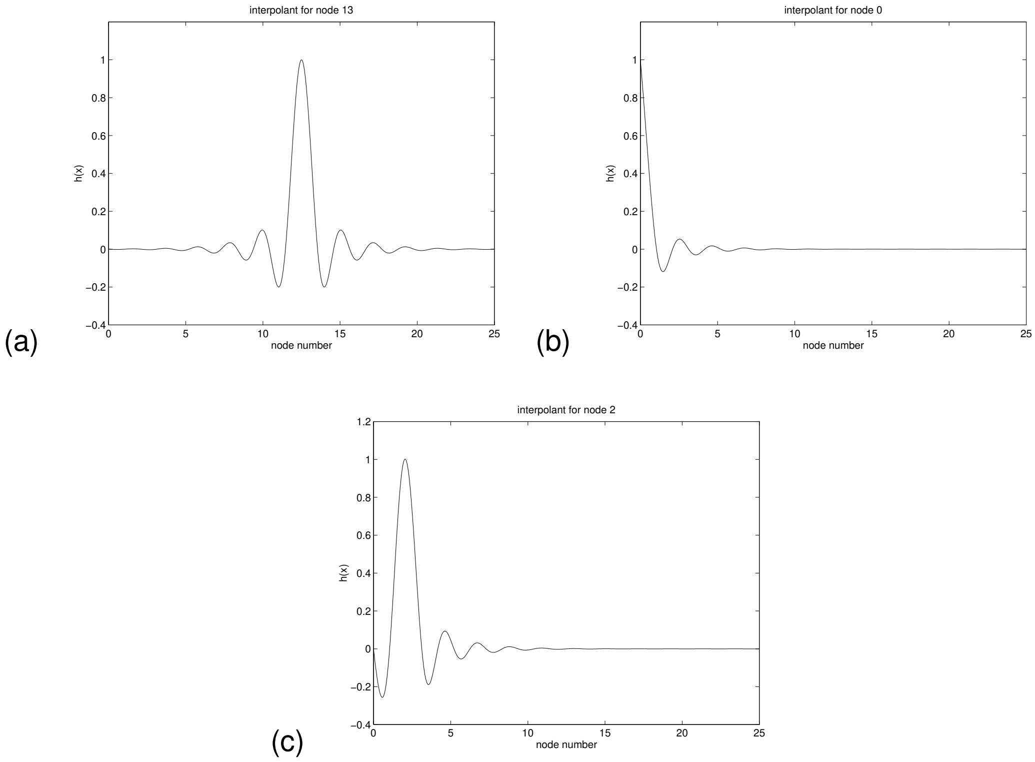


Figure 3-6: Graphs showing typical interpolation functions for a uniform 1-D element. The interpolant for an internal node is shown in (a), the interpolant for an endpoint is show in (b), and near-border node in (c). Note that the nodes near the boundary have slightly different interpolant profiles. These interpolants were constructed from linear combinations of Gaussian basis functions as described in the text.

including a normalization term in h_i in Equation 3.15,

$$h_i(\mathbf{x}) = \frac{\displaystyle\sum_{k=1}^{m} a_{ik} g_k(\mathbf{x})}{\displaystyle\sum_{j=1}^{m} \sum_{k=1}^{m} a_{jk} g_k(\mathbf{x})} \quad . \tag{3.17}$$

For most of the examples in this thesis, we will use the simpler, non-conforming interpolants, primarily because the integrals for mass and stiffness can be computed analytically. The differences between conforming and nonconforming interpolants do not affect the matching results reported in this thesis. On the one hand, non-conforming interpolants can be used to model an infinite-support, potential field of features. On the other hand, conforming interpolants can be used to model an object's particular shape by using a support-map. Both types of elements are formulated in the next section.

3.4.4 Formulating 2-D Elements

For the sake of illustration we will now give the mathematical details for a two-dimensional implementation. The extension to dimensions three and higher is straight-forward and provided in Appendix A.

Before formulating these 2-D elements, we will first need to provide some basic integrals. The first three are infinite integrals of exponentials (adapted from [45]).

$$\int_{-\infty}^{\infty} e^{-px^2+2qx}\,dx = \sqrt{\frac{\pi}{p}}\,e^{\frac{q^2}{p}} \tag{3.18}$$

$$\int_{-\infty}^{\infty} x\,e^{-px^2+2qx}\,dx = \frac{q}{p}\sqrt{\frac{\pi}{p}}\,e^{\frac{q^2}{p}} \tag{3.19}$$

$$\int_{-\infty}^{\infty} x^2\,e^{-px^2+2qx}\,dx = \frac{1}{2p}\sqrt{\frac{\pi}{p}}\left(1 + \frac{2q^2}{p}\right)\,e^{\frac{q^2}{p}} \tag{3.20}$$

We can then use these integrals to derive the 2-D infinite Gaussian basis function integrals:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g_i(\mathbf{x})\,g_j(\mathbf{x})dx\,dy = \pi\sigma^2\,\sqrt{g_{ij}} \tag{3.21}$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x\,g_i(\mathbf{x})\,g_j(\mathbf{x})dx\,dy = \frac{\pi\,\sigma^2}{2}\,(x_i + x_j)\,\sqrt{g_{ij}} \tag{3.22}$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x\,y\,g_i(\mathbf{x})\,g_j(\mathbf{x})dx\,dy = \frac{\pi\,\sigma^2}{4}\,(x_i + x_j)\,(y_i + y_j)\,\sqrt{g_{ij}} \tag{3.23}$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^2\,g_i(\mathbf{x})\,g_j(\mathbf{x})dx\,dy = \frac{\pi\,\sigma^4}{4}\,(y_i + y_j)\,\left(1 + \frac{(x_i + x_j)^2}{2\,\sigma^2}\right)\,\sqrt{g_{ij}} \tag{3.24}$$

2-D Element with Infinite Support

We begin by assembling a 2-D interpolation matrix from the shape functions developed in Equation 3.15:

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} h_1 & \dots & h_m & 0 & \dots & 0 \\ 0 & \dots & 0 & h_1 & \dots & h_m \end{bmatrix}. \tag{3.25}$$

Substituting into Equation 3.5 and multiplying out we obtain a mass matrix for the feature data:

$$\mathbf{M} = \int_A \rho \mathbf{H}^T \mathbf{H} dA = \begin{bmatrix} \mathcal{M} & 0 \\ 0 & \mathcal{M} \end{bmatrix}, \tag{3.26}$$

where the m by m submatrix $\mathcal{M}$ is positive definite symmetric. The elements of $\mathcal{M}$ have the form:

$$m_{ij} = \rho \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sum_{k,l} a_{ik}\, a_{jl}\, g_k(\mathbf{x})\, g_l(\mathbf{x})\, dx\, dy. \tag{3.27}$$

We then integrate and regroup terms:

$$m_{ij} = \rho\, \pi\, \sigma^2 \sum_{k,l} a_{ik}\, a_{jl}\, \sqrt{g_{kl}} \tag{3.28}$$

where $g_{kl} = g_k(\mathbf{x}_l)$ is an element of the $\mathbf{G}$ matrix in Equation 3.16.

This can be rewritten in matrix form:

$$\mathcal{M} = \rho\, \pi\, \sigma^2 \mathbf{A}^T \mathcal{G} \mathbf{A} = \rho\, \pi\, \sigma^2 \mathbf{G}^{-1} \mathcal{G} \mathbf{G}^{-1}, \tag{3.29}$$

where the elements of $\mathcal{G}$ are the square roots of the elements of the $\mathbf{G}$ matrix in Equation 3.16.

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:

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x} h_1 & \dots & \frac{\partial}{\partial x} h_m & 0 & \dots & 0 \\ 0 & \dots & 0 & \frac{\partial}{\partial y} h_1 & \dots & \frac{\partial}{\partial y} h_m \\ \frac{\partial}{\partial y} h_1 & \dots & \frac{\partial}{\partial y} h_m & \frac{\partial}{\partial x} h_1 & \dots & \frac{\partial}{\partial x} h_m \end{bmatrix}, \tag{3.30}$$

where the partials are of the form

$$\frac{\partial}{\partial x} h_i = \frac{1}{\sigma^2} \sum_{k=1}^{m} (x_k - x)\, a_{ik}\, g_k(\mathbf{x}) \tag{3.31}$$

and

$$\frac{\partial}{\partial y} h_i = \frac{1}{\sigma^2} \sum_{k=1}^{m} (y_k - y)\, a_{ik}\, g_k(\mathbf{x}). \tag{3.32}$$

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

$$\mathbf{C} = \beta \begin{bmatrix} 1 & \alpha & 0 \\ \alpha & 1 & 0 \\ 0 & 0 & \xi \end{bmatrix}. \tag{3.33}$$

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:

$$\alpha = \frac{\nu}{1-\nu}, \qquad \beta = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}, \quad \text{and} \quad \xi = \frac{1-2\nu}{2\,(1-\nu)}. \tag{3.34}$$

Substituting into Equation 3.5 and multiplying out we obtain a stiffness matrix for the 2-D feature data:

$$\mathbf{K} = \int_A \mathbf{B}^T \mathbf{C} \mathbf{B} dA = \begin{bmatrix} \mathbf{K}_{aa} & \mathbf{K}_{ab} \\ \mathbf{K}_{ba} & \mathbf{K}_{bb} \end{bmatrix} \tag{3.35}$$

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

$$k_{aa_{ij}} = \beta \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sum_{k,l} a_{ik}\, a_{jl} \left[\frac{\partial g_k}{\partial x} \frac{\partial g_l}{\partial x} + \xi \frac{\partial g_k}{\partial y} \frac{\partial g_l}{\partial y} \right] dx\, dy. \tag{3.36}$$

Integrate and regroup terms:

$$k_{aa_{ij}} = \pi\beta \sum_{k,l} a_{ik}\, a_{jl} \left[\frac{1+\xi}{2} - \frac{\left(\hat{x}_{kl}^2 + \xi \hat{y}_{kl}^2\right)}{4\sigma^2} \right] \sqrt{g_{kl}}, \tag{3.37}$$

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

$$k_{bb_{ij}} = \pi\beta \sum_{k,l} a_{ik}\, a_{jl} \left[\frac{1+\xi}{2} - \frac{\left(\hat{y}_{kl}^2 + \xi \hat{x}_{kl}^2\right)}{4\sigma^2} \right] \sqrt{g_{kl}}. \tag{3.38}$$

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

$$k_{ab_{ij}} = \beta \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sum_{k,l} a_{ik}\, a_{jl} \left[\alpha \frac{\partial g_k}{\partial x} \frac{\partial g_l}{\partial y} + \xi \frac{\partial g_k}{\partial y} \frac{\partial g_l}{\partial x} \right] dx\, dy. \tag{3.39}$$

When integrated this becomes:

$$k_{ab_{ij}} = -\frac{\pi\beta(\alpha+\xi)}{4\sigma^2} \sum_{k,l} a_{ik}\, a_{jl}\, \hat{x}_{kl}\, \hat{y}_{kl} \sqrt{g_{kl}}. \tag{3.40}$$

2-D Element with Finite Support

Given a support function we can "cut" the finite element sheet into any shape. We do this by defining a support function $s(\mathbf{x})$ that is zero anywhere outside the shape region R, and greater than zero inside the shape region. Thus the support function can be used to define both the shape and the thickness of the elastic model.

We will now use the conforming interpolants h_i of Equation 3.17 in our formulation of a 2-D finite element model. The equivalent integral for the mass matrix Equation 3.35 becomes:

$$m_{ij} = \rho \int_R s(\mathbf{x})\, h_i(\mathbf{x})\, h_j(\mathbf{x}) dx dy \tag{3.41}$$

Assuming that there is a large number of pixels, this is approximated by the following discretization:

$$m_{ij} = \rho \sum_{\mathbf{x} \in R} s(\mathbf{x}) \frac{\sum_{k,l} a_{ik} a_{jl} g_k(\mathbf{x}) g_l(\mathbf{x})}{\left(\sum_k \hat{a}_k g_k(\mathbf{x}) \right)^2} \tag{3.42}$$

where $\hat{a}_k = \sum_i a_{ik}$.

The partial derivatives for the conforming interpolants h_i take the form:

$$\frac{\partial}{\partial x} h_i(\mathbf{x}) = \frac{\displaystyle\sum_{k,l} (x - x_l)\ [\hat{a}_k\, a_{il}\ -\ \hat{a}_l\, a_{ik}]\ g_l(\mathbf{x})\, g_k(\mathbf{x})}{\sigma^2 \left(\displaystyle\sum_k \hat{a}_k g_k(\mathbf{x})\right)^2}, \tag{3.43}$$

and

$$\frac{\partial}{\partial y} h_i(\mathbf{x}) = \frac{\displaystyle\sum_{k,l} (y - y_l)\ [\hat{a}_k\, a_{il}\ -\ \hat{a}_l\, a_{ik}]\ g_l(\mathbf{x})\, g_k(\mathbf{x})}{\sigma^2 \left(\displaystyle\sum_k \hat{a}_k g_k(\mathbf{x})\right)^2}. \tag{3.44}$$

The elements of the stiffness matrix that correspond with those described in Equation 3.36 become discrete integrals of the form:

$$k_{aa_{ij}} = \beta \sum_{\mathbf{x} \in R} s(\mathbf{x}) \left[\frac{\partial h_i}{\partial x} \frac{\partial h_j}{\partial x} + \xi \frac{\partial h_i}{\partial y} \frac{\partial h_j}{\partial y} \right]. \tag{3.45}$$

Similarly

$$k_{bb_{ij}} = \beta \sum_{\mathbf{x} \in R} s(\mathbf{x}) \left[\frac{\partial h_i}{\partial y} \frac{\partial h_j}{\partial y} + \xi \frac{\partial h_i}{\partial x} \frac{\partial h_j}{\partial x} \right] \tag{3.46}$$

and

$$k_{ab_{ij}} = \beta \sum_{\mathbf{x} \in R} s(\mathbf{x}) \left[\alpha \frac{\partial h_i}{\partial x} \frac{\partial h_j}{\partial y} + \xi \frac{\partial h_i}{\partial y} \frac{\partial h_j}{\partial x} \right]. \tag{3.47}$$

In pseudocode, the algorithm for computing the 2-D finite-support mass and stiffness matrices is as follows:

for each **x** $\in R$ if $s(\mathbf{x}) > 0.0$ do *values and partials of basis functions*
  for each node i do
    $g\,[i] = g_i(\mathbf{x})$
    $g_x\,[i] = (x - x_i) g\,[i] \,/\, \sigma^2$
    $g_y\,[i] = (y - y_i) g\,[i] \,/\, \sigma^2$
  end

  $h\ =\ A\, g$ *multiply basis vectors by coefficients*
  $h_x\ =\ A\, g_x$
  $h_y\ =\ A\, g_y$

$a = \sum_i h[i]$ *interpolants must sum to 1*
$b = a^2 \sum_i h_x[i]$
$c = a^2 \sum_i h_y[i]$

for each node i do *fill vectors of interpolants and partials*
  $g[i] = h[i]/a$
  $g_x[i] = h_x[i]/a - h[i]/b$
  $g_y[i] = h_y[i]/a - h[i]/c$
end

$G = G_{xx} = G_{xy} = G_{yy} = 0.0$ *zero interpolant and partial matrices*
  for each node i do *fill matrices of products*
    for each node j do
      $G[i][j] = G[i][j] + s(\mathbf{x})g[i]g[j]$
      $G_{xx}[i][j] = G_{xx}[i][j] + s(\mathbf{x})g_x[i]g_x[j]$
      $G_{yy}[i][j] = G_{yy}[i][j] + s(\mathbf{x})g_y[i]g_y[j]$
      $G_{xy}[i][j] = G_{xy}[i][j] + s(\mathbf{x})g_x[i]g_y[j]$
    end
  end
end

$M_{aa} = \rho G$ *fill mass and stiffness submatrices*
$K_{aa} = G_{xx} + \xi G_{yy}$
$K_{bb} = G_{yy} + \xi G_{xx}$
$K_{ba}^T = K_{ab} = \alpha G_{xy} + \xi G_{xy}^T$

3.5 Summary

We have formulated a family of physical models that use the feature data itself as nodes in building a finite element; our stiffness and mass matrices use the positions of image feature points as the finite element nodes. Through the use of Galerkin surface approximation, we can avoid sampling problems and incorporate outside information such as feature connectivity and distinctiveness. The eigenvectors or *modes* of the resulting matrices form an intrinsic coordinate system for the shape and its natural deformation.

Building on this formulation, the next three chapters will develop the main system components: modal matching, modal descriptions, and modal combinations of models.

Chapter 4

Modal Matching

In this chapter we will describe modal matching, a new method for determining corresponding features on similar shapes. Modal matching utilizes a shape-intrinisic coordinate system for robust correspondence computation. This intrinsic coordinate system is built from the eigenvectors of the new physical shape model, and offers useful properties: invariance to scaling and translation, and robustness to nonrigid deformation and noise. We will first formulate the fundamental correspondence method, and then extend it it two main areas. First, we introduce the notion of multiresolution matching; this allows us to more efficiently match up shapes that have potentially hundreds of features. Second, we derive a rotation-invariant modal matching formulation.

4.1 Determining Correspondences

To determine correspondences, we first compute mass and stiffness matrices for both feature sets. These matrices are then decomposed into eigenvectors ϕ_i and eigenvalues λ_i as described in Section 3.4.2. The resulting eigenvectors are ordered by increasing eigenvalue, and form the columns of the modal matrix $\mathbf{\Phi}$:

$$\mathbf{\Phi} = [\phi_1 \mid \ldots \mid \phi_{2m}] = \begin{bmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_m^T \\ \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_m^T \end{bmatrix} \tag{4.1}$$

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

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

4.1.1 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 eigenmodes will represent the rigid body modes of two translations and a rotation. These first three columns of each modal matrix are therefore discarded to make the correspondence computation invariant to differences in rotation and translation.

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

$$\begin{array}{l} \mathbf{\Phi}_1 = [\underbrace{\phi_{1,1} \mid \phi_{1,2} \mid \phi_{1,3}} \mid \underbrace{\phi_{1,4} \mid \ldots \mid \phi_{1,p}} \mid \underbrace{\phi_{1,p+1} \mid \ldots \mid \phi_{1,2m}}] \\ \qquad\quad \begin{array}{ccc} \text{rigid body} & \text{intermediate} & \text{high-order} \\ \text{modes} & \text{modes} & \text{modes} \end{array} \\ \mathbf{\Phi}_2 = [\overbrace{\phi_{2,1} \mid \phi_{2,2} \mid \phi_{2,3}} \mid \overbrace{\phi_{2,4} \mid \ldots \mid \phi_{2,p}} \mid \overbrace{\phi_{2,p+1} \mid \ldots \mid \phi_{2,2n}}] \end{array} \tag{4.2}$$

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

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

$$\bar{\mathbf{\Phi}} = [\phi_4 \mid \ldots \mid \phi_p] = \begin{bmatrix} \bar{\mathbf{u}}_1^T \\ \vdots \\ \bar{\mathbf{u}}_m^T \\ \bar{\mathbf{v}}_1^T \\ \vdots \\ \bar{\mathbf{v}}_m^T \end{bmatrix}, \tag{4.3}$$

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.

4.1.2 Computing Correspondence Affinities

Using a modified version of an algorithm described by Shapiro and Brady [127], 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:

$$\mathbf{z}_{ij} = \| \bar{\mathbf{u}}_{1,i} - \bar{\mathbf{u}}_{2,j} \|^2 + \| \bar{\mathbf{v}}_{1,i} - \bar{\mathbf{v}}_{2,j} \|^2. \tag{4.4}$$

The affinity measure for the i^{th} and j^{th} points, z_{ij}, will be zero for a perfect match and will increase as the match worsens. Using these affinity measures, we can easily identify which features correspond to each other in the two images by looking for the minimum entry in each column or row of $\mathbf{Z}$. Shapiro and Brady noted that the symmetry of an eigendecomposition requires an intermediate sign correction step for the eigenvectors ϕ_i. This is due to the fact that the direction (sign) of eigenvectors can be assigned arbitrarily. Readers are referred to [126] for more details about this.

To obtain accurate correspondences the Shapiro and Brady method requires three simple, but important, modifications. First, only the generalized features that match with the greatest certainty are used to determine the deformation; the remainder of the correspondences are determined by the deformation itself as in our previous method. By discarding affinities greater than a certain threshold, we allow for tokens that have no strong match. Second, as described earlier, only the low-order twenty-five percent of the eigenvectors are employed, as the higher-order modes are known to be noise-sensitive and thus unstable [9]. Lastly, because of the reduced basis matching, similarity of the generalized features is required in both directions, instead of one direction only. In other words, a match between the i^{th} feature in the first image and the j^{th} feature in the second image can only be valid if z_{ij} is the minimum value for its row, and z_{ji} the minimum for its column. Image points for which there was no correspondence found are flagged accordingly.

4.1.3 Implementation Details: Mode Reordering and Selection

In cases with low sampling densities or with large deformations, the mode ordering can vary slightly even though shapes may be similar. Such cases require an extra step in which neighborhoods of similarly-valued modes are compared to find the best matching modes. Furthermore, certain low-order modes will match each other quite well while other modes will not match as well (primarily due to the multiple eigenvalue problem mentioned earlier). We will use distances between modes to select only the best-matching subset of modes to use in computing feature correspondence affinities. As will be described in the next chapter, these mode distances can also be used directly for determining rough shape similarity.

One simple heuristic for measuring modal distance proceeds as follows. We first find the centroids and moments for both shapes. Using the axes of intertia to define the outer corners of a bounding rectangle, we then sample a coarse rectangular grid of points. If the two shapes are at different orientations then we include an extra step in which the two models are rotated into rough alignment.

The mode shape is then interpolated for each point in this rectilinear grid. We utilize an interpolated modal matrix that describes each mode's shape for the at the regular grid points:

$$\hat{\mathbf{\Phi}} = \hat{\mathbf{H}}\mathbf{\Phi}. \tag{4.5}$$

In this equation, the interpolation matrix $\hat{\mathbf{H}}$ relates the displacement at the nodes (original features) to displacements at the rectangular grid point feature locations $\mathbf{x}_i$:

$$\hat{\mathbf{H}} = \begin{bmatrix} \mathbf{H}(\mathbf{x}_1) \\ \vdots \\ \mathbf{H}(\mathbf{x}_n) \end{bmatrix}, \tag{4.6}$$

where each submatrix $\mathbf{H}(\mathbf{x}_i)$ is a $2 \times 2m$ interpolation matrix as in Eq. 3.25.

Given modes interpolated for both objects, we wish to select the subset of corresponding modes whose similarity is strongest to within a threshold. This is done by either taking the dot product or Euclidean distance between interpolated mode vectors ϕ_i. For each of the first shape's nonrigid modes, we check over a neighborhood of the second shape's modes (in our experience, a neighborhood of three is sufficient). Distances should be computed for both the positive and negative directions of the eigenvectors. Finally, if we want to allow for symmetry, then we need to take the dot product for the four flip-symmetries of the rectangular grid.

For this thesis, problems with sampling density were avoided by taking the dot product of the *normalized* eigenmodes. As a result, the eigenmode match threshold was invariant to sampling density and scaling. Sometimes it may be useful to instead normalize for aspect ratio differences between the two shapes.

4.2 Multiresolution Models

When there are possibly hundreds of feature points for each shape, computing the FEM model and eigenmodes for the full feature set can become non-interactive. For efficiency, we can select a subset of the feature data to build a lower-resolution finite element model and then use the resulting eigenmodes in finding the higher-resolution feature correspondences. The procedure for this is as follows.

First, a subset of m feature points is selected to be finite element nodes. This subset can be a set of particularly salient features (*i.e.,* corners, T-junctions, and edge mid-points) or a randomly selected subset of (roughly) uniformly-spaced features. As before, a FEM model is built for each shape, eigenmodes are obtained, and modal truncation is performed as described in Section 4.1.1. The resulting eigenmodes for the two shapes are then matched and sign-corrected using the lower-resolution models' affinity matrix.

With modes matched for the feature subsets, we now proceed to finding the correspondences for the full sets of features. To do this, we use Equation 4.5 to obtain interpolated modal matrices. In this case, the interpolation matrix $\hat{\mathbf{H}}$ relates the displacement at the nodes (original features) to displacements at the full-resolution feature locations $\mathbf{x}_i$; we thereby compute matrices that describe each mode's shape for the full set of features in terms of the features used to build the low-resolution model.

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

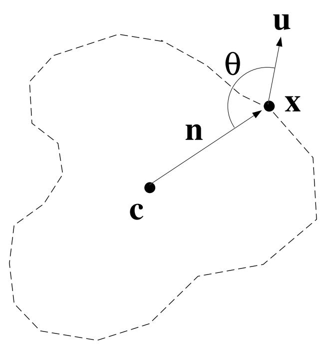


Figure 4-1: Transforming a modal displacement vector **u** $= (u, v)$ into (θ, r). The angle θ is computed relative to the vector **n** from the object's centroid **c** to the nodal point **x**. The radius r is simply the length of **u**.

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

$$\phi_i = [u_1 \ldots u_m, v_1 \ldots v_m]^T \tag{4.7}$$

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 4-1. 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 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.*:

$$\mathbf{z}_{ij} = \| \bar{\theta}_{1,i} - \bar{\theta}_{2,j} \|^2. \quad (4.8)$$

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.

Chapter 5

Modal Descriptions

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

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

5.1 Recovering Modal Descriptions via Modal Alignment

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

$$\mathbf{u}_i = \mathbf{x}_{1,i} - \mathbf{x}_{2,i}, \tag{5.1}$$

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

Recalling that $\mathbf{U} = \mathbf{\Phi}\tilde{\mathbf{U}}$, and using the identity of Equation 3.10, we find:

$$\tilde{\mathbf{U}} = \mathbf{\Phi}^{-1}\mathbf{U} = \mathbf{\Phi}^T\mathbf{M}\mathbf{U}. \tag{5.2}$$

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

In the first and simplest approach, we are given the nodal displacements $\mathbf{u}_i$ at the matched nodes, and we set the loads $\mathbf{r}_i$ at unmatched nodes to zero. We can then solve the equilibrium equation, $\mathbf{KU} = \mathbf{R}$, where we have as many knowns as unknowns. Modal displacements are then obtained via Eq. 5.2. This approach yields a closed-form solution, but it has the disadvantage that we have assumed that forces at the unmatched nodes are zero.

5.1.1 Reducing the Degrees of Freedom via Modal Truncation

One way around this zero-force assumption is to solve a mode-truncated version of the physical system. Assume that we have found correspondences for p of the m nodes. We can reduce the degrees of freedom by discarding $m - p$ of the high-frequency modes in Equation 5.2, and then solve for the modes via a matrix inverse. This method proceeds as follows.

Given that we know correspondences for some of the nodes, we reorder the columns of the matrix $\mathbf{\Phi}^{-1}$:

$$\left[\begin{array}{c|c} \mathbf{\Phi}^{-1}_{known} & \mathbf{\Phi}^{-1}_{unknown} \\ & \end{array}\right] \left[\begin{array}{c} \mathbf{U}_{known} \\ \mathbf{U}_{unknown} \end{array}\right] = \left[\begin{array}{c} \tilde{\mathbf{U}}_{unknown} \\ 0 \end{array}\right], \tag{5.3}$$

where $\mathbf{U}_{known}$ is the vector of known displacements at the p matched nodes, $\mathbf{U}_{unknown}$ is the vector of unknown nodal displacements, and $\tilde{\mathbf{U}}_u nknown$ is the vector of desired modal amplitudes. In this formulation, it is assumed that the amplitudes for discarded modes are zero.

We now regroup terms:

$$\left[\begin{array}{c|c} \mathbf{\Phi}^{-1}_{known} & 0 \\ & \mathbf{I} \end{array}\right] \left[\begin{array}{c} \mathbf{U}_{known} \\ 0 \end{array}\right] = \left[\begin{array}{c|c} \mathbf{I} & \mathbf{\Phi}^{-1}_{unknown} \\ 0 & \end{array}\right] \left[\begin{array}{c} \tilde{\mathbf{U}}_{unknown} \\ \mathbf{U}_{unknown} \end{array}\right]. \tag{5.4}$$

The desired mode amplitudes can now be obtained directly by inverting the right-hand matrix. Note that we again obtain a solution in closed-form, but we have assumed that the modal displacements $\tilde{\mathbf{u}}_i = 0$, for $i > p$.

5.1.2 Strain-minimizing Least Squares Solution

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. To achieve this, we find mode amplitudes that minimize the strain energy:

$$E_I = \frac{1}{2}\tilde{\mathbf{U}}^T \mathbf{\Omega}^2 \tilde{\mathbf{U}}. \tag{5.5}$$

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

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

$$E = \underbrace{\left[\mathbf{U} - \mathbf{\Phi}\tilde{\mathbf{U}}\right]^T\left[\mathbf{U} - \mathbf{\Phi}\tilde{\mathbf{U}}\right]}_{\text{squared fitting error}} + \underbrace{\lambda\tilde{\mathbf{U}}^T\mathbf{\Omega}^2\tilde{\mathbf{U}}.}_{\text{strain energy}} \tag{5.6}$$

As can be seen, this strain term directly parallels the smoothness functional employed in regularization [139].

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

$$\tilde{\mathbf{U}} = \left[\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{\Omega}_2\right]^{-1}\mathbf{\Phi}^T\mathbf{U}. \tag{5.7}$$

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

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

$$\tilde{\mathbf{U}} = \left[\mathbf{\Phi}^T\mathbf{W}^2\mathbf{\Phi} + \lambda\mathbf{\Omega}^2\right]^{-1}\mathbf{\Phi}^T\mathbf{W}^2\mathbf{U} \tag{5.8}$$

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

5.1.3 The Dynamic Solution

So far, we have described methods for finding the modal displacements that directly deform and align two feature sets. It is also possible to solve the alignment problem by physical simulation, in which the finite element equations are integrated over time until equilibrium is achieved. In this case, we solve for the deformations at each time step via the *dynamic equation* (Eq. 3.12). In so doing, we compute the intermediate deformations in a manner consistent with the material properties that were built into the finite element model. The intermediate deformations can also be used for physically-based morphing.

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

$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + k(\mathbf{x}_{1,i} + \mathbf{u}_i(t) - \mathbf{x}_{2,i}), \tag{5.9}$$

where k is an overall stiffness constant and $\mathbf{u}_i(t)$ is the nodal displacement at the previous time step. These loads simulate "ratchet springs," which are successively tightened until the surface matches the data [49].

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

$$\ddot{\tilde{u}}_i(t) + \tilde{d}_i \dot{\tilde{u}}_i(t) + \omega_i^2 \tilde{u}_i(t) = \tilde{r}_i(t), \tag{5.10}$$

where the $\tilde{r}_i(t)$ are components of the transformed load vector $\tilde{\mathbf{R}}(t) = \mathbf{\Phi}^T \mathbf{R}(t)$. These independent equilibrium equations can be solved via an iterative numerical integration procedure (*e.g.,* Newmark method [9]). The system is integrated forward in time until the change in load energy goes below a threshold δ, *e.g.*:

$$\| \mathbf{R}(t+\Delta t) - \mathbf{R}(t) \|^2 < \delta^2. \tag{5.11}$$

The loads $\mathbf{r}_i(t)$ are updated at each time step by evaluating Equation 5.9.

5.2 Coping with Large Rotations

If the rotation needed to align the two sets of points is potentially large, then it is necessary to perform an initial alignment step before recovering the modal deformations. Orientation, position, and (if desired) scale can be recovered in closed-form via quaternion-based algorithms described by Horn [52] or by Wang and Jepson [150].[1]

[1] While all the examples reported here are two-dimensional, it was decided that for generality, a 3-D orientation recovery method would be employed. For 2-D orientation recovery problems, simply set $z = 0$.

The following is a summary of Horn's orientation recovery method. 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 that take features in one image and line them up with their counter parts in the other.

Once the two images have been aligned in this way, the features may not be completely aligned. We can now solve for the modal deformations $\tilde{\mathbf{U}}$ that 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 used to align the images, we may want to go back and use this alignment information to determine correspondences for the unmatched features, or recompute correspondences for 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 the two images, where $\mathbf{x}_{1,i}$ corresponds to $\mathbf{x}_{2,i}$. We first find the centroids for these two point sets:

$$\bar{\mathbf{x}}_1 = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_{1,i} \quad \text{and} \quad \bar{\mathbf{x}}_2 = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_{2,i}. \tag{5.12}$$

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

$$\mathbf{x'}_{1,i} = \mathbf{x}_{1,i} - \bar{\mathbf{x}}_1 \quad \text{and} \quad \mathbf{x'}_{2,i} = \mathbf{x}_{2,i} - \bar{\mathbf{x}}_2. \tag{5.13}$$

We are now ready to recover the unit quaternion $\mathbf{q} = q_0 + iq_x + jq_y + kq_z$ that describes the rotation that orients $\mathbf{x}_1$ with $\mathbf{x}_2$. This is done by first building the symmetric matrix

$$\mathbf{M} = \begin{bmatrix} (S_{xx} + S_{yy} + S_{zz}) & S_{yz} - S_{zy} & S_{zx} - S_{xz} & S_{xy} - S_{yx} \\ S_{yz} - S_{zy} & (S_{xx} - S_{yy} - S_{zz}) & S_{xy} + S_{yx} & S_{zx} + S_{xz} \\ S_{zx} - S_{xz} & S_{xy} + S_{yx} & (-S_{xx} + S_{yy} - S_{zz}) & S_{yz} + S_{zy} \\ S_{xy} - S_{yx} & S_{zx} + S_{xz} & S_{yz} + S_{zy} & (-S_{xx} - S_{yy} + S_{zz}) \end{bmatrix} \tag{5.14}$$

where

$$S_{xx} = \sum_{i=1}^{n} x'_{1,i}\, x'_{2,i}, \quad S_{xy} = \sum_{i=1}^{n} x'_{1,i}\, y'_{2,i}, \quad \textit{etc.} \tag{5.15}$$

We then find the most positive eigenvalue for the matrix $\mathbf{M}$. Horn has shown that the eigenvector associated with this eigenvalue represents the unit quaternion $\mathbf{q}$ that 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:

$$s = \left(\sum_{i=1}^{n} \| \mathbf{x'}_{2,i} \|^2 \Big/ \sum_{i=1}^{n} \| \mathbf{x'}_{1,i} \|^2 \right)^{\frac{1}{2}}. \tag{5.16}$$

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 $\bar{\mathbf{x}}_2$:

$$\mathbf{x}_0 = \bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1. \tag{5.17}$$

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

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

$$\mathcal{R} = \begin{bmatrix} (q_0^2 + q_x^2 + q_y^2 + q_z^2) & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 2(q_y q_x + q_0 q_z) & (q_0^2 - q_x^2 + q_y^2 - q_z^2) & 2(q_y q_z - q_0 q_x) \\ 2(q_z q_x - q_0 q_y) & 2(q_z q_y + q_0 q_x) & (q_0^2 - q_x^2 - q_y^2 + q_z^2) \end{bmatrix}. \tag{5.18}$$

5.2.1 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:

$$\bar{\mathbf{x}}_1 = \sum_{i=1}^{n} w_i \mathbf{x}_{1,i} \Big/ \sum_{i=1}^{n} w_i \quad \text{and} \quad \bar{\mathbf{x}}_2 = \sum_{i=1}^{n} w_i \mathbf{x}_{2,i} \Big/ \sum_{i=1}^{n} w_i \; . \tag{5.19}$$

Computing the scale factor also includes the weights

$$s = \left(\sum_{i=1}^{n} w_i \| \mathbf{x}'_{2,i} \|^2 \Big/ \sum_{i=1}^{n} w_i \| \mathbf{x}'_{1,i} \|^2 \right)^{\frac{1}{2}} ; \tag{5.20}$$

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

$$S_{xx} = \sum_{i=1}^{n} w_i x'_{1,i}\, x'_{2,i}, \quad S_{xy} = \sum_{i=1}^{n} w_i x'_{1,i}\, y'_{2,i}, \quad \textit{etc.} \tag{5.21}$$

5.2.2 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 $\mathbf{R}$ that deform the features in the first image towards their corresponding positions in the second image.

Since we have introduced an additional rotation, translation, and scale, we must modify

Equation 5.1 so as to measure distances between features in the correct coordinate frame:

$$u_i = (\frac{1}{s}\mathcal{R}^T \left[\mathbf{x}_{2,i} - \mathbf{x}_0 - \bar{\mathbf{x}}_1\right] + \bar{\mathbf{x}}_1 - \mathbf{x}_{1,i}). \tag{5.22}$$

Through the initial alignment step, we have essentially reduced virtual forces between corresponding points; the spring equation accounts for this force reduction by inverse transforming the matched points $\mathbf{x}_{2,i}$ into the finite element's local coordinate frame. The modal amplitudes $\tilde{\mathbf{U}}$ are then solved for via a matrix multiply (Eq. 5.2) or by solving the dynamic system (Eq. 3.12).

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

$$\mathbf{u}(\mathbf{x}) = s\mathcal{R}\mathbf{H}\mathbf{\Phi}\tilde{\mathbf{U}} \tag{5.23}$$

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

This method of rigid alignment and deformation is shown for the two flat tree shapes in Figure 5-1. This figure shows how the two shapes can be progressively 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.

5.3 Comparing Modal Descriptions

Once the mode amplitudes have been recovered, we can compute the strain energy incurred by these deformations by plugging into Equation 5.5. This strain energy can then be used as a similarity metric. As will be seen in the next section, we may also want to

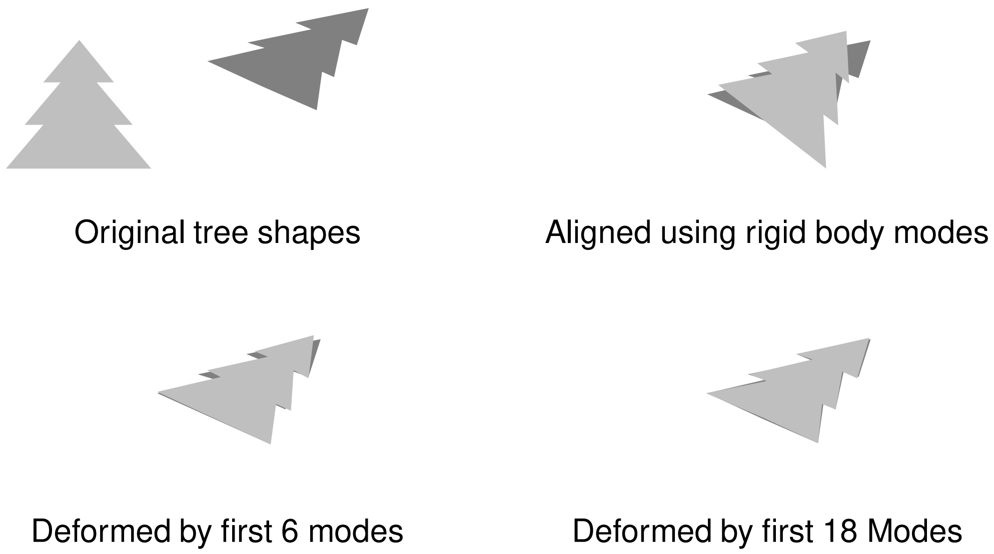


Figure 5-1: Using rigid body modes and then deformations to align two flat tree shapes. Given correspondences between corner points on both trees, we use Horn's algorithm to first recover the rigid body modes and align the two trees. The trees are aligned further using modal deformations that are recovered 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.

compare the strain in a subset of modes deemed important in measuring similarity, or the strain for each mode separately. The strain associated with the i^{th} mode is simply:

$$E_{mode_i} = \frac{1}{2}\tilde{u}_i^2 {\omega_i}^2. \tag{5.24}$$

Since each mode's strain energy is scaled by its frequency of vibration, there is an inherent penalty for deformations that occur in the higher-frequency modes.

If a metric distance function is desired, then this simple energy measure needs to be modified: strain does not satisfy one of the three axioms for a metric space[146]:

Minimality:

$$\delta(A,B) \geq \delta(A,A) = 0.$$

Symmetry:

$$\delta(A,B) = \delta(B,A).$$

The triangle inequality:

$$\delta(A,B)+\delta(B,C)\geq\delta(A,C).$$

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

There is an additional property that proves useful in defining a metric space, segmental additivity:

$$\delta(A,B)+\delta(B,C)=\delta(A,C),$$

if B is on the line between A and C.

To satisfy segmental additivity, we can take the square root of the strain energy:

$$\delta=\left(\frac{1}{2a}\sum_i\tilde{u}_i^2{\omega_i}^2\right)^{\frac{1}{2}}, \tag{5.25}$$

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

5.3.1 Modal Prototypes

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

As will be demonstrated in the image database experiments of Chapter 7, we can use distance to prototypes to define a low-dimensional space for efficient shape comparison. In such a scenario, a few prototypes are selected to span the variation of shape in each category. Every shape in the database is then aligned with each of the prototypes using modal matching, and the resulting modal strain energy is stored as an n-tuple, where n is the number of prototypes. Each shape in the database now has a coordinate in this "strain-energy-from-prototypes" space; shapes can be compared simply in terms of their Euclidean distance in this space.

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

5.3.2 Mode-Similarity-Space

As was described in 4.1.3, we can measure the distance between modes and determine how similar two shape's modes are without computing feature correspondences. This suggests an alternative coordinate space for describing distances between prototypes: mode-similarity-space. In this formulation, we match and compare modes; consequently, for each shape in the database we tally the number of modes that match each prototype's modes. Typically, mode distances are computed for only the lowest-order 25% of nonrigid modes. As before, these tallies are stored as coordinates in an n-dimensional similarity space; thus, shape similarity is proportional to the Euclidean distance in this space.

Finally, if two shapes have no modes falling within the reasonable tolerance for similarity, then the shapes will be flagged as "no modes the same." The lack of modal similarity is a strong clue that attempting correspondence and alignment is unreasonable; the shapes are probably from different categories.

Chapter 6

Modal Combinations of Models

We are given features in the original image and the recovered modal amplitudes $\tilde{\mathbf{U}}$ that describe how to align them with features in the second image. We now want to use this alignment information to warp and/or animate images. For the image warping, we want to deform the whole image based on the displacements calculated at the feature points scattered in the image. To do this, we will need to generate flow fields. For images, a flow field is a dense 2-D vector field showing where features or pixels move from one frame to another.

Given that we already have correspondences, there are standard techniques for solving this warping and interpolation problem, but it is proposed that the finite element interpolation functions h_i be used because:

1. The underlying physical model enforces elastic and smoothness constraints in warping pixels that lie between features.

2. The modal representation gives separate parametric "control knobs" that allow for selecting the types of deformations that can occur (affine *vs.* rigid *vs.* nonrigid deformations). This has the useful side-effect that image transformations can be scheduled, *e.g.,* rigid alignment first, affine next, *etc.*

3. The modal representation gives a frequency-ordered description of the deformations needed to align the two feature sets. By discarding highest-frequency modes, the image warp can be made more robust with respect to noise.

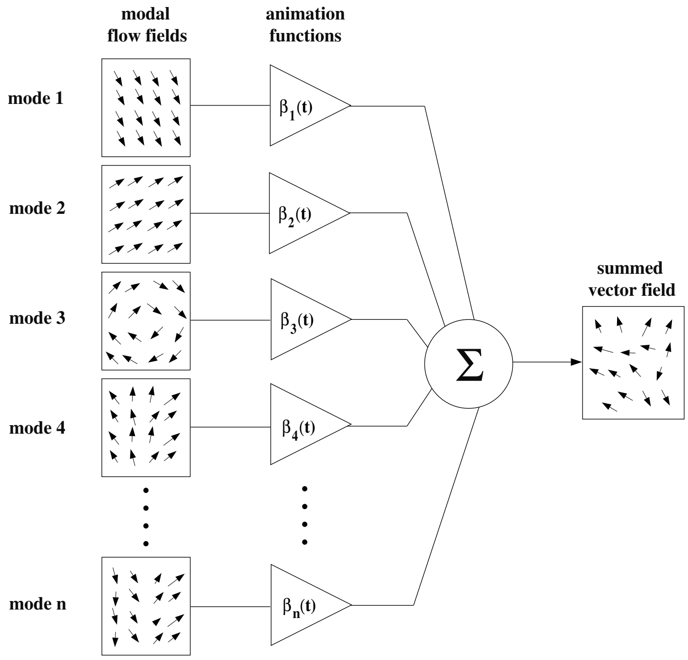


Figure 6-1: The total flow field for an image warp can be represented as the sum of many *modal* flow fields. The mixture of these modal flow fields is modulated by the animation functions $\beta_i(t)$.

Using the core technology of modal matching will yield an improved physically-based framework for linear-combinations-of-views. In the rest of this chapter we first develop the notion of *modal image warping*, where image deformations are described in terms of the linear-superposition of modal flow fields. We then formulate algorithms for describing categories of objects in terms of linear combinations of modally-warped examples, and for characterizing non-rigid motions in terms of warps from extremal views. The resulting formulation will also be useful for the computer graphics technique of image metamorphosis.

6.1 Modal Image Warping

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 Figure 6-1.

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 frames, these flow fields can be modulated by a function $\beta_i(t)$. These functions act as animation control knobs, choreographing the mixture of modal flow fields.

In general, these animation functions should satisfy the boundary conditions, *i.e.*:

$$\begin{aligned} \beta_i(t) = 0, &\quad t = 0 \\ \beta_i(t) = 1, &\quad t = 1 \\ 0 \leq \beta_i(t) \leq 1, &\quad 0 \leq t \leq 1 \end{aligned} \tag{6.1}$$

Animation functions can be simple linear ramps, sigmoid curves, splines, or any other functions, as long as these conditions are met.

6.1.1 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 $\mathbf{x}$:

$$\mathbf{u}(\mathbf{x}) = \mathbf{H}(\mathbf{x})\mathbf{U}. \tag{6.2}$$

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 Equation 6.2, thereby computing the vector field. As a result, the vector field for each non-rigid 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} non-rigid mode's vector field can be expressed in terms of the FEM displacement interpolation function:

$$\mathbf{u}(\mathbf{x}) = \tilde{u}_i \mathbf{H}(\mathbf{x})\phi_i. \tag{6.3}$$

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 need to include an additional alignment step in our method as described in the next section.

6.1.2 Modal Flow with Additional Alignment Step

If the rotation or translation needed to align the two shapes is large, then additional rigid alignment parameters are recovered as described in Section 5.2. We need to modify the modal image warping method to incorporate this additional alignment information. In the modified method, we are given the original image, the modal amplitudes $\tilde{\mathbf{U}}$, and the following additional alignment information:

$\mathbf{x}_0$	translation vector
$\mathbf{q}$	unit quaternion defining orientation
s	scale factor
$\bar{\mathbf{x}}_1$ and $\bar{\mathbf{x}}_2$	centroids

As before, we want to use the recovered alignment information to warp and/or animate images through the use of image flow fields. To do this, we will need to include the additional rotation, scale, and translation in our system, as shown in Figure 6-2. The animation functions $\alpha_1(t)$, $\alpha_2(t)$, $\alpha_3(t)$, act as control knobs for the translation, scale, and rotation, respectively. The animation functions $\alpha_1(t)$ and $\alpha_3(t)$ satisfy the same conditions described for the $\beta_i(t)$ of Equation 6.1. The animation function for scale is different; it satisfies:

$$
\begin{aligned}
\alpha_2(t) &= \tfrac{1}{s}, \quad t = 0 \\
\alpha_2(t) &= 1, \quad t = 1 \\
\alpha_2(t) \text{ is between } \tfrac{1}{s} &\text{ and } 1, \quad 0 \leq t \leq 1
\end{aligned}
\tag{6.4}
$$

The separation of rigid body from non-rigid deformation modes makes the warping more robust to large rotations – this is due to the use of quaternions rather than linearized rotation. The incremental rotation path for a warp can be defined in quaternion space using spherical linear interpolation, as is described by Shoemake[130]. As mentioned earlier, separation of rigid and non-rigid warping allows for “scheduled” warping, where groups of modes can be interpolated sequentially in time (for instance, translate first, then rotate,

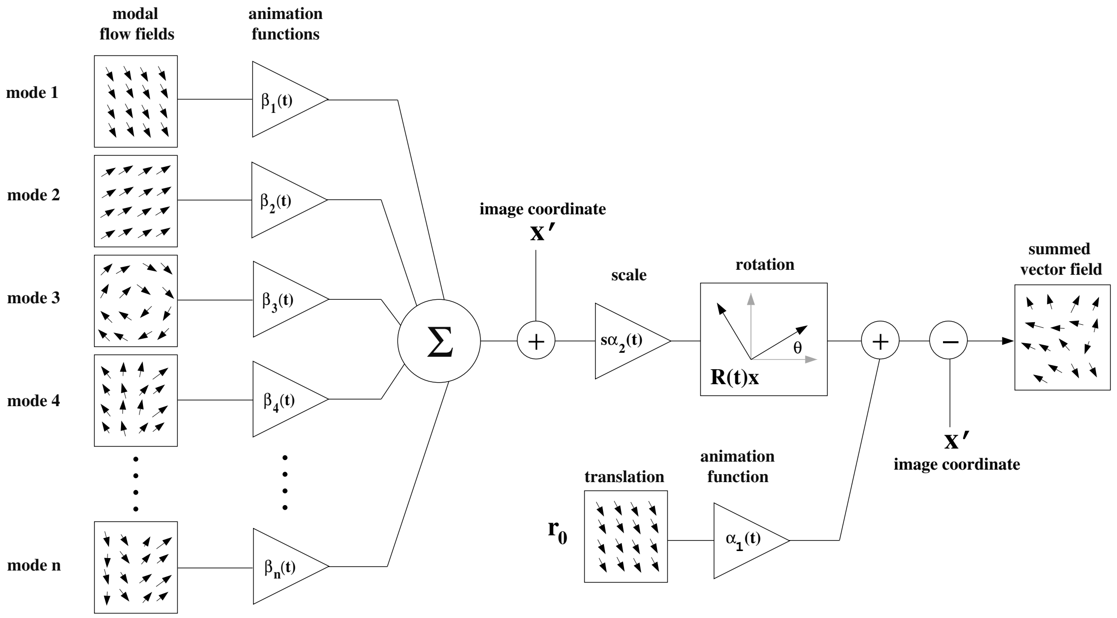


Figure 6-2: Image warping diagram modified to include the extra alignment step. The total modal flow field for an image warp passes through an extra rotation, scale, and translation stage. The flow field mixture is modulated by the animation functions $\alpha_i(t)$ and $\beta_i(t)$. Note that image coordinates $\mathbf{x}$ are translated into the local coordinate system defined by the centroid: $\mathbf{x}' = \mathbf{x} - \bar{\mathbf{x}}_1$.

then scale, and finally deform).

The memory required for storing many modal vector fields may be prohibitive. We can get around this by directly generating the summed flow field:

$$\mathbf{u}_{summed}(\mathbf{x}, t) = \alpha_1(t)\mathbf{x}_0 + \alpha_2(t)s\mathcal{R}\left[\mathbf{x}' + \mathbf{H}(\mathbf{x})\mathbf{\Phi}\bar{\beta}\tilde{\mathbf{U}}\right] - \mathbf{x}', \tag{6.5}$$

where $\bar{\beta}$ is a diagonal matrix, with the modulation functions $\beta_i(t)$ along the diagonal. Note that image coordinates $\mathbf{x}$ are translated into the local coordinate system defined by the centroid: $\mathbf{x}' = \mathbf{x} - \bar{\mathbf{x}}_1$.

6.2 Morphing: Physically-based Linear Combinations of Models

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 pixel 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 [154].

6.2.1 Morphing between two images

Using modal matching, feature matching and smooth alignment can be computed efficiently and automatically — rather than requiring hours of artist's labor. To morph between two base images, we first recover the modal deformations that align the two images as was described in Chapter 5. We then proceed to warp and combine the two images, and thereby generate an image metamorphosis.

To generate a morph, we utilize a modal image warping function $\mathbf{W}(\mathbf{B}, \tilde{\mathbf{U}}, \mathcal{A}, t)$ that descibes the warping at time t, where $\mathbf{B}$ is a base image, $\tilde{\mathbf{U}}$ is a vector of the modal amplitudes used to warp the image, and $\mathcal{A}$ is a vector of animation functions. In actuality, we will be morping between two base images $\mathbf{B}_1$ and $\mathbf{B}_2$. The animation functions for the first base image are the complements of those for the second, *e.g.*,

$$\beta_{\mathbf{B}_1,i}(t) = 1 - \beta_{\mathbf{B}_2,i} \tag{6.6}$$

is the relationship between the animaton functions for the i^{th} modes taking the first base image to the second image and *vice versa*. These animation functions are assembled into the vectors $\mathcal{A}_1$ and $\mathcal{A}_2$.

A modal morph is then defined as a weighted combination of the two base images:

$$\mathbf{I}_{morph} = \gamma(t)\mathbf{W}(\mathbf{B}_1, \tilde{\mathbf{U}}, \mathcal{A}_1, t) + (1 - \gamma(t))\mathbf{W}(\mathbf{B}_2, \tilde{\mathbf{U}}, \mathcal{A}_2, t), \tag{6.7}$$

where $0 \geq \gamma(t) \leq 1.0$ is a weighting function that controls the mixture of the two images. Given this framework, we can interpolate any image that lies between two example images.

6.2.2 Linear combinations of models

In the linear combinations of models scheme, we define an object class as all possible morphs in the basis defined by a collection of base images $\mathbf{B}_i$. Take for instance, the case where we have two basis images $\mathbf{B}_1$ and $\mathbf{B}_2$. Assume that we have the modal parameters that align the first image with the second (we will denote this vector as $\tilde{\mathbf{U}}_{1\to 2}$). All members of the class defined by the two base images can be obtained by varying the parameter t in Equation 6.7.

To map a candidate image $\mathbf{C}$ into this class, we attempt to synthesize it from the basis images. We begin by matching up image features between the candidate and each basis image, and then recovering the mode amplitudes $\tilde{\mathbf{U}}_i$ that warp each basis image $\mathbf{B}_i$ into alignment with $\mathbf{C}$. We can express the mixture of the two basis images needed to synthesize the model as:

$$t = 1 + \frac{\delta(\mathbf{B}_1, \mathbf{C})^2 - \delta(\mathbf{B}_2, \mathbf{C})^2}{2\delta(\mathbf{B}_1, \mathbf{B}_2)^2}, \tag{6.8}$$

where the function δ is the square root of strain distance metric of Equation 5.25. Given t and $\tilde{\mathbf{U}}_{1\to 2}$, we use Equation 6.7 to synthesize the model in terms of the two basis images. This formulatation enforces the constraint that the deformations used to synthesize the model be along the line between the two basis images.

Rather than using just two basis images to synthesize new images, we can generalize this concept to many basis images [11, 102, 101, 147]. In this case, we employ a modal morph defined by a weighted combination of n basis images:

$$\mathbf{I}_{morph} = \sum_i^n \psi_i \mathbf{W}(\mathbf{B}_i, \tilde{\mathbf{U}}_i, \mathcal{A}, 1), \tag{6.9}$$

where $\tilde{\mathbf{U}}_i$ is the vector mode amplitudes describing how to deform the i^{th} basis image with the candidate image, and ψ_i is a scalar controlling the mixture of the warped basis images:

$$\psi_i = \frac{1}{n-1}\left(1 - \frac{\delta(\mathbf{B}_i, \mathbf{C})}{\sum_j^n \delta(\mathbf{B}_j, \mathbf{C})}\right). \tag{6.10}$$

6.3 Example-based Shape Comparison

Recall that since we have a physical model, similarity is measured in terms of the strain energy needed to align an image with a prototype. If we use modal warps to align the grayscale image with the prototype image, we can then incorporate a second measure which accounts for the pixel-by-pixel difference between them. This two term comparison measure is similar to the one employed by Ullman and Basri, where not only the necessary deformation but also the "goodness of fit" are used to measure similarity.

If we align two images, then we can employ a distance metric that includes a measure of the pixel-wise differences between them [147]:

$$d = d_I(\mathbf{I}, \mathbf{B}, \mathbf{s}) + \lambda\delta(\mathbf{I}, \mathbf{B}) \tag{6.11}$$

The first term d_I measures the residual distance between the pixel values in the target image $\mathbf{I}$, and a basis image $\mathbf{B}$ after modal deformation $\tilde{\mathbf{U}}$, over the region defined by the support function $\mathbf{s}(x, y)$. The second term is the square root of strain metric of Equation 5.25.

For the linear combination of views paradigm, d_I measures the pixel-by-pixel difference between the synthesized image and original image, and the second term becomes a weighted sum of the strain energy needed to align each basis image with the target image. This is equivalent to determining if the target image lies within the convex hull defined by the basis images.

6.3.1 Phase Plots for Nonrigid Motion Understanding

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. The resulting strain-energy can be used to generate a phase-plot, where each axis represents distance from an extremal view.

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 approach to describing motion is related to the view-based shape recognition proposals of Ullman and Basri [147] and Poggio, *et al.* [103]. 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 [32]. 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.

Modal matching provides the needed framework 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 as will be demonstrated in the next chapter.

Chapter 7

Experiments

This chapter describes experiments in using the modal matching framework in a number of applications. We will first present experiments in using modal matching for finding corresponding features on shapes, and then use these correspondences as input to an alignment and description module. Describing deformations in terms of modal parameters makes it possible to pin-point the types of deformation needed to align shapes, thereby making it possible to determine relationships between objects and categories. These ideas are then extended to solve problems in shape-based image database search and shape categorization. Finally, the concept of physically-based linear-combinations-of-views is demonstrated on problems in nonrigid and articulated motion description.

7.1 Correspondence Experiments

In this section we will first illustrate the method on a few classic problems, and then demonstrate its performance on real imagery. In each example the feature points are treated independently; no connectivity or distinctiveness information was employed. Thus the input to the algorithm is a cloud of feature points, not a contour or 2-D form. The mass and stiffness matrices were then computed, and the $\mathbf{M}$-orthonormalized eigenvectors determined. In cases where there were greater than 100 feature points, a roughly uniform subsampling of features was used as input to the multiresolution matching scheme described in Section 4.2. Finally, correspondences were obtained as described earlier.

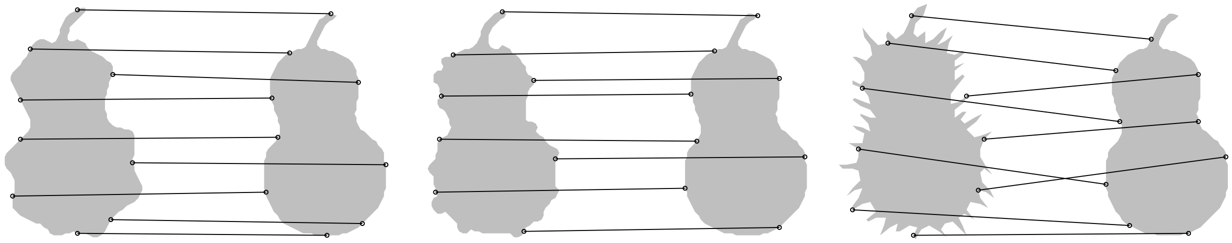


Figure 7-1: Correspondence obtained for bumpy, warty, and prickly pears. Roughly 300 silhouette points were matched from each pear. Because of the large number of data points, only two percent of the correspondences are shown in this figure.

Figure 7-1 shows a classic example taken from [110]. 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. A roughly-uniform subsampling of approximately one quarter of the features was used to build a finite element model. Correspondences were then computed for all the features 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. This is because the low-order symmetries (eigenvectors) of all the pears are similar. The fact that the low-order descriptions are similar allows us to recognize that the shapes are "the same," and to establish a good point-to-point correspondence.

Figure 7-2(a) shows two wrenches: one prototypical straight-handled wrench, and the other a bent wrench. The first 28 modes were computed for both wrenches, and were compared to obtain the correspondences shown. The fact that the two wrenches have similar low-order symmetries (eigenvectors) allows us to recognize that two shapes are closely related, and to easily establish the point correspondences. Roughly 100 silhouette points were matched from each wrench. For clarity, only a few of the computed correspondences are shown in the figure.

Figure 7-3(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 7-3(b). Roughly 400 points were sampled from each hand silhouette and correspondences were computed using the first 32 modes. As in the previous example, only two percent of

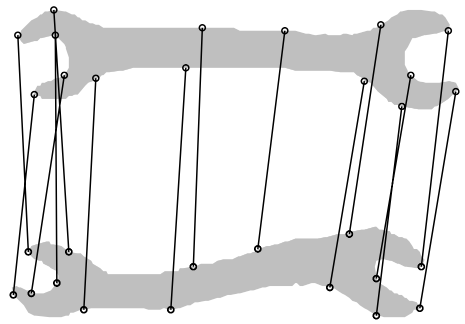


Figure 7-2: Correspondence found for two wrenches, one straight and one bent, together with the obtained correspondence. The 28 low-order modes were computed for each tree and then correspondences were determined using the algorithm described in the text.

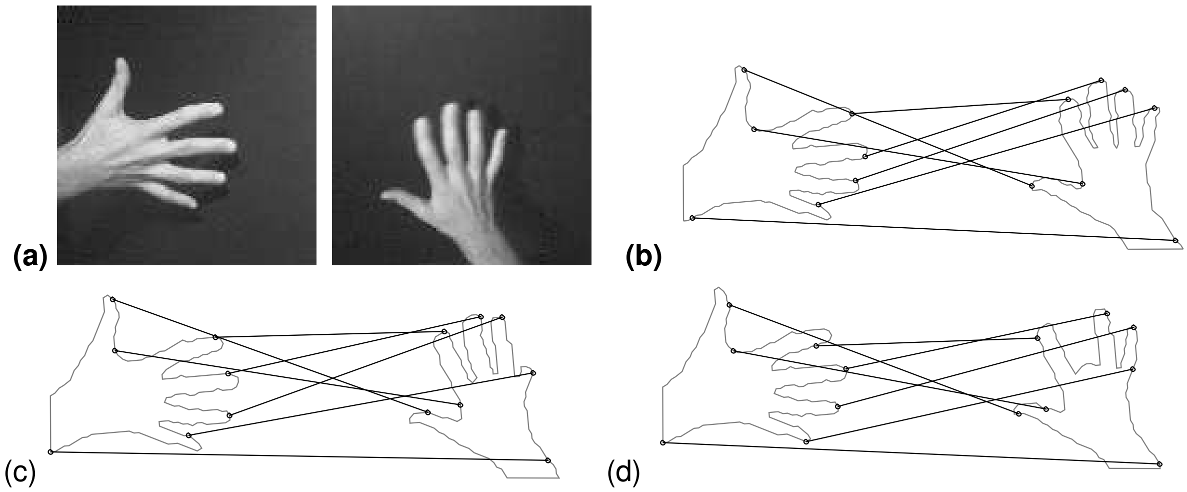


Figure 7-3: (a) Two hand images, (b) correspondences between silhouette points, (c),(d) correspondences after digital surgery. Roughly 400 points were sampled from each hand silhouette. Correspondences were computed for all points using the first 32 modes. For clarity, only correspondences for key points are shown in this figure.

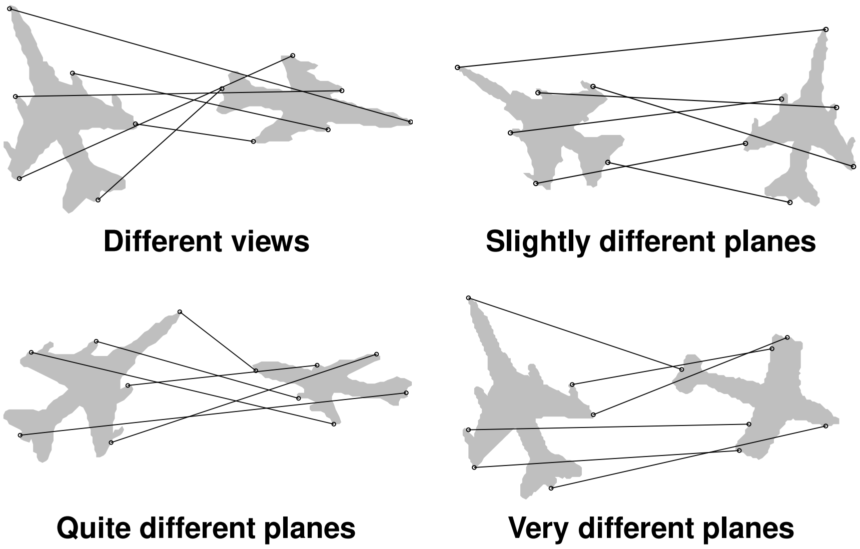


Figure 7-4: Correspondence obtained for outlines of different types of airplanes. The first example shows the correspondences found for different (rotated in 3D) views of the same fighter plane. The others show matches between increasingly different airplanes. In the final case, the wing position of 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.

the correspondences are shown.

Figures 7-3(c) and (d) show the same hand data after digital surgery. In Figure 7-3(c), the little finger was almost completely removed; despite this, a nearly perfect correspondence was maintained. In Figure 7-3(d), the second finger was removed. In this case a good correspondence was still obtained, but not the most natural given our knowledge of human bone structure.

The next example, Figure 7-4, uses outlines of three different types of airplanes as seen from a variety of different viewpoints (adapted from [155]). 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.

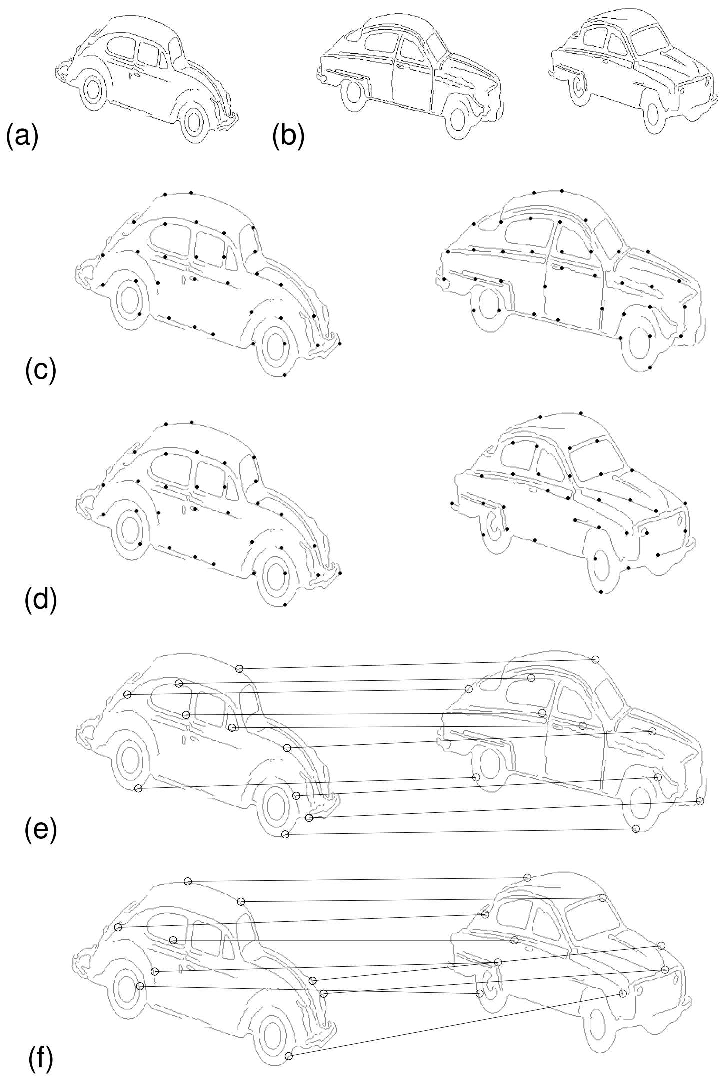


Figure 7-5: Finding correspondence for one view of a Volkswagen (a) and a two views of a Saab (b) taken from [147]. Each car has well over 1000 edge points. Note that both silhouette and interior points can be used in building the model. As described in the text, when there are a large number of feature points, modal models are first built from a uniform sub-sampling of the features as is shown in (c,d). In this example, roughly 35 points were used in building the finite element models. Given the modes computed for this lower-resolution model, we can use modal matching to compute feature matches for the higher-resolution. Correspondences between similar viewpoints of the VW and Saab are shown in (e), while in (f) a different viewpoint is matched (the viewpoints differ by 30°). Because of the large number of data points, only a few of the correspondences are shown in this figure.

Our final example is adapted from [147] and utilizes multi-resolution modal matching to efficiently find correspondences for a very large number of feature points. Figure 7-5 shows the edges extracted from images of two different cars taken from varying viewpoints. Figure 7-5(a) depicts a view of a Volkswagen Beetle (rotated 15^o from side view) and Figure 7-5(b) depicts two different views of a Saab (rotated 15^o and 45^o). If we take each edge pixel to be a feature, then each car has well over 1000 feature points.

As described in Section 4.2, when there are a large number of feature points, modal models are first built from a roughly uniform sub-sampling of the features. Figures 7-5(c) and 7-5(d) show the subsets of between 30 and 40 features that were used in building the finite element models. Both silhouette and interior points were used in building the model.

The modes computed for the lower-resolution models were then used as input to an interpolated modal matching which paired off the corresponding higher-resolution features. Some of the strongest corresponding features for two similar views of the VW and Saab are shown in 7-5(e). The resulting correspondences are reasonable despite moderate differences in the overall shape of the cars. Due to the large number of feature points, only a few of the strongest correspondences are shown in this figure.

In Figure 7-5(f), the viewpoints differ by 30^o. Overall, the resulting correspondences are still quite reasonable, but this example begins to push the limits of the matching algorithm. There are one or two spurious matches; *e.g.*, a headlight is matched to a sidewall. We expect that performance could be improved if information about intensity, color, or feature distinctiveness were included in our model.

7.2 Alignment and Description

Figure 7-6 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 7-4. 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 5.2. The modal strain energy was computed using Equation 5.5.

The graphs in Figure 7-6 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 7-6(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 7-6(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 7-6(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 7-6(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.

7.2.1 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 7-7 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,

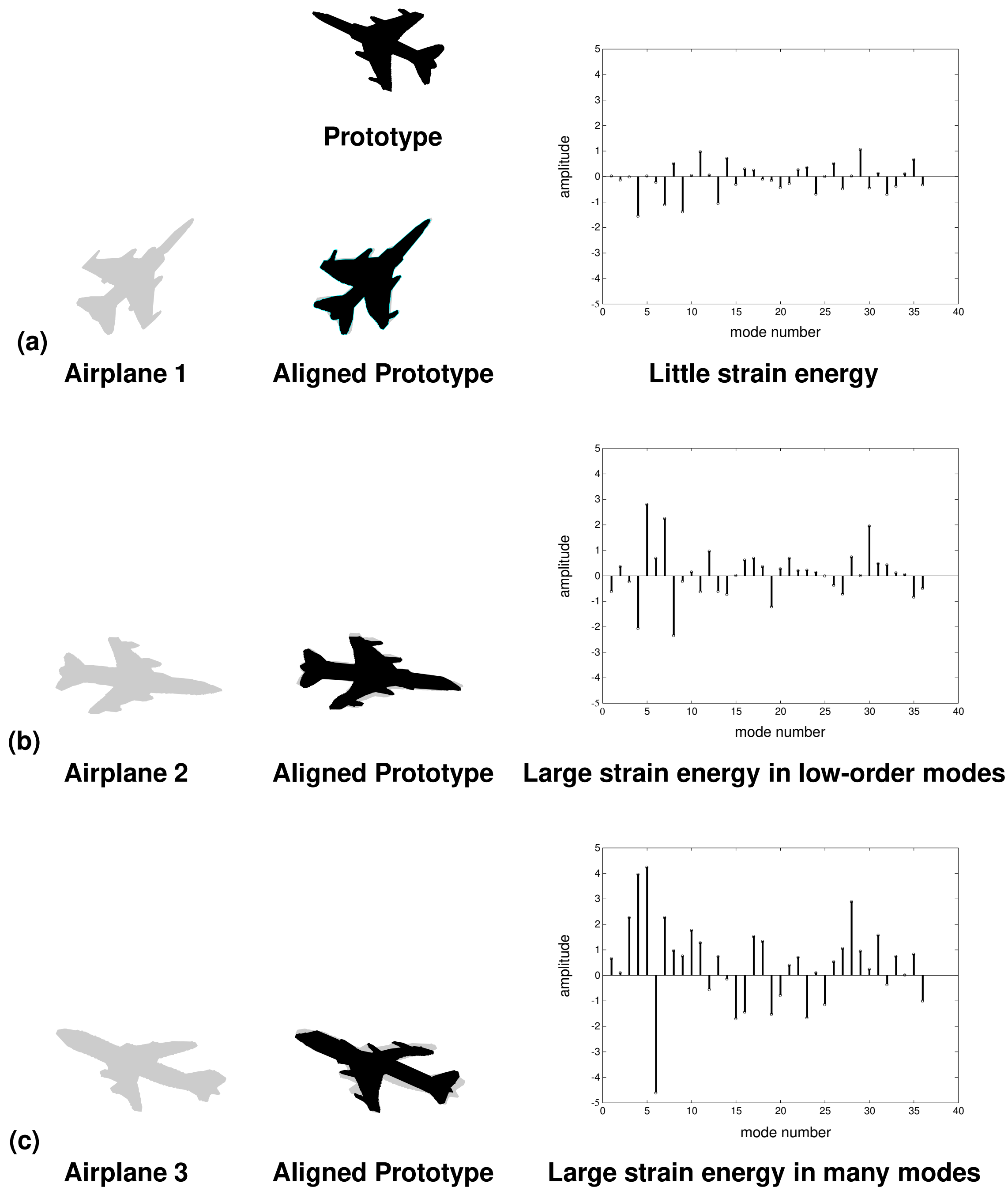


Figure 7-6: Describing planes in terms of a prototype. The graphs show the 36 mode amplitudes used to align the prototype with each target shape. (a) shows that similar shapes can be aligned with little deformation; (b) shows that viewpoint changes produce mostly low-frequency deformations, and (c) shows that to align different shapes requires both low and high frequency deformations.

we want to compare different hand tools. The prototype is a wrench, and the two target objects are a bent wrench and hammer. Silhouettes were extracted from the images, and thinned down to approximately 80 points per contour. Using the strongest matched contour points, we then recovered the first 22 modal deformations that warp the prototype onto the other tools. A rotation, translation, and scale invariant alignment stage was employed as detailed in Section 5.2.

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 7-8 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 7-8) cannot be described simply in terms of a few deformations of the wrench.

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

7.3 Recognition of Objects and Categories

In the next example (Figures 7-9 and 7-10) we will use modal strain energy to compare three different prototype tools: a wrench, hammer, and crescent wrench. As before, silhouettes were first extracted and thinned from each tool image, and then the strongest corresponding contour points were found. Mode amplitudes for the first 22 modes were recovered and used to warp each prototype onto the other tools. Total CPU time per trial

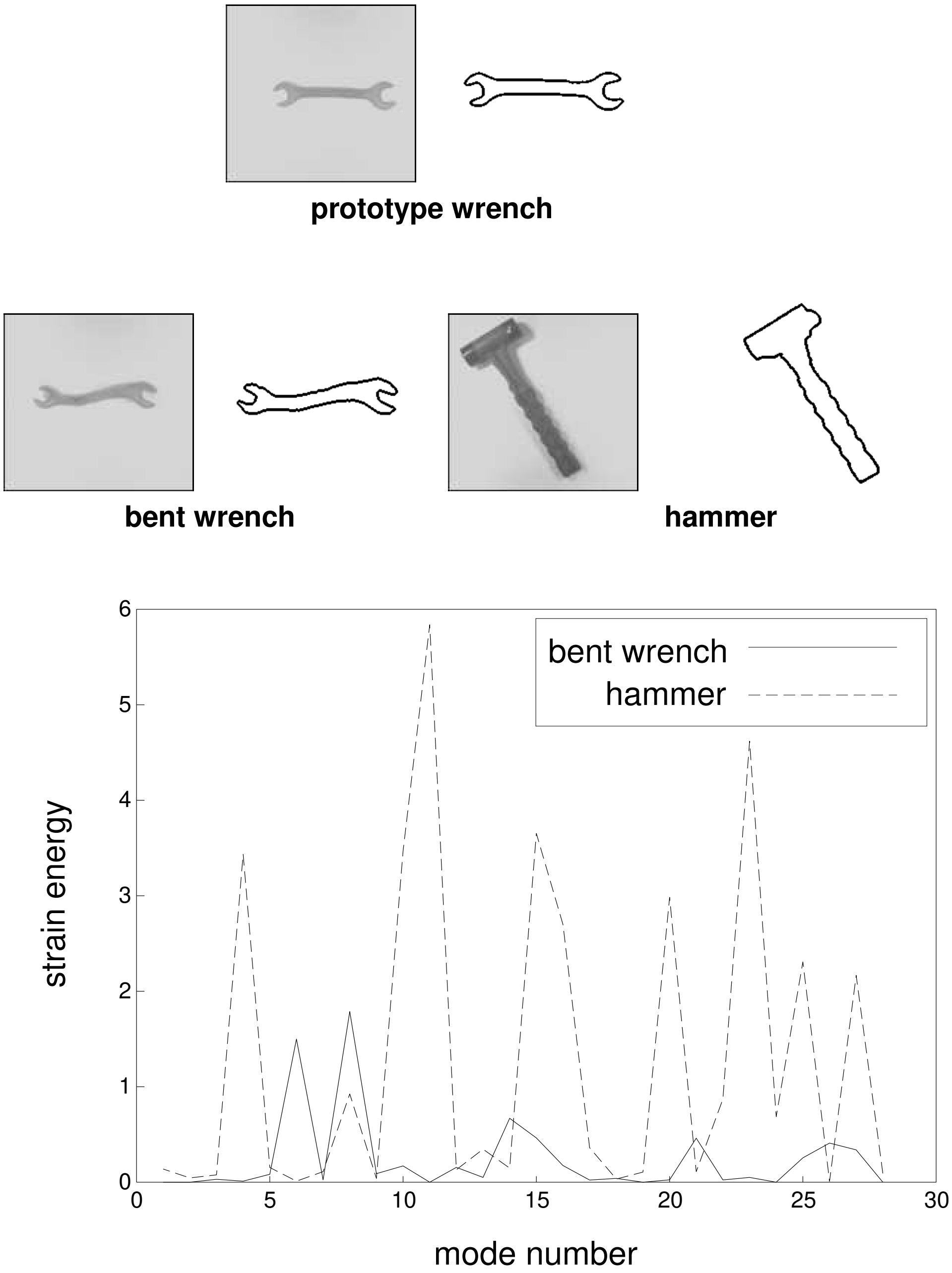


Figure 7-7: Describing a bent wrench and a hammer in terms of modal deformations from a prototype wrench. Silhouettes were extracted from the images, and then the strongest corresponding contour points were found. Using these matched contour points, the first 28 modal deformations that warp the prototype's contour points onto the other tools were then recovered and the resulting strain energy computed. A graph of the *modal strain* attributed to each modal deformation is shown at the bottom of the figure.

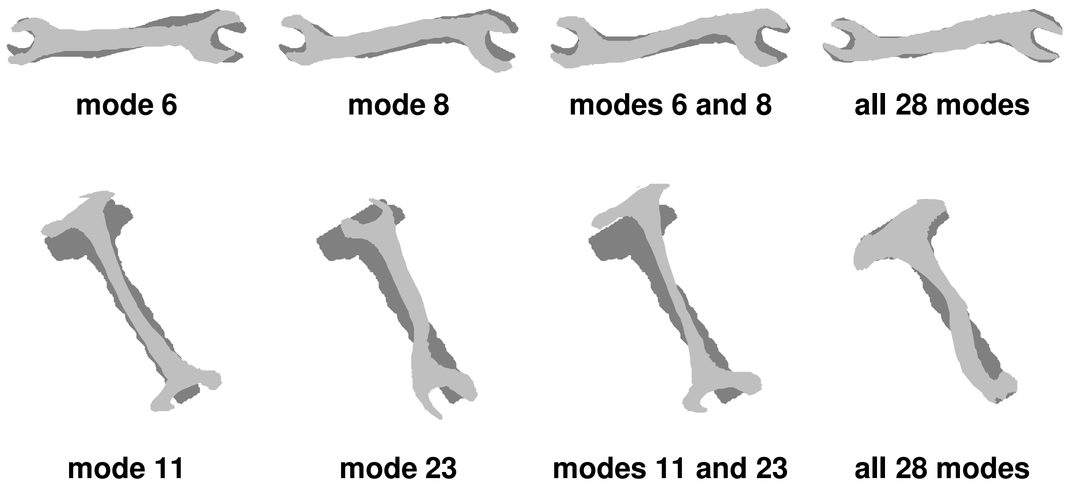


Figure 7-8: Using the two modes with largest strain energy to deform the prototype wrench to two other tools. The figures demonstrates how the top two highest-strain modal deformations contribute to the alignment of a prototype wrench to the bent wrench and a hammer of Figure 7-7.

(match and align) averaged 11 seconds on an HP 735 workstation.

The resulting modal strain energy was then used as a similarity metric in Photobook, an image database management system developed at the MIT Media Lab [95]. Using Photobook, the user selected the image at the upper left, and the system retrieved the remaining images sorted by strain energy (shape similarity) from left to right, top to bottom. The similarity measure is shown below each image.

Figure 7-9 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 7-10 continues this example using as prototypes the hammer and a single-ended wrench. Again, the modal strain energy that results from deforming the prototype to each tool is shown below each image.

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

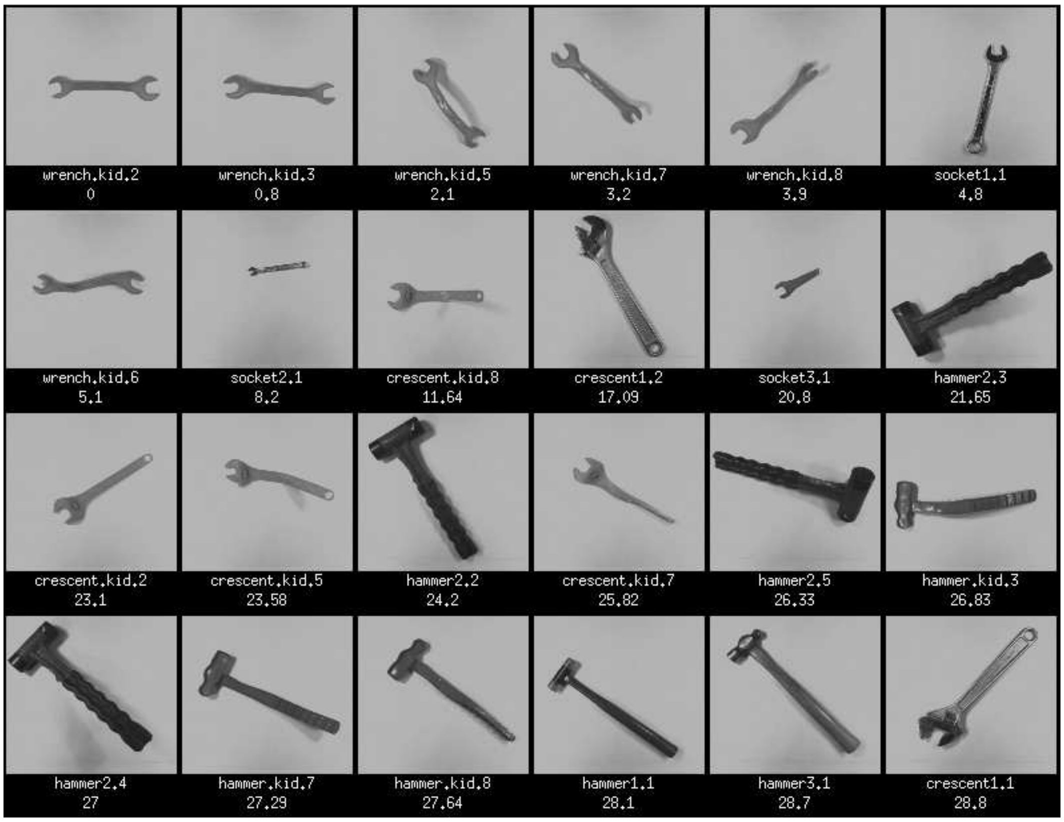


Figure 7-9: Using modal strain energy to compare a prototype wrench with different hand tools. The user selected the image at the upper left, and the Photobook system returned the remaining images sorted by similarity (strain energy) from left to right, top to bottom. As in Figure 7-7, silhouettes were first extracted from each tool image, and then the strongest corresponding contour points were found. Mode amplitudes for the first 22 modes were recovered and used to warp the prototype onto the other tools. The modal strain energy that results from deforming the prototype to each tool is shown below each image in this figure.

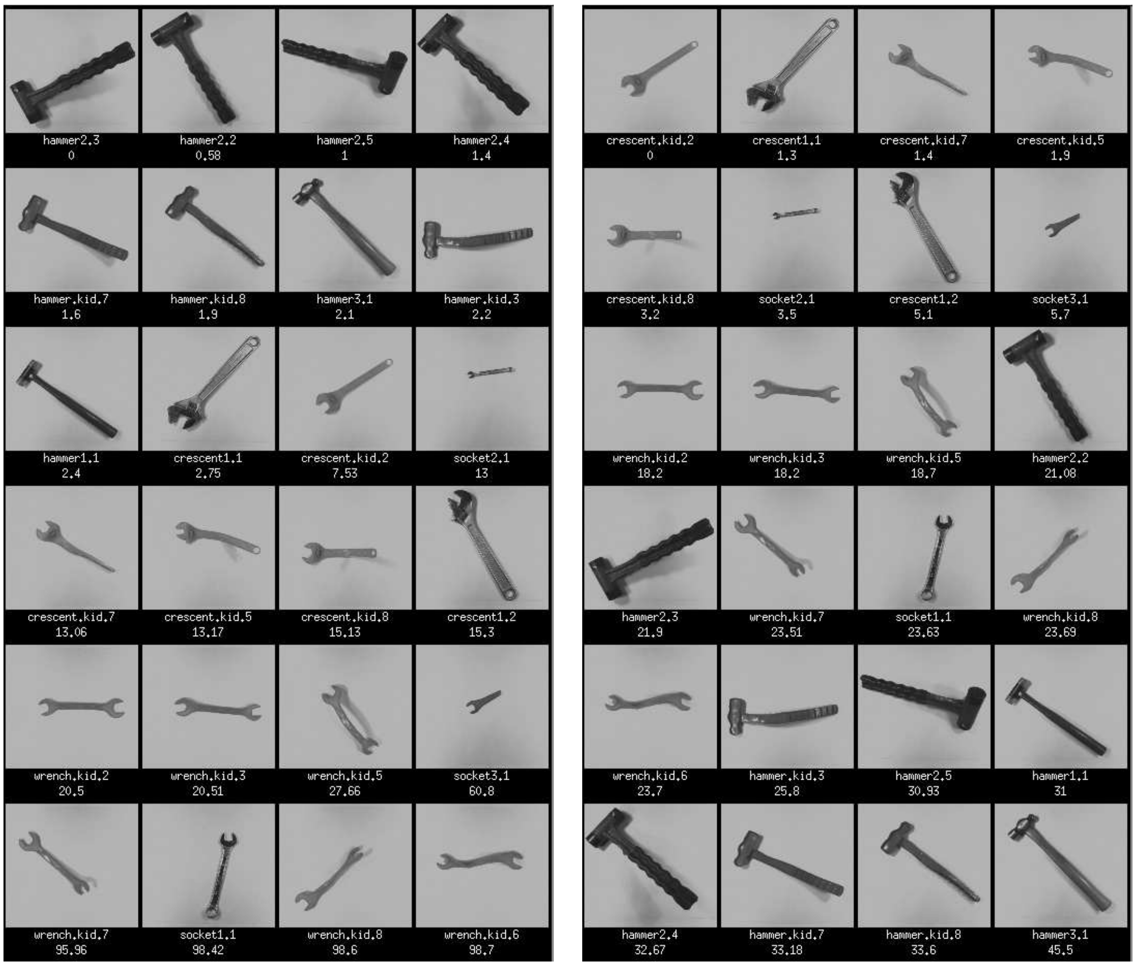


Figure 7-10: Using modal strain energy to compare a hammer with different hand tools, and a prototype crescent wrench with different hand tools. The user selected the image at the upper left; the remaining images were sorted by similarity from left to right, top to bottom. The modal strain energy that results from deforming the prototype to each tool is shown below each image.

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*.

7.4 Structuring Image Databases for Interactive Search

The last set of examples shows progress towards structuring image databases into categories in terms of a few prototype shapes. The images in this experimental database are digitized from children's field guides [1, 46]. Currently, there are 96 images in the database: 12 rabbits and 74 tropical fish. Each image depicts an animal from nearly the same canonical viewpoint (side view). The goal is to structure a potentially huge image database in such a way as to allow multiple *interactive* shape-based searches.

We use the prototype-based shape description method formulated in Section 5.3.1, where each shape's strain-energy distance to the prototypes was precomputed and stored for interactive search later. First, for each image, a support map and edge image was computed, a finite-support shape model was built, and then the eigenmodes were determined as before. For the shapes in this experiment, approximately 60-70 finite element nodes were chosen so as to be roughly-regularly spaced across the support region.

Each shape in the database is then modal matched to seven prototype images. There were two rabbit prototypes (shown in Figure 7-11) and five fish prototypes (shown in Figure 7-12). The prototype images were selected by a human operator so as to span the range of shapes in the database. For rabbit prototypes, we chose one seated rabbit (Fig. 7-11(a)) and one standing rabbit (7-11(b)). For fish prototypes, we chose prototypes that span the

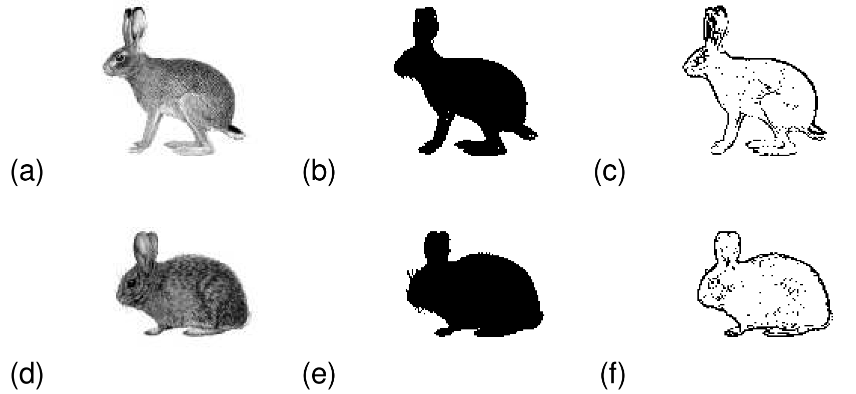


Figure 7-11: The two prototype rabbit images (a) a European Hare, and (d) a Desert Cottontail. Their associated support maps are shown in (b,e) and their edge maps in (c,f). The original color images were digitized from [1].

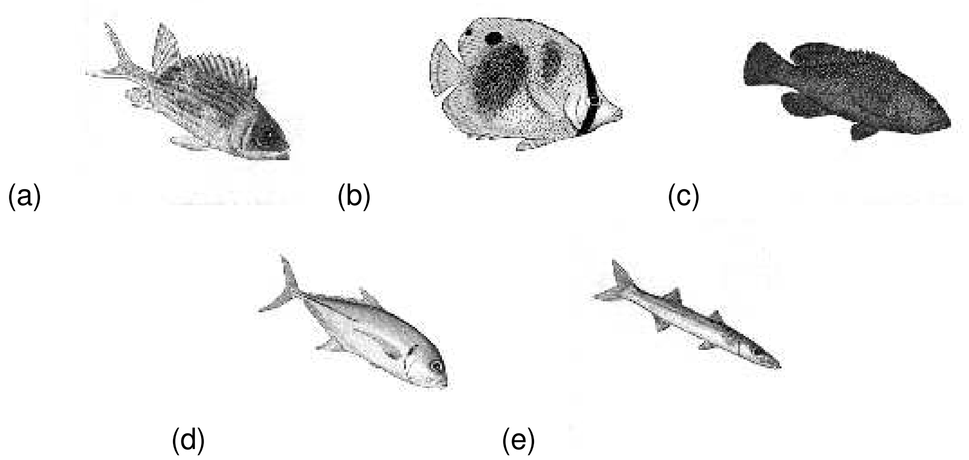


Figure 7-12: The five prototype fish used in the image database experiments: (a) Squirrel Fish, (b) Spot Fin Butterflyfish, (c) Coney, (d) Horse Eye Jack, and (e) Southern Sennet.

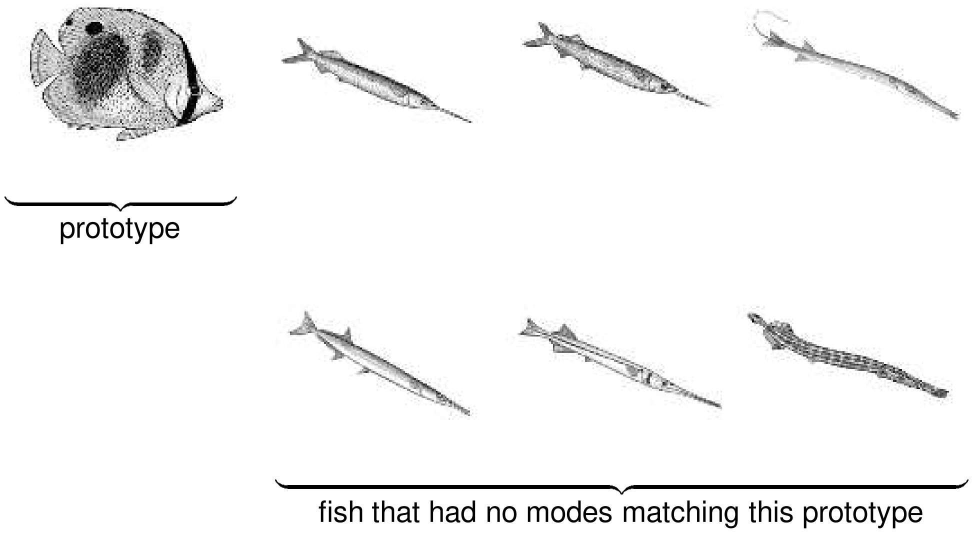


Figure 7-13: Six fish had no modes that came within tolerance of matching modes for the Butterfly Fish prototype in Figure 7-12(b), and are clearly not in the Butterfly Fish category.

range from skinny fish (Fig. 7-12(e)), to fat fish (7-12(b)), and from smooth fish (7-12(c)) to prickly or pointy-tailed fish (7-12(a,d)).

Not all shapes in the database have similar modes (similarity is measured to within a threshold). For instance, shapes from the rabbit class do not have similar modes to those shapes in the fish class, and *vice versa*. Sometimes, as is shown in Figure 7-13, even shapes within the same category do not have similar modes. In this particular case, the modes of the wide-bodied, Butterfly Fish prototype of Fig. 7-12 did not match well with the modes of the most narrow-bodied fish. When modes are nowhere near being similar, no attempt at alignment and strain energy computation is made. Such shapes are simply flagged as being "not at all similar" to a particular shape prototype.

When a shape's modes match a particular prototype, then the modal alignment parameters are recovered, and the strain energy is computed. The resulting modal strain energies are stored as a seven-tuple that maps the shapes into a space where each axis represents deformation from one of the prototypes.

Figure 7-14 shows a scatter plot of the two-dimensional "rabbit subspace." The graph's x-axis depicts the square-root of strain energy needed to align the European Hare prototype with each rabbit shape, while the y-axis shows the energy needed to align the Desert

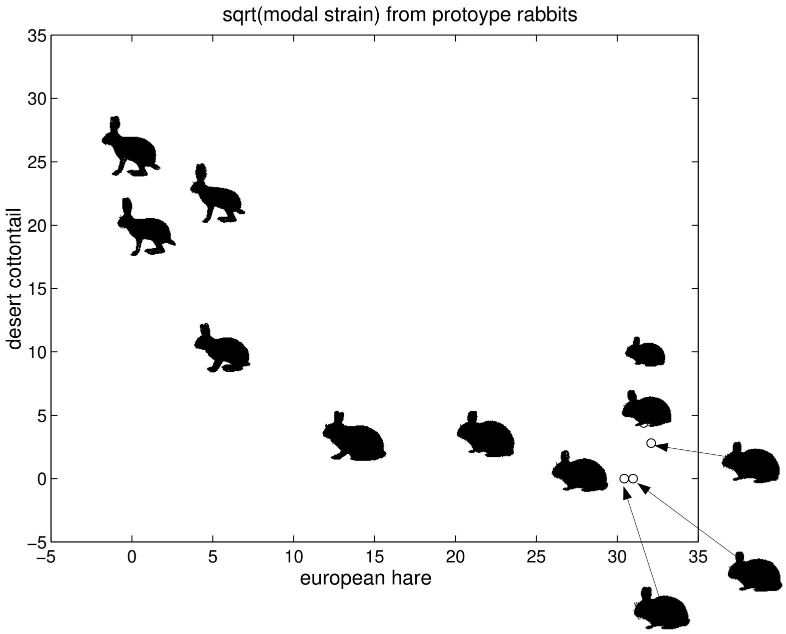
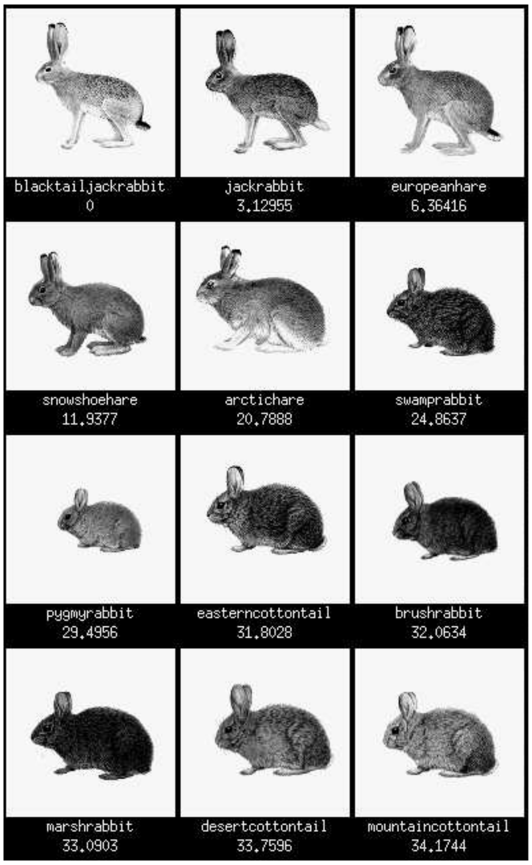


Figure 7-14: Scatter plot of square-root modal strain energy for rabbit prototypes used in the image database experiment. Each axis depicts the square-root of strain energy needed to align a shape with a rabbit prototype. Thus each rabbit shape has a coordinate in this space. The rabbits are clustered in terms of their 2-D shape appearance: long-legged, standing rabbits cluster at the top-left of the graph, while short-legged, seated rabbits cluster at the bottom right. There are two rabbits that map between clusters, showing the smooth ordering from long-legged, to medium-legged, to short-legged rabbits in this view-space.

Cottontail prototype with each rabbit shape. Each rabbit shape has a coordinate in this strain-energy-from-prototype subspace. As can be seen, the rabbits are clustered in terms of their 2-D shape appearance: long-legged, standing rabbits cluster at the top-left of the graph, while short-legged, seated rabbits cluster at the bottom right. There are two rabbits that map between clusters, showing the smooth ordering from long-legged, to medium-legged, to short-legged rabbits in this view-space.

The next examples demonstrate how the full seven-dimensional coordinate system can be used for interactive database search. Interactive database searches were conducted using the Photobook system [95] on HP 735 workstation. In this system, the user can select example images and then the computer displays a number of “similar” images ordered in terms of a similarity metric. In this case, the similarity metric was Euclidean distance in

(a)


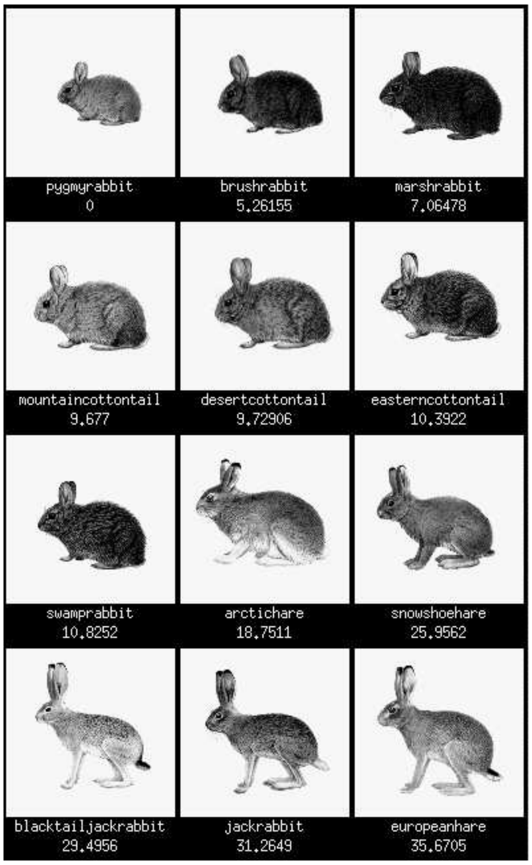
(b)


Figure 7-15: Searching an image database for similarly-shaped rabbits in the Photobook system. Using the prototype-based coordinate system described in the text, we can search the image database for similarly shaped animals. The user chooses an example image, and the system retrieves other rabbit shapes, ordered in terms of their similarity to the user's example. This figure shows two examples of the ordering that resulted in such searches: (a) Blacktail Jackrabbit, and (b) Pygmy Rabbit. The system retrieved a reasonable ordering of similar rabbits that progressed from long-legged to short-legged rabbits. The number shown below each image in the figure is the Euclidean distance in prototype strain-space.

our strain-energy-to-prototype space.

Figure 7-15 shows the results of searches for similarly shaped rabbits. In Fig. 7-15(a), a Blacktail Jackrabbit was selected. The most similar rabbit was a Jack Rabbit, the next a European Hare, *etc.*. The matches are shown in order, starting with the most similar. The number shown below each image in the figure is the Euclidean distance in the prototype strain-space. As can be seen, the system returned a reasonable ordering, starting with other long-legged rabbits, moving to medium-legged rabbits, and finally short-legged (or seated) rabbits. Figure 7-15(b) shows similar performance for a search based on the Pygmy Rabbit shape.

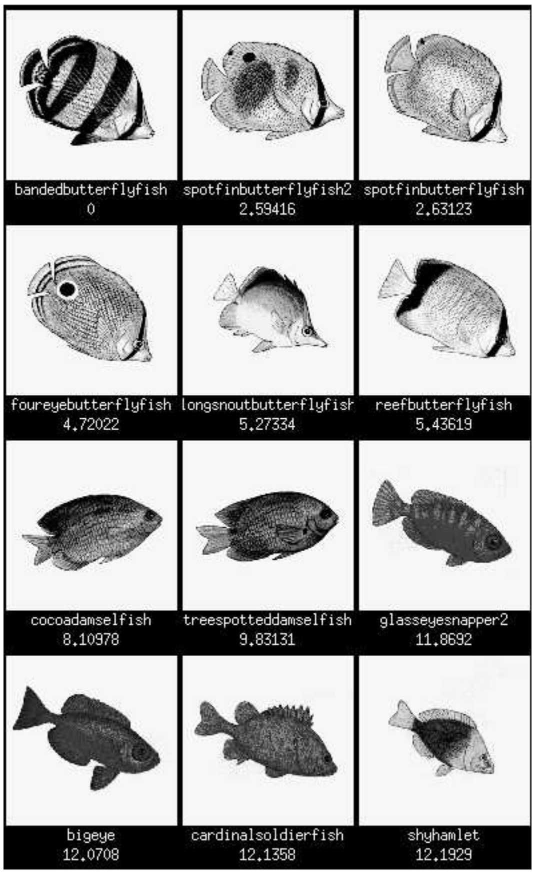
As suggested in Chapter 5, the similarity of two shapes' modes can be measured directly, thus avoiding the need to recover alignment parameters. In our experiments, there appears to be only slightly improved performance in using strain energy over this more easily-computed modal similarity.

The database searches in Figures 7-16 and 7-17 were conducted using distance in mode-similarity-space. In Fig. 7-16(a), a Banded Butterflyfish was selected. The matches are shown in order, starting with the most similar. Based on mode-similarity-distance, the system retrieved the animal shapes that were closest to the Banded Butterfly Fish shape (other Butterfly Fish, and other fat-bodied fish). In the second search, shown in Fig. 7-16(b), a Trumpet Fish was selected. In this case, the system retrieved similar long and skinny fish. In both searches, the fish judged "most similar" by the system appeared on the same page in the field guide, and in the same taxonomic category.

Figure 7-17 continues this example, this time searching for shapes most similar to a Crevalle Jack and a Dog Snapper. Again, the matches are shown in order, starting with the most similar. The shapes most similar to a Crevalle Jack (Fig. 7-17(a)) are other fish with similar body and tail shapes. In this case, the system rates Jolt Head Porgy over a closer relative (Yellow Jack). This is fairly reasonable, since all are closely-related, open water fish.

In the last example, Fig. 7-17(b), the user selected a Dog Snapper. Again the system rated fish from the same pages in the field guide as "most similar." In each example, search and display took less than a second on an HP 735.

(a)



(b)

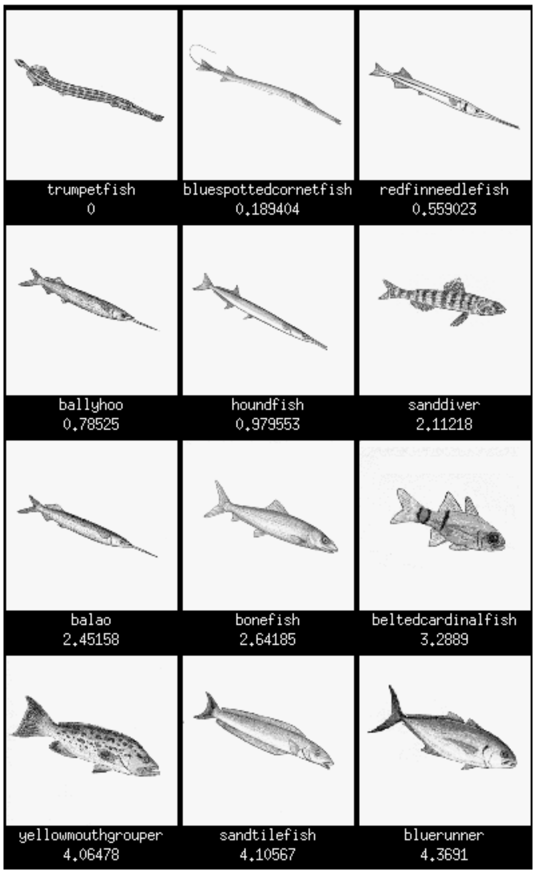
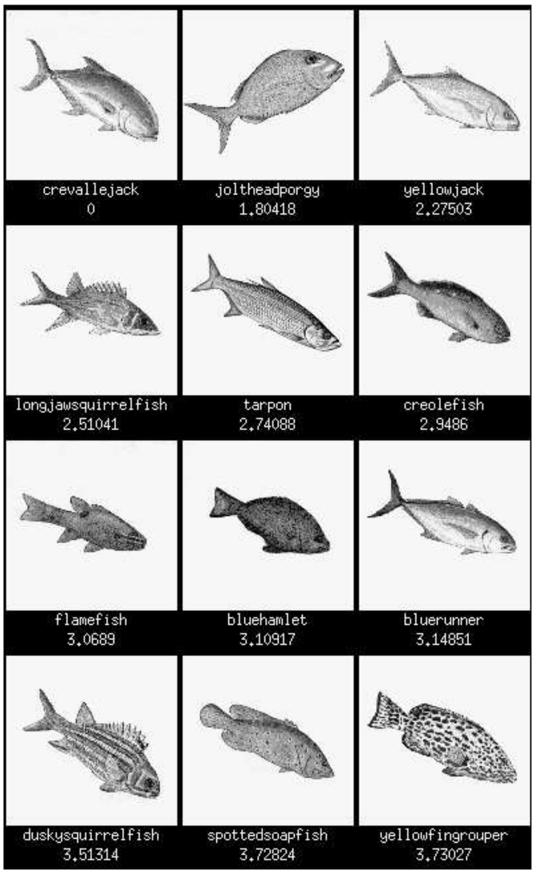
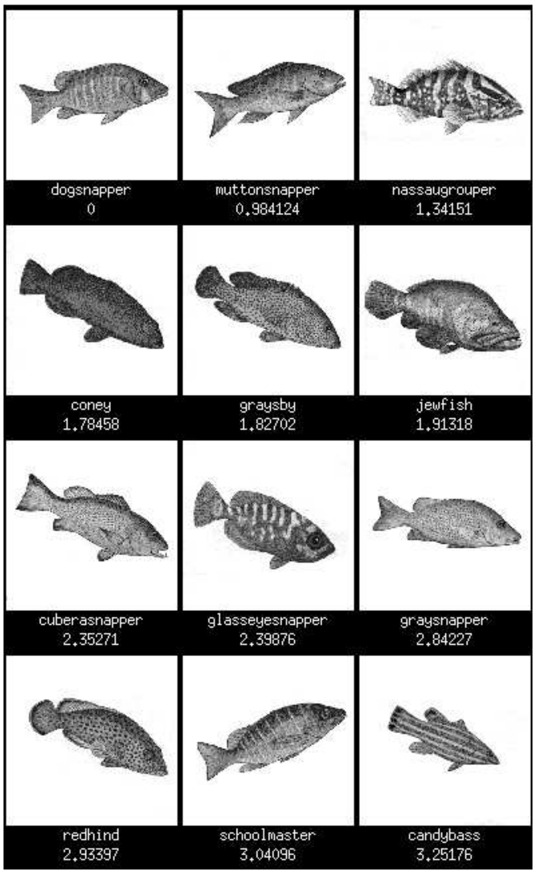


Figure 7-16: Searching an image database for similarly-shaped fish. In this example, distance in mode-similarity-space was used as a shape similarity metric. This figure shows two examples of the ordering that resulted in searches for similar fish: (a) a Banded Butterfly Fish, and (b) a Trumpet Fish. The matches are shown in order, starting with the most similar. Based on mode-similarity-distance, the system retrieved the animal shapes that were closest to the Banded Butterfly Fish shape (other Butterfly Fish, and other fat-bodied fish). In the second search the system retrieved similar long and skinny fish. In both searches, the fish judged "most similar" by the system appeared on the same page in the original field guide book, and in the same taxonomic class.



(a)



(b)

Figure 7-17: Searching an image database for similarly-shaped fish (continued). Two more examples of the ordering that resulted in searches for similar fish: (a) a Crevalle Jack, and (b) a Dog Snapper. Again, the matches are shown in order, starting with the most similar. The shapes most similar to a Crevalle Jack are other fish with similar body shapes and pointed tails (other open water fish). In the second search the user selected a Dog Snapper. The system rated fish from the same page in the field guide as "most similar." In each example, search and display took less than a second on an HP 735.

Database queries like those just described were performed for each of the 72 fish images in the database. Overall, another fish from the same page in the field guide was judged as most similar over 70% of the time. This type of similarity judgement performance is an encouraging result, since fish appearing on the same page in the field guide are usually in the same taxonomic category. To gain enhanced performance in capturing animal taxonomies, we suspect that modal matching would need to be part of a combined system that includes local feature and color information.

In related work, Strachan [132] has developed a prototype fish recognition system based on both shape and color discriminants. The ultimate goal is to develop a vision system that can perform on-board sorting of fish for commercial fishing boats. The shape component of the system uses two simple descriptors: aspect ratio, and area ratio of the front and back halves of the fish. In the preliminary system, it was assumed that the fish was laying flat, long axis aligned with the horizontal, and that there was only one fish present on a uniform background.

Under these restricted conditions, Strachan deals with deformation by using a grid-based technique. To place this deformable grid, a medial axis is first extracted (based on the contour from a threshold image), and then a regular grid laid out perpindicular to the medial axis at ten regular intervals. A combined shape and color discriminant analysis was done using the widths of the fish at each of the ten stations along the medial axis, and the average color in each grid rectangle. This preliminary system has been tested for over 20 different fish varieties, getting promising recognition results ($>$ 90%).

Unfortunately, the underlying shape representation is not rotation invariant, cannot handle occlusion or clutter, and is not really robust to deformation (the grid layout is somewhat arbitrary and noise-prone due to reliance on medial axis extraction). A modal model could be used to overcome many of these limitations. In addition, modal models could be used to build aspect graphs that encode 3-D rotation, thereby loosening the restriction that the fish lay flat and horizontal.

7.5 Modal Combinations of Models

As can be seen from previous image database experiments, a subset of prototype shapes can be used to define a low-dimensional coordinate system for shape, resulting in an interactive framework for shape-based searches of large image databases. This concept will now be extended to the linear-combinations-views framework.

Figure 7-18(a) shows a *shape space* defined by three prototype models. As before, correspondences were determined and similarity (strain energy) was computed. Each edge is labeled with its associated strain. Traveling along an edge in this triangle performs a linear blend, using the modal deformations, from one prototype model to another. Thus, each edge of the triangle describes a family of models which can be represented as linear combinations of the two prototypes. Similarly, we can describe an entire family of shapes by moving around inside the triangle defined by three models.

Adding a fourth model to the triangle creates a pyramid, unless the new model can be exactly described as a linear combination of the prototype models. Figure 7-18(b) shows how the fourth plane model was 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 missing nacelles, for instance). As a result, the fourth model does not lie in the plane defined by the three base models. The distance between the new plane and the triangle of base shapes is the similarity between the new plane and the *class* of shapes defined by linear combinations of the prototype models. Using this similarity measure, we can decide whether or not the new shape is a member of the class defined by the prototype models.

The same methods can also be extended to model grayscale information. Figure 7-22 shows the modal blending of two hand images as we move along the edge between the two base images (a) and (b). This image warping is accomplished by describing the deformation of the whole image in terms of the displacements calculated at feature points (typically

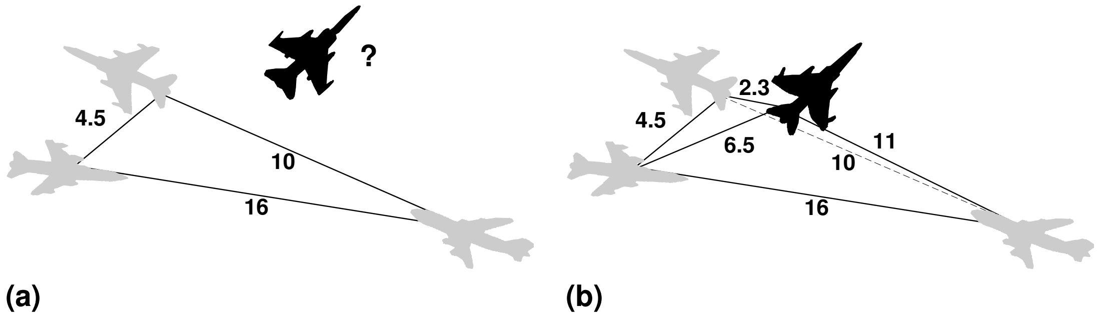


Figure 7-18: Given three gray models in (a), we can define a triangle with edge lengths proportional to the amount of strain needed to align each model; thus, each model is a vertex in this triangle. When we encounter a new model (shown in black), we want see how it can be synthesized from deformed versions 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.

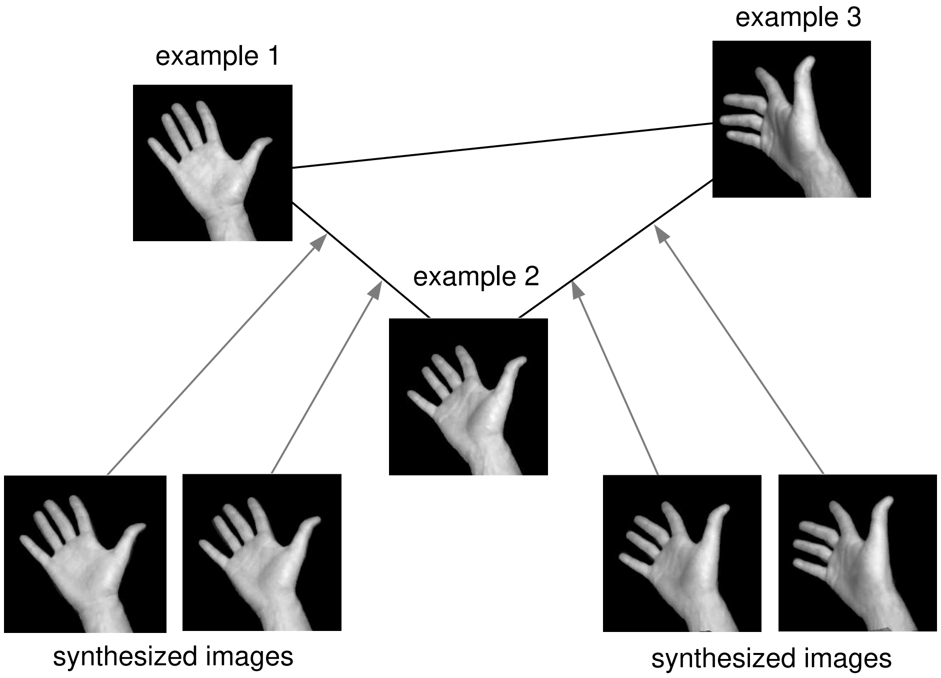
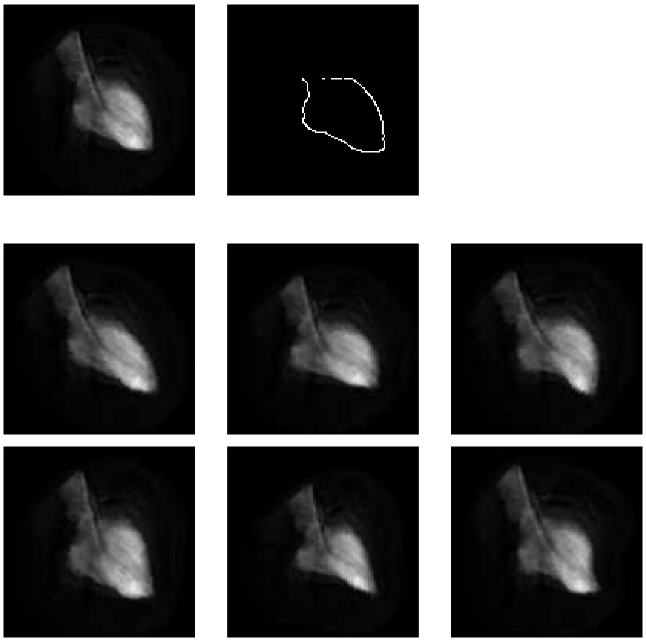


Figure 7-19: Physically-based linear combinations of images. Using three example images, we form a triangle that describes a shape space. Modal matching is used to determine corresponding features, and modal flow fields are used to warp and linearly combine the examples to synthesize intermediate, novel images. By using a combined distance metric, we can measure a shape’s similarity in terms of: (a) the deformation energy needed to synthesize the candidate shape from examples, and (b) the pixel-by-pixel differences between the synthesized image and the candidate image.



(a)

(b)

Figure 7-20: An original image and its extracted contour are shown in (a). The first six nonrigid modes for the heart image are shown in (b).

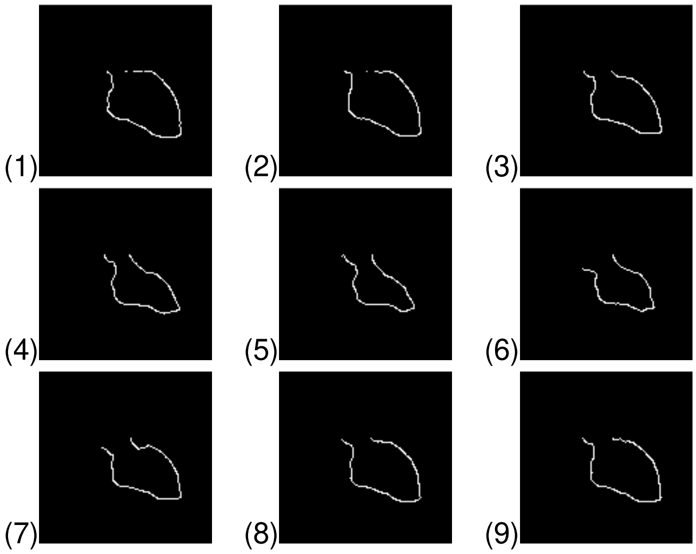
edge points) for which correspondences have been determined. These displacements can be calculated by generating flow fields using the finite element interpolation functions.

7.5.1 Nonrigid and Articulate Motion Tracking

Fig. 7-20(a) shows an X-ray image of a heart ventricle together with its bounding contour. Fig. 7-20(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. 7-21(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 approximately 60 contour points from each image. Point correspondence 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,



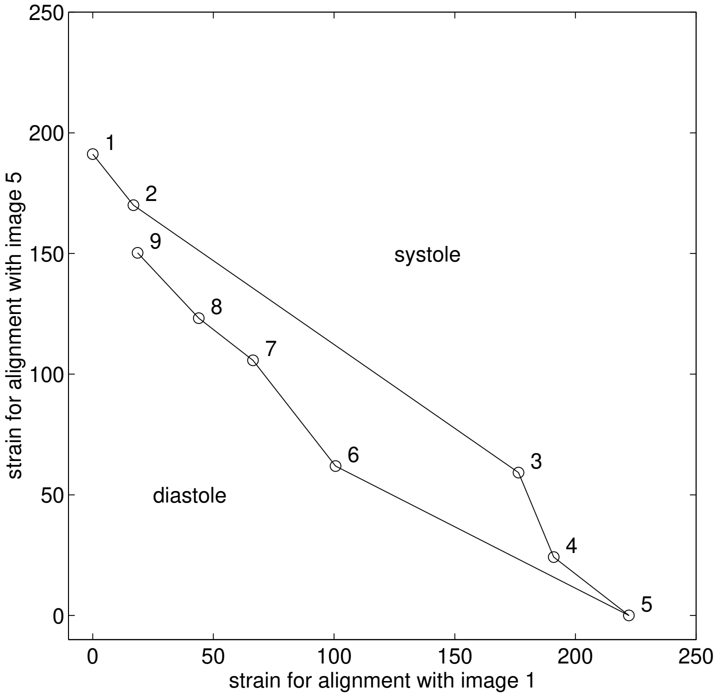


Figure 7-21: Representing a beating heart in terms of warps of extremal views. Given a modal model for the heart as shown in Fig. 7-20, we code a series of frames (1-9) in which the bounding contours deform as the heart beats. Frames 1 and 5 represent extremal views of the heart's deformation, and are 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.

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. 7-21. 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 [129].

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. 7-22(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. 7-22(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. 7-23 shows the first nine nonrigid modal warps used to deform the first extremal view in Fig. 7-22(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. As will be discussed in the next chapter, our modal framework can be extended to model articulated shapes and anisotropic materials. In addition, we will describe methods for physics-based active part detection along the lines of [57].

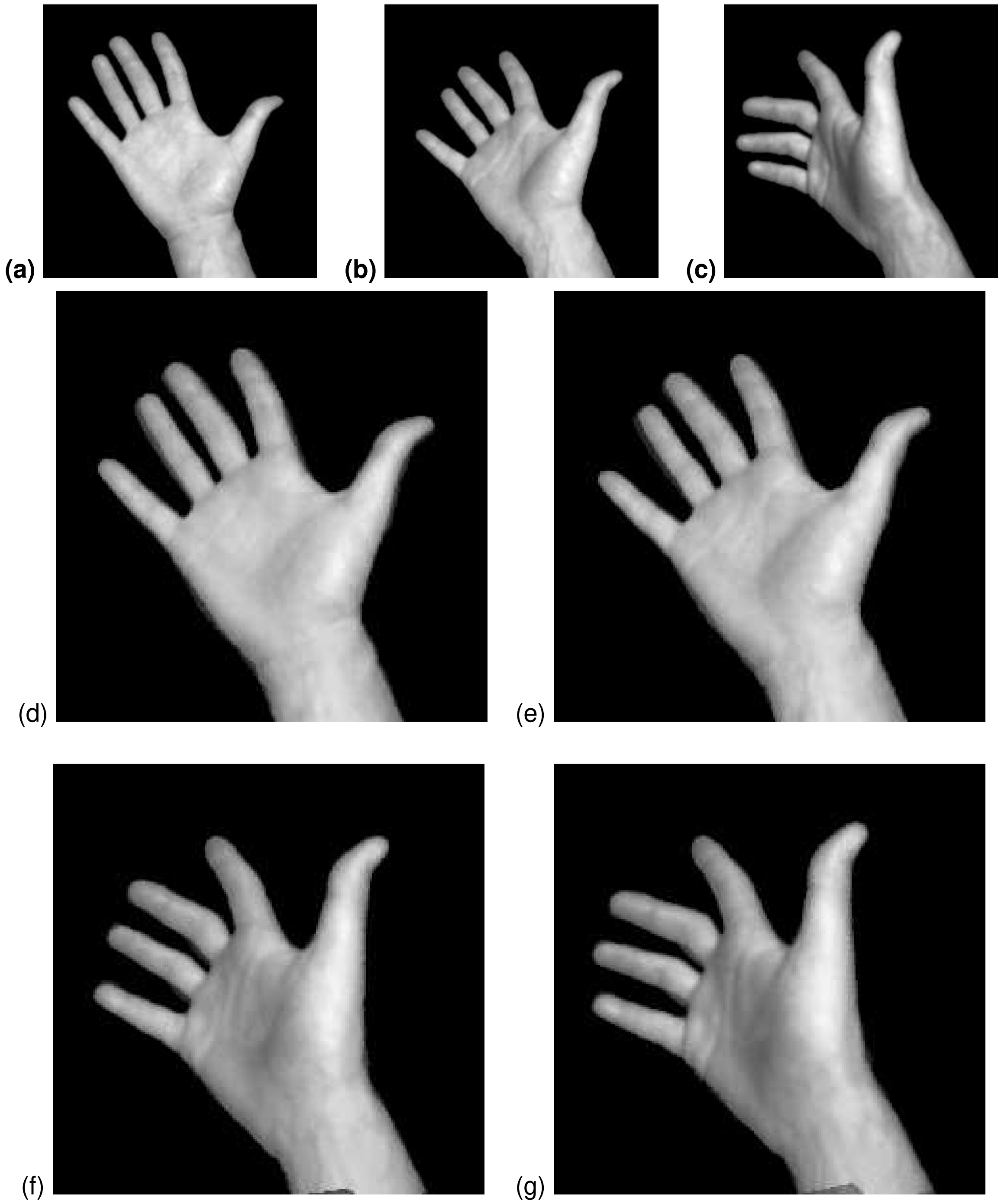


Figure 7-22: Modal image synthesis. Given example images (a,b,c) and their corresponding features, we can use modal flow fields to synthesize intermediate images as linear combinations of modal-deformed versions of the examples. Images (e,f) were generated by morphing between examples (a) and (b), and (g,h) were generated by morphing between (b) and (c). In-between images were produced by the image warping system described in Chapter 6.

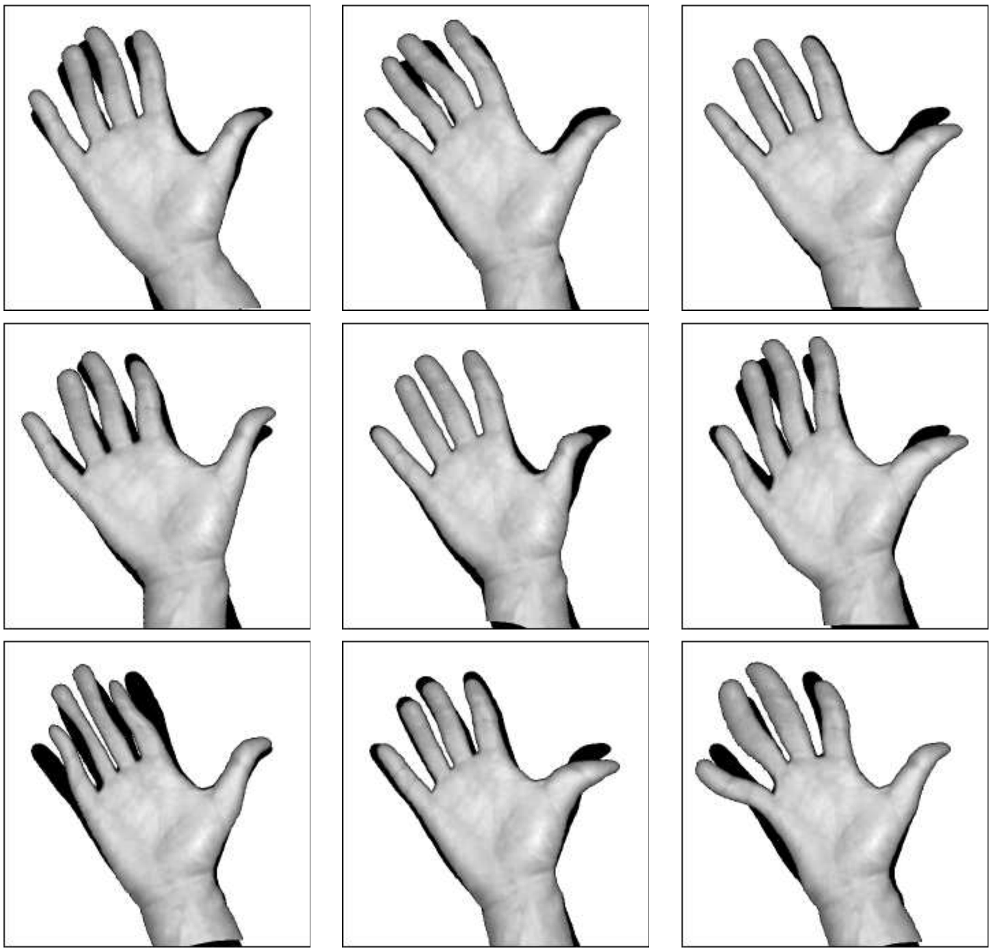


Figure 7-23: The first nine nonrigid modal warps for the first extremal view in Fig. 7-22(a). The metamorphosis between extremal views is described in terms of modal warps for an 2-D elastic hand shape using the algorithm described in the text. The warped grayscale images are drawn over the original hand image silhouette (shown in black).

Chapter 8

Discussion

8.1 Shape-based Image Database Search

One of the main motivations for this research was to provide improved shape representations for query by image content. While the shape comparison algorithms developed in the machine vision and pattern recognition communities can serve as a good starting point for developing shape-based image database search methods, retrieval by shape is still considered to be one of the most difficult aspects of content-based image search [39]. The results presented in the previous chapter for databases of animals and hand tools represent some clear improvements over previous shape-based search methods.

IBM's Query By Image Content system (QBIC) [39, 85] is perhaps the most advanced image database system to date; it is available as a commercial product. QBIC can perform searches that combine information about shape, color, and texture. For its shape-based search, the system utilizes a whole cadre of shape features: area, circularity, eccentricity, major axis of inertia, and higher-order algebraic moment invariants. These shape features are combined into one feature vector, and shape similarity is measured using a weighted Euclidean distance metric. As input, the system assumes non-occluded, planar shapes that are represented as a binary image. Shape-based search in QBIC is primarily contour-based, and cannot deal well with nonrigid deformation. Algebraic moment invariants [136] were intended for modeling rigid objects only. In addition, the higher moments are dominated by points that are furthest from the centroid; therefore, they are highly susceptible to outliers. Similar moments do not necessarily guarrantee perceptually similar shapes.

Other shape indexing schemes have been based on local boundary features [79, 48], and are therefore not very robust to noise, scale, and sampling. Another system, proposed by Chen [26] identified 2-D aircraft shapes using elliptic Fourier descriptors. Because it is Fourier descriptor-based, Chen's system suffers from problems with samping and parameterization. Jagadish introduced a multidimensional indexing scheme that offered the advantage that it could index images much faster than previous techniques [56]. However, the system had limited descriptive power, because the shape similarity measure was too simple (the area difference between two shapes) and the underlying shape representation was polyhedral (representing shapes in terms of K-d-b trees of overlapping minimum bounding rectangles).

In contrast to previous formulations, the FEM integrals used in the modal model formulation provide greater robustness to sampling, outliers, and missing data. Furthermore, modal models provide quasi-invarinace to different types of nonrigid deformation, while also providing an ordered, orthogonal, *encoding* of the nonrigid deformation that relates a candidate shape to a shape prototype or shape category. Finally, as was demonstrated in Figure 7-5, the modal representation does not have the requirement that a shape be represented as a contour or binary image.

Besides being able to select example images, users can make a sketch of what they are looking for in QBIC. A reduced edge-map representation is then computed and corellated in eight by eight blocks. Shape similarity is then measured as the sum of the corellation scores for each local block. While this representation is handy for detecting the presence of a shape in an image, and is somewhat robust to occlusion, it is prohibitively slow, and cannot deal well with nonrigid deformation. Modal matching has yet to be tested in a sketch-based search scenario. It is likely that modal matching would do better at dealing with deformation, but may not perform better under occlusion.

8.1.1 Matching Human Similarity Judgments

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

For human shape similarity judgments, sometimes scale and rotation invariance are important, other times not [82]; it is therefore desirable to duplicate this performance in our image database search algorithms. In QBIC, the weighted metric allows for subset selection, and thus it provides selective invariance to size and orientation. Modal matching also provides this invariance to size and orientation, but unlike any of the shape representations used in QBIC, a modal representations can also be made invariant to affine deformations, and thus selectively invariant to changes in camera viewpoint.

In an effort to better duplicate human similarity judgments, an experimental version of QBIC was developed and tested [116]. The extended system incorporates curvature and turning angles, in addition to the Hausdorff distance [53], and parametric curve distance and its first and second derivatives. Unfortunately, none of these metrics performs consistently well, particularly when there is nonrigid deformation. Curvature descriptors, while invariant to rotation and translation, are exterior-based, and therefore unstable and sampling-dependent [82]. The Hausdorff distance is very sensitive to outliers and is reliable for compact shapes only. Finally, parametric curve representations are highly susceptible to outliers, sampling, and parameterization problems.

As mentioned above, the modal formulation avoids many of these problems with robustness and nonrigid deformation. More importantly, the modal representation provides deformation "control knobs" that correspond qualitatively with human's notions of perceptual shape similarity [90, 96]. Shape is thought of in terms of an ordered set of deformations from an initial shape: starting with bends, tapers, shears, and moving up towards higher-frequency wiggles.

8.1.2 Speed of Image Database Search

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

Until recently, shape-based search for a QBIC database with over 1000 images could take over 40 seconds. To improve performance, a truncated principal components analysis is computed for all the feature vectors, and truncated to only the first two principal components. Database searches are done using a candidate shape's projection onto these two components. This speeds database search time up so as to make it negligible (search is now bound by I/O).

As demonstrated in the previous chapter, the modal matching system search speeds are comparable. Searches on a database containing nearly 100 image takes less than a second (including image display) on an HP 735 workstation. Without the use of category information, search time in our system scales linearly on the number of shapes in the database. The notion of building up protoype-based modal categories for content-based search is a unique and powerful idea. It allows databases to be structured into taxonomic trees for inference, and can be used for further improvement of search time/performance. Furthermore, the underlying energy-based representation has a clear connection to statistical estimation and learning, as will be explored further in the next section.

8.2 Modal Models, Estimation, and Learning

As has been pointed out by Boult [20], there has been a well understood link between energy formulations and statistical estimation for over 25 years now. Splines were perhaps some of the first "physically-based" models employed in statistical estimation [62]; they are particularly well-suited to modeling data sampled from a Markov Random Field

(MRF), where Gaussian noise has been added [15, 44]. The same principles hold true for regularization [15, 47, 104, 139], where the energies of a physical model can be related directly with measurement and prior probabilities used in Bayesian estimation [134, 43].

Following [20, 106], energy-based formulations can be related to maximum likelihood estimation in the following way. Suppose we are given a set of N observations y_i, each having an error that is independently random and normally distributed around $y(x)$, and each having the known variance σ^2. We assume that these observations were generated by known x, and by a known function $y(x) = y(x; \mathbf{a})$. We want to estimate the unknown parameters $\mathbf{a}$ that maximize the joint probability P of the observations, namely:

$$P = \prod_{i=1}^{N} \exp\left(-\frac{[y_i - y(x_i; \mathbf{a})]^2}{2\sigma^2}\right). \tag{8.1}$$

Maximimizing the product of the probabilities is equivalent to minimizing its negative logarithm:

$$-\ln P = \sum_{i=1}^{N} \frac{[y_i - y(x_i; \mathbf{a})]^2}{2\sigma^2}. \tag{8.2}$$

Thus, if we express our energy measure as $E = -\ln P$, then minimizing the energy of a spline or regularization model is equivalent to maximum likelyhood estimation.

In modal models, as in regularization, a direct connection can be drawn between the uncertainty measures familiar in statistical estimation and the energy formulation of our physical model. In statistical estimation, we need to combine knowledge about the uncertainty of our measurement data with confidence in our prior model. In our energy-based paradigm, the confidence in the prior model is expressed both as the model's initial resting shape (the mean) and the local material stiffnesses of the model (the covariance).

If we assume independent distribution of measurement data, the stiffness matrix and inverse covariance matrix are equivalent [29, 76]. Similarly, there is an inverse relationship between the eigenmode frequencies and the principal variances (modulo a scale factor) [77]. In this thesis, we used modal strain energy for comparing objects. This strain energy is computed by taking the dot product of the modal amplitudes with the modal frequencies. The modal amplitudes are decoupled, each having a "variance" that is inversely proportional to the mode's eigenvalue. Our modal strain energy measure therefore yields a Mahalonobis distance metric that can be directly related to multi-dimensional Gaussian probability distributions.

Thus it is plausible to use principal components analysis to learn an shape category's eigenmodes from training data. Using a modal model as an initial estimate, we would then iteratively learn the "true" modes via recursive principal components analysis [87, 88]. As a result, we would obtain a regularized learning scheme in which the initial covariance matrix, $\mathbf{\Omega}^{-2}$ is iteratively updated to incorporate the observed modal parameter covariances.

8.3 Material Properties, Preset Parameters, and Robustness

8.3.1 Material Properties

For modal matching, we have assumed an isotropic material of uniform thickness, and that both shapes are modeled by the same material. Under these assumptions, how the material parameters are chosen has no observed affect on the modal matching correspondence algorithm. What happens if we loosen these constraints somewhat, allowing the two shapes to be modeled of different materials?

As was detailed in Section 3.4, our finite element formulation has three parameters that describe the material: mass density ρ, Young's modulus of elasticity E, and the Poisson ratio ν. The first two parameters uniformly scale the mass and stiffness matrices respectively. Changing these material parameters results in a uniform scaling of the eigenvalues; however, it does not affect either the mode ordering or the mode shapes.[1] Since our correspondence algorithm relies on the eigenvalue order not their scale, varying ρ and/or E will have no affect on the modal matching algorithm.

Changing the Poisson ratio, however, does change the mode shapes. Intuitively, this third parameter controls a shape's resistance to shear. As a result, allowing two shapes to have different values of ν can alter the resulting mode shapes and their ordering, thereby hampering the correspondence computation.

[1]This holds true if we replace the $\mathbf{M}$-orthonormality constraint, $\mathbf{\Phi}^T\mathbf{M}\mathbf{\Phi} = \mathbf{I}$, with the orthonormality constraint, $\mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{I}$.

8.3.2 Preset Parameters

Aside from material properties, the modal matching system has three preset parameters: the FEM interpolation function widths, the threshold for what constitutes a matched mode, and the threshold for picking out strong correspondences from the affinity matrix. Experiments with each of these have yielded the following observations and heuristics:

Basis Function Widths

In the formulation used in the experiments, the finite element basis function widths were constant for all nodes of a shape. It was assumed that the shape's nodes were roughly evenly-spaced, and thus it was reasonable to set the basis function size to the average distance from each feature to its nearest neighbor. Under this assumption, eigenmode shapes and magnitudes are scale-invariant — only the eigenmode's eigenvalues change as the object is scaled.

If inter-feature distances vary greatly, then it may be necessary to use variable-width Gaussian interpolants to avoid an ill-conditioned finite element model. This problem was not dealt with in the thesis, and it is not clear how variable-width basis functions will affect the invariance of eigenmode ordering and shape over scale.

Lastly, in the hand experiments it has been noticed that if the basis function widths get large enough with respect to overall object size, then there will be a coupling between the finger tips. In other words, the Gaussians get wide enough to glue the fingers together. As a result, the modes for the hand will be "mitten modes," rather than modes that wiggle the fingers as if they were independent parts attached at the palm of the hand. To solve this problem, we could modify the interpolant at finger tip nodes to exclude the basis functions at other finger tips.

Eigenmode Match Threshold

As was mentioned in Chapter 4, there is a sign ambiguity in matching eigenvectors, and as shapes get more and more different, their modes can change ordering. Thus, before feature correspondences can be computed, eigenmodes for each shape have to be matched and sign-corrected. Given all of the modes computed for both objects, we select the subset of corresponding low-order modes whose dot product falls below a threshold. These matched modes form the object-centered coordinate system used for computing feature correspondence.

For this thesis, problems with sampling density were avoided by taking the dot product of the *normalized* eigenmodes. As a result, the eigenmode match threshold was invariant to sampling density and scaling. Nonetheless, the threshold was still *ad hoc*, and was set at 0.7 for the experiments described in Chapter 7. Experience has shown that this threshold is somewhat conservative; threshold values between 0.6 and 0.8 work well in our experiments.

Feature Affinity Threshold

Once a subset of best-matching eigenmodes has been selected, feature correspondence affinities are computed. This is done by measuring Euclidean distance in the generalized coordinate system formed by the eigenmodes. Features whose modal distance falls below a threshold are considered to be strong matches and are used as input to the alignment and description routines.

While the correspondence measure is scale-invariant and robust to deformation, there can be problems in setting a hard threshold that does not account for the number of modes included in the match. This can be easily circumvented by multiplying the affinity threshold by the number of modes included in computing the feature correspondences.

Setting this threshold conservatively large will allow only a few matches to get through to the alignment phase, and thus only low-order deformations can be recovered. A less-conservative threshold will let more matches through for alignment, but it may pass a few incorrect matches. By including match strengths in the modal alignment recovery, it is possible to weigh stronger matches more heavily than weaker ones. For details, see the weighted least squares formulation described in Section 5.1.2.

In practice, if higher-order alignment is needed (*i.e.,* for image metamorphosis), it is reasonable to employ a multistage process: (1) obtain low-order alignment using only conservative matches, (2) recompute the finite element model using the low-order deformed features, (3) obtain high-order alignment using liberal matches. Alternatively, additional feature correspondences can be found in a manner similar to snakes: by using the physical model as a smoothness constraint as described in Chapter 4.

8.3.3 Robustness to Noise and Missing Features

As the experiments with matching pears, cars, airplanes, and hands indicate, modal matching is relatively robust to noise. This robustness is the direct result of the stability of low-order eigenmode shape in the face of noise and nonrigid deformation. This property extends to small missing parts, as was evidenced by the missing fingers in the hand matching experiments and missing nacelles in the airplane matching experiments.

However, the modal matching algorithm has the limitation that it cannot reliably match largely occluded or partial objects. For instance, if a shape is cut in half by occlusion, then the algorithm will treat the single object as two separate shapes, with two separate sets of eigenmodes. Furthermore, the eigenmodes cannot be used to match parts of shapes to whole shapes — even if the part is distinctive. For example, in the current system it would be difficult to identify a person by only seeing the top half of the silhouette, since modes for half a shape can be quite different from the modes for the whole shape. To get around this problem the algorithm would need to utilize some combination of global (modal) measures and local (feature) measures to match distinctive substructures of shapes.

8.3.4 Robustness to Rotation

Using the special formulation described in Chapter 4, modal matching and description can be made invariant to in-plane rotations. In our experiments we have observed that modal matching is robust for out-of-plane rotations of up to ± 15 degrees. In fact, the modal matching system often performs well up to 45 degree rotations (for examples see the plane matching and hand matching experiments). In general, however, the feature correspondences begin to degrade for out-of-plane rotations of greater than ± 15 degrees; out-of-plane rotation drastically changes the overall aspect-ratio for the shape. This causes the sequencing and shape of low-order modes to change as the aspect ratio changes.

There are two heuristics that could be used to extend modal matching's immunity to larger out-of-plane rotations. The first method is simple: while matching up eigenmodes, look for matches over a wider range of modes. The second method is more complicated: prior to matching up eigenmodes for the two shapes, normalize each shape's eigenmodes by the shape's inverse aspect-ratio. Both of these heuristics operate under the assumption that in-plane rotation and nonrigid deformation between views would be relatively small.

8.4 Limitations

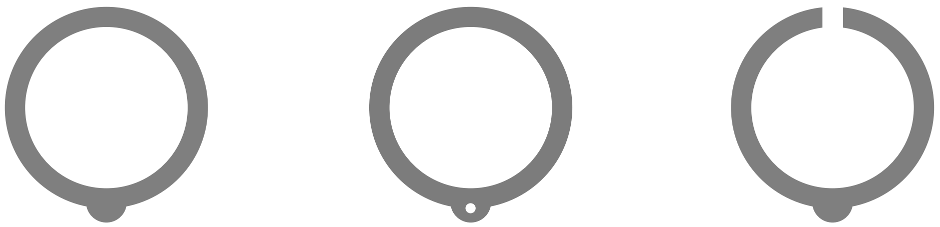
8.4.1 Segmentation and Detection

Currently, the algorithm assumes that the shape has been segmented from the background. For the experiments, shapes were segmented from the scene simply via image thresholding.

8.4.2 Computational Efficiency of Building Modal Models

The model building phase is slow since it requires the computation of a stiffness matrix and eigenmodes for each shape. Two well-known numerical techniques — subspace methods [9] and perturbation methods — can be used for computing these more efficiently. Model building efficiency is not a critical issue for the image database application, since it was assumed that shape representations are precomputed and used repeatedly. It may be more important for image morphing or for real-time vision applications. In such applications, better efficiency can be achieved via the multiscale technique formulated in Section 4.2 and demonstrated in the car matching experiment.

The current implementation requires that correspondences are found by explicitly matching up eigenvectors. There may be a way to measure similarity that does not first require solving this optimization problem. Given the precomputed modes for a prototype, it may be possible to just see if these modes directly diagonalize the stiffness matrix for a candidate shape.



(a) (b) (c)

Figure 8-1: A change in topology can sometimes affect modal matching performance, particularly when the difference in topology occurs at a larger scale. Although the three gaskets shown above may in some ways appear similar, each has a different topology. For the first two gaskets (a,b), the difference in topology occurs at a small scale. The low-order modes are relatively unaffected by this difference, and therefore, modal matching performance will be relatively undegraded. For the third gasket (c), a cut yields a difference in topology at a larger scale. In this case, modal matching will degrade because the low-order mode shapes for (c) differ from those for the other two gaskets.

8.4.3 Multiple Eigenvalues and Symmetry

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

Conversely, multiple eigenvalues are clues to symmetry. Thus, we could use this information to detect when there may be more than one valid shape correspondence. The most severe cases of this are a square (four possible alignments) or a circle (an infinite number of possible valid alignments).

The modal matching system can be used to match mirrored shapes if the rotation invariant formulation is employed. Mirror symmetry would need to be treated as a special case during the modal alignment stage. The correct alignment for a mirrored shape would require less strain, since it would not require an object to fold over on itself.

8.4.4 Topology and Low-order Modes

A change in topology can sometimes adversely affect modal matching performance, when the difference in topology occurs at a larger scale. This is because large scale topological differences can alter the low-order mode shapes used for modal matching, making it difficult to match similar but topologically different shapes. On the other hand, topological differences that occur at small scales have little or no affect on the low-order modes, therefore modal matching performance in such cases remains unaffected.

Figure 8-1 demonstrates this concept on three gasket shapes. While in some ways, these three gaskets appear to be similar, their topology differs. The first two gaskets Fig. 8-1(a,b) have a topological difference that occurs at a small enough scale, in that (b) has a small drilled in the tab where (a) does not. Since this type of difference has a minimal affect on the low-order mode shapes, the accuracy of correspondence would be relatively unaffected.

In contrast, the third gasket Fig. 8-1(c) has a cut in it that yields a difference in topology at a larger scale. If the gap created by this cut is larger than the basis function radius used in building our finite element model, then the low-order mode shapes for gasket (c) will differ from those for the other two gaskets. This in turn makes it difficult to use modal matching to find corresponding features on (a,c) or (b,c).

In general, if the topological difference between two shapes occurs at or below the scale of basis function radii (*e.g.,* small cracks or small holes), then modal matching will work because the basis functions are sufficiently large to smooth in gaps. If the gaps are larger, however, then modal matching would need to be coupled with a local correspondence algorithm.

8.5 Future Work

8.5.1 Trainable Modal Descriptions

It is possible to train the system to select a weighted subset of modes that are crucial when dealing with a particular category of objects, or when viewing in a particular context. This can be achieved by computing the principal components for the shape deformation over a training set (see Section 2.4.2). These context-specific principal variations can then be used to select a subset of deformation modes that will be used to adequately capture the variation over the training set.

Adequacy can be determined by projecting the principal components into the general modal eigenspace. It can be shown that these principal components can be spanned by some subset of the modal vibrations. The amount of each vibration mode needed to achieve this projection can be used to weigh how much of each vibration mode is considered important for object matching in a particular category or context.

8.5.2 Recursive Update of Physical Model Parameters

In a similar vein, a modal model could be recursively refined to capture the material properties of the object being observed [98]. Suppose that we have a set of eigenvectors (vibration modes) that describe how our prototype shape will deform under general conditions — we have assumed certain material properties: isotropic material, elasticity, Poisson ratio, and mass. The resulting *a priori* finite element stiffness matrix can be updated to incorporate observations as they are acquired. Similar work towards combining observations with our modal matching formulation has been undertaken recently by Cootes [29].

8.5.3 Automatic Part Detection

The current modal matching system has no notion of parts (for instance, in the hand motion example the hand was modeled as one continuous rubber sheet). However, the modal model could be extended to infer parts from example image sequences in much the same way suggested for recursively updating material properties.

Take, as an example, tracking a human hand gesture. The system starts out tracking the entire hand shape using a single homogeneous rubber sheet. As the hand moves through a gesture, ridges of high stress/strain energy will occur at part boundaries (knuckles, joints). These ridges are clues to where to cut the single rubber model into parts. Part boundaries could be added in such a way as to optimally reduce the strain energy needed to track the hand shape through the motion sequence. A similar technique for tracking and inferring human body parts based on fuzzy clustering has recently been proposed by Kakadiaris, *et al.* [57].

It may also be possible to infer parts from static images to obtain decompositions along the lines of [51, 66, 63]. To do this, the algorithm would first build a homogeneous rubber sheet model for the shape, and then randomly apply forces at extremal nodes (*e.g.,* finger tips). As each force is applied, the method would keep track of large stress/strain contours. A region of frequent stress/strain would be used as evidence for a part boundary.

8.5.4 Automatic Example Selection and Shape Categories

In the current image database system, a human operator selects a few example shapes that approximately span a category of shapes. System performance is therefore dependent on the user's ability to select an adequately diverse and sufficient set of examples. We could devise an unsupervised method for selecting the prototype shapes based on modal matching and modal strain. Such an energy-minimization scheme has parallels in theories for human category representation in terms of prototypes [111].

A first-cut algorithm for this would first randomly choose a prototype shape, and then sequentially compare the other shapes to it by matching modes and computing modal strain. If a shape is encountered that has eigenmodes matching none of the current prototypes, and/or there is no prototype that is close in terms of strain needed for alignment, then that shape would be used as an additional prototype. This would be repeated until all of the shapes in the database were accounted for (within some tolerance).

Unfortunately, this greedy algorithm selects prototypes dependent on the ordering of shapes in the database. Furthermore, it cannot guarantee that an optimal set of prototypes will be chosen. To get around this problem, the algorithm would need to do a second pass to look at how the shapes clustered in prototype strain-space. New "generic" prototypes would then be chosen from each cluster, or an average prototype could be synthesized from each cluster. The database would then be restructured using these new prototypes.

8.5.5 Modal Aspect Graphs

The above-mentioned principles can be used to automatically generate aspect graphs for a shape. As before, there would be issues of determining generic views and updating prototypes. In addition, the physically-based combinations of views paradigm of Chapter 6 could be employed to model pixel-by-pixel appearance. Given that pixel-by-pixel ap-

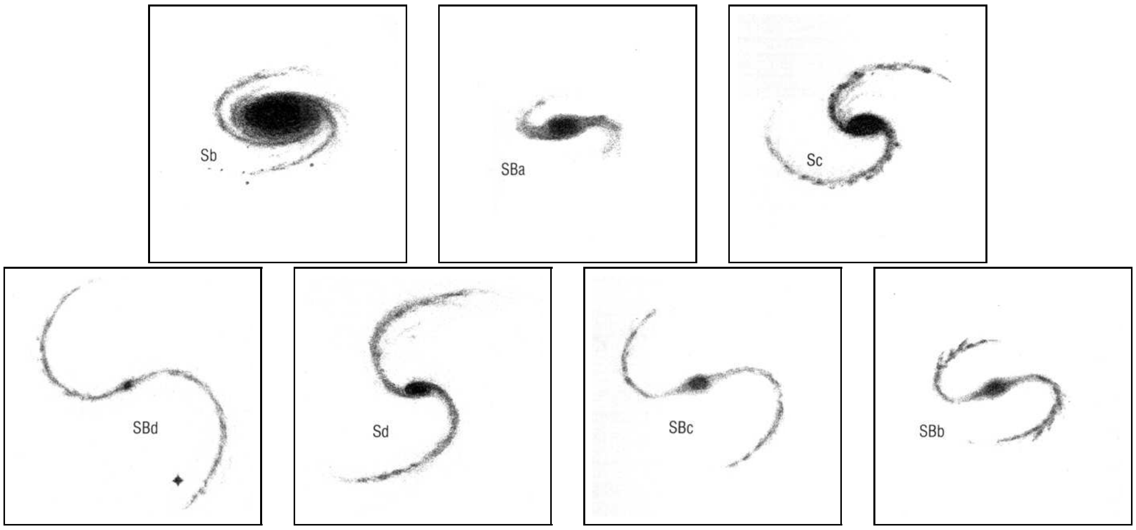


Figure 8-2: Spiral galaxies: deformed shapes that can be described in terms of their density patterns (taken from [40]). Using the method described in the text, modal models can be modified to incorporate such grayscale density information. This would make it possible to match, compare, and categorize these shapes in terms of modal deformations.

pearance can vary independently of shape deformation, we may also want to compute an eigenpicture representation for each generic view (see Section 2.4.3). The resulting representation would more efficiently capture grayscale variations due to lighting.

8.5.6 Including Grayscale in Modal Models

Rather than computing separate representations for an object's shape and pixel-by-pixel appearance, it may be appropriate to build a combined model. Three-dimensional modal models could be used to represent image features in terms of (x, y, i) components, where i represents the image intensity at the image location (x, y). However, some care would be required in using edge features, since the intensity values at edge points tend to be unstable.

We can avoid this problem by using an alternative formulation in which image intensity information could instead be included in the model by varying the finite element support function $s(x)$ for a two-dimensional element (see Section 3.4.4). This would have the effect of varying the thickness of the shape based on the image brightness throughout the shape. The resulting density representation would allow us to represent and match up structures like those found in the spiral galaxies of Figure 8-2.

8.5.7 Psychophysical Study of Deformed Shape Similarity

As in Saund's PhD thesis [115], it would be useful to do a small-scale psychophysical study to determine how human subjects order deformed shapes based on their similarity. Instead of arranging shapes in terms of features, subjects would be asked to arrange them in terms of overall shape similarity.

In Saund's "arrange the shapes" study, he reported that some subjects attempted to order the shapes by identifying a few prototype shapes, and then classifying the others according to which prototypes they were most similar to. The fish or hand tools image masks could be used as stimuli in the study. The subjects would be asked to arrange the shapes in order of similarity from a test shape. This would help verify our image database shape ordering tool, or point to areas where the modal matching ordering algorithm needs refinement.

8.5.8 3-D Modal Matching

The results described in this thesis can be easily extended to matching 3-D structures in volumetric data sets, and also the space-time volumes generated by video sequences. A 3-D finite element formulation is provided in Appendix A, although some issues regarding rotation-invariance in 3-D remain to be worked out. To avoid large rotations, perhaps shapes can be initially aligned as in [76] or moments of inertia methods can be used to obtain a rough initial alignment.

8.5.9 Extension to Other Signal Domains

The modal matching concept can be applied to deformed signal shapes in dimensions other than two. For instance, the method may be useful for matching up oil well-log data, seismic data, or other one-dimensional geological data. The methods described in this thesis could also be applied to higher-dimensional shapes; an N-dimensional finite element formulation is provided in Appendix A.

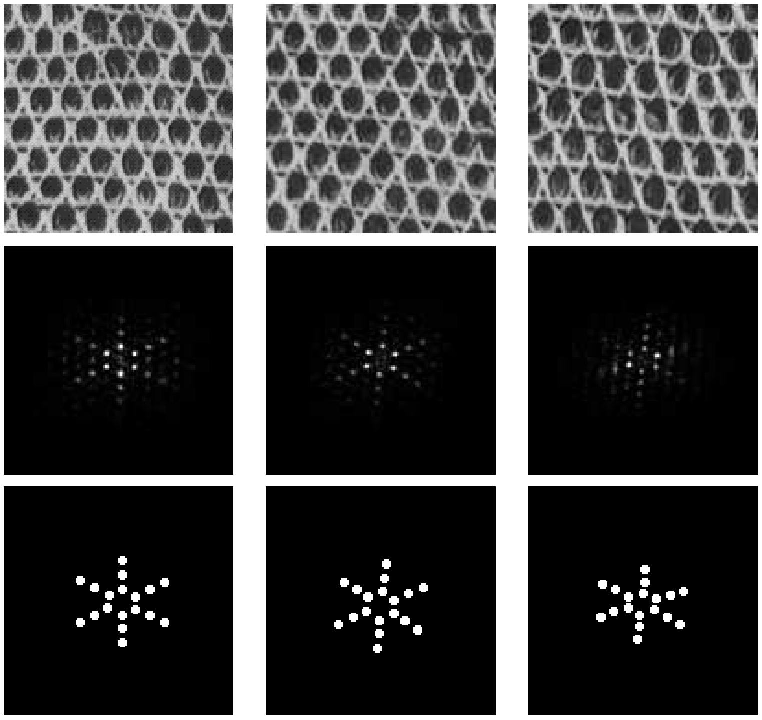


Figure 8-3: Using modal matching to compare and/or synthesize periodic textures. Given images of a periodic texture like lizard skin, we use the corresponding DFT magnitude images as input to a peak detection routine. The resulting peaks are a good reduced-description of the harmonic component of the texture [100]. These peaks rotate and deform due to changes in viewpoint or due to stretching of the skin over the lizard's body, and can be used as input to the modal matching algorithm described in the text. The DFT and magnitude peak images were provided by Fang Liu.

In applying modal matching to other types of signal shapes, it may sometimes be necessary to employ an intermediate representation. In the sound domain, for instance, the signal might be represented as a constant-Q spectrogram. Details of the application to sound are provided in Appendix B.

Figure 8-3 shows another application where an intermediate representation would be useful: matching and classifying periodic textures. Given images of a periodic texture like the lizard skin, we use the corresponding DFT magnitude images as input to a peak detection routine [100]. The extracted peaks serve as an intermediate representation for the harmonic component of the texture. In general, the overall pattern of these magnitude peaks can deform (rotate, stretch, shear, *etc.*) due to changes in viewpoint or due to stretching of the skin over the lizard's body. Modal matching could be used to match up these deformed DFT peak patterns, and then to describe what kinds of deformations relate the patterns. Such a framework may prove useful for texture classification.

Lastly, may be possible to resynthesize textures given deformed DFT peaks, phase information, plus the non-deterministic (random) components of the texture. This would make it possible to generate novel textures via modal morphing of the DFT peaks.

Chapter 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 [122, 126, 127] 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, not only does the transformation to modal space allow for automatically establishing correspondence between clouds of feature points; the same modes (and the underlying FEM model) allow for describing the deformations that take the features from one position to the other. The amount of deformation required to align the two feature clouds can be used for shape comparison and description, and to warp the original images for alignment and sensor fusion. The power of this method lies primarily in its ability to unify the correspondence and comparison tasks within one representation.

By using Gaussian interpolants as our trial functions for Galerkin approximation, we can easily formulate finite elements for any dimension. A very useful aspect of multidimensional Gaussians is that they are factorizable; this not only reduces computational cost, it also has useful implications for VLSI hardware and neurobiological implementations.

We have also introduced a physically-motivated linear-combinations-of-models scheme, where the computer synthesizes an image of the object in terms of a weighted combination of modally deformed prototypes. In contrast to previous linear-combinations-of-models techniques, our method offers the advantages that: (a) it uses the physical model to enforce smoothness constraints for better image warping, and (b) it provides a set of orthogonal deformation parameters, making it possible to select the types of deformations that can occur. For the computer graphics application of image metamorphosis, this has the useful side-effect that image transformations can be scheduled, *e.g.,* rigid alignment first, affine next, *etc.*

Finally, we note that the descriptions computed are canonical, and vary smoothly even for very large deformations, thus allowing the descriptions to be used directly for object recognition (as illustrated by the recognition experiments of Chapter 7). Because the deformation comparisons are physically-based, we can determine whether or not two shapes are related by a simple physical deformation and thereby identify shapes that appear to be members of the same category.

Bibliography

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

[2] Y. Amit, U. Grenander, and M. Piccioni. Structural image restoration through deformable templates. Technical report, MIT, April 1990.

[3] R. Arnheim. *Art and Visual Perception*. University of California Press, 1974.

[4] R. Bajcsy and S. Kovačič. Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing*, 46(1):1–21, 1989.

[5] D. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[6] D. Ballard. Strip trees: A hierarchical representation for curves. *Communications of the ACM*, 24(5):310–321, May 1981.

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

[8] H. Barrow, J. Tenebaum, R. Bolles, and H. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. of the Fifth IJCAI*, pages 659–663, 1977.

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

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

[11] D. Beymer, A. Shashua, and T. Poggio. Example based image analysis and synthesis. Technical Report TR 1431, MIT, 1993.

[12] I. Biederman. Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, 94(2):115–147, 1987.

[13] E. Binaghi, I. Gagliardi, and R. Schettini. Indexing and fuzzy logic-based retrieval of color images. In *Proc. Visual Database Systems II, IFIP Transactions A-7*, pages 79–92, 1990.

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

[15] A. Blake and A. Zisserman. *Visual Reconstruction*. M.I.T. Press, 1987.

[16] H. Blum. *A Transformation for extracting new descriptors of shape*. MIT Press, 1967.

[17] F. Bookstein. The line skeleton. *Computer Graphics and Image Processing*, 11:123–137, 1979.

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

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

[20] T. Boult, S. Fenster, and T. O'Donnell. Physics in a fantasy world vs. robust statistical estimation. In *Proc. NSF Workstop on 3D Object Recognition*, New York, NY, November 1994.

[21] M. Brady and H. Asada. Smoothed local symmetries and their implementation. *International Journal of Robotics Research*, 3(3):36–61, 1984.

[22] C. Broit. *Optimal Registration of Deformed Images*. PhD thesis, University of Pennsylvania, 1981.

[23] R. Brooks. Model-based three-dimensional interpretations of two-dimensional images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 5(2):140–149, March 1983.

[24] D. J. Burr. A dynamic model for image registration. *Computer Vision, Graphics and Image Processing: Image Understanding*, 15:102–112, 1981.

[25] S. Chen and L. Williams. View interpolation for image synthesis. *Computer Graphics*, 27(2):279–288, August 1993.

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

[27] 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.

[28] L. Cohen and I. Cohen. A finite element method applied to the new active contour models and 3-D reconstruction from cross sections. In *Proc. Third International Conference on Computer Vision*, December 1990.

[29] T. Cootes. Combining point distribution models with shape models based on finite element analysis. In *Proc. British Machine Vision Conference*, 1994.

[30] 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.

[31] T. Cootes and C. Taylor. Active shape models – ‘smart snakes’. In *Proc. British Machine Vision Conference*, pages 266–275, 1992.

[32] T. Darrell and A. Pentland. Space-time gestures. In *Proc. Computer Vision and Pattern Recognition*, pages 335–340, June 1993.

[33] 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.

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

[35] D. Ellis. *A Perceptual Representation of Audio*. Master’s thesis, MIT Electrical Engineering and Computer Science Department, 1992.

[36] D. Ellis, 1994. Personal communication.

[37] D. Ellis and B. Vercoe. A perceptual representation of sound for auditory signal separation. In *Proc. of the 125th meeting of the Acoustical Society of America*, Ottawa, May 1993.

[38] I. Essa, S. Sclaroff, and A. Pentland. *Directions in Geometric Computing*, chapter Physically-based Modeling for Graphics and Vision. Information Geometers, U.K., 1992. R. Martin, Ed.

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

[40] T. Ferris. *Galaxies*. Sierra Club Books, New York, NY, 1987.

[41] M. A. Fischler and R. A. Eschlager. The representation and matching of pictorial structures. *IEEE Trans. Computers*, 22:67–92, 1973.

[42] H. Freeman and I. Chakravarty. The use of characteristic views in the recognition of three-dimensional objects. In *Pattern Recognition in Practice*, E. Gelsema and L. Kanal (ed.), 1980.

[43] D. Geiger and F. Girosi. Parallel and deterministic algorithms from MRFs: Surface reconstruction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(5):674–693, May 1991.

[44] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6(11), November 1984.

[45] I. Gradstein and I. Ryshik. *Summen-, Produkt-, und Integral Tafeln*, volume 2. Verlag Harri Deutsch, 1981.

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

[47] W. Grimson. An Implementation of a Computational Theory for Visual Surface Interpolation. *Computer Vision, Graphics, and Image Processing*, 22:39–69, 1983.

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

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

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

[51] D. Hoffman and W. Richards. Parts of recognition. *Cognition*, 18:65–96, 1984.

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

[53] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the hausdorff distance. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.

[54] M. Ioka. A method of defining the similarity of images on the basis of color information. Technical Report RT-003 0, IBM Tokyo, 1989.

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

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

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

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

[59] Kaniza. *Organization in Vision: Essays in Gestalt Perception*. Holt Reinhart Winston, 1982.

[60] P. Karaolani, G. Sullivan, and K. Baker. Active contours using finite elements to control local scale. In *Proc. British Machine Vision Conference*, pages 481–487, 1992.

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

[62] G. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation and on stochastic processes and smoothing by splines. *Annals of Mathematical Statistics*, 41(2):495–502, February 1970.

[63] B. Kimia, A. Tannenbaum, and S. Zucker. Toward a computational theory of shape: An overview. Technical report, McGill University, 1989.

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

[65] J. Koenderink and A. Van Doorn. The internal representation of shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[66] J. Koenderink and A. Van Doorn. Dynamic shape. *Biological Cybernetics*, 53:383–396, 1986.

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

[68] M. Leyton. A Processs-Grammar for Shape. *Artificial Intelligence*, 34:213–247, 1988.

[69] S. Librande. *Example-based Character Drawing*. Master's thesis, MIT Media Laboratory, September 1992.

[70] 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.

[71] H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[72] D. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publications, 1985.

[73] D. Lowe. Three-dimensional object recognition from single two dimensional images. *Artificial Intelligence Journal*, 31:355–395, 1987.

[74] K. Mardia and I. Dryden. The statistical analysis of shape data. *Biometrika*, 76(2):271–281, 1989.

[75] K. Mardia, J. Kent, and A. Walder. Statistical shape models in image analysis. In *Proc. of the 23rd Symposium on the Interface*, pages 550–557, 1991.

[76] J. Martin, A. Pentland, and R. Kikinis. Shape analysis of brain structures using physical and experimental modes. In *Proc. Computer Vision and Pattern Recognition*, June 1994.

[77] J. Martin, A. Pentland, R. Kikinis, and S. Sclaroff. Characterization of pathological shape deformations. Technical report, MIT Media Laboratory, Perceptual Computing Section, Cambridge, MA, 1995.

[78] T. McInerney and D. Terzopoulos. A finite element model for 3-D shape reconstruction and nonrigid motion tracking. In *Proc. Fourth International Conference on Computer Vision*, May 1993.

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

[80] F. Mokhtarian and A. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(8):789–805, 1992.

[81] K. Mori, M. Kidodi, and H. Asada. An iterative predictor correction method for automatic stereo comparison. *Computer Vision, Graphics and Image Processing: Image Understanding*, 2:393–401, 1973.

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

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

[84] C. Nastar. Analytical Computation of the Free Vibration Modes: Application to Non Rigid Motion Analysis and Animation in 3D Images. Technical Report 1935, INRIA — Rocquencourt, 78153 Le Chesnay Cedex, France, June 1993.

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

[86] J. Oden and J. Reddy. *An Introduction to the Mathematical Theory of Finite Elements*. John Wiley and Sons, 1976.

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

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

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

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

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

[92] A. Pentland, B. Horowitz, and S. Sclaroff. Non-rigid motion and structure from contour. In *Proc. IEEE Workshop on Visual Motion*, October 1991.

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

[94] A. Pentland, R. Picard, G. Davenport, and R. Welsh. The BT/MIT project on advanced image tolls for telecommunications: An overview. In *Proc. ImageCom93, Second International Conf. on Image Communications*, Bordeaux, France, March 1993.

[95] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. In *Proc. SPIE Conf. on Storage and Retrieval of Image and Video Databases II*, volume 2185, San Jose, CA, February 1994.

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

[97] A. Pentland and J. Williams. Good vibrations : Modal dynamics for graphics and animation. *Computer Graphics*, 23(4):215–222, 1989.

[98] A. Pentland and J. Williams. The perception of nonrigid motion: Inference of material properties and force. In *Proc. International Joint Conference on Artificial Intelligence*, August 1989.

[99] E. Persoon and K. Fu. Shape discrimination using Fourier descriptors. In *Proc. Second IJCPR*, pages 126–130, 1974.

[100] R. Picard and F. Liu. A new Wold ordering for image similarity. In *Proc. IEEE Conf. on Acoustics, Speech, and Signal Processing*, Melbourne, Australia, April 1994.

[101] T. Poggio and R. Brunelli. A novel approach to graphics. Technical Report A.I. Memo No. 1354, Artificial Intelligence Lab, MIT, Cambridge, MA, February 1992.

[102] T. Poggio and S. Edelman. A network that learns to recognize three-dimensional objects. *Nature*, 343:263–266, January 1990.

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

[104] T. Poggio, V. Torre, and C. Koch. Computational Vision and Regularization Theory. *Nature*, 317:314–319, September 1985.

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

[106] W. Press, Brian Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK, 1988.

[107] L. Quam. Hierarchical warp stereo. In *Proc. Image Understanding Workshop*, pages 149–155, October 1984.

[108] E. Reiner, L. Abbey, T. Moran, P. Papmichalis, and R. Schafer. Characterization of normal human cells by pyrolis-gas-chromatography mass specttrometry. *Biomedical Mass Spectrometry*, 6(11):491–498, 1979.

[109] E. Reiner and G. Kubica. Predictive value of pyrolis-gas-liquid chromatography in the differentiation of mycobacteria. *Amercian Review of Respiratory Disease*, 99:42–49, 1969.

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

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

[112] A. Rosenfeld. Axial representations of shape. *Computer Vision, Graphics and Image Processing*, 33:156–173, 1986.

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

[114] D. Sankoff and J. Kruskal. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, 1983.

[115] Eric Saund. *The Role of Knowledge in Visual Shape Representation*. PhD thesis, MIT Dept. of Brain and Cognitive Sci., October 1988.

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

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

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

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

[120] S. Sclaroff and A. Pentland. Search by shape examples: Modeling nonrigid deformation. In *Proc. Twenty-Eighth Annual Asilomar Conference on Signals*, Nov 1994.

[121] S. Sclaroff and A. Pentland. Modal Matching for Correspondence and Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, in press.

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

[123] T. Sederberg, P. Gao, G. Wang, and H. Mu. 2D shape blending: An intrinsic solution to the vertex path problem. *Computer Graphics*, 27(2):15–18, August 1993.

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

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

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

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

[128] A. Shashua. *Geometry and Photometry in 3-D Visual Recognition*. PhD thesis, MIT Artificial Intelligence Laboratory, November 1992.

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

[130] K. Shoemake. Animating rotation with quaternion curves. *Computer Graphics*, 19(3):245–252, 1985.

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

[132] N. Strachan. Recognition of fish species by colour and shape. *Image and Vision Computing*, 11(1):2–10, 1993.

[133] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[134] R. Szeliski. *Bayesian Modeling of Uncertainty in Low-Level Vision*. Kluwer Academic Publishers, 1989.

[135] C. Tappert, C. Suen, and T. Wakahara. The state of the art in on-line handwriting recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(8):787–808, 1990.

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

[137] D. Terzopoulos. Multilevel computational processes for visual surface reconstruction. *Computer Vision, Graphics, and Image Processing*, 24:52–96, 1983.

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

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

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

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

[142] D'Arcy Thompson. *On Growth and Form*. Cambridge University Press, 1988.

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

[144] M. Turk. *Interactive-Time Vision: Face Recognition as a Visual Behavior*. PhD thesis, MIT Media Laboratory, September 1991.

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

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

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

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

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

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

[151] D. Widrow. The rubber mask technique, parts I and II. *Pattern Recognition*, 5:175–211, 1973.

[152] A. Witkin, D. Terzopoulos, and M. Kass. Signal matching through scale space. *International Journal of Computer Vision*, 1:133–144, 1987.

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

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

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

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

Appendix A

A FEM Formulation for Higher-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:

$$\mathcal{H} = \left[\begin{array}{cccc} h_1 & h_2 & \ldots & h_m \end{array} \right]. \tag{A.1}$$

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:

$$\frac{\partial}{\partial x_i}\mathcal{H} = \left[\begin{array}{ccc} \frac{\partial}{\partial x_i}h_1 & \frac{\partial}{\partial x_i}h_2 & \ldots \frac{\partial}{\partial x_i}h_m \end{array} \right]. \tag{A.2}$$

A.1 Formulating a 3-D Element

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

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} h_1 & \dots & h_m & & & & & & \\ & & & h_1 & \dots & h_m & & & \\ & & & & & & h_1 & \dots & h_m \end{bmatrix} = \begin{bmatrix} \mathcal{H} & & \\ & \mathcal{H} & \\ & & \mathcal{H} \end{bmatrix}. \tag{A.3}$$

where m is the number of finite element nodes.

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

$$\mathbf{M} = \int_V \rho \mathbf{H}^T \mathbf{H} dV = \begin{bmatrix} \mathcal{M} & 0 & 0 \\ 0 & \mathcal{M} & 0 \\ 0 & 0 & \mathcal{M} \end{bmatrix} \tag{A.4}$$

where the m by m submatrix $\mathcal{M}$ is positive definite symmetric. The elements of $\mathcal{M}$ have the form:

$$m_{ij} = \rho \int_V \sum_{kl} a_{ik}\, a_{jl}\, g_k(\mathbf{x})\, g_l(\mathbf{x}) dV \;=\; \rho\, \pi^{\frac{3}{2}} \sigma^3 \sqrt{g_{ij}}. \tag{A.5}$$

where a_{ik} are the interpolation coefficients, g_k and g_l are 3-dimensional Gaussian basis functions, and $g_{kl} = g_k(\mathbf{x}_l)$ is an element of the $\mathbf{G}$ matrix in Equation 3.16.

This can be rewritten in matrix form:

$$\mathcal{M} = \rho\, \pi^{\frac{3}{2}}\, \sigma^3 \mathbf{A}^T \mathcal{G} \mathbf{A} = \rho\, \pi^{\frac{3}{2}}\, \sigma^3 \mathbf{G}^{-1} \mathcal{G} \mathbf{G}^{-1}, \tag{A.6}$$

where the elements of $\mathcal{G}$ are the square roots of the elements of the $\mathbf{G}$ matrix as in Equation 3.16.

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:

$$\begin{gathered} \epsilon_{xx} = \frac{\partial u}{\partial x} \qquad \epsilon_{yy} = \frac{\partial v}{\partial y} \qquad \epsilon_{zz} = \frac{\partial w}{\partial z} \\ \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \qquad \gamma_{xz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \qquad \gamma_{yz} = \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \end{gathered} \tag{A.7}$$

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:

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x}\mathcal{H} & & \\ & \frac{\partial}{\partial y}\mathcal{H} & \\ & & \frac{\partial}{\partial z}\mathcal{H} \\ \frac{\partial}{\partial y}\mathcal{H} & \frac{\partial}{\partial x}\mathcal{H} & \\ \frac{\partial}{\partial z}\mathcal{H} & & \frac{\partial}{\partial x}\mathcal{H} \\ & \frac{\partial}{\partial z}\mathcal{H} & \frac{\partial}{\partial y}\mathcal{H} \end{bmatrix} \tag{A.8}$$

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

$$\beta \begin{bmatrix} 1 & \alpha & \alpha & & & \\ \alpha & 1 & \alpha & & & \\ \alpha & \alpha & 1 & & & \\ & & & \xi & & \\ & & & & \xi & \\ & & & & & \xi \end{bmatrix}. \tag{A.9}$$

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:

$$\alpha = \frac{\nu}{1-\nu}, \qquad \beta = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}, \quad \text{and} \quad \xi = \frac{1-2\nu}{2\,(1-\nu)}. \tag{A.10}$$

It is possible to use other material matrices which model anisotropic materials. 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.

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

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

Note that this stiffness matrix $\mathbf{K}$ is symmetric. Since we use Gaussian basis functions, its actually the case that the submatrices themselves are symmetric.

The elements of $\mathbf{K}_{11}$ have the form:

$$k_{11_{ij}} = \beta \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sum_{k,l} a_{ik}\, a_{jl} \left[\frac{\partial g_k}{\partial x} \frac{\partial g_l}{\partial x} + \xi \left(\frac{\partial g_k}{\partial y} \frac{\partial g_l}{\partial y} + \frac{\partial g_k}{\partial z} \frac{\partial g_l}{\partial z} \right) \right] dx\, dy\, dz. \tag{A.12}$$

Integrate and regroup terms:

$$k_{11_{ij}} = \pi^{\frac{3}{2}} \sigma \beta \sum_{k,l} a_{ik}\, a_{jl} \left[\frac{1+\xi}{2} - \frac{\hat{x}_{kl}^2 + \xi \left(\hat{y}_{kl}^2 + \hat{z}_{kl}^2 \right)}{4\sigma^2} \right] \sqrt{g_{kl}}, \tag{A.13}$$

where $\hat{x}_{kl} = (x_k - x_l)$, $\hat{y}_{kl} = (y_k - y_l)$, and $\hat{z}_{kl} = (z_k - z_l)$. Similarly, the elements of $\mathbf{K}_{22}$ and $\mathbf{K}_{33}$ have the form:

$$k_{22_{ij}} = \pi^{\frac{3}{2}} \sigma \beta \sum_{k,l} a_{ik}\, a_{jl} \left[\frac{1+\xi}{2} - \frac{\hat{y}_{kl}^2 + \xi \left(\hat{x}_{kl}^2 + \hat{z}_{kl}^2 \right)}{4\sigma^2} \right] \sqrt{g_{kl}}, \tag{A.14}$$

and

$$k_{33_{ij}} = \pi^{\frac{3}{2}} \sigma \beta \sum_{k,l} a_{ik}\, a_{jl} \left[\frac{1+\xi}{2} - \frac{\hat{z}_{kl}^2 + \xi \left(\hat{x}_{kl}^2 + \hat{y}_{kl}^2 \right)}{4\sigma^2} \right] \sqrt{g_{kl}}. \tag{A.15}$$

The elements of $\mathbf{K}_{12}$ have the form:

$$k_{12_{ij}} = \beta \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sum_{k,l} a_{ik}\, a_{jl} \left[\alpha \frac{\partial g_k}{\partial x} \frac{\partial g_l}{\partial y} + \xi \frac{\partial g_k}{\partial y} \frac{\partial g_l}{\partial x} \right] dx\, dy\, dz. \tag{A.16}$$

When integrated this becomes:

$$k_{12_{ij}} = -\frac{\pi^{\frac{3}{2}}\beta(\alpha+\xi)}{4\sigma}\sum_{k,l} a_{ik}\, a_{jl}\, \hat{x}_{kl}\, \hat{y}_{kl}\sqrt{g_{kl}}. \tag{A.17}$$

Similarly, the elements of $\mathbf{K}_{13}$ and $\mathbf{K}_{23}$ have the form:

$$k_{13_{ij}} = -\frac{\pi^{\frac{3}{2}}\beta(\alpha+\xi)}{4\sigma}\sum_{k,l} a_{ik}\, a_{jl}\, \hat{x}_{kl}\, \hat{z}_{kl}\sqrt{g_{kl}}. \tag{A.18}$$

and

$$k_{23_{ij}} = -\frac{\pi^{\frac{3}{2}}\beta(\alpha+\xi)}{4\sigma}\sum_{k,l} a_{ik}\, a_{jl}\, \hat{y}_{kl}\, \hat{z}_{kl}\sqrt{g_{kl}}. \tag{A.19}$$

A.2 Formulating Higher-Dimensional Elements

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

$$\mathbf{M} = \int_V \rho \mathbf{H}^T\mathbf{H} dV = \begin{bmatrix} \mathcal{M} & & & \\ & \mathcal{M} & & \\ & & \ddots & \\ & & & \mathcal{M} \end{bmatrix} \tag{A.20}$$

where the submatrix $\mathcal{M}$ is positive definite symmetric. The elements of this submatrices are computed by using n-dimensional Gaussian basis functions in Equation A.5.

In general, the n-dimensional mass submatrix takes the form:

$$\mathcal{M} = \rho\, \pi^{\frac{n}{2}}\, \sigma^n \mathbf{A}^T \mathcal{G} \mathbf{A} = \rho\, \pi^{\frac{n}{2}}\, \sigma^n \mathbf{G}^{-1} \mathcal{G} \mathbf{G}^{-1}. \tag{A.21}$$

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

$$\begin{bmatrix} \epsilon_{x_1x_1} & \gamma_{x_1x_2} & \gamma_{x_1x_3} & \cdots & \gamma_{x_1x_{n-1}} & \gamma_{x_1x_n} \\ & \epsilon_{x_2x_2} & \gamma_{x_2x_3} & \cdots & \gamma_{x_2x_{n-1}} & \gamma_{x_2x_n} \\ & & \epsilon_{x_3x_3} & \cdots & \gamma_{x_3x_{n-1}} & \gamma_{x_3x_n} \\ & & & \ddots & \vdots & \vdots \\ & & & & \epsilon_{x_{n-1}x_{n-1}} & \gamma_{x_{n-1}x_n} \\ & & & & & \epsilon_{x_nx_n} \end{bmatrix}. \tag{A.22}$$

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:

$$\boldsymbol{\epsilon}^T = [\ \underbrace{\epsilon_{x_1x_1}, \cdots \epsilon_{x_nx_n}}_{\text{diagonal elements}}\ ,\ \underbrace{\gamma_{x_1x_2}, \gamma_{x_1x_3}, \cdots \gamma_{x_{n-1}x_n}}_{\text{upper triangle elements}}\ ], \tag{A.23}$$

where

$$\epsilon_{x_ix_i} = \frac{\partial u_i}{\partial x_i} \quad \text{and} \quad \gamma_{x_ix_j} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}, \tag{A.24}$$

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

To get the n-dimensional stiffness matrix, we use a generalized version of Equation A.9. Assuming an isotropic material, the n-dimensional material matrix $\mathbf{C}$ will take the form:

$$\beta \left[\begin{array}{cccc|cccc} 1 & \alpha & \dots & \alpha & & & & \\ \alpha & 1 & \dots & \alpha & & & & \\ \vdots & \vdots & \ddots & \vdots & & & & \\ \alpha & \alpha & \dots & 1 & & & & \\ \hline & & & & \xi & & & \\ & & & & & \xi & & \\ & & & & & & \ddots & \\ & & & & & & & \xi \end{array}\right] \tag{A.25}$$

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

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:

$$\mathbf{B}(\mathbf{x}) = \left[\begin{array}{ccccc}
\frac{\partial}{\partial x_1}\mathcal{H} & & & & \\
& \frac{\partial}{\partial x_2}\mathcal{H} & & & \\
& & \frac{\partial}{\partial x_3}\mathcal{H} & & \\
& & & \ddots & \\
& & & & \frac{\partial}{\partial x_n}\mathcal{H} \\ \hline
\multicolumn{1}{c|}{\frac{\partial}{\partial x_2}\mathcal{H}} & \frac{\partial}{\partial x_1}\mathcal{H} & & & \\
\multicolumn{1}{c|}{\frac{\partial}{\partial x_3}\mathcal{H}} & & \frac{\partial}{\partial x_1}\mathcal{H} & & \\
\multicolumn{1}{c|}{\vdots} & & & \ddots & \\
\multicolumn{1}{c|}{\frac{\partial}{\partial x_n}\mathcal{H}} & & & & \frac{\partial}{\partial x_1}\mathcal{H} \\ \hline
\multicolumn{1}{c|}{} & \multicolumn{1}{c|}{\frac{\partial}{\partial x_3}\mathcal{H}} & \frac{\partial}{\partial x_2}\mathcal{H} & & \\
\multicolumn{1}{c|}{} & \multicolumn{1}{c|}{\vdots} & & \ddots & \\
\multicolumn{1}{c|}{} & \multicolumn{1}{c|}{\frac{\partial}{\partial x_n}\mathcal{H}} & & & \frac{\partial}{\partial x_2}\mathcal{H} \\ \hline
\multicolumn{5}{c}{etc.}
\end{array} \right]. \tag{A.26}$$

The stiffness matrix for a general n-dimensional element can be computed directly:

$$\mathbf{K} = \int_V \mathbf{B}^T \mathbf{C} \mathbf{B} dV = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \dots & \mathbf{K}_{1n} \\ \mathbf{K}_{12}^T & \mathbf{K}_{22} & \dots & \mathbf{K}_{2n} \\ \vdots & \vdots & & \vdots \\ \mathbf{K}_{1n}^T & \mathbf{K}_{2n}^T & \dots & \mathbf{K}_{nn} \end{bmatrix}. \tag{A.27}$$

As before, each of the submatrices is symmetric.

Lastly, the general formulae for the elements of the submatrix $\mathbf{K}_{pq}$ are as follows. If $p = q$, then

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

otherwise

$$k_{pq_{ij}} = -\frac{\pi^{\frac{n}{2}} \sigma^{n-4} \beta (\alpha + \xi)}{4} \sum_{k,l} a_{ik}\, a_{jl}\, \hat{x}_{p_{kl}}\, \hat{x}_{q_{kl}} \sqrt{g_{kl}}. \tag{A.29}$$

Appendix B

Sound

Nearly any signal can be represented as an n-dimensional shape; thus, the modal matching method could be used to analyze and manipulate the signal in terms of features and/or signal shapes. To handle sound, for instance, we could map sound energy (a spectrogram) into a two dimensional mass distribution with some elastic interconnections. Given such a framework, we could then employ each of the modal matching and manipulation techniques described in this thesis. For instance, the representation could be used to "morph" sounds for analysis, recognition, and sound synthesis.

Figure B-1 shows an example of deformed signals in the sound domain. Here we can define "shape" as peaks found in a time-frequency plot (spectrogram). In Figure B-1(a,b) we have two different instances of the 1-D signal for the spoken word *yellow*. Differences in these sounds give rise to deformations of formant shapes (shown as white ridges) in the Constant-Q spectrograms for these sounds as is shown in Figure B-1(c,d).

In a constant-Q spectrogram, the time-frequency kernel shape used to transform sound energy to image intensity is meant to mimic the variation in frequency bandwidth found in mammalian cochleae [37]. In the constant-Q spectrograms shown in the figure, image intensity is an inverse function of magnitude (highest energy is shown as black, lowest as white), and the two image axes are time and frequency.

Ellis has formulated a *tracks* representation, where a spectrogram is represented in terms of energy peaks and phase information extracted from a spectrogram. These tracks serve as a convenient reduced representation for sounds. Ellis has shown that perceptually

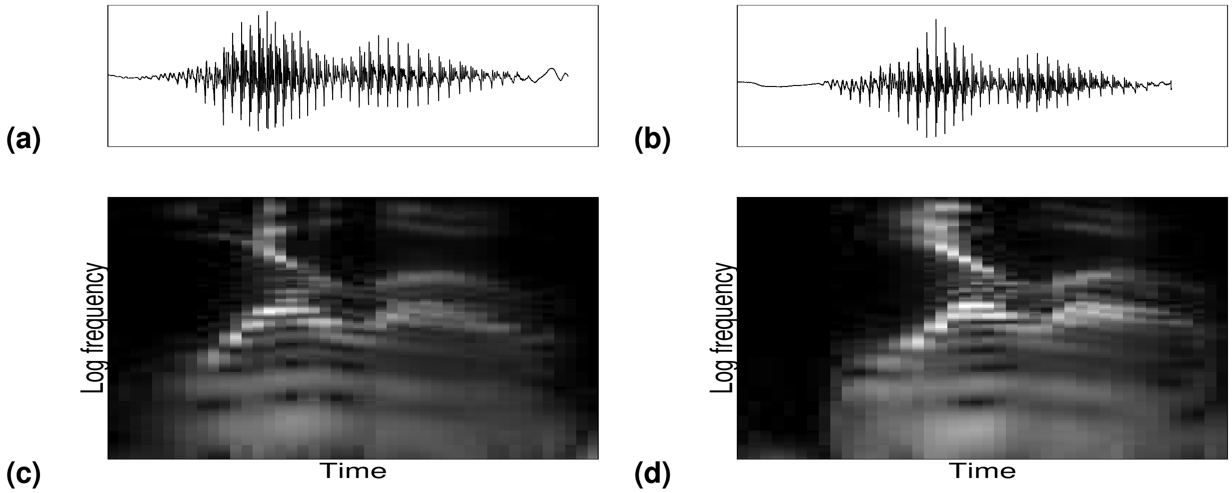


Figure B-1: An example of matching deformed signals in another domain. In this example we have two different instances of the 1-D signal for the spoken word *yellow* (a,b). Differences in these sounds give rise to deformations of formant shapes (white ridges) in the Constant-Q spectrograms for these sounds as is shown in (c,d). Note that the time axis goes from left to right, and that the frequency plot is shown on a log scale to preserve harmonic structure. To measure similarity between these two sounds, we could match up these formant ridges, and then measure the amount and types of deformations needed to align the two spectrograms.

similar sounds could then be resynthesized by playing back just the tracks [35] (rather than the full signal). The input to the modal matching and morphing paradigm would be a track representation such as this. To measure similarity between these two sounds, we would then match up these tracks, and then measure the amount and types of deformations needed to align the two spectrograms.

For synthesis, the concept is simple: given such a framework, we could match and morph between sound effects. Once a set of tracks has been deformed, it can then be resynthesized for playback. For instance, we should be able to smoothly morph between the sounds produced by two different types of balls bouncing. Thus we could produce new sound effects from existing tracks. Another application would be speech, where we could morph between words or between speakers of different ages, genders, *etc.* Another interesting application in speech is modeling formant deformation which is due to coarticulation of phonemes.

Coarticulation in Speech

One could naively think of speech production as the process of stringing phonemes together to form words. Stated in such terms, the problem of producing intelligible speech is mainly a control problem, where there is a sequence of phonemes to be produced, and the brain generates a control plan/schedule for their production via gestural patterns — within the physical limitations and perturbations of the vocal tract. Even when we cast aside basic considerations about cross-speaker differences like gender, age, dialect, and native language, and context dependencies like emotion, stress, and emphasis, phoneme shape still varies a great deal depending on the phonemes produced before and after it. Nearby sounds actually deform each phoneme (represented as tracks in the spectrogram) from its ideal shape. This effect of temporal interleaving and overlap among nearby gestures is known as coarticulation.

It is clear that 2-D deformations of an elastic sheet can be used to model many of the deformations observed in formants; however, not all 2-D vibration modes are consistent with the detailed physics of speech. Take for instance modes which stagger the formant onsets or offsets (shear, for instance). Though onsets could be delayed slightly (depending on frequency) the fact that the same power sources are generating the formants rules out staggering like this. Another example is rotation: in the worst case, this means that the production mechanism could transpose time and frequency, which is not plausible. An interesting unlawful mode is compression: this corresponds with a sound medium change (*i.e.,* water or air). Modes which keep onsets aligned are more likely to be lawful: translation, scaling, bending, and tapering frequency over time are good examples. What we want in general is to choose a subset of p lawful modes, and then match and morph sounds using this mode-truncated system.

From an implementation standpoint, it is unclear what would be the appropriate time-frequency kernel for building the spectrograms. Though the constant-Q kernel offers the advantage that it mimics what is known about mammals' hearing, modally deforming the tracks of a constant-Q spectrogram may not yield satisfactory resynthesis. This type of spectrogram finely resolves the frequency of the lowest harmonics, while higher harmonics are more finely resolved in time (making pitch cycles visible). Therefore, simple deformations would transform the tracks in a way which would not preserve the formants' periodicity [36]. This leaves the open question of which type of filter to use: narrow-band, wide-band, or constant-Q — in so doing, we would need to balance resolution in frequency with resolution in time.

A fixed-bandwidth, wideband analysis looks promising, since it yields a "spot" representation, where the features are a set of isolated points corresponding to local maxima in the time-frequency plane. These spots would then be centers for the Gaussian interpolation functions used to describe the finite element. Thus the spots would be connected in an infinite rubber sheet, with mass concentrated at the feature points. This representation could then be matched and warped to generate a set of modified (time, frequency, energy) spots which could then be resynthesized. The trick in resynthesis would be to assure that the features are remote enough that phase interactions would not be perceptually important [36].