Report Number: WUCSE-2006-57

2006-01-01

Manifold Learning for Natural Image Sets, Doctoral Dissertation August 2006

Richard Souvenir

The field of manifold learning provides powerful tools for parameterizing high-dimensional data points with a small number of parameters when this data lies on or near some manifold. Images can be thought of as points in some high-dimensional image space where each coordinate represents the intensity value of a single pixel. These manifold learning techniques have been successfully applied to simple image sets, such as handwriting data and a statue in a tightly controlled environment. However, they fail in the case of natural image sets, even those that only vary due to a single degree of freedom, such as... **Read complete abstract on page 2.**

Manifold Learning for Natural Image Sets, Doctoral Dissertation August 2006

Richard Souvenir

Complete Abstract:

The field of manifold learning provides powerful tools for parameterizing high-dimensional data points with a small number of parameters when this data lies on or near some manifold. Images can be thought of as points in some high-dimensional image space where each coordinate represents the intensity value of a single pixel. These manifold learning techniques have been successfully applied to simple image sets, such as handwriting data and a statue in a tightly controlled environment. However, they fail in the case of natural image sets, even those that only vary due to a single degree of freedom, such as a person walking or a heart beating. Parameterizing data sets such as these will allow for additional constraints on traditional computer vision problems such as segmentation and tracking. This dissertation explores the reasons why classical manifold learning algorithms fail on natural image sets and proposes new algorithms for parameterizing this type of data.

2006-57

Manifold Learning for Natural Image Sets, Doctoral Dissertation August 2006

Authors: Richard Souvenir

Abstract: The field of manifold learning provides powerful tools for parameterizing high-dimensional data points with a small number of parameters when this data lies on or near some manifold. Images can be thought of as points in some high-dimensional image space where each coordinate represents the intensity value of a single pixel. These manifold learning techniques have been successfully applied to simple image sets, such as handwriting data and a statue in a tightly controlled environment. However, they fail in the case of natural image sets, even those that only vary due to a single degree of freedom, such as a person walking or a heart beating. Parameterizing data sets such as these will allow for additional constraints on traditional computer vision problems such as segmentation and tracking. This dissertation explores the reasons why classical manifold learning algorithms fail on natural image sets and proposes new algorithms for parameterizing this type of data.

Type of Report: Other

WASHINGTON UNIVERSITY

THE HENRY EDWIN SEVER GRADUATE SCHOOL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MANIFOLD LEARNING FOR NATURAL IMAGE SETS

by

Richard M. Souvenir, M.S. Computer Science, B.S. Applied Science

Prepared under the direction of Professor Robert Pless

A dissertation presented to the Henry Edwin Sever Graduate School of
Washington University in partial fulfillment of the
requirements for the degree of

DOCTOR OF SCIENCE

August 2006

Saint Louis, Missouri

WASHINGTON UNIVERSITY
THE HENRY EDWIN SEVER GRADUATE SCHOOL
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ABSTRACT

MANIFOLD LEARNING FOR NATURAL IMAGE SETS

by

Richard M. Souvenir

ADVISOR: Professor Robert Pless

August 2006

Saint Louis, Missouri

The field of manifold learning provides powerful tools for parameterizing high-dimensional data points with a small number of parameters when this data lies on or near some manifold. Images can be thought of as points in some high-dimensional image space where each coordinate represents the intensity value of a single pixel. These manifold learning techniques have been successfully applied to simple image sets, such as handwriting data and a statue in a tightly controlled environment. However, they fail in the case of natural image sets, even those that only vary due to a single degree of freedom, such as a person walking or a heart beating. Parameterizing data sets such as these will allow for additional constraints on traditional computer vision problems such as segmentation and tracking. This dissertation explores the reasons why classical manifold learning algorithms fail on natural image sets and proposes new algorithms for parameterizing this type of data.

Contents

List of Tables

List of Figures

Acknowledgments

I would very much like to thank my adviser, Dr. Robert Pless, for his support of my graduate study through our many discussions and collaborative efforts. It has been a privilege to work under him and I hope that I can one day mentor students as effectively as he did for me. I would also like to acknowledge the other members of my committee, Dr. Sally Goldman, Dr. Tao Ju, Dr. Samuel Wickline and Dr. Weixiong Zhang, who have offered their time and expertise to assist in the evaluation of my research.

I would like to thank Gazihan Alankus, Michael Dixon, Matthew Hampton, Nathan Jacobs, Christine Julien, Tobias Mann, Qilong Zhang, and, especially, Jamie Payton for lending valuable assistance throughout my progression towards this graduate degree. As friends and colleagues, they have always been willing to share their time by answering questions or providing critiques on practice talks and paper drafts.

I appreciate very much the efforts of the department faculty and staff who have applied their time and talent to develop an intellectually stimulating environment for teaching, learning, and research. I would like to specifically acknowledge Peggy Fuller, Jean Grothe, Myrna Harbison, Sharon Matlock, and Stella Sung whose efforts ensure that the department functions effectively.

Finally, I would especially like to thank my parents, Yves and Elvire L. Souvenir, for all of their love and support.

The research represented in this dissertation was supported generously by the National Science Foundation Graduate Research Fellowship Program under grant number DGE-0202737.

Richard M. Souvenir

Washington University in Saint Louis
August 2006

Chapter 1

Introduction

> In a very large part of morphology, our essential task lies in the comparison of related forms rather in the precise definition of each; and the deformation of a complicated figure may be a phenomenon easy of comprehension, though the figure itself may have to be left unanalyzed and undefined.
>
> —D'Arcy Thompson, [61]

Generalized video analysis is a difficult problem. For humans, understanding and describing the contents of a moving scene can be, at best, subject to personal bias, and, at worst, ambiguous. Consider the images in Figure 1.1. These are sample frames from a video captured in the Human Performance Laboratory of Dr. Jack Ensberg at the Washington University Medical School. Taken as a whole, understanding and describing this scene is an easy task. If asked to describe this video, most people would answer that this video depicts a woman running on a treadmill. Moreover, such a description would suffice to allow a listener to visualize a scene similar to the video being described. However, now let us consider any single frame from this

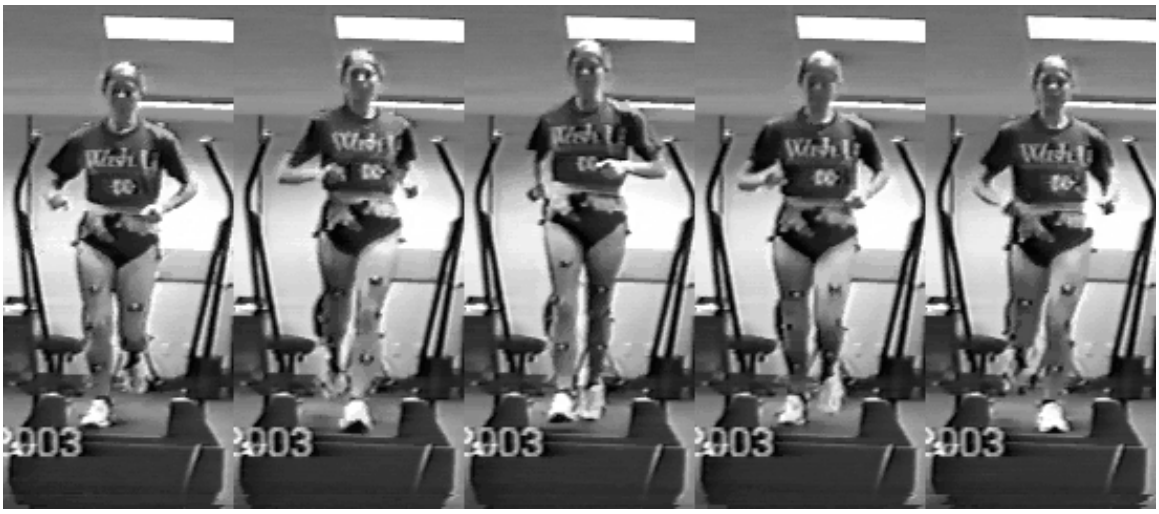


Figure 1.1: Sample video frames from a woman running on a treadmill.

video. If asked to describe this single frame, the task becomes more ambiguous. The image could be described on a semantic level by reporting the relative position of the body parts. Another description could include directly reporting the pixel values of the image. Also, if some metadata, such as the frame number or time-stamp, was available, this could be used to provide some information. The choice of description is arbitrary and, in general, would not provide enough information for a listener to understand the contents of the scene.

Now, let us consider the sample frames in Figure 1.2. These frames depict an image set of a clock at various times. In this case, the first question, describing the contents of the video, is again quite easy. Most people would describe this video as a working clock. However, in this case, the second question of describing any single frame is also quite easy. Most people would simply read the time off the clock to describe the content of the image. In this example, the most obvious and strongest model of the video is "working clock" and this model is meaningfully parameterized by the time of day. This information suffices to give a reasonable representation of the image data.

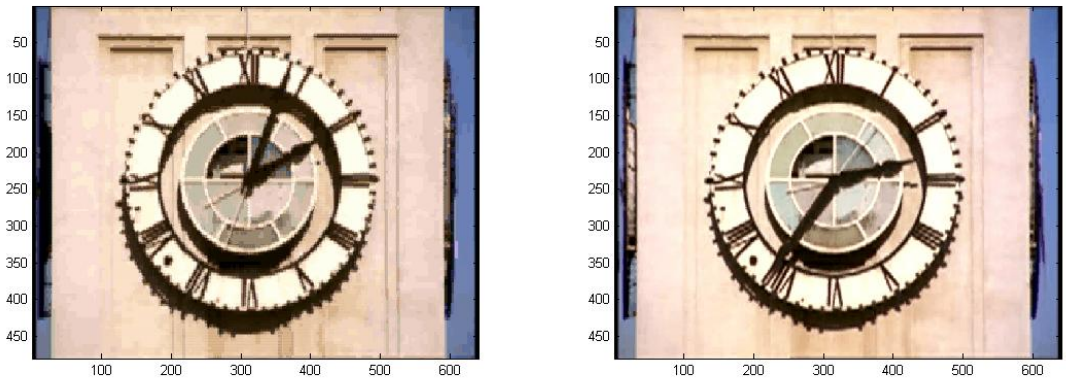
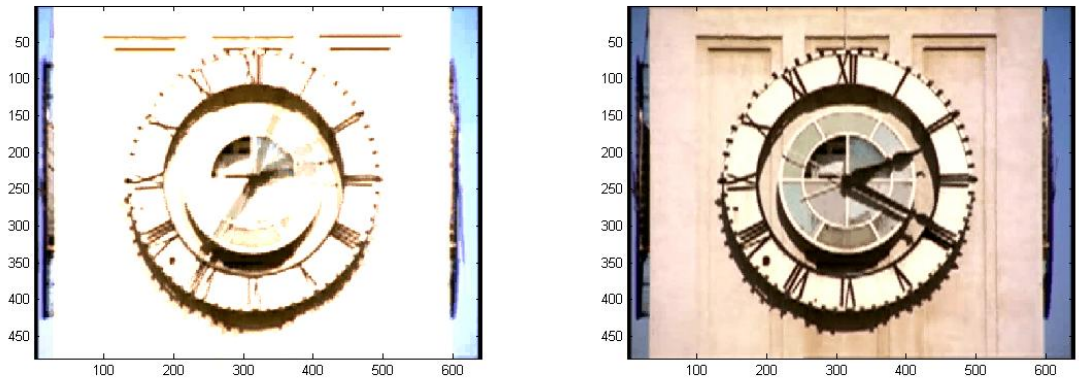


Figure 1.2: Sample video frames from a clock over time.

In computer vision, the exercises described above are generally divided into a set of separate, but related tasks.

- *Segmentation:* separating an object of interest from the background.
- *Recognition:* finding an object of interest in a scene.
- *Tracking:* maintaining location information for an object of interest over time in a video.
- *Registration:* finding corresponding points or features in multiple images.

It is the underlying hypothesis of the work in this dissertation that learning good models with meaningful parameterizations simplifies the computer vision tasks described above and extends automated video analysis to scenes of low image quality and/or complex (and possibly ambiguous) semantic representation. For example, let us reconsider the previous clock example and a simple, yet possibly ambiguous, computer vision task. Supposed the task was to find the "average" of the two images depicted in Figure 1.2. Figure 1.3 shows two possible results. Figure 1.3(a) shows



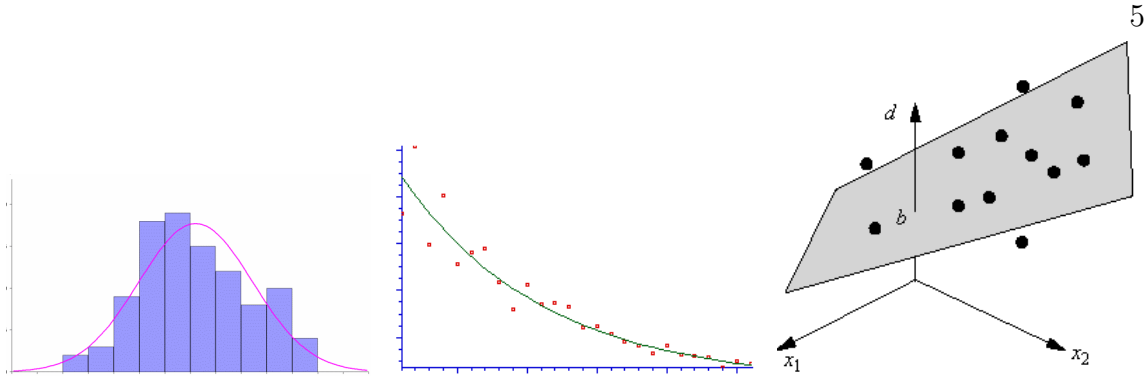
(a) Image-based average

(b) Model-based average

Figure 1.3: Two possible "average" images of the of the frames depicted in Figure 1.2.

the typical computer vision result obtained by taking the average pixel intensity at each location. Figure 1.3(b) shows the result by employing additional information about the operation of clocks and averaging the times represented by each image and displaying an image which most closely represents this "average." In this case, one could argue that by using the strong clock model, the second result is more reasonable and more useful than the first. One could then say that this computer vision task was improved by using a strong model. This example worked well mainly because our semantic understanding of images depicting clocks is quite strong. In this dissertation, we will consider automated methods for improving common vision tasks using data modeling where, unlike this example, human intervention is not required.

General data modeling forms the backbone of many sub-disciplines of engineering, mathematics, and science. The fact that this premise has found acceptance in many other areas of work indicates that it is a promising approach to problems in modeling natural images. The remainder of this introduction reviews data modeling methods for general data sets and images.



(a) Normal distribution over 1-D data (b) Curve fitting over 2-D data (c) Plane fitting over 3-D data

Figure 1.4: (a) A curve of the normal distribution is fit to histogram data. (b) A parametric curve is fit to 2D data points. (c) A plane is fit to 3D data points.

1.1 Data Modeling

Finding compact, meaningful parametric representations for data sets is quite common. Especially for data sets that are large or whose constituent points are of high dimensionality, data modeling can provide a way to more easily understand, describe, and visualize the data. Figure 1.4 shows some example data points and the models that provide good representations.

Figure 1.4(a) shows an example of fitting a normal distribution to a 1-dimensional histogram. The curve can be represented by the mean and variance and this provides a simpler representation of the (possibly large amount of) histogram data. Figure 1.4(b) shows an example of fitting a curve to a set of points which lie in 2-dimensions. This parametric curve provides a compact representation of the input data. In addition, by having such a robust model, it is now possible to interpolate among points from the input set and accurately estimate the remainder of the data from the universe which could also be included in the set. Finally, Figure 1.4(c) shows an example of fitting a plane to a set of 3D points. This is similar to the example in Figure 1.4(b). In these

two examples the property of dimensionality reduction is highlighted. In Figure 1.4(c), it suffices to know the parameters of the plane and the 2-dimensional coordinates of the input points on that plane to accurately describe each original 3-dimensional point. That is, by fitting this plane to the data, it requires less information to describe each point. This dimensionality reduction thus provides a more compact representation of the original data. The goal of this dissertation is to find such representations for image data and use those representations to improve common vision tasks. In the next section, we describe, in detail, the most widely used method for dimensionality reduction of large data sets whose constituent components are of high dimensionality.

1.1.1 Principal Component Analysis

One of the most widely used geometric data modeling techniques is Principal Component Analysis (PCA) [36]. PCA finds the linear subspace that best represents the input data. Figure 1.5 shows an example of the 1-dimensional linear subspace (in red) that best represents the 2-dimensional data set.

Given an input data set, $\mathcal{X}$, which is a finite subset of $\mathcal{R}^D$, PCA computes a function f which projects each image onto a set of basis vectors. The input set, $\mathcal{X}$, is used to derive a set of orthonormal basis images $\vec{b}_1, \vec{b}_2, \ldots \vec{b}_d$. The function f which maps an point x in $\mathcal{R}^D$ to a set of coefficients in $\mathcal{R}^d$ is:

$$f(x) = (x^\top \vec{b}_1, x^\top \vec{b}_2, \ldots, x^\top \vec{b}_d) = (c_1, c_2, \ldots, c_d) \tag{1.1}$$

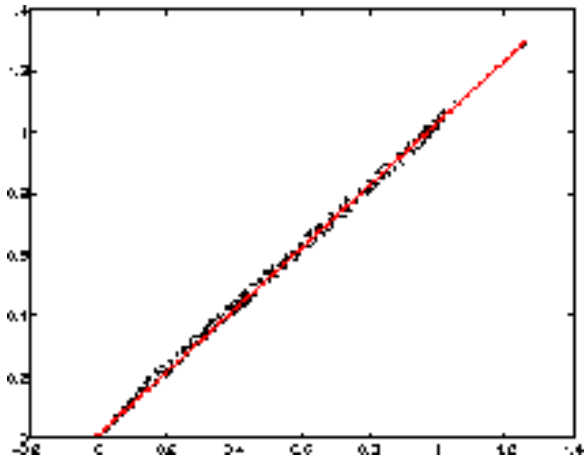


Figure 1.5: The PCA basis vector (in red) that best represents this 2-dimensional data set

One of the major advantages of PCA is that although the basis images are defined based upon an Eigen-analysis of the input data set $\mathcal{X}$, the function f is defined for all possible points of D dimensions:

$$f_{PCA} : \mathcal{R}^D \longrightarrow \mathcal{R}^d \tag{1.2}$$

The projection function f of PCA remains well defined for points that are not present in the original set $\mathcal{X}$. Also, the inverse function is defined as well, so that any point in the coefficient space can be mapped to a specific high-dimensional point by a linear combination of basis vectors:

$$f_{PCA}^{-1}(c_1, c_2, \ldots c_d) = c_1\vec{b}_1 + c_2\vec{b}_2 \ldots + c_d\vec{b}_d \tag{1.3}$$

1.2 Image Data Modeling

Images can be thought of as points in some high-dimensional image space where each coordinate represents the intensity value of a single pixel. In this framework, an intensity image from a standard 5 megapixel camera would be a point in a 5,000,000 dimensional space (or 15,000,000 if you consider RGB values). Grouping or analyzing points in such a high-dimensional space is rather difficult, as explained by the so-called "curse of dimensionality." Therefore, finding low-dimensional representations of this type of data is a natural solution.

Data modeling, using dimensionality reduction, in images has been used for many tasks. In this section, we will introduce how dimensionality reduction, specifically PCA, has been used with images and discuss the limitations which will motivate the work of this thesis.

1.2.1 PCA for Recognition

PCA is the foundation for the well-known Eigenfaces [64] algorithms for face recognition. Figure 1.6 shows sample images from the Olivetti Research Laboratory (ORL) face database [52] and Figure 1.7 shows the first 4 eigenfaces, which are also known as principal vectors or basis images. The general eigenfaces methods works by finding the coefficients for each of the images in the labeled input set. Then, when a new image is introduced to the algorithm, a function (similar to the one described in Equation 1.2) is applied to find the "closest" image(s) from the training set. The test image is then labeled. The details for finding "nearby" images and labeling the test

Figure 1.6: Sample images from the Olivetti Research Laboratory (ORL) face database.

image vary among the different algorithms. However, one thing to note is that all of the faces are generally in the same position and the top eigenfaces capture the main causes of differences among the image set.

PCA has also been extensively used in the subbranch of computer vision and machine learning known as *content-based image retrieval (CBIR)* [57]. In CBIR, the idea is to find images from a database based on user input. The user input is generally in the form of text, related pictures, sketches, or relevance scoring of returned images. The basic idea [43, 42] for using PCA in CBIR is that images of similar objects will have similar PCA coefficients and thus reduces the search space of relevant images in the, usually large, database. This is followed by various optimization steps to return the closest, or best match.

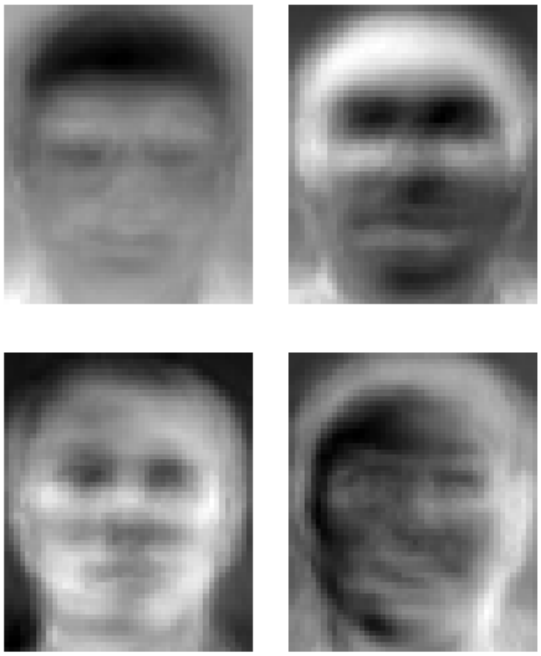


Figure 1.7: The first four eigenfaces of the ORL face database.

1.2.2 Limitations of PCA on Images

The previous examples demonstrate the use of PCA on images for recognition. This implies that PCA representations of image sets can be useful descriptors for individual images. However, one can now ask *how* useful are these representations and what is their effectiveness on typical image sets. Recall the clock example and strong representation model which could be used to represent each image. Can we get similar results with PCA? Figure 1.8 shows a sample data set where a statue was placed atop a motorized turntable and viewed from a camera on an elevating arm. In this case, there are two degrees of freedom, namely the rotation of the turntable and the elevation of the camera arm. Figure 1.9 shows the first 5 PCA basis images and reconstruction of single image using progressively more PCA basis images.

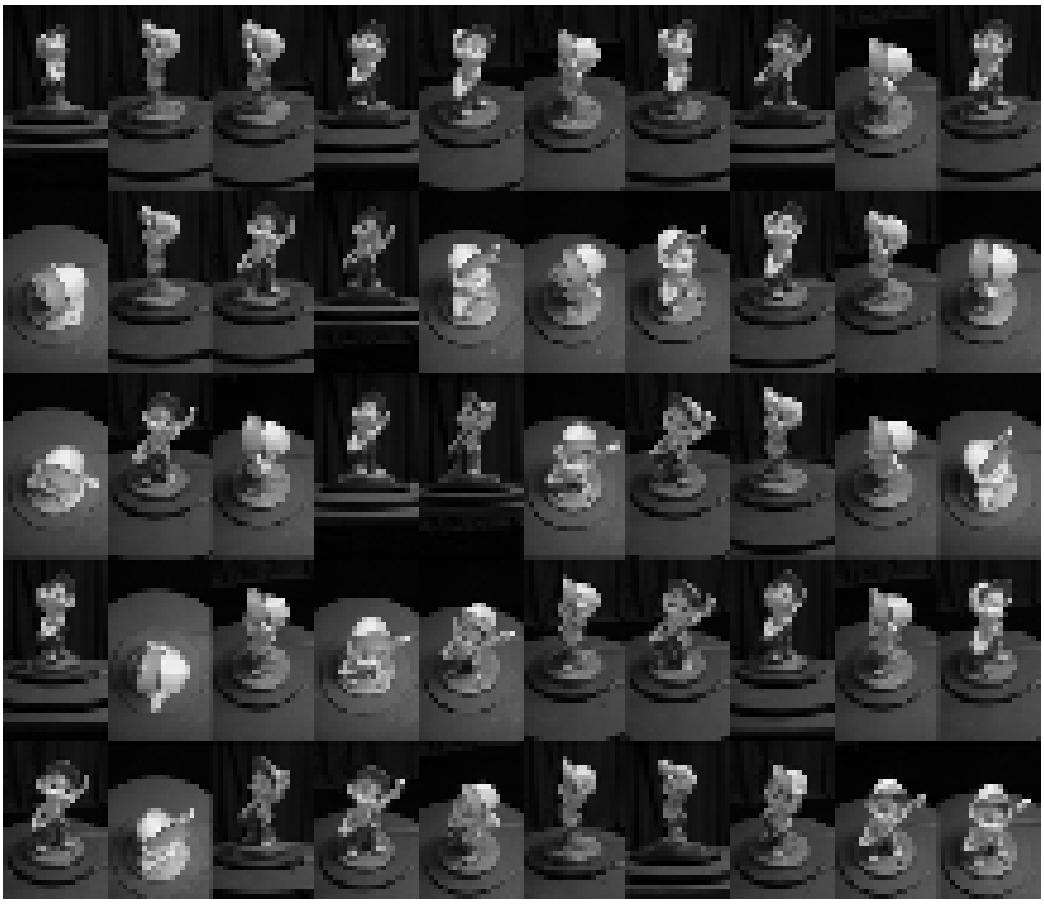
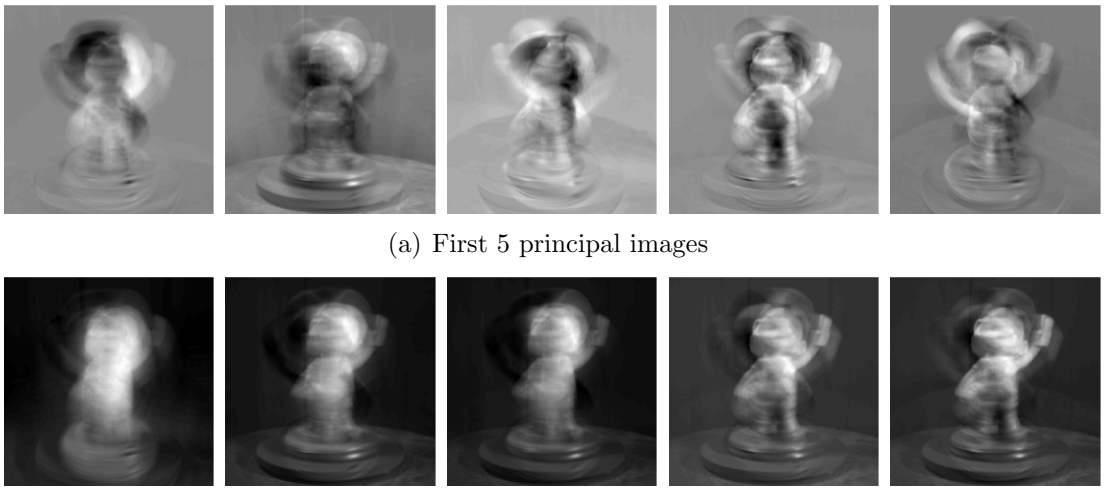


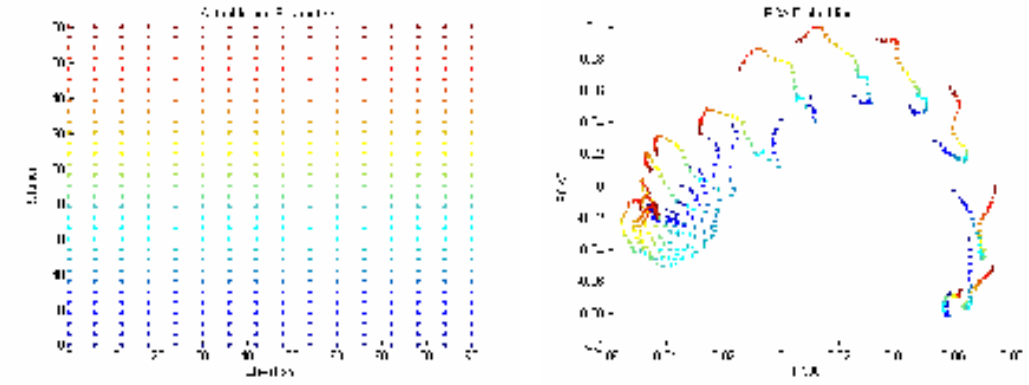
Figure 1.8: Sample images from a statue data set with two degrees of freedom, namely rotation and camera elevation.



(a) First 5 principal images

(b) Image reconstruction using PCA

Figure 1.9: (a) The first 5 principal images of the data set depicted in Figure 1.8. (b) The reconstruction of one frame of the original image set as the linear combination of progressively more principal images.



(a) Input data

(b) Input data

Figure 1.10: In each graph, each point represents an original frame the image set depicted in Figure 1.8. (a) Known rotation and elevation for each image. (b) The plot of the first two principal coefficients for each image. It is clear that PCA has failed to learn the intrinsic parameterization of this data set.

Figure 1.9 shows the PCA decomposition of this statue data set. Despite the fact that this image set has only two degree of freedom which are both easy to describe, it takes many principal components to reconstruct any of the original images effectively. Figure 1.10 shows a plot of the first two PCA coefficients for this data set. In each graph, each point represents an original frame from the image set. Figure 1.10(a) plots the known rotation and elevation for each image and Figure 1.10(b) shows the plot of the first two principal coefficients for each image. The first graph shows the expected output. In this case, it appears that PCA does not provide a compact, reasonable representation of this data set.

PCA models every image as the linear combination of a set of basis images. However, this is not usually the type of image change that underlies natural image variations. Natural changes to images, for example those due to variation in pose or shape deformations, are very poorly approximated by changes in linear basis functions. In the remainder of this dissertation, we discuss more recent methods for dimensionality reduction, demonstrate that, in general, they also perform poorly on images, and provide extensions and new algorithms to improve data modeling for natural image sets.

1.3 Contributions

The work of this dissertation makes the methods of nonlinear dimensionality reduction applicable to natural image sets and provides the framework for image manifold learning. Specifically, the major contributions include:

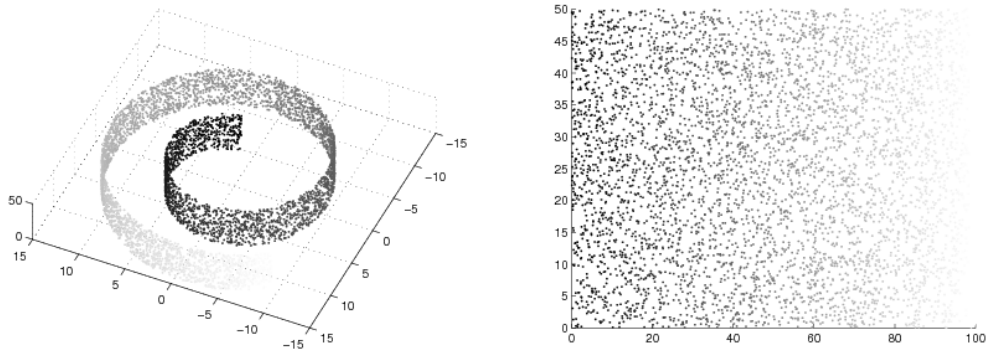
- Applying a statistical framework to quantify the differences between related images.
- Using the constraints implied by this framework to improve common vision tasks such as image registration, segmentation, and interpolation.
- Developing a novel algorithm for parameterizing complex topologies found in natural image sets.

Chapter 2

Manifold Learning

Manifold learning, or *nonlinear dimensionality reduction*, is the counterpart to PCA (described in the previous chapter) which aims to find a low dimensional parameterization for data sets which lie on *nonlinear* manifolds in a high-dimensional space. Figure 2.1 shows a classic example of manifold learning on a synthetic data set. Figure 2.1(a) depicts the so-called "Swiss Roll" data set which consists of 20,000 3-dimensional points. Intuitively, this data set can be visualized as points drawn off a rolled-up sheet of paper. While each of these points can be described by three coordinates (or more), there is an underlying two-dimensional representation. The goal of manifold learning is to automatically learn this representation. The expected output of manifold learning algorithms is shown in Figure 2.1(b).

This chapter describes the current state of manifold learning and the set of proposed algorithms in the field. Next, applications of these existing algorithms to images is then examined.



(a) 3D Input Data (b) Low-dimensional Embedding

Figure 2.1: Manifold learning example using the "Swiss Roll" data set which consists of 20,000 3-dimensional points. While each of these points can be described by three coordinates as shown in (a), there is an underlying two-dimensional representation as shown in (b). The goal of manifold learning is to automatically learn this representation. In this figure, the intensity of the points represents distance along the curved "axis."

2.1 Overview of Methods

Manifold learning has been described in various ways throughout the literature. For the purposes of this dissertation, we choose the following problem statement:

> Given an input set $\mathcal{X}$, which is a finite subset of $\mathcal{R}^D$, for some dimension D, learn a parameterization which produces a mapping function $f : \mathcal{X} \longrightarrow \mathcal{R}^d$ which preserves *some properties* of the structure of $\mathcal{X}$.

This problem statement is intentionally vague. The choice of which properties of the input set to preserve in order to learn the low-dimensional parameterization is quite varied. The remainder of this section focuses on current methods for manifold learning and describes the properties of the input data set which each preserves. There exist a number of algorithms for manifold learning which we will briefly review in the next section. However, we will focus on two of the most popular algorithms, Isometric

Feature Mapping (Isomap) [60] and Locally Linear Embedding (LLE) [50]. These two algorithms, developed in parallel, but independently, signaled the beginning of a surge of interest in nonlinear dimensionality reduction.

2.1.1 Isomap

Isomap is a manifold learning algorithm which preserves geometric features of the input set. Specifically, the goal of Isomap is to return an isometric mapping,

$$f : \mathcal{X} \longrightarrow \mathcal{Y} \tag{2.1}$$

for $\mathcal{X} \subset \mathcal{R}^D$, $\mathcal{Y} \subset \mathcal{R}^d$, and $d << D$ where, for all pair of points $X_i \in \mathcal{X}$ and $X_j \in \mathcal{N}(X_i)$,

$$|X_i - X_j|_2 = |Y_i - Y_j|_2 \tag{2.2}$$

and $\mathcal{N}(X)$ is defined as the set of points which comprise the neighborhood of X. That is, Isomap returns an embedding where the distances of local neighbors in the original space is preserved. Below, we describe the main steps of the algorithm.

Given: A set of points $\mathcal{X} \subset \mathcal{R}^D$

1. Compute the distance between all pairs of points (traditionally using the L_2 norm distance.)

2. Define the set of points which comprise the neighborhood, $\mathcal{N}(X_i)$ for each point, $X_i \in \mathcal{X}$. This is typically done in one of two ways:

   - *k-nearest neighbors.* Select the k closest points to X_i.
   - *ϵ-ball.* Select all points $X_j \in \mathcal{X}$ such that $|X_j - X_i|_2 < \epsilon$.

3. Define a graph with a node for each input point, X_i and weighted undirected edges connecting each node to the nodes corresponding to the points in $\mathcal{N}(X_i)$. For each edge, the weight equals the corresponding distance between the input points.

4. Solve for the all-pairs shortest paths on this sparse graph to calculate a complete pair-wise distance matrix.

5. Solve for the low-dimensional embedding $\mathcal{Y} \subset \mathcal{R}^d$, using Multidimensional Scaling (MDS) [37] (described below). d is the dimension of the low-dimensional embedding and can be chosen as desired, but, ideally, is the number of degrees of freedom in the image set.

Output: $\mathcal{Y}$, the low-dimensional embedding of $\mathcal{X}$

Multidimensional Scaling (MDS)

The final step of Isomap requires embedding using MDS. The basic idea behind MDS is to convert a matrix of pair-wise distance into absolute coordinates in some (typically low) dimensional space. Below, we formally describe the procedure.

Given an $n \times n$ matrix D, such that $D(i,j)$ is the desired squared distance from point i to point j:

1. Define $\tau = -HDH/2$, (H is the centering matrix: $H = I - \vec{e}\vec{e}^{\top}/n$, where $\vec{e} = [1, 1, \cdots, 1]^{\top}$).

2. Let $s_1, s_2, \ldots$ be the (sorted in decreasing order) eigenvalues of τ, and let $v_1, v_2, \ldots$ be the corresponding (column) eigenvectors. The matrix

   $Y = [\sqrt{s_1}v_1 | \sqrt{s_2}v_2 | \ldots \sqrt{s_k}v_k]$ has row vectors which are the coordinates of the best k-dimensional embedding.

The matrix $YY^{\top}$ is the best rank k approximation to τ (with respect to the L_2 matrix norm). This process finds the k-dimensional coordinates that minimize:

$$\sum_{ij} \left(|Y_i - Y_j|_2^2 - D(i,j) \right)^2 .$$

2.1.2 LLE

LLE is another manifold learning algorithm which makes different assumptions in order to learn a low-dimensional embedding. LLE attempts to represent the manifold

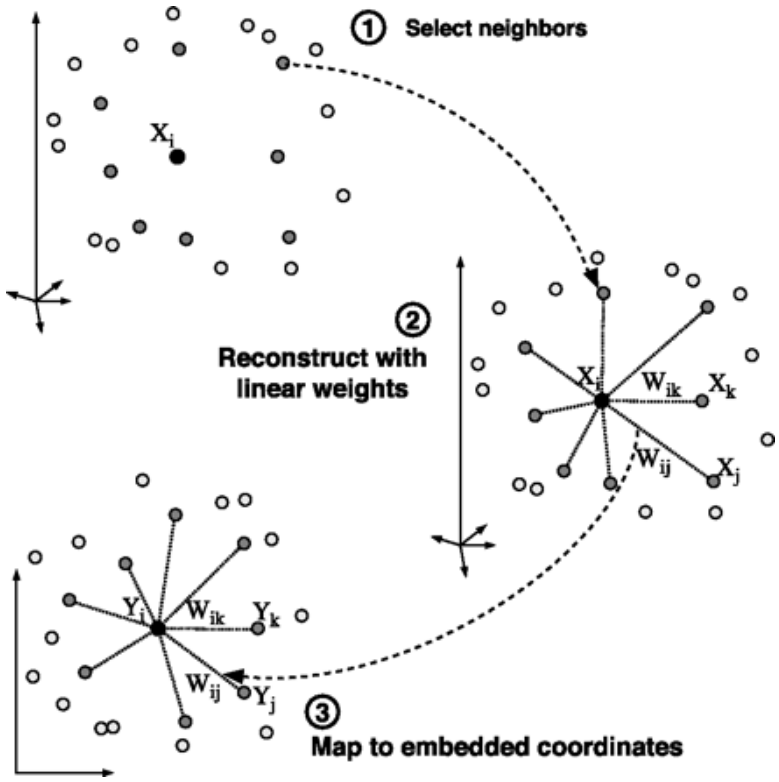
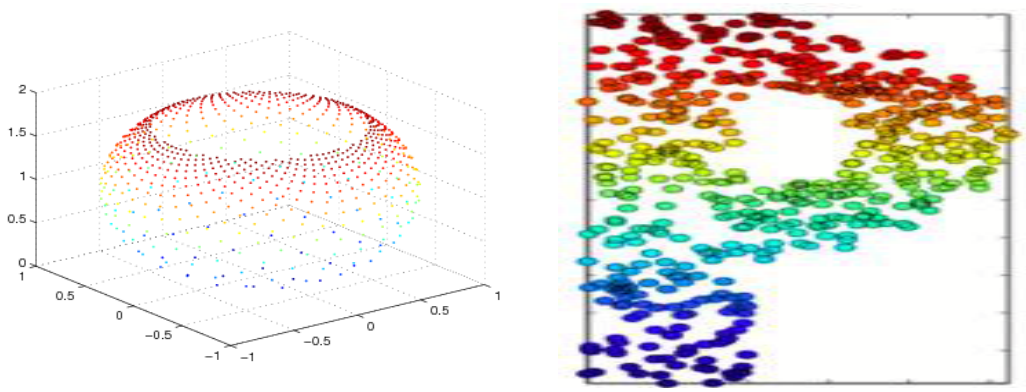


Figure 2.2: Diagram of the LLE algorithm. The three main steps are: (1) define the neighborhood for each point, (2) solve for the reconstruction weights, and (3) learn an embedding which preserves the reconstruction weights. Image obtained from `http://www.cs.toronto.edu/~roweis/lle/algorithm.html`



(a) "Fishbowl" data set (b) Non-convexity example

Figure 2.3: Examples of data sets for which early manifold learning algorithms fail. Table 2.1 lists a variety of manifold learning algorithms, some of which can accurately parameterize data sets such as these.

locally by reconstructing each input point as weighted combination of its neighbors. Below we formally describe the algorithm and show a diagram in Figure 2.2.

1. Define $\mathcal{N}(X_i)$ for each point in $\mathcal{X}$. As in step 3 of Isomap (above), the k-nearest neighbors or ϵ-ball methods can be used.

2. Solve for the reconstruction weights, W, where $W(i,j)$ represents the weight of X_j to reconstruct X_i. For $X_j \notin \mathcal{N}(X_i), W(i,j) = 0$. Normalize each row of W, such that $\sum_j W(i,j) = 1$, for each row i.

3. Learn the embedding coordinates $\mathcal{Y}$ using the weights W by solving an eigenproblem. Define $M = (I - W)' \times (I - W)$. Set $\mathcal{Y}$ to be the the eigenvectors of M corresponding to the d smallest eigenvalues after discarding the smallest (with eigenvalue of zero.)

Table 2.1: Manifold learning algorithms

Algorithm Class	**Examples**
Isomap variants	ST-Isomap [35] Continuum Isomap [22, 74] Landmark Isomap [21] Conformal Isomap [21]
Charting	Manifold Charting [10] Non-linear CCA & PCA [65]
Self-Organizing Maps	[1, 7, 66]
Graph Spectral Methods	Laplacian Eigenmaps [3] Kernel Eigenmaps [11] Hessian Eigenmaps [23] Locality Preserving Projections [30]
Supervised Methods	Local Fisher Embedding [20] Supervised LLE [19]
Other	Diffusion Maps [16] Manifold Tangent Learning [4] Proximity Graphs [12] Semidefinite Embedding [69] Stochastic Neighbor Embedding [32] Local Smoothing [44]

2.1.3 Other Methods

The original Isomap and LLE algorithms worked well for data sets such as the "Swiss Roll" in Figure 2.1. However, due to the assumptions made by these algorithms, certain data sets are not parameterized very well. Figure 2.3 shows sample data sets where either Isomap or LLE failed. Some of the reasons include highly curved manifolds, "short-circuits", and non-convexity. A number of incremental algorithms have been devised to deal with these and other complexities in manifold learning. Table 2.1 represents a list of many of these algorithms for manifold learning. The breadth of attempted solutions highlight the broad interest in the problem.

The variants of Isomap all follow the general steps described in Section 2.1.1. ST-Isomap can be applied to data with a temporal component, such as frames of a video, and works by modifying the local neighborhood structure and distance matrix to reduce the distance to both spatially and temporally adjacent points. Landmark Isomap trades off accuracy for speed by only using a subset of the points for the embedding step. Conformal Isomap addresses a sampling problem into to faithfully embed data sets such as the "fishbowl" depicted in Figure 2.3.

Another algorithm, Semidefinite Embedding (SDE), is related to Isomap in that the goal of the method is to provide an isometric embedding. In fact, the final step, embedding using MDS, is identical. The major difference is in the construction of the similarity matrix between all pairs of input points. SDE applies semidefinite programming to learn this kernel matrix. This method does not fail in the case of non-convexity like Isomap and can correctly parameterize the "P shape" in Figure 2.3.

Self-Organizing Maps (SOMs) [49], also known as Kohonen Feature Maps, precede most of the algorithms in Table 2.1. Intended as a visualization tool for high-dimensional data, the use of SOMs for manifold learning was not discovered until later. SOMs follow the general framework of artificial neural networks trained using competitive learning [51] to discover a low-dimensional embedding for the input data.

The charting methods represent the high-dimensional manifold as a set of overlapping "charts" or "patches". The chart sizes may either be fixed or expandable until some assumption is violated, such as local planarity. In contrast to Isomap and LLE, most of these methods do not provide a globally consistent parameterization of the input data set, but rather parameterizations within local regions.

All of the algorithms described so far are largely unsupervised, however, there exist a related class of semi- and fully supervised manifold learning algorithms which are variants of LLE. These algorithms generally retain the assumption that each point can be represented as a local combination of its neighbors, however introduce supervision in order to output smoother manifolds and better global representations. All of the work described in this dissertation will focus on unsupervised methods for manifold learning.

2.2 Summary of Manifold Learning Methods

The previous section described the major algorithms for manifold learning. Even though all of these methods work well on data sets, such as the "Swiss Roll" depicted in Figure 2.1, it is important to highlight some of the major limitations of manifold learning algorithms, in general. First, these methods define a mapping from the original data set to $\mathcal{R}^d$. That is, the result is a mapping

$$f : \mathcal{X} \longrightarrow \mathcal{R}^d$$

and not, as might be more convenient,

$$f : \mathcal{R}^D \longrightarrow \mathcal{R}^d.$$

This means that once the embedding of an data set $\mathcal{X}$ is computed, for $X' \notin \mathcal{X}$, the value of $f(X')$ is not well defined. Additionally, the inverse mapping is also problematic. For a point $Y \in \mathcal{R}^d$, if Y is not in the set of points defined by $f(\mathcal{X})$,

then $f^{-1}(Y)$ is also not well defined. Although approaches have been proposed to compute these "out of sample" projections [5], this remains, both theoretically and practically, a challenge for nonlinear dimensionality reduction techniques.

In summary, there are many varied approaches to manifold learning. While each algorithm makes different assumptions about the input data and the embedding mapping, all of these algorithms generally follow a simple two-step framework.

- Define the local neighborhood in the high-dimensional input space. This can be done explicitly using the k-nearest neighbors or ϵ-ball methods as previously described or implicitly using charts or diffusion distance.
- Extend the local constraints to learn a global low-dimensional parameterization.

Images can be thought of as points in a high-dimensional image space where each coordinate represents the intensity value of a single pixel, so a set of images, or a video, could be considered as a set of points in such a space. In the remainder of this dissertation, we explore how existing manifold learning algorithms have performed on image data and how this performance can be improved.

Chapter 3

Image Manifold Learning

This dissertation focuses on understanding the image changes which are the result of simple scene or object changes. Figure 3.1 revisits the data set discussed in Chapter 1 showing sample frames from a woman running on a treadmill in a laboratory. As previously discussed, the cause of the image change is easily describable, however the changes taking place in each image are rather complex. In this chapter, we apply the tools of manifold learning to image sets such as these to see if we can learn the simple representation, or underlying parameters.

3.1 Previous Work

Manifold learning has been successful at finding natural parameterizations, or "perceptual organizations" [60], of a variety of different image sets. Figure 3.2 shows an example of using Isomap for pose estimation of rigid body motions [45]. In this experiment, a statue was placed atop a turntable and viewed from a camera on an elevating arm. In this case, there are two degrees of freedom, namely the rotation

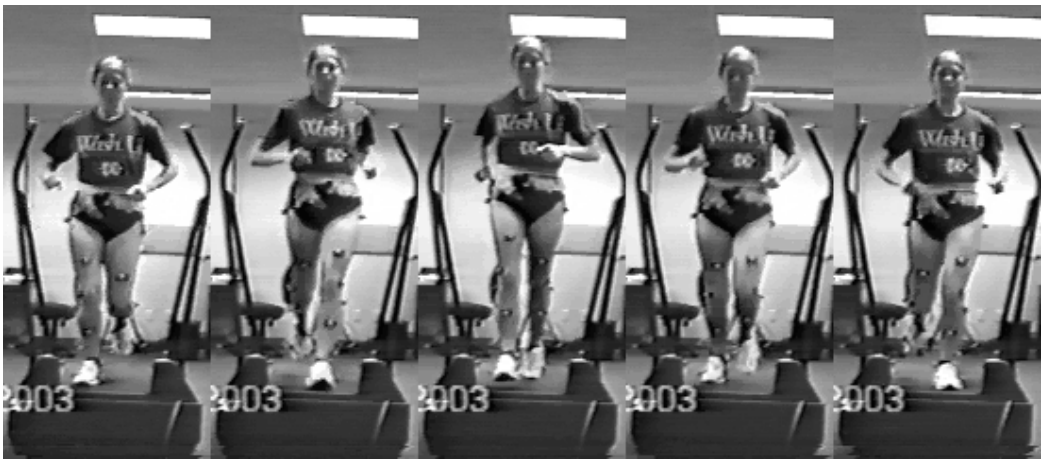
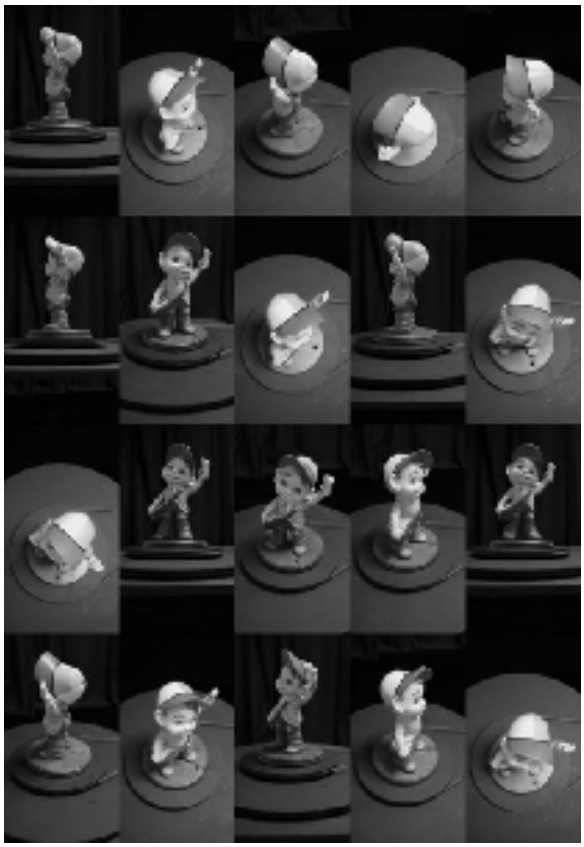


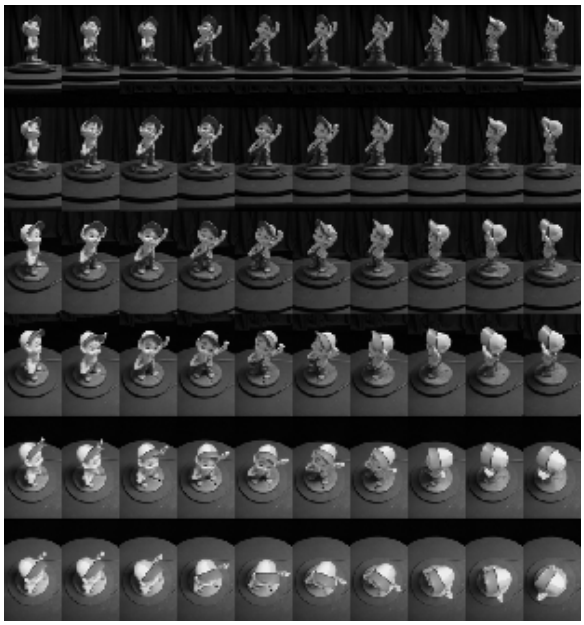
Figure 3.1: These are example frames of a woman running a treadmill.

of the turntable and the elevation of the camera arm. In this experiment, the pose of the statue in a set of unordered images was estimated by embedding the images with Isomap, labeling a small subset of the images with known rotation and elevation, and interpolating those known values using the embedding coordinates. Figure 3.3 depicts a planar arrangement of fish contours which serves as a useful visualization of biomedical image data sets [39]. By arranging the unordered contours, previously undiscovered similarities between various fish shapes were elucidated. Figure 3.4 shows the arrangement of a limited set of articulated hand poses [60]. This data set consists of images undergoing various amounts of both wrist rotation and finger extension. These results, which were one of the earliest applications of manifold learning on images, show how these methods can, at times, recover perceptually meaningful organizations of data.

In these examples, the "perceptual organizations" returned by the various algorithms seem to correspond to the intuitive interpretation of these images sets. Thus, it seems as if these manifold learning techniques overcome the limitations of PCA for the analysis of images, as described in Chapter 1. However, the changes seen in



(a) Input Data



(b) Ordered Results

Figure 3.2: In [45], the unordered images (a) were arranged using Isomap (b) to obtain pose estimates of the object. The mean embedding error over all 1800 images is only 6.98° for the rotation, and 2.97° for the camera elevation.

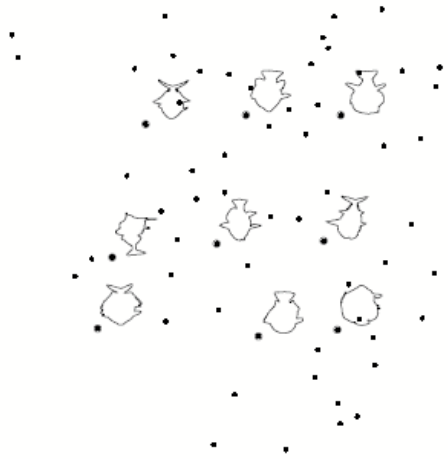


Figure 3.3: In [39], contours of various fish species were arranged using Isomap for visualization purposes.

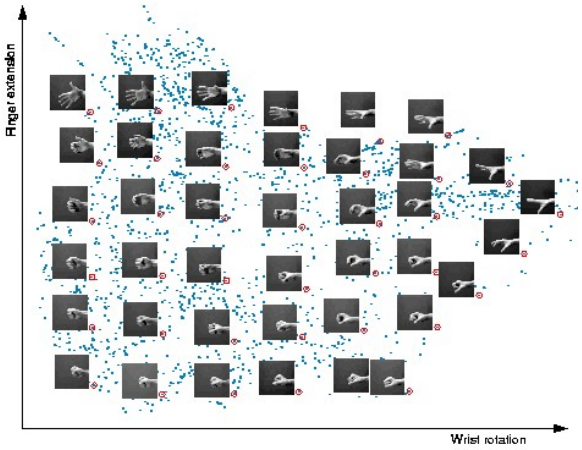


Figure 3.4: In [60], wrist images were generated by making a series of opening and closing movements of the hand at different wrist orientations to demonstrate how Isomap captures the intrinsic degrees of freedom.

natural image sets are more complex than the simple transformations demonstrated in the previous images.

The example in Figure 3.5 depicts four frames from an MR acquisition of a heart. MRI data is typified by large image sets which are often noisy. An image of a particular subject may vary for a number of reasons, including noise inherent in the sensor itself, motion of the subject during data capture, and time-varying effects of contrast agents that are used to highlight particular types of tissue. These variations are difficult to parameterize in a very general setting, but for a particular subject, the images are likely to lie on a low dimensional manifold.

The image set depicted in Figure 3.5 contains real-time cardiac MR images, captured during a 60 ms window during the systolic part of consecutive heartbeats. The data set includes 180 such images from the same patient.

The variation in these images has three causes. First, there is motion induced in the heart due to the breathing of the patient. In the sample images in Figure 3.5 this can be most easily seen by the differences in the relative positions of the heart (the object in the center of the scene) and the liver (the lighter object at the bottom of the scene.) Second, there is a general increase in the lighting intensity of the scene as time passes. This is due to a contrast agent that was administered to the patient which is slowly permeating through the tissues. Third, the MRI images are noisy. Figure 3.5 shows the 2-D Isomap embedding and a plot of the residual variance indicating that two parameters capture most of the information of this data set.

It appears that the Isomap algorithm has performed well for this particular data set. The results shown in the residual plot imply that these images lie on or near

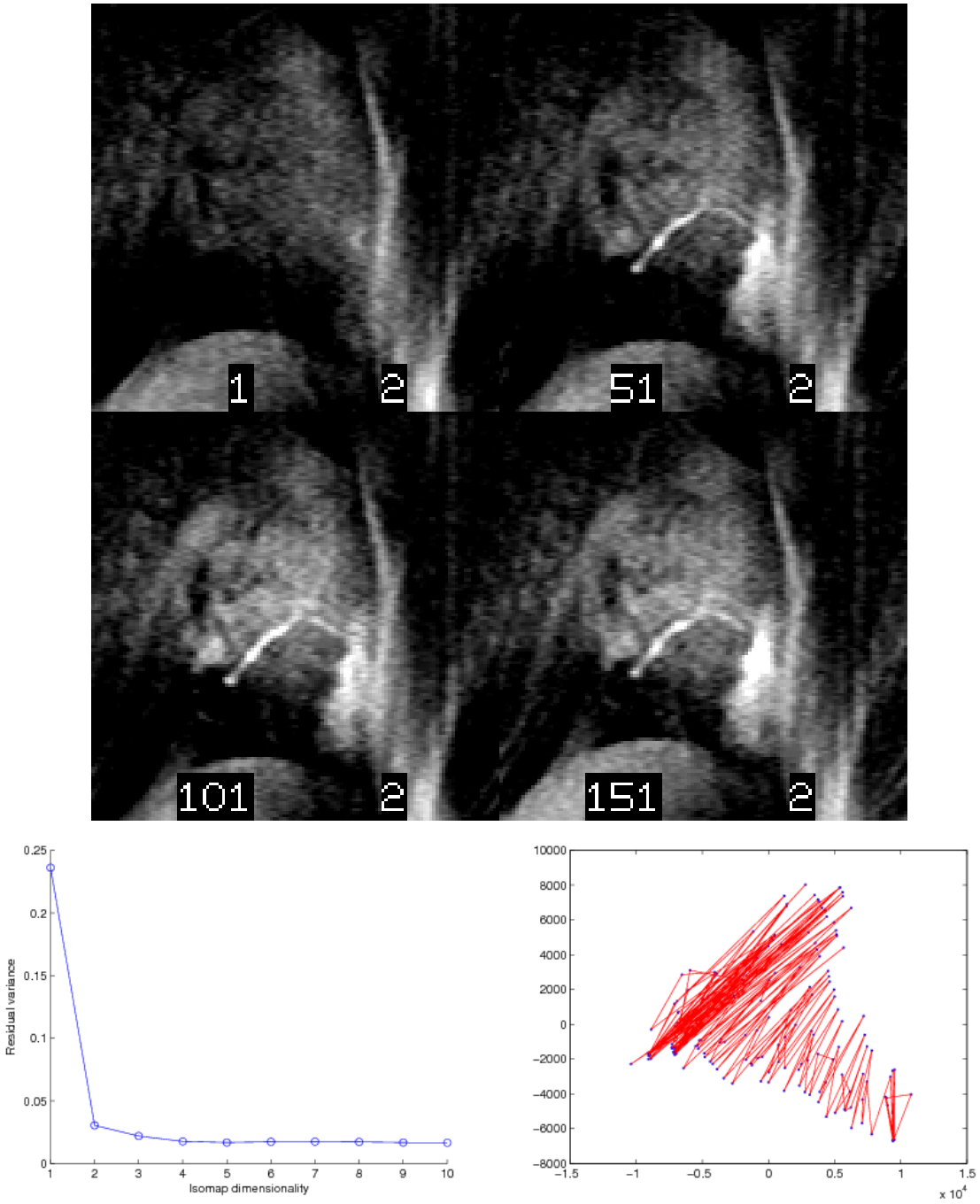


Figure 3.5: Four samples of a sequence of MR images, and the associated Isomap embedding (using $k = 8$ neighbors.) The plot of Isomap dimension versus residual error indicates that 2 dimensions suffice to capture most of the distance information. Each blue dot corresponds to an original frame from the video and the red line connects the images in the original order.

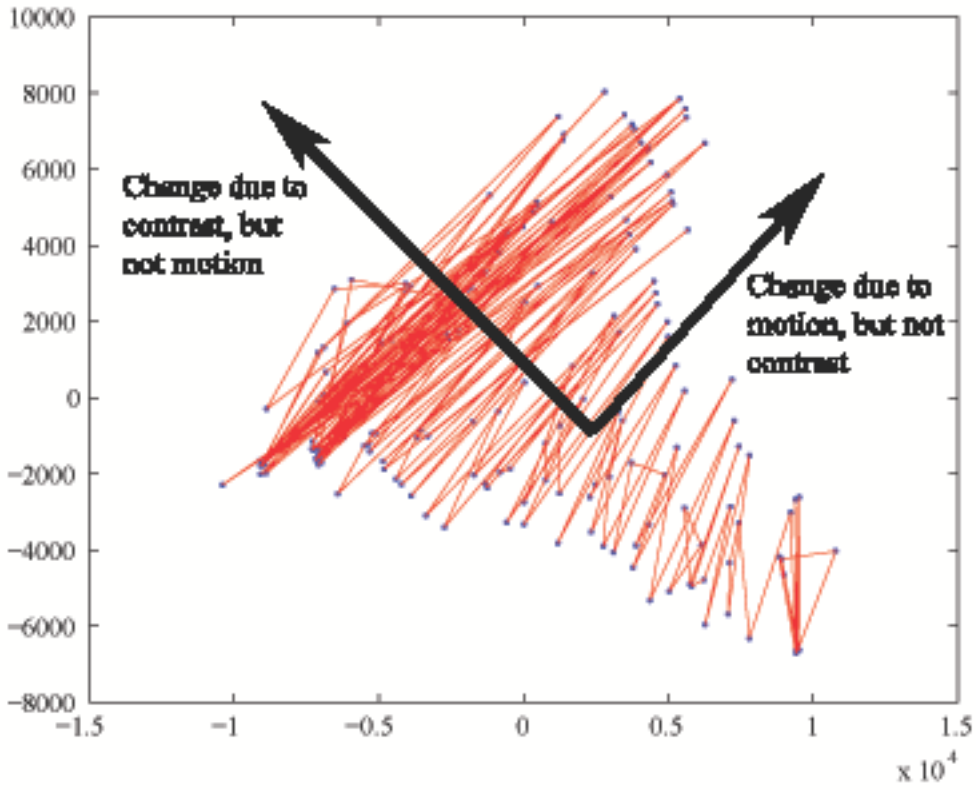


Figure 3.6: Isomap embedding of the video depicted in Figure 3.5. Each blue dot corresponds to an original frame from the video and the red line connects the images in the original order.

some manifold which can be accurately parameterized with two values. These results, however, do not appear to provide the most reasonable parameterization of this set. In Figure 3.6, this Isomap embedding is labeled with two axes which correspond to the major changes of this data set.

There are two main problems with this embedding. First, the natural $x-$ and $y-$ axes do not correspond to the known degrees of freedom of this data set. That is, a change in either parameter does not correspond to a natural change in the images in the data set. Second, the "scaling effect" seen as the trajectory is followed (in this case, from bottom right to top left) does not correspond to changes in the object of interest.

As the images are acquired the heart undergoes a rhythmic non-rigid deformation as a result of the breathing of the patient. Since the breathing rate of the patient is consistent, the amount of motion of the heart in neighboring frames should be similar. However, in this graph, the points spread apart along the "axis" of motion implying a greater distance (or more motion) for frames later sequence. Ideally, this embedding would provide two parameters which correspond to the major degrees of freedom of this data set. In addition, the distances between the points in the embedding should correspond to the amount of change of the image transformation.

It is worthwhile to point out that, for this particular example, the points in the embedding could be easily warped to solve these two main problems. However, as we are focused on unsupervised methods for this general problem, it is important to discover automated solutions. The remainder of this dissertation focuses on solving the problems introduced by "real" or natural image sets and proposes a set of methods which promise to make manifold learning useful for computer vision.

3.2 Learning Meaningful Parameterizations

In the previous section, we showed a cardiac MR example where traditional manifold learning methods fail to provide an adequate representation for the dominant cause(s) of change in the input set. The focus of this section is on the failure modes of current manifold learning techniques to provide natural parameterizations for data sets derived from images. The problem is that image sets (i.e., videos) which appear to lie on or near some manifold in image space are not parameterized as such.

In this work, one of the underlying themes will be providing *minimal parameterizations* for data sets. For natural image data sets, we would like to provide parameterizations which are minimal, natural, and descriptive. Part of this research will be to formally define this notion and provide minimal parameterizations for data sets which arise from natural video.

It is important to differentiate minimal parameterizations from intrinsic dimensionality estimates. There has been some recent work which estimates the intrinsic dimensionality of data sets which lie on or near some manifold. The methods presented in [38, 31] are based on well-understood principles from statistics and provide information similar to the residual error results obtained from Isomap. These types of results partially address our goal of minimality, however, do not provide interpretable parameters.

To address the problem of current algorithms not providing a meaningful parameterization of the image data, it is reasonable to ask two questions.

- Do these data sets, in fact, lie on some image manifold?
- If so, why do manifold learning algorithms fail for this type of data?

Our approach to this problem examines the general manifold learning framework described in Section 2.2. A fundamental operation in most manifold learning algorithms is finding a set of "neighboring" points for each of the input points. In the case of data lying in a high dimensional space, the most natural distance metric is the Euclidean distance. For video, when each image is considered a vector in some high-dimensional

space, the Euclidean distance metric between two images corresponds to the sum-of-squared differences of pixel intensities. This is the distance metric typically used in manifold learning on images. The rest of this chapter examines the consequences of using the Euclidean distance metric and suggests other distance metrics which may be more appropriate for various image sets.

3.3 Image Distance Functions

A formal theory of the statistics of natural images and natural image variations gives tools for defining relevant image distance metrics. We postulate that for natural image data sets, a small number of distance metrics are useful for many important applications. In this chapter, we propose a set of distance measures that correspond to the most common causes of transformation in image sets and gives examples of how these significantly improve the parameterization of natural image sets.

Our exploration starts by considering again the data set of a woman running on a treadmill, captured by a standard camcorder. Figure 3.7 shows the output of the Isomap algorithm using the most common image distance metric, the sum of the squared pixel intensity differences.

It appears as if the Isomap algorithm separates the image variations into two components. Tangential motion in the coefficient space (moving around the circle) corresponds to changes in the phase of the the running cycle. Radial motion in the coefficient space encodes the residual variation. In this sequence this most naturally corresponds to the left to right position of the runner on the treadmill. Figure 3.8

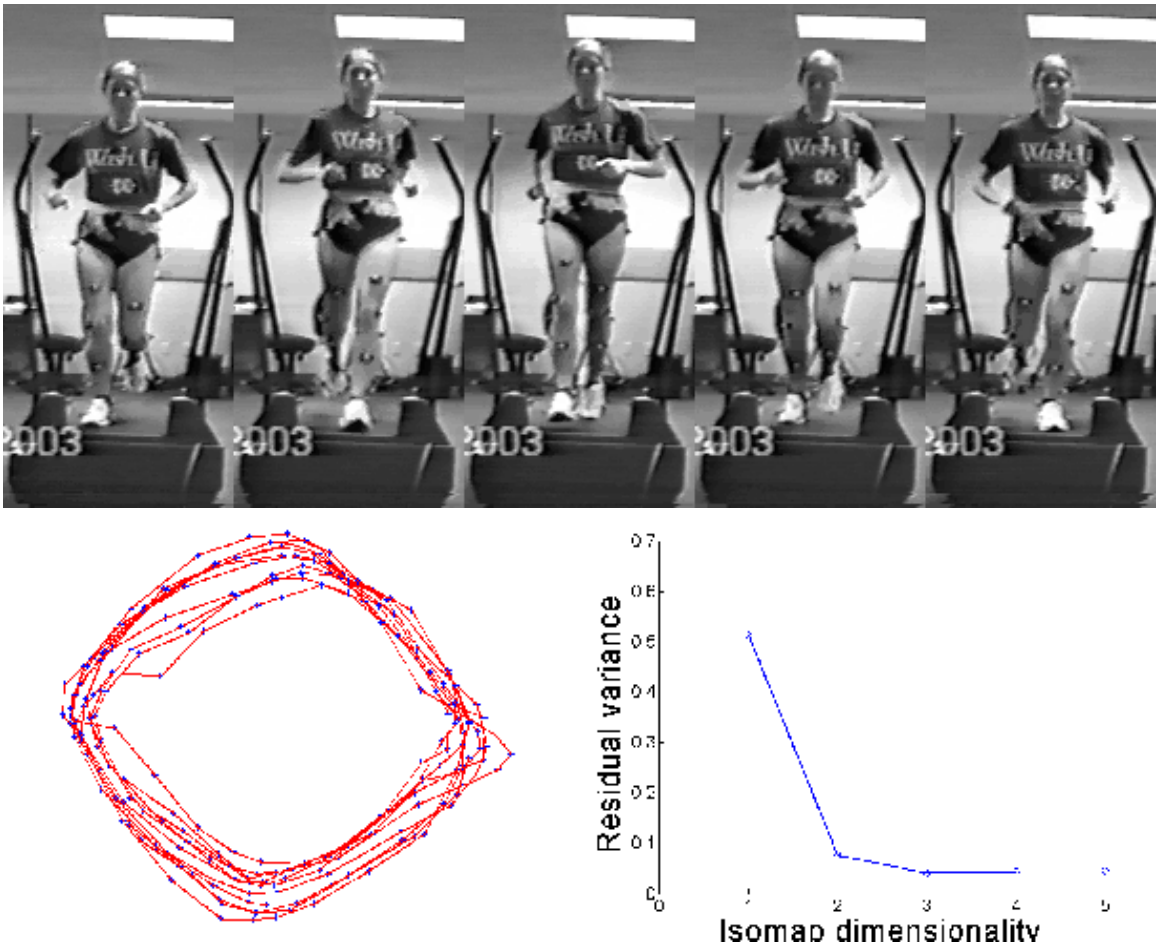


Figure 3.7: Sample frames of a video depicting a woman running on a treadmill. (Bottom left) The 2-D Isomap embedding of this data set where each blue dot corresponds to the an original frame and the red line depicts the original time sequence. (Bottom right) Plot of residual error shows that two dimensions capture most of the information in these local distance measurements.

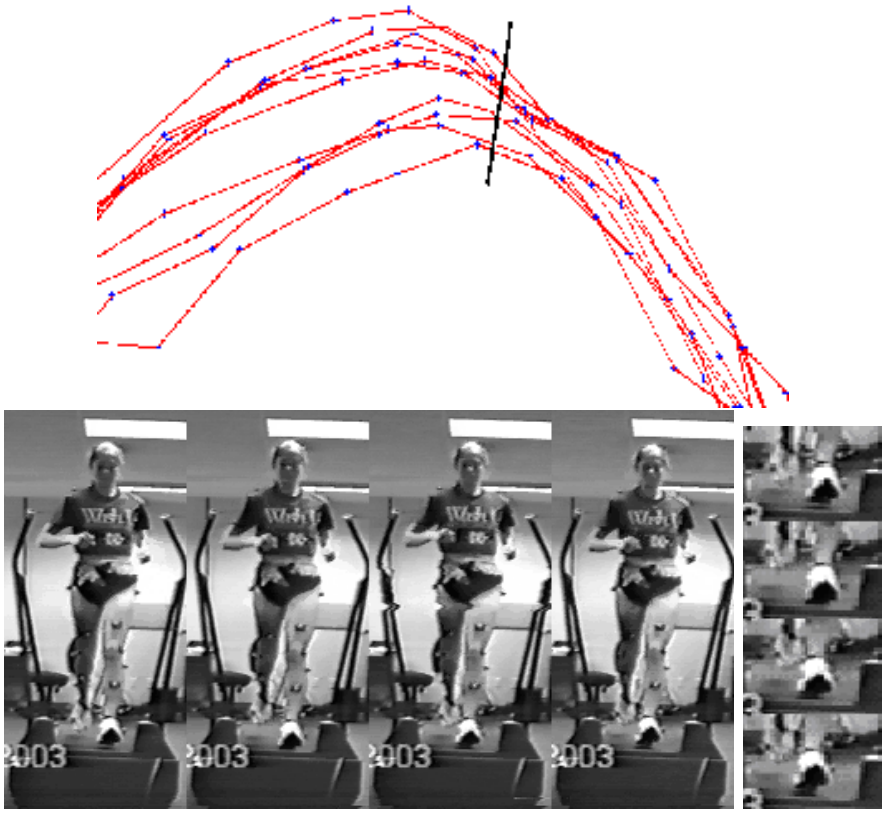


Figure 3.8: An expanded view of the top of the trajectory shown in Figure 3.7. The radial variation show images taken at the same part of the running cycle. The dominant variation here is translation to the left, and can be seen most clearly in the enlarged view of the feet shown at the right.

shows an expanded view of the Isomap embedding, and the image set generated by moving radially through the coefficient space. In this case, with additional *post hoc* analysis, Isomap decomposes the data set into its main types of variation. However, this leads us to ask a number of questions.

- Is it possible to automatically obtain readily interpretable low-dimensional embeddings?
- What different types of image variation are likely to arise in natural images?
- Can manifold learning be improved in order to relate the structure of these embeddings to type of image variation?

In order to address these questions, we examine one of the fundamental operations of the manifold learning algorithms described in Chapter 2. Defining the local neighborhood, or finding neighboring points, of each point in the original input space is a key (implicit or explicit) step of each of these algorithms. In the remainder of this chapter, we will consider how modifying this operation can address some of the current limitations of manifold learning algorithms on natural data sets.

In the general case, there has been some related work which considers finding "neighbors" in some feature space rather than the original input space. In [15], Kernel Isomap is presented which modifies the resulting pair-wise similarity matrix to address the topological stability problems addressed in [2]. In [29], the argument presented is that algorithms which use local information to find a global embedding (e.g., Isomap, LLE, etc.) can be described as Kernel PCA [54] with different kernel functions. The work of this dissertation frames this image differencing problem in terms of the changes of the object of interest.

3.3.1 Pattern Theory, Image Variation and Distance Metrics

Pattern Theory builds upon the characterization of shapes defined by Grenander [26] and encodes variations in shapes of natural objects as the results of applying elements of a small group of transformations. The core of the research in Pattern Theory has been to develop tools to define probability distributions over these transformations. However, we notice that in many videos of particular objects, the set of observed transformations is quite limited. For example, a human runner cycles through a particular set of 3D shapes (related to one another through the action of diffeomorphic transformation) as she/he passes in front of a camera (varying the rigid transform relating the 3D object to the camera). When the observed deformations lie on such a low-dimensional manifold, developing distance measures that approximate geodesic distances along the manifold suffices to discover interesting structures.

Deformable template analysis [63, 62] is one instantiation of pattern theory that applies to images. Small deformations in the neighborhood of a particular image $I_a \in \mathcal{R}^D$ can be expressed in terms of three components:

- image motion (rigid and non-rigid),
- photometric changes, and
- noise.

Following a presentation is adapted from [63], these variations can be expressed as:

$$I_{ah}(\vec{p}) = I_a(\vec{p} - hv(\vec{p})) + h\sigma^2 z(\vec{p}) + N(\vec{p}), \tag{3.1}$$

Here, the first term uses a displacement field v to define the spatial motion of image regions (pixel $\vec{p}$ in image I_a provides support for the pixel $\vec{p} + hv(\vec{p})$ in image $I_{ah}(\vec{p})$), the second term uses an additive term z defined at each pixel and an overall scaling factor σ^2 to specify variations in image appearance not accounted for by motion (lighting changes for example), and the third term describes imaging noise, which ought to be independent of the magnitude of the overall transformation h.

In developing our distance measures, we also distinguish between global motion patterns caused by changes in the camera orientation or translations of the object, and local motions caused by the non-rigid object deformations. Our goal for this section, then, is to propose distance measures that approximate geodesic distances along each group of transforms:

Transform Group	Distance Measure
Rigid Motions / Projection Changes	Global Motion Estimates
Non-rigid Motions	Local Motion Estimates
Intensity Variation	Local Contrast Change Estimates
Image Noise	Euclidean Distance

In the cardiac MR example, which motivated much of the work of this dissertation, there are two major but unrelated causes of change in the image set. Therefore, for each of the proposed distance metrics, one desirable property would be invariance to orthogonal changes between images.

More formally, consider an image set $\mathcal{I}$ that is parameterized by two parameters (b, c) (which might stand for breathing and contrast), and suppose that $I(b, c)$ defines the noise-free image that would be generated for a particular part of the breathing

cycle and contrast permeation. We would like to define a distance measures d_1, d_2 such that Isomap applied to $\mathcal{I}$ using d_1 gives a 1-D parameterization such that image $I(b, c)$ is mapped to the real number b, and Isomap applied to $\mathcal{I}$ using d_2 gives a 1-D parameterization such that image $I(b, c)$ is mapped to the real number c. In addition to being as insensitive to noise as possible, this requires:

- $d_1(I(b, c), I(b + \delta, c)) = \delta$
- $d_1(I(b, c), I(b, c + \epsilon)) = 0$
- $d_2(I(b, c), I(b + \delta, c)) = 0$
- $d_2(I(b, c), I(b, c + \epsilon)) = \epsilon$

That is, we require that each distance accurately measure small variations in the deformation parameter it is assigned to capture, and furthermore the distance measure should be invariant to the image changes caused by small changes in the other mode of deformation. Because only nearby points are used in the Isomap procedure, it is not required that the distance measures are globally invariant to the other deformation. Depending upon the application, weaker conditions may suffice, such as requiring the image distance to be monotonic with respect to (rather than equal to) an intrinsic parameterization of the deformation.

The remainder of this section describes the traditional Euclidean distance measure and introduces our novel metrics.

Euclidean Distance Measure

The most common implementations of manifold learning algorithms when used with image data start by computing the Euclidean distance (square root of the sum of the squares of the pixel intensities) between each pair of images. Define $||I_a - I_b||_2$ to be the Euclidean distance between of two images. Does this this distance measure have any concrete interpretation with respect to our deformation models?

If I_a and I_b are separate images of the same object (under the same deformation), then, from Equation 3.1, $v(\vec{p})$, and $z(\vec{p})$ are uniformly zero, and the Euclidean distance between I_a and I_b is:

$$\sum_{\vec{p}} ||I_a(\vec{p}) - I_b(\vec{p})||_2 = \sum_{\vec{p}} N(\vec{p})$$

If this noise is i.i.d, Gaussian and zero-mean, then the Euclidean distance $||I_a - I_b||^2$ is a negative log-likelihood that the two images are of the same object. That is, under this model of image formation, the distance measure commonly used in Isomap is most directly a measure of how unlikely it is that they are the same image, rather than a measure of how different the two images are. The following sections consider different definitions of image distances, so that the image embedding function may be based on local distances more closely tied to magnitude of the image deformations.

Rigid Motion

Some changes to the imaging geometry lead to globally consistent image transformations; rigid translations of an object lead to translations and scale changes of the

image, and changing camera parameters (calibration and zoom) are well approximated by affine image warping. Measuring the magnitude of these changes between two images can be expressed as finding the image warp that makes those images the most similar.

For example, we can express the allowable warping of an image I_b as AI_b, for $A \in T$, where T represents a class of allowable transforms that define a global motion across the image (such as affine warps). Then, the distance measure can be written as the magnitude of the transform that minimizes the image difference:

$$||I_a - I_b||_T = || \arg\min_{A \in T}(I_a - AI_b)||$$

However, manifold learning techniques are most relevant to the understanding of non-rigid motions. To understand non-rigid motion in natural data sets, it is sometimes important to ignore the image distances caused by rigid motion. A rigid motion-invariant distance measure can be written:

$$||I_a - I_b||_{\text{invar(T)}} = \min_{A \in T} ||I_a - AI_b||$$

Non-Rigid Motion

For the case of unknown non-rigid transformations, the generic class of diffeomorphic deformations is a natural choice of transform groups. These deformations may not have a global structure, so we propose to measure the magnitude of the transformation by accumulating measures of local motion over the image.

One implementation of this is to define a distance measure that uses the response of a collection of Gabor filters to estimate local motions. Complex Gabor filters are applied to the same positions in both images, and the phase difference of the complex response is summed over all locations. Given two images I_a, I_b and $G_{(\omega,\{V|H\},\sigma)}$ which is defined to be the 2D complex Gabor filter with frequency ω, oriented either vertically or horizontally, with σ as the variance of the modulating Gaussian, the distance can be expressed as:

$$\begin{aligned} ||I_a - I_b||_M &= \sum_{x,y} \Psi(G_{(\omega,V,\sigma)} \otimes I_a, G_{(\omega,V,\sigma)} \otimes I_b) \\ &\quad + \Psi(G_{(\omega,H,\sigma)} \otimes I_a, G_{(\omega,H,\sigma)} \otimes I_b) \end{aligned}$$

where Ψ returns the absolute value of the phase difference of the pair of complex Gabor responses.

This distance function is dependent upon the choices of ω, and σ. The wavelength of the Gabor filter should be at least twice as large as the image motion caused by small deformations, and σ can be chosen as approximately the wavelength. In practice, this metric is surprisingly robust to the choice of σ.

Because it is based on the phase of the local image structure, this image distance measure is robust to small changes in the local contrast. Furthermore, because the Gabor filters are computed over small regions of the image, the effect of pixel noise is minimized.

Other distance measures are appropriate when the shape of the object is defined by its silhouette and the object can be cleanly segmented from the background. If the segmentation is robust, a distance metric is invariant to *any* changes in illumination

or contrast as long as it relies only on binary valued data. For a pair of images I_a, I_b with point sets falling inside the silhouette P_a, P_b, we can employ the symmetric Hausdorff distance, $h(P_a, P_b)$. Extending this to become an affine invariant distance measure requires an additional minimization step:

$$||I_a - I_b||_A = \min_{A \in T} h(P_a, AP_b),$$

where AP_b is the point set of the second image after deformation by an affine transform A.

Intensity Variation

For image sets derived from an object undergoing intensity changes (e.g., contrast changes, lighting, shading, and fog), we exploit a different function of the Gabor filter bank responses. Given two images I_a, I_b and $G_{(\omega,\{V|H\},\sigma)}$ which is defined earlier, the image distance can be expressed as:

$$\begin{aligned} ||I_a - I_b||_C &= \sum_{x,y} \Big| |G_{(\omega,V,\sigma)} \otimes I_a| - |G_{(\omega,V,\sigma)} \otimes I_b| \Big| \\ &\quad + \Big| |G_{(\omega,H,\sigma)} \otimes I_a| - |G_{(\omega,H,\sigma)} \otimes I_b| \Big| \end{aligned}$$

where $|\cdot|$ returns the magnitude of a complex value.

Small motions of an image region may change the phase a Gabor filter response, but do not significantly affect the magnitude of the filter response, so while this distance measure is closely related to that designed for non-rigid motions, it has the desirable property of being largely invariant to those small motions.

3.3.2 Extending Manifold Learning to Images

In this section, we illustrate the use of the proposed distance metrics from the previous sections on two example application domains: a bird flying against a blue sky towards the camera and the cardiac MRI data set described in Section 3.1.

Rigid and Deformable Motion

We consider a data set of a flying bird captured against a clear sky. This data set exhibits two important properties. First, the clear sky background allows very simple and robust segmentation of the bird. Second, there exists an obvious dominant motion – the wings flapping. The wing flapping is a non-rigid deformation that is complicated to parameterize without an explicit bird dynamics model. Furthermore, the bird is flying past the camera, so the rigid transformation relating the bird and camera position is continuously changing. Therefore, the variability in this data set is a combination of rigid and non-rigid motions. These properties of the input data set suggest that using the previously described Hausdorff distance measure may elucidate relevant structures within the Isomap embedding.

Isomap is performed on this data set using the symmetric Hausdorff distance and $k = 8$ neighbors. This gives the embedding shown in Figure 3.9. There is a circular motion in the trajectory caused by the cyclic nature of the data. However, there is also a larger scale consistent radial motion, caused by image differences that arise from the approach of the bird toward the camera. Thus the Isomap embedding automatically de-couples the cyclical, non-rigid component of the bird motion from the rigid component of the bird approaching the camera. To highlight this effect, the right

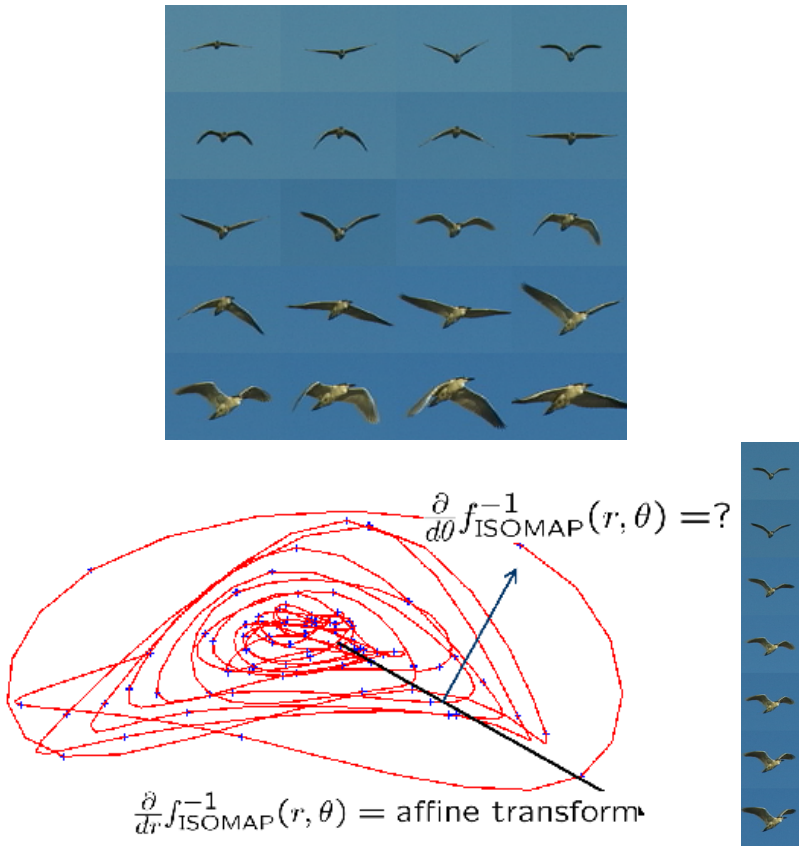


Figure 3.9: (Top) Sample images of a video sequence of a bird flying across the sky. (Bottom left) The Isomap embedding of this set of images. Moving radially in the embedding corresponds, locally, to an affine transformation of the image that depends only on the relative position of the bird to the camera. The transform required to move tangentially in the Isomap space varies by location and requires a motion model of the bird. (Bottom right) Images closest to the dark radial arrow.

side of Figure 3.9 shows the images closest to a radial line in the Isomap embedding. The images nearest this line are approximately related by a rigid transformation.

In order to emphasize the deformable motion of the bird, we desire the distance function to ignore, as much as possible, variation caused by anything other than the deformable motion. Small rigid transformations of an object lead to locally affine distortions of the image, so here we consider the affine-invariant Hausdorff distance measure.

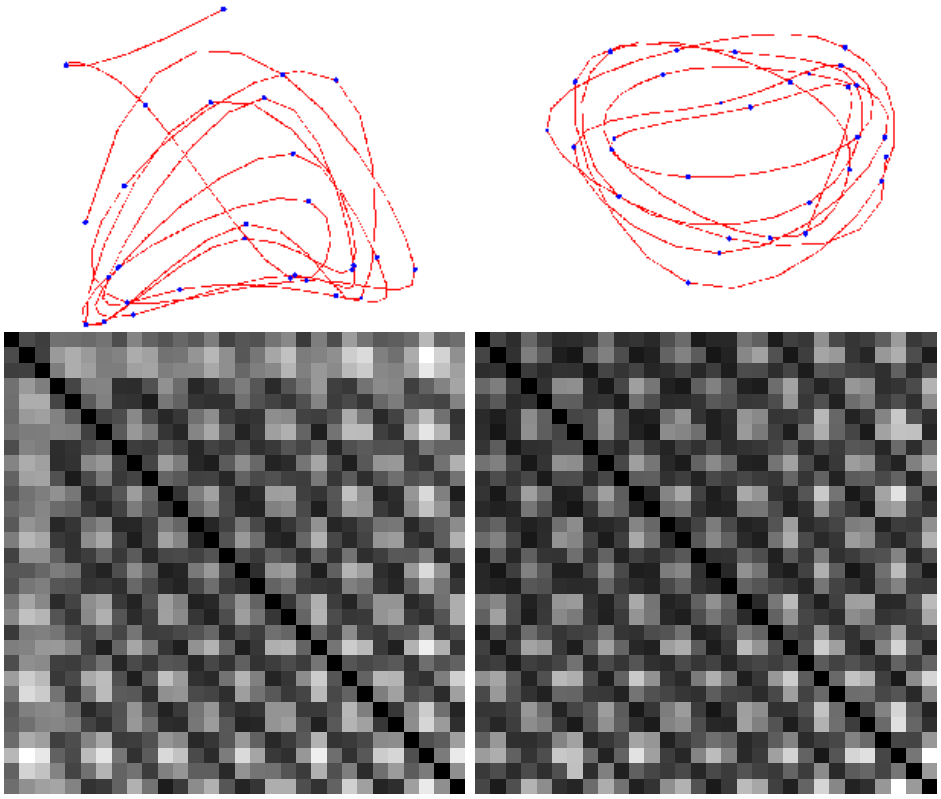


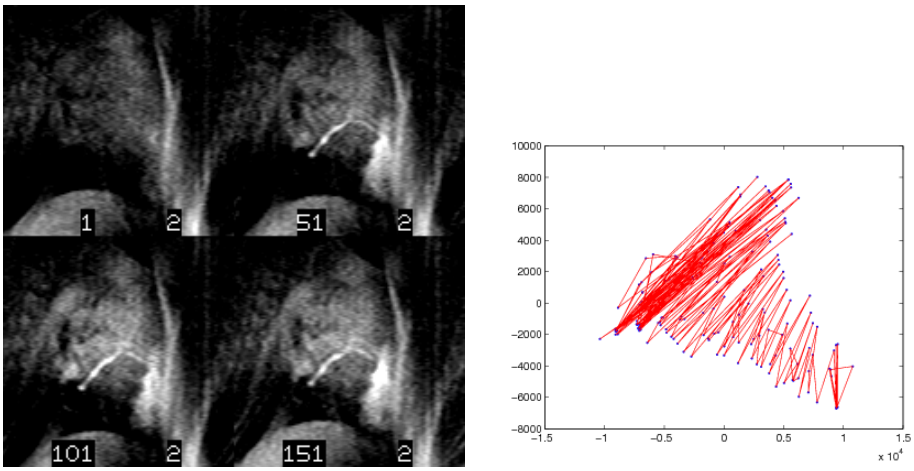
Figure 3.10: (Top left) Isomap embedding of the bird image sequence where each image is a dot and the line connects the images in order. (Bottom left) the complete distance matrix defined by the Euclidean distance metric. (Right) The plots for the same sequence using the affine-invariant distance measure.

Figure 3.10 shows the result of using this affine-invariant distance measure. This defines a more clearly cyclic mapping of the bird images, emphasizing the variation in the non-rigid deformation of the bird shape by minimizing the image distance due to rigid variation. In addition, the lower (darker) values along the diagonals in the distance matrix using the novel distance metric show a periodicity which reflects the regularity of the flapping wings of the bird. Finally, the solution for the best-fitting affine matrix A between two images offers an image warping operator for interpolating between images. This could form the basis for better "out-of-sample" inverse projections, and could be used to create more realistic image interpolation.

Deformable Motion and Contrast Changes

In this section, we revisit the cardiac MR data set described at the end of Section 3.1. This image set contains real-time cardiac MR images, captured during a 60 ms window during the systolic part of consecutive heartbeats. The data set includes 180 such images from the same patient. The variation in these images has three causes. First, between images there is variation in the position of the heart (and liver, which is visible at the bottom of the images) due to the compression of the chest cavity during breathing. Second, a contrast agent is slowly permeating through the tissues. Third, the MRI images are noisy. The direct application of Isomap (using the sum of the squared pixel intensity differences and $k = 8$ neighbors) to this image set is shown in Figure 3.11.

Distance functions that measure variation due to breathing motion and are invariant to the contrast changes or vice-versa — instead of the Euclidean image distance which varies due to both effects — give distance measures that are more isometric to the



(a) Sample Frames (b) Direct Isomap Embedding

Figure 3.11: Four samples of a sequence of MR images, and the associated Isomap embedding (using $k = 8$ neighbors.)

underlying manifold parameters. We use the pair of functions based on Gabor filter responses previously proposed. Embedding the images in one dimension using Isomap with these distance functions gives each image two coordinates. These coordinates are plotted at the right of Figure 3.12. The y-axis corresponds to the embedding based upon the Gabor filter phase difference (which measures local motions, but is largely invariant to contrast changes), and correlates to the different deformations of the chest cavity. The x-axis variation is based upon the Gabor filter magnitude change (which measures local contrast changes, but is largely invariant to small motions). Because the contrast change is due to a contrast agent permeating through the tissue, this is related to the original ordering of the data.

In Section 3.1, we discussed the major flaws of the Isomap embedding, specifically that the coordinates did not correspond with the major causes of image change and the "scaling effect" which is a side effect of using the Euclidean distance. The embedding shown in Figure 3.12 using our novel distance metrics correct both of these problems.

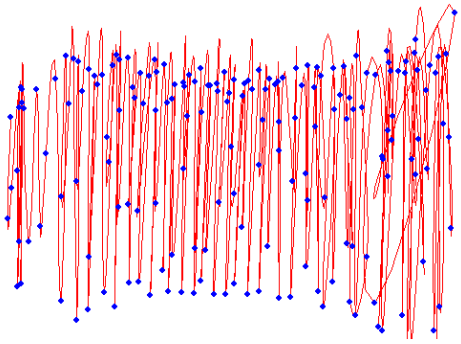


Figure 3.12: Isomap embedding using a contrast invariant distance measure based on local Gabor phase. Note how the embedding aligns itself with two concrete degrees of freedom.

3.4 Summary

Techniques such as Isomap and LLE are important tools in processing large video and image collections. These general statistical tools need to be specialized in order to take advantage of properties of natural images and deformations because interesting image data sets are almost never linear combinations of other images, as is the underlying assumption when using PCA. A small set of image transformation primitives gives powerful tools for registration of many different kinds of data sets.

Pattern Theory provides a framework for defining relevant image distance metrics. Specifically, for natural image data sets a small number of distance metrics are useful for most of the important applications. This research focused on the image transformations specified by the Pattern Theory framework, namely imaging noise, lighting changes, rigid motion and non-rigid motion and devised a set of distance metrics which correspond to each of the major modes of image transformation. These distance metrics are easy to compute and invariant to the other transform groups. They are also quite general. That is, it is not necessary to have a strong model of the

exact transform (i.e., it is not necessary to have an explicit model of the stride of the woman on the treadmill), rather it is only necessary to know the type of transform (e.g. diffeomorphic deformation) in order to apply the distance metric. In Chapter 4, we show some applications which exploit these natural parameterizations.

Chapter 4

Applications of Image Manifold Learning

In Chapter 2 we introduced manifold learning and described a number of algorithms. In Chapter 3, we showed how these algorithms generally fail in the case of natural image sets, but can be extended to provide natural parameterizations. In this chapter, we describe how these natural parameterizations can be used to improve important vision tasks such as learning motion models, interpolation, noise reduction, and segmentation. We generally employ Isomap as the manifold learning algorithm of choice. However, in most instances, our methods can be generalized to the other algorithms described in Chapter 2.

4.1 Learning Motion Models

Understanding non-rigid deformations remains a challenging problem in the analysis of images. Here, we consider a special case of this problem: given many images of

an object undergoing an unknown (but small) set of deformations, characterize the deformations and give each image parameters which describe how they have been deformed.

This problem is quite common in the medical imaging community, and one important application is cardiac MR imagery. Complete low-resolution images can be captured in modern MRI machines in about 60 ms. MR imagery taken of the same patient varies primarily due to a small number of causes such as, deformation of the heart during the heartbeat, deformation of the heart due to breathing, and (potentially) contrast agents permeating slowly through the tissues.

Current common practice for diagnostic cardiac MRI is to isolate the effects of the heartbeat by triggering the MR image capture at the same part of the heartbeat cycle and to isolate the effects of breathing by asking the patient to hold their breath. This leaves images which vary only because of the permeating contrast agent. For this application, we provide an image-based method to improve the quality of images captured during this procedure in the presence of these undesired motions by learning the model of image deformation. In the future, this could alleviate the necessity that the patient hold their breath and would allow certain cardiac MRI procedures to be performed on unconscious patients who are unable to hold their breath.

A gated-cardiac image set consist of images taken at the same part of the heartbeat cycle. These images, for any specific patient, fall near (but because of image noise, not exactly on) a 2-D manifold within the space of images. This manifold has natural parameters, the breathing cycle and the permeation of the contrast agent. Our

goal can be phrased as learning a mapping between the image set and these natural parameters, and then learning a model of the image deformation caused by breathing.

Because these image sets can be parameterized by a small number of parameters, reasoning about such image sets is an ideal application of dimensionality reduction. For this application, we demonstrate the use of manifold learning with the intelligent selection of distance functions to automatically parse an image set and parametrize each image by its magnitude of variation due to different causes. After the images are so parameterized, surprisingly simple and naive approaches suffice to capture complex non-rigid deformations. Furthermore, the mapping of the parameterization onto a specific image deformation defines natural interpolation and projection models, lacking in classical manifold learning algorithms.

4.1.1 Selecting Image Distance Metrics

In Chapter 3, we demonstrated the value of choosing appropriate distance metrics when using Isomap to parameterize image data. The axes used to define the coordinates in a traditional parameterizations are difficult to interpret, so instead we seek to find a pair of distance measures matched to the causes of the image deformation such that the parameterizations using each distance measure correlates with only one of the causes of the deformation.

For the example case of a gated cardiac MRI data in this application, there are two causes of motion, that manifest themselves locally as image motion and contrast change. We will use the Gabor-filter based distance metrics we previously described in Chapter 3.

Local Image Deformation Distance

1. Given images I_1, I_2
2. Define $G_{(\omega,\{V|H\},\sigma)}$ to be the 2D complex Gabor filter with frequency ω, oriented either vertically or horizontally, with σ as the variance of the modulating Gaussian.
3. $$\begin{aligned} D_{(\omega,\sigma)} &= \sum_{x,y} \Psi(G_{(\omega,V,\sigma)} \otimes I_1, G_{(\omega,V,\sigma)} \otimes I_2) \\ &\quad + \Psi(G_{(\omega,H,\sigma)} \otimes I_1, G_{(\omega,H,\sigma)} \otimes I_2), \end{aligned}$$

   where Ψ returns the phase difference of the pair of complex Gabor responses.

Local Image Contrast Distance

1. Given images I_1, I_2
2. Define $G_{(\omega,\{V|H\},\sigma)}$ to be the 2D complex Gabor filter with frequency ω, oriented either vertically or horizontally, with σ as the variance of the modulating Gaussian.
3. $$\begin{aligned} D_{(\omega,\sigma)} &= \sum_{x,y} \big| |G_{(\omega,V,\sigma)} \otimes I_1| - |G_{(\omega,V,\sigma)} \otimes I_2| \big| \\ &\quad + \big| |G_{(\omega,H,\sigma)} \otimes I_1| - |G_{(\omega,H,\sigma)} \otimes I_2| \big|, \end{aligned}$$

   where $|\cdot|$ returns the magnitude of a complex value.

We now discuss methods to use the Isomap parameterization of the image set, and in particular give methods for the analysis of a data set including image which have undergone an unknown spatial deformation. The advantage given by Isomap is that the magnitude of this deformation is known, and the images can be re-ordered by their deformation.

These results derive from a gated cardiac MRI study, with 180 images taken from an unknown part of the patient's breathing cycle. Using these distance functions effectively gives a 1-D parameterization of the image set, with the free parameter corresponding monotonically with the breathing cycle.

4.1.2 Extracting Deformation Groups

For the class of image sets generated by multiple examples of an object undergoing a non-rigid transform, we address the principal shortcoming of manifold learning algorithms, namely the inability to extract meaning for the low-dimensional coordinates and perform an inverse projection from a point not in the original set to a new point on the image manifold. By using an appropriate distance measure, as described earlier, the images are sorted relative to their major deformation. In order to solve for the parameters of this deformation, our method takes the following steps:

- Select an appropriate distance measure
- Use Isomap to find an ordering for the images
- Find point correspondences between images
- Extend point correspondences into image warps

Point Tracking

The main benefit to sorting the points relative to the deformation instead of using unsorted images is that point tracking is simplified. In general, point tracking is easier if the putative corresponding points are closer together. Naive methods such as iterative closest point matching [6] can effectively track points through hundreds of frames. Here, we use a simple feature tracker [40] which makes an initial guess of point correspondences and uses RANSAC [25] to improve the solution.

Thin Plate Splines

A thin-plate spline [8] is a two-dimensional interpolation function whose name refers to a physical analogy involving the bending of a thin sheet of metal. Given an arbitrary set of points in $\mathcal{R}^2$ and some function $f(x, y)$ evaluated at those points, the thin plate minimizes what is known as the "bending energy" function:

$$\int\int_{\mathcal{R}^2} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dxdy$$

Thin-plate splines have been used frequently in image analysis. This construct has been used with velocity encoded MR images [53], to calculate cardiac strain from MR images [27], and analyzing bone structure on radio-graphs [14].

The pervasiveness of this construct in the image analysis domain indicates that thin-plate splines provide a natural way to move from point correspondences to entire image warps. Let $P_t(i)$ be the coordinate of the i-th tracked point in frame t. The

thin plate spline warping function is the function f that minimizes the bending energy above and simultaneously maps all points $P_1(i)$ in the first frame exactly onto their corresponding points $P_t(i)$ in the t-th frame. That is,

$$\forall_i f(p_{1,i}) = p_{t,i},$$

and for all image points (x, y) that were not tracked in the first image, the function f maps them in such a was that the overall mapping minimizing the distortion measured by the bending energy. Figure 4.1 shows a representation of a thin-plate spline capturing the image deformation for two cardiac MR images. Using the image distance measure described above, the images represented in Figures 4.1a and b had a high inter-image distance. The thin-plate spline overlaid on these images represents the deformation of one image to the other. Figure 4.1c shows the result of transforming the image in 2b to that in 2a.

4.1.3 Evaluation

In a test to demonstrate the effectiveness of this method, we applied our method to the entire set of 180 MR images. To demonstrate that the motion model was learned for each image, we solved for the image deformation of each image relative to the first image in dataset. Figure 4.2 shows the results of capturing the unknown deformation with and without using the Isomap sorting step. In this experiment, point correspondences and image warps (to a reference frame) were applied to the set using the original ordering and the ordering obtained by Isomap sorting. *Mutual Information* was used to calculate how similar the warped frames were to the reference

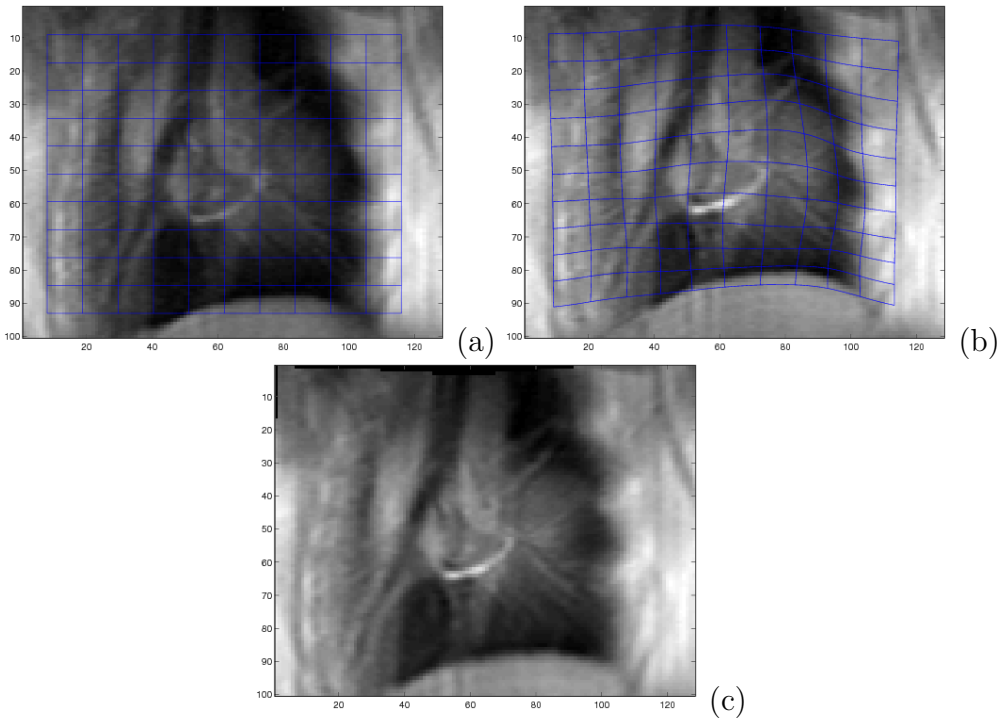


Figure 4.1: Using thin-plate splines to model deformations.

frame. Mutual Information [17, 68, 70] is a widely-used metric in medical image registration [46].

The results in Figure 4.2 demonstrate that by using our framework, we are able to accurately learn the motion model and, thus, more easily register all of the images. In this application, we illustrated the use of Isomap as a preprocessing tool for the analysis of deformable image sets where the deformation is unknown. By using our novel image distance metrics, it became possible to separate the effects of different causes of image variation. This allows the images to be parameterized by their unknown deformation, which facilitates modeling the deformation.

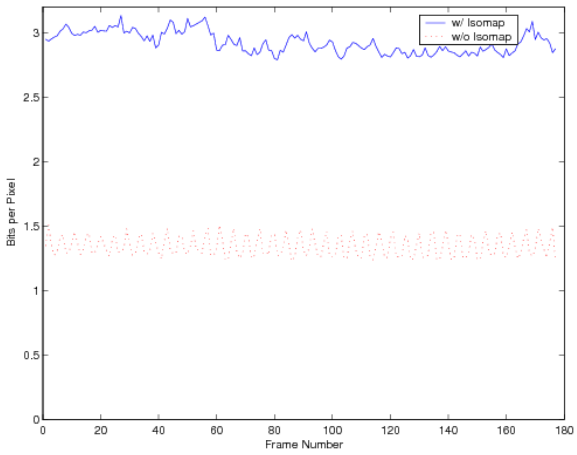


Figure 4.2: Mutual information (in bits) of frames in a video sequence warped to a reference image with and without the use of the Isomap sorting step.

4.2 Image Interpolation on a Nonlinear Manifold

Typical biomedical image sets, such as the sample heart MR frames shown in Figure 4.3, suffer from poor resolution both spatially and temporally. Improvements in the imaging apparatus and protocols have helped to mitigate these problems, but current diagnostic biomedical images and videos are still of low quality. In this application, we describe a method for biomedical image deformation analysis which incorporates all of the video data simultaneously in order to facilitate inter-image interpolation and de-noising. This method exploits the common condition that a set of images has an low-dimensional manifold structure.

The primary hurdle in this problem concerns the key limitation to manifold analysis of images. Namely, the lack of projection functions from the low-dimensional embedding space back to image space for samples not in the original training set. There has been some work on this problem, in the general case, where linear interpolation of nearby images is performed using generalized radial basis functions [24]. However, these

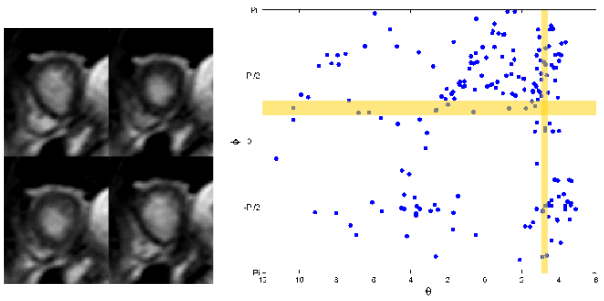


Figure 4.3: Sample frames from a heart MR sequence and the 2D Isomap embedding from this sequence. Images with similar y-coordinates in the embedding were captured near the same part of the heartbeat cycle; similar x-coordinates means the images were captured near the same part of the breathing cycle. In this application, we characterize image deformations in terms of these manifold coordinates.

function approximation methods do not generalize to approximating image manifolds where the major cause of image change is non-rigid deformation. By employing our framework of embedding using novel distance metrics, we integrate explicit models of image deformation with manifold representations of image data sets.

4.2.1 Manifold of Deformation Fields

We seek to build a parametric representation of the deformation fields which characterize the image variation on the manifold. In this section, we describe the free-form deformation (FFD) [55] model, then illustrate how to describe an image manifold using a parameterized model of image deformations. Finally we use this model to reconstruct images corresponding to arbitrary points on the manifold.

Deformation Model

Using the FFD model, the resulting deformations can be written as the 2D tensor product of standard one-dimensional cubic B-splines:

$$f(x,y) = \sum_{m=0}^{3}\sum_{n=0}^{3} B_m(\tilde{x})B_n(\tilde{y})\Phi_{m+i,n+j} \tag{4.1}$$

where Φ denotes a $n_x \times n_y$ lattice of control points which parameterize the FFD, i, j denote the indices of the control points, and $\tilde{x}, \tilde{y}$ represent the relative positions of x, y in lattice coordinates (i.e. relative to the B-spline control point grid).

Commonly, when optimizing the FFD deformation to fit noisy image data, it is important to impose a smoothness term to prevent artifacts such as folding. A common smoothing term is:

$$\mathcal{S}[f] = \int \left[\left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial xy}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 \right] dx\ dy \tag{4.2}$$

Since the global transformation f_{global} can still be represented by rigid or affine transformation of the control lattice of FFD, we simply use FFD to model the transformation f between two images.

Integrating Manifold Constraints

In this section, we propose a method to solve for deformation fields in all images of a data set simultaneously. This offers a powerful new constraint by penalizing variation in the FFD relative to nearby images in the manifold.

We illustrate the algorithm on an image set which varies due to two main causes. By using Isomap and our motion-specific image distance metrics, we automatically parameterize all images and obtain u and v, the 2D image manifold coordinates. The deformation fields on the manifold are also a function of u and v, and our goal is to describe these deformation fields explicitly by a set of FFD control points. Figure 4.3 shows example images and the Isomap embedding of a 100 frame cardiopulmonary image data set, for which the image changes are caused by the heart beating and breathing of the patient.

Given such an image data set $\mathcal{I} = \{\mathrm{I}_1, \mathrm{I}_2, \cdots, \mathrm{I}_N\}$ the transformation between any image I_i and the reference image I_{ref} is denoted by a transformation f_i. This transformation is the FFD transformation described in the last section and defined by the motion of $n_x \times n_y$ lattice of control points: $\Phi_i = \{\phi_{i,1},\ \phi_{i,2},\ \cdots,\ \phi_{i,n_x n_y}\}$. The set Φ contains all of the control points in all images, where $\phi_{i,j}$ is the j-th control point in the i-th image. However, image i is associated with manifold coordinates (u_i, v_i), so we can express these control points as a function of the manifold coordinates (u_i, v_i). We do this with another FFD to express the variation of the j-th control point as a function of the manifold coordinates (u, v). We parameterize this FFD with variables Θ_j, and there is one such FFD for each control point used to define the image warps.

Summarizing, we express the transformation of image i as a function $\mathcal{F}(u_i, v_i)$. In order to express a transformation, $\mathcal{F}$ describes the process of creating image warping FFD control points as $\{\Theta_1(u_i, v_i), \Theta_2(u_i, v_i), \ldots \Theta_{n_x n_y}(u_i, v_i)\}$. These control points define the warp for image i. The only free variables in this system are the parameters of the mapping Θ_j between manifold coordinates and the control point positions. Thus, to solve for the deformation fields, we minimize the following joint energy

functional, over the set $\mathcal{I}$ of all N images, with respect to Θ (which affects the $\mathcal{F}$ term):

$$\mathbf{E}[\mathcal{I}, \mathrm{I}_{ref}; \mathcal{F}] = \sum_{i=1}^{N} \mathcal{D}[\mathrm{I}_i^{\mathcal{F}(u_i,v_i)}, \mathrm{I}_{ref}] + \lambda \mathcal{S}[\mathcal{F}(u_i, v_i)] \tag{4.3}$$

where $\mathcal{D}$ measures the error between image I warped according to the manifold coordinates and the reference image (using SSD, correlation or mutual information), and the second term corresponds to the regularizer defined as:

$$\begin{aligned}\mathcal{S}[\mathcal{F}] = \sum_{i=1}^{N} \int [& \left(\frac{\partial^2 \mathcal{F}(u_i, v_i)}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 \mathcal{F}(u_i, v_i)}{\partial xy} \right)^2 \\ & + \left(\frac{\partial^2 \mathcal{F}(u_i, v_i)}{\partial y^2} \right)^2] dxdy + \gamma \sum_{j=1}^{n_x n_y} \int [\left(\frac{\partial^2 \mathcal{F}_j}{\partial u^2} \right)^2 \\ & + 2 \left(\frac{\partial^2 \mathcal{F}_j}{\partial uv} \right)^2 + \left(\frac{\partial^2 \mathcal{F}_j}{\partial v^2} \right)^2] dudv \end{aligned} \tag{4.4}$$

where γ is a weighting parameter. The regularizer (the corollary to Equation 4.2 in the single image case) ensures the smoothness of the deformation fields. The first term constrains the transformation between images to be smooth, while the second term penalizes the large local variations of control points Φ on manifold space.

The optimal deformation fields are found using gradient descent minimization of Equation 4.3 with respect to Θ. To avoid the high computation cost associated with the transformation complexity required to capture the deformation fields, one can use a multi-resolution approach [18] in which the resolution of the control points mesh Φ increases along with the image resolution, in a coarse-to-fine manner.

Figure 4.4: An artificial data set constructed by composing two deformations (illustrated on the left), a non-rigid variation and a rigid translation. Eight sample frames are shown on the right

4.2.2 Interpolation on an Image Manifold

Once manifold deformation fields $\mathcal{F}$ are obtained, we can calculate the transformation of any image on the manifold with respect to the reference image. Given two images I_p and I_q, as well as their associated transformation functions f_p and f_q respectively, we can furthermore approximate the transformation $f_{p,q}$ between these two images by computing the thin-plate spline [8] on image point position correspondences $\{f_p(x,y) \leftrightarrow f_q(x,y)\}$.

Now, given a query point p on the image manifold, we want to approximate its associated image I_p. The approximation $\hat{\mathrm{I}}_p$ can be obtained, in a straightforward manner, by transforming the image I_i closest on the manifold to position p, that is $\hat{\mathrm{I}}_p = \mathrm{I}_i^{f_{i,p}}$. To reduce the bias introduced by transforming a single image, one could consider $\hat{\mathrm{I}}_p$ as a weighted sum of images transformed from neighbors by using natural neighbor interpolation [56]

$$\hat{\mathrm{I}}_p = \sum \omega_i(p)\ \mathrm{I}_i^{f_{i,p}} \tag{4.5}$$

where $\omega_i(p)$ is the natural neighbor coordinate of image I_p with respect to the image I_i.

4.2.3 Evaluation

We examine our approach on artificially generated data, for which we control the deformation parameters. We also tested this method on a noisy cardiopulmonary MRI sequence. All of the experiments were initialized with the same set of parameters. The computation of deformation fields is performed in a multi-resolution fashion by successive refinement of the FFD $\mathcal{F}(u_i, v_i)$ using control points resolutions of 4×4, 7×7 and finally 13×13. The resolution of the FFD $\mathcal{F}_j$ controlling image control points is kept as 5×5.

We generated an artificial 100 frame data set by defining a shape and deforming it using a non-rigid deformation and a rigid translation. Thus, this data set has a 2D manifold structure, indexed by the magnitude of each deformation. The two deformations and eight sample frames of this data set are depicted in Figure 4.4. After learning the image manifold coordinates using Isomap, the images on convex hull on the manifold were chosen as the training image set. The goal was to reconstruct the remaining images from the data set.

To assess the quality of the interpolation results, we calculated the mean and variance of the SSD between the ground truth and the reconstructed images:

$$SSD_i = \frac{1}{n}\sqrt{\sum (\mathrm{I}_{ref} - \mathrm{I}^{f_i})^2} \tag{4.6}$$

where n is the number of pixels on the image. Another measure used to assess the quality of interpolation results, especially in medical image analysis, is known as the normalized mutual information (NMI) [59]. This represents the amount of *overlay* between two images.

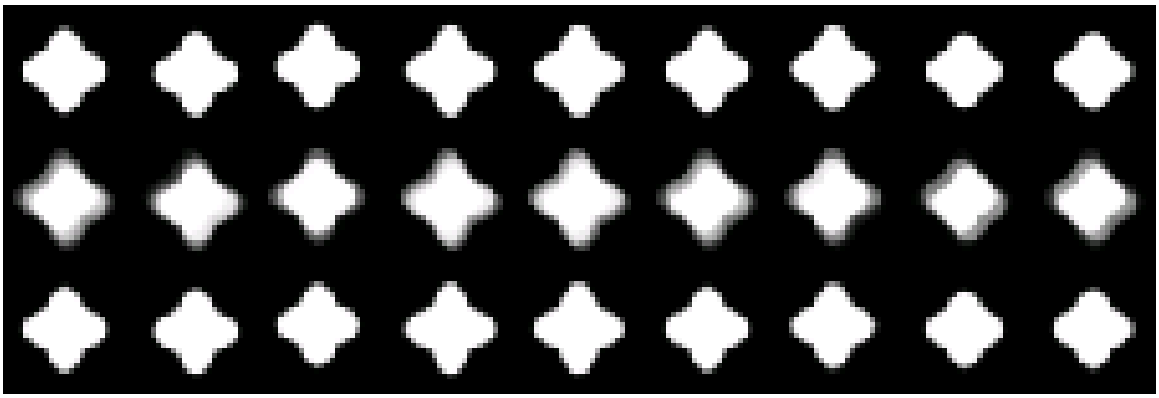


Figure 4.5: Results on the artificial "star" data set. (1st row) Ground truth results. (2nd row) Results by directly interpolating between natural neighbors. (3rd row) Results using our method.

Table 4.1: Comparison for artificial "star" data set experiment between natural neighbor interpolation and our method. Our method improves both the SSD error measure, and the NMI similarity measure. Each cell shows the mean and variance over all of the images.

Method	SSD	NMI
Direct	0.90 ± 0.42	1.80 ± 0.048
Our Method	0.37 ± 0.12	1.91 ± 0.013

The large image changes make this a challenging data set for standard image interpolation techniques. The top row of Figure 4.5 shows the ground truth results for this synthetic data set. The middle row gives results for direct natural neighbor interpolation on the image manifold. We show the results of using our method in the third row. The SSD and NMI measures shown in Tables 4.1 show the increased performance of our method versus the direct method.

We applied our method to the heart MR sequence depicted in Figure 4.3. Figure 4.6 shows examples of two sets of deformation fields. The top half shows three deformation fields relating four images along a vertical strip of the cardiopulmonary image manifold which roughly corresponds to variations in the heartbeat cycle, but not the breathing cycle. The bottom half shows images and deformation fields corresponding

to a images that are translated due to breathing. Finally, Figure 4.7 shows the results of interpolating an image for the manifold coordinates corresponding to a given image, illustrating its de-noising properties.

This application integrates tools for manifold learning with standard models of image deformation. For important image sets such as cardiopulmonary MR data, using this manifold structure regularizes the solution to the deformation fields relating all images. Better deformation fields offer diagnostic value in measuring heart volume and dynamics and support image de-noising. Furthermore, for the case of image manifolds which vary due to image deformation, this offers the ability to reconstruct images for arbitrary manifold positions, lifting an important limitation of nonlinear manifold learning algorithms.

4.3 Image De-noising

In this section, we describe how using meaningful embeddings improves image de-noising. The image set we consider is a cardiac-gated MRI sequence of a heart. To acquire these images, the MRI pulses are triggered at the same point in consecutive heart beats until enough pulses are captured to reconstruct an image. Each image is created in this way, and the data set includes 180 such images from the same patient. The variation in these images has three causes: variation in the position of the heart and liver due to breathing, lighting changes due to a contrast agent slowly permeating through the tissue, and noise.

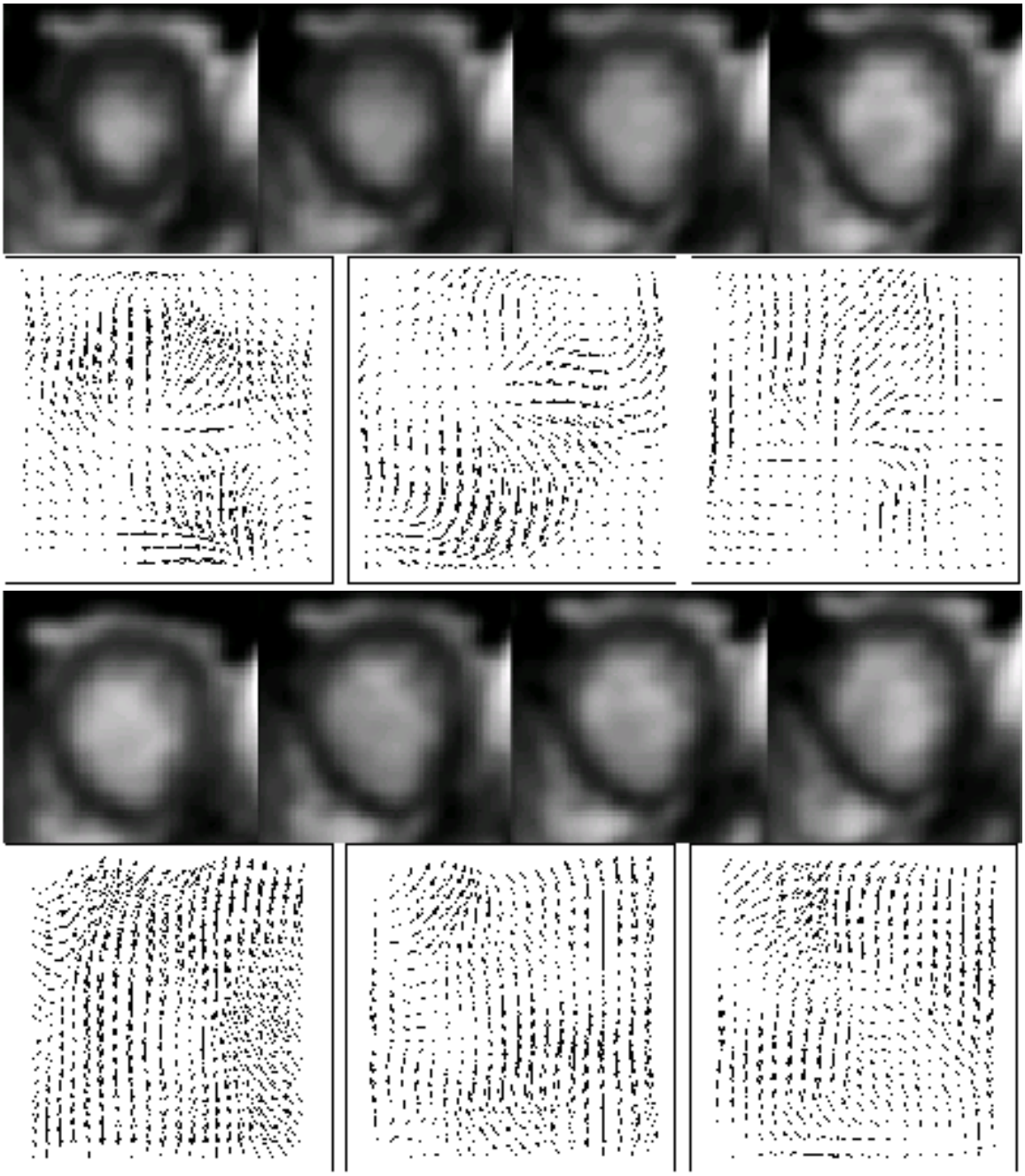


Figure 4.6: Two examples of a series of images, and the deformation fields computed using the manifold constraints. The top row shows the expansion during a heart beat; the bottom row shows the dominant translation due to breathing.

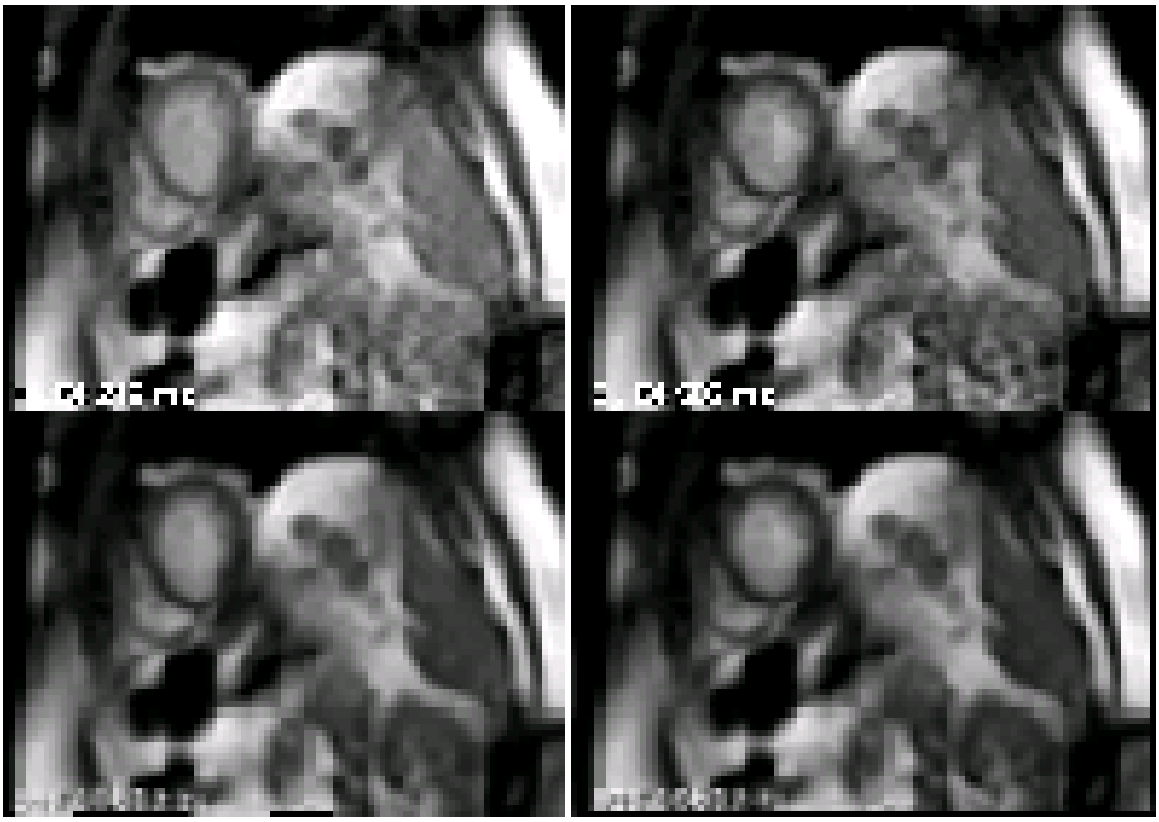


Figure 4.7: An example of de-noising using learned deformation fields on the image manifold. On the top row are the original images and the bottom row shows the de-noising possible by correctly warping the nearby images.

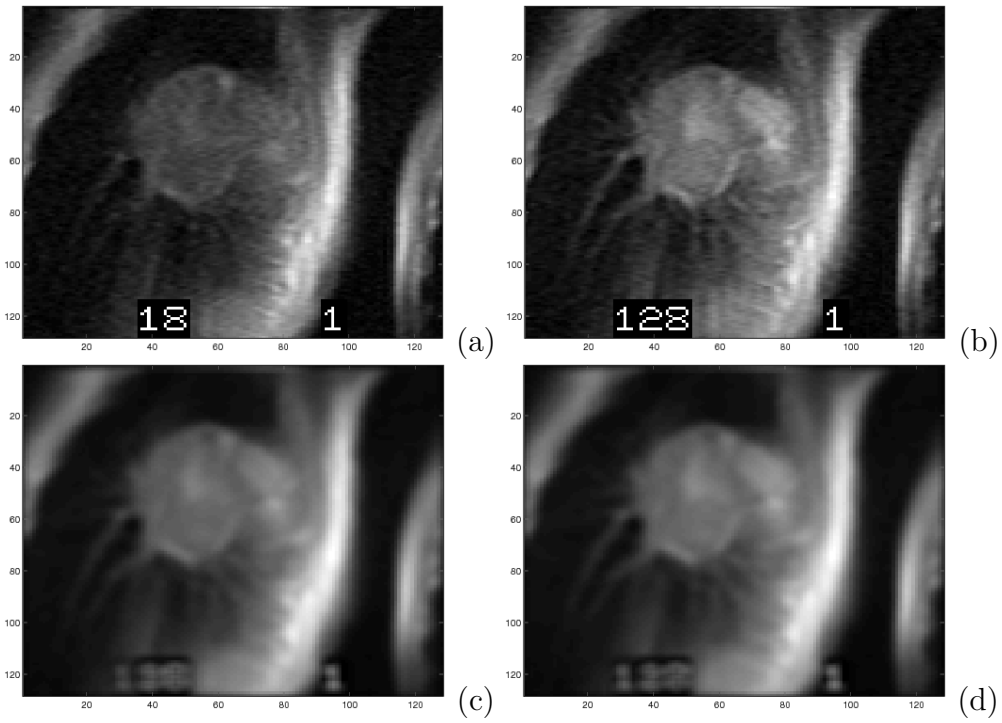


Figure 4.8: Reducing MR image noise by blurring with neighboring (in Isomap order) images. Images (a) and (b) are (originally) temporally distant and differ due to the effect of a contrast agent. Images (c) and (d) are the result of applying a blur using nearby (in Isomap order) images. Minimal motion blur is introduced using this method.

It is common practice to blur consecutive images in a video to remove image noise. However, due to the significant motion between temporally adjacent frames, this introduces motion blur, which is visually unappealing and partially defeats the purpose of de-noising the images. However, by reordering the images based on their ordering on the image manifold, the motion in between images is generally small enough that motion blur is not introduced. Figure 4.8 shows two consecutive images (after reordering using Isomap) and the result of blurring. In the original ordering, these are not proximal frames. In fact, due to the effects of the contrast agent used in the procedure, there is a difference in the average intensity of each image, which can clearly be seen in Figure 4.8b where the vasculature is more pronounced.

Figure 4.9 depicts another example of image de-noising using neighbors along the image manifold. The images whose projections onto the y-axis are similar are taken at essentially the same part of the breathing cycle. A video sequence that plays the original images in the order they appear when projected onto the y-axis shows a very slow deformation, because the frames are ordered by what part of the breathing cycle they capture. Taking a window of 10 consecutive frames within this movie (all of which have similar y-axis projections) gives 10 images of the heart at the same part of the breathing cycle. These can be averaged together to de-noise the image without introducing motion blur. One frame of this sliding window average is shown at the bottom of figure 4.9.

4.4 Image Segmentation

By using the transformation-specific distance metrics, it is possible to obtain embeddings which can be the basis for improved segmentation of noisy cardiac MR data.

In some recent work, algorithms were developed to incorporate image manifold constraints in image segmentation using both active contours [75] and level sets [76]. These methods simultaneously segment every image from data sets which lie on low-dimensional image manifolds. These methods rely heavily the meaningful low-dimensional embeddings described in the previous chapter.

For this test, an artificial data set was constructed by defining a shape and deforming it through a composition of a non-rigid deformation and a rigid translation along diagonal direction. Thus, this data set has a 2D manifold structure indexed by the

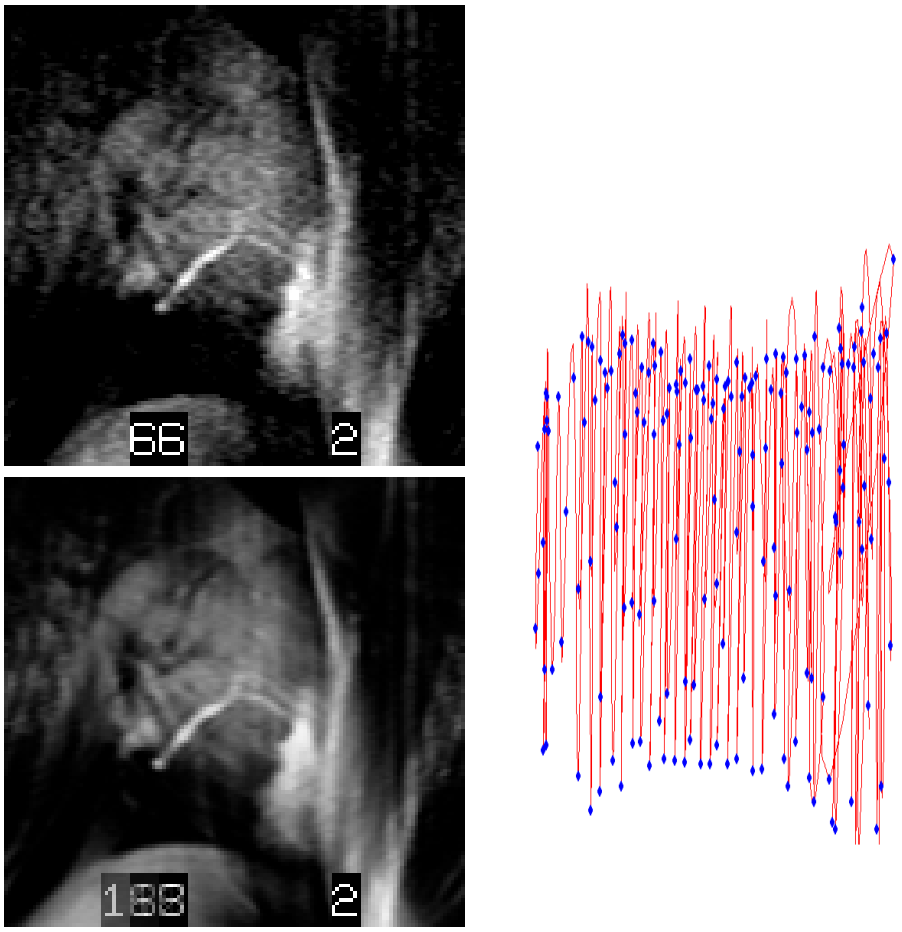


Figure 4.9: (Right) Isomap embedding using the Gabor-based metric. (Top left) A sample image from the cardiac MR image set. (Bottom left) The result of averaging 10 images with similar y-component values. Since the y component encodes motion of the object in the image, averaging images with similar y-components does not result in spatial blurring, but does minimize pixel noise in individual images.

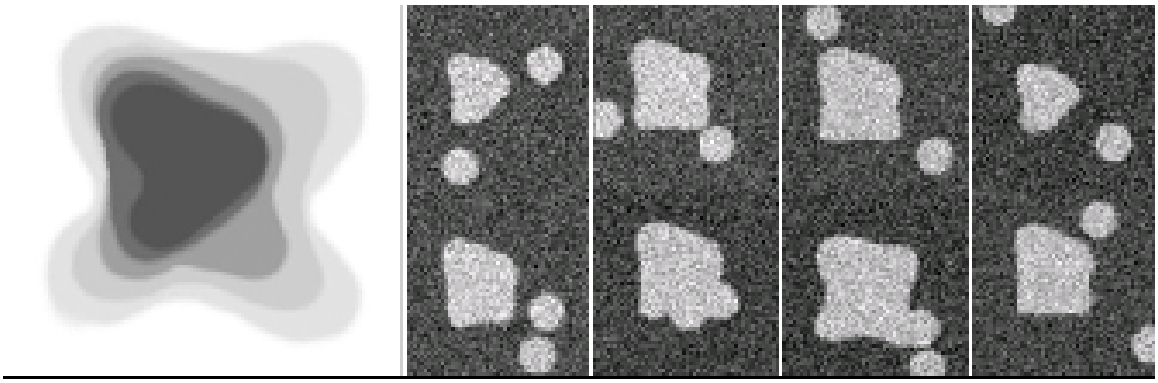


Figure 4.10: Artificial data set generated by composing a non-rigid shape deformation (left) and a diagonal translation. (Right) Eight examples from 100 generated images with SNR 10dB. Random white patches are added to the images to make the segmentation more challenging.

magnitude of each deformation. One hundred images were created and each was then corrupted by additive white Gaussian noise and the introduction of small random white patches. Figure 4.10 depicts the shape deformation and eight selected frames among the 100 generated images.

The noise in the image and the random patches make this a challenging data set for traditional level set approaches to converge to the correct boundary. The first row of Figure 4.11 shows the iso-contours of the final estimate of ϕ. The second row gives the contours which are the results of applying the our algorithm, which exploits the manifold structure of these images. Note our proposed method is very robust to the added noise. Conventional level set methods fail to detect the correct object boundaries.

This framework was also applied to segmenting cardiac MR images of the left ventricle. Figure 4.12 show examples of the segmentation result for eight consecutive frames.

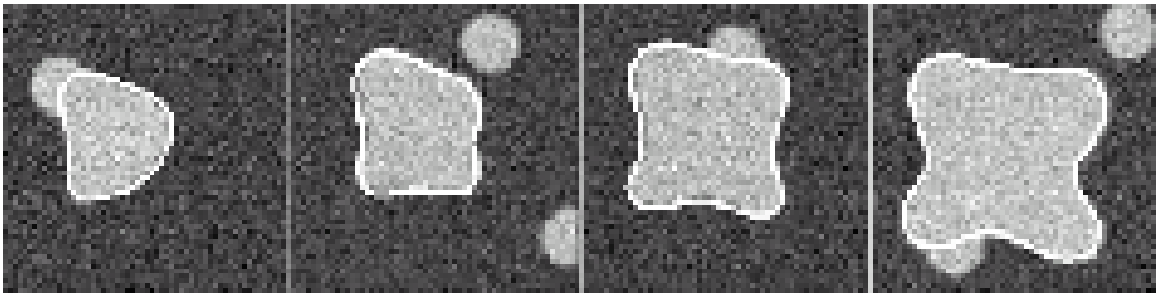


Figure 4.11: Example segmentation results from the level set approach described in [76] using image manifold constraints. Without being informed by the image manifold, conventional level set methods fail on this data set.

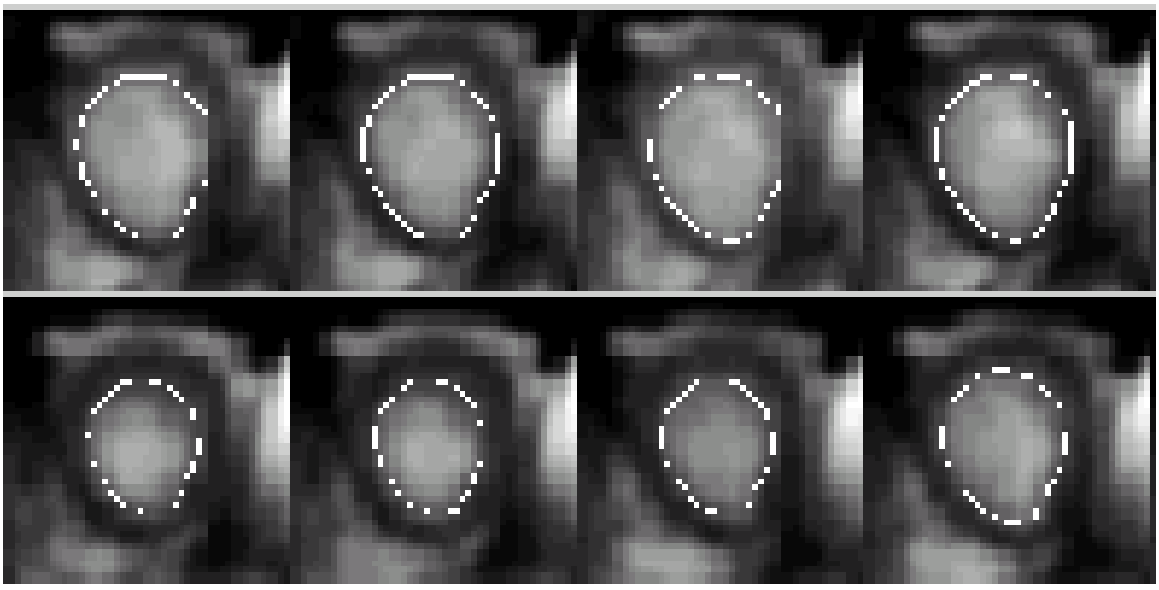


Figure 4.12: Segmentation examples of cardiac MR images.

In this work, it was discovered that the manifold structure was most important for images that are especially low contrast or noisy. This was shown by comparing the results from single-image level set segmentation with the two proposed level set segmentation methods. Figure 4.13 gives examples of images where the manifold based solutions differ significantly from the single image solution. In the first two cases, the manifold constraints show a significant improvement — the single image solutions are incorrect because of insufficient contrast. The last two cases show segmented shape boundaries that are different than the single image segmentation, which may reflect more accurately the correct boundary, although it is difficult to quantify the improvement (see details on the figure).

These results demonstrate how integrating manifold learning can improve the simultaneous segmentation of large image sets, if those image sets lie on some low-dimensional manifold and meaningful parameters can be learned. The natural applications of this likely lie primarily in medical imaging, particularly cardiopulmonary images.

The segmentation results shown in this section are not part of the work of this dissertation, however, the framework described in Chapter 3 forms the basis for this approach.

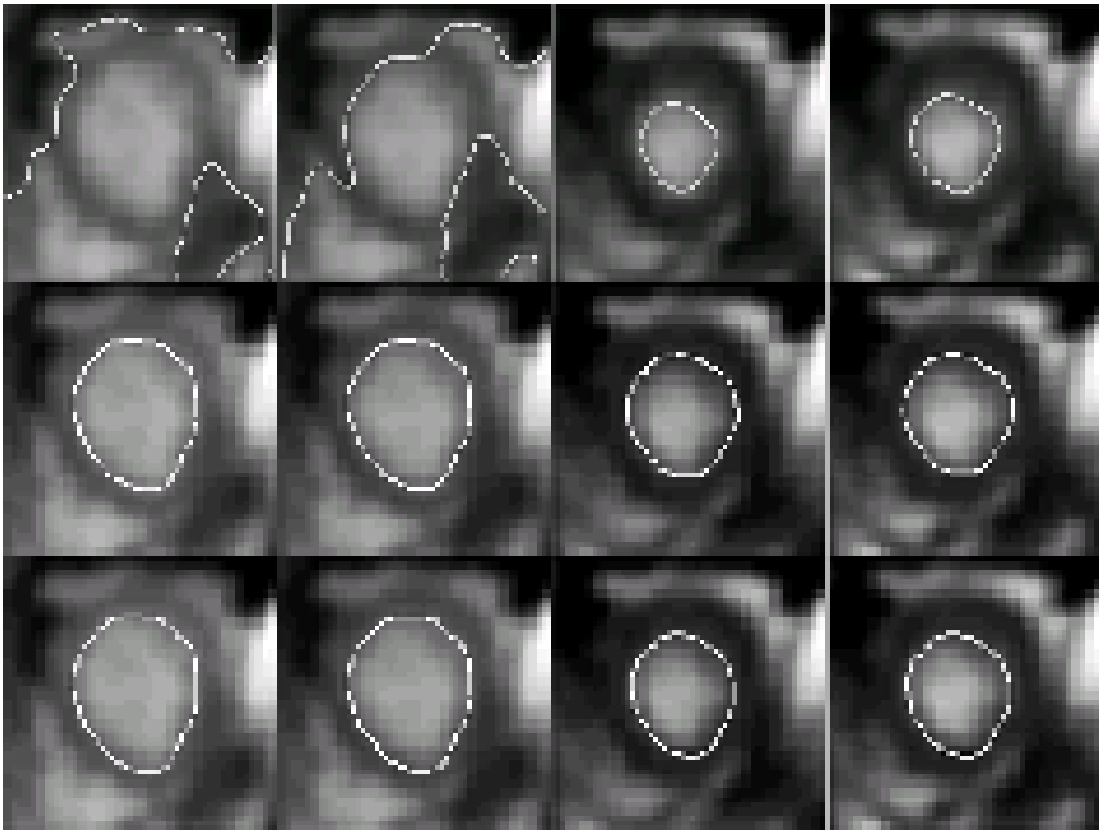


Figure 4.13: A comparison of single image based segmentation (top) and the simultaneous solution for all image using the 4D level set method (middle) and the 2D multilayer level set method (bottom). In the left two images, the single image solution fails because of low contrast, on the right manifold based solution differs and draws a perceptually more reasonable boundary.

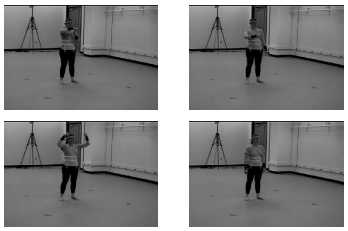
Chapter 5

Complex Topologies

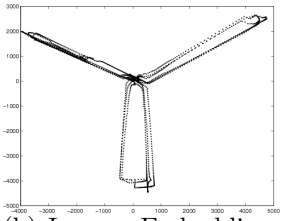
In the previous chapters, we discussed extending current manifold learning techniques in order to provide minimal parameterizations for natural image sets. This framework was using to improve common vision tasks such as registration and segmentation. In this chapter, we explore manifold topologies found in natural image sets and show how correctly parameterizing these types of manifolds can lead to algorithms which are relevant not only to researchers in computer vision, but the broader pool of machine learning.

In order to visual this problem and motivate this work, consider the video depicted in Figure 5.1(a). This video consists of an actor performing a series of basketball referee signals. The actor performed each of the three signals (technical foul, jump ball, and carrying) three times in the sequence. The data set consisted of 2212 frames of $x-$, $y-$, and $z-$coordinates for 175 markers. Each frame therefore represented a point in some 525-dimensional space (175 markers * 3 coordinates).

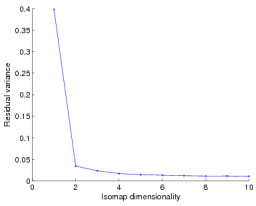
The Isomap embedding of this video data into two dimensions is depicted in Figure 5.1(b). Figure 5.1(c) shows the residual variance of the Isomap embedding for the



(a) Example Frames



(b) Isomap Embedding



(c) Residual Variance

Figure 5.1: The actor in the video performed a series of 3 basketball signals (technical foul, jump ball, and carrying) 3 times. (a) The images show examples of each signal plus the actor in the neutral position. (b) The original Isomap embedding of the 2212 frame motion capture data. (c) An output graph which estimates how well Isomap matches the original point distances. In this case, there is no significant gain by embedding into more than two dimensions.

given dimensionality. For this data set, there is no significant increase in embedding accuracy by using more than two dimensions. In other words, this data set can be described well by two parameters, namely the $x-$ and $y-$ coordinates in the embedding space. One question which then arises, is how meaningful are these coordinates? While, visually, one can see that Isomap has "discovered" the three different motions, it is not so clear that, given the $x-$ and $y-$ coordinates of an individual frame, one could know anything about the frame this point represents. For instance, it would be difficult to determine if two points were examples of the same motion or what phase of a given motion a frame represented. The rest of this section describes a method for parameterizing data sets such as this. The result is a novel algorithm which is generally useful for nonlinear, nonparametric subspace clustering.

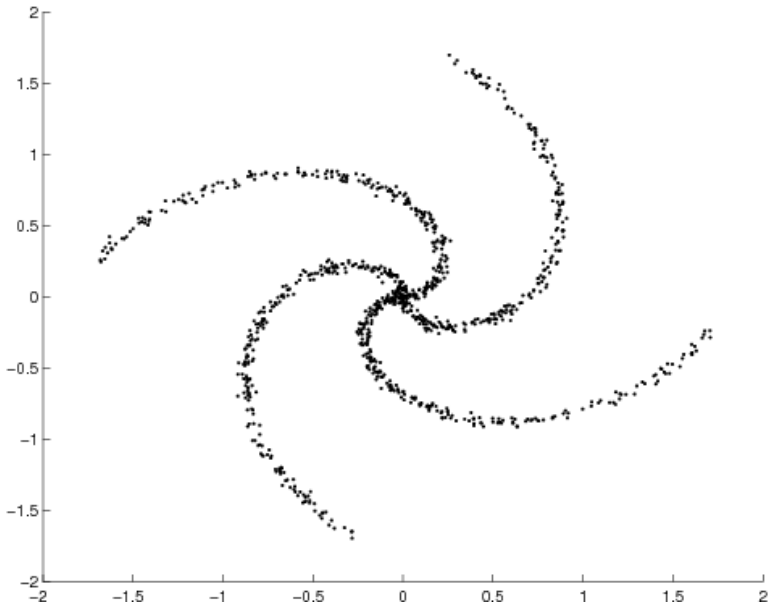


Figure 5.2: An example of data points drawn from intersecting manifolds.

5.1 Manifold Clustering

Up to this point in this dissertation, each of the data sets considered lies on low-dimensional manifolds that are nonlinear subspaces of the (high-dimensional) input data space. Current manifold learning approaches, as described in Chapter 2 seek to explicitly or implicitly define a low-dimensional embedding of the data points that preserves some properties (such as geodesic distance or local relationships) of the high-dimensional point set. However, when the input data points are drawn from multiple (low-dimensional) manifolds, such as the video from Figure 5.1(a), many manifold learning approaches suffer. In fact, if there is significant overlap in the manifolds, as shown in Figure 5.2, prior methods fail because these methods implicitly assume that connected data points lie along a single manifold.

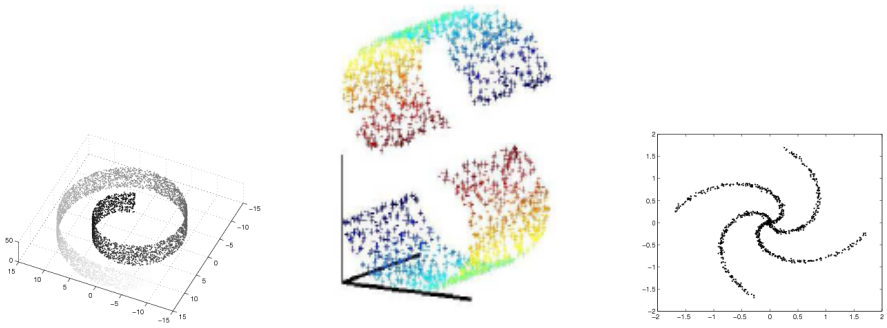
The desire would then be to partition an input data set into clusters where each cluster contains data points from a single low-dimensional manifold.

Figure 5.3 shows three example data sets. Figure 5.3(a) shows the so-called "swiss roll" data set which consists of 3D points drawn from a 2D manifold. All of the manifold learning algorithms previously described provide reasonable parameterizations for data sets such as this. Figures 5.3(b) and 5.3(c) depict cases when the input data points are drawn from multiple (low-dimensional) manifolds. For data sets such as these, many manifold learning approaches suffer. In the case where the multiple manifolds are separated by a gap (as in Figure 5.3(b)), techniques such as Isomap may discover the different manifolds as different connected components in the local neighborhood graph, and spectral clustering techniques [3] may identify and cluster each manifold based upon the optimization of certain objective functions. However, if there is significant overlap in the manifolds as shown in Figure 5.3(c), prior methods fail because these methods implicitly assume that connected data points lie along a single manifold.

Before we describe the algorithm, we review related methods and discuss the limitations of existing algorithms on data sets where the points come from multiple, intersecting manifolds. In Section 5.1.3 we show results of our algorithm on artificially generated data, human motion capture data, and image data.

5.1.1 Related Work

Clustering data into multiple, underlying models is a well-studied problem in many fields of science and engineering. Most approaches apply statistical methods to fit



(a) "Swiss roll" data set (b) "gapped" data set (c) "4-arm spirals" data set

Figure 5.3: Three data sets with different topologies. (a) shows the classic "Swiss Roll" example which can be parameterized by all manifold learning algorithms. (b) shows an example of a manifold with a "gap." Some algorithms parameterize this as two separate manifolds and some learn the global structure. (c) shows an example of intersecting manifolds, which is type of data considered in this chapter.

each data point to a parametric model, such as the Expectation-Maximization (EM) method for fitting Gaussian distributions.

There has been some recent work which is more closely related to the problem addressed here which considers the segmentation of data drawn from intersecting linear subspaces. Two algorithms, the K-subspaces algorithm [33] and generalized principal component analysis (GPCA) [67] model this type of input data. In Section 5.1.3, we describe these methods in more detail and compare the performance of our algorithm on artificially generated data.

These also exists recent work in manifold learning which has focused on data sets whose points do not lie on a single, simple manifold. [4] describe the *multi-manifold problem* where the data comes from an underlying, possibly large set of low-dimensional manifolds. The authors exploit smoothness constraints in the tangent spaces on different parts of the manifold in order to learn manifolds for under-sampled data. [71]

present the *multi-class manifold* problem. This considers data sets whose points come from a single, underlying low-dimensional manifold; however, this manifold is sampled in such a way that large "gaps" are introduced and the data set is fragmented. The algorithm presented described by [71] addresses a failure mode of Isomap [2] where, for data sets with certain properties, any neighborhood size selected either "short-circuits" the true manifold or only learns the manifold for a subset of the data points. The authors demonstrate an algorithm for learning the underlying manifold by differentiating between intra- and inter- fragment distances. Similarly, [28] describe a method where data sets whose points lie on disjoint manifolds are embedded on a single coordinate system.

In this work, we consider the case where the data is drawn from multiple, underlying, intersecting manifolds with unknown parametric forms. In addition, these manifolds are not necessarily fragments of a single underlying manifold and may have differing topology or dimensionality.

5.1.2 The k-Manifolds Algorithm

Our goal is to partition an input data set into clusters where each cluster contains data points from a single low-dimensional manifold. We start by assuming that the number and dimensionality of the low-dimensional manifolds are known. We define the manifold clustering problem as:

> *Given a set of points* $\{X_1, X_2, \cdots, X_n\}$ *derived from* k *intersecting nonlinear manifolds where* $X_i \in \mathcal{R}^D$ *for some dimension* D*, output the set of labels* $\{c_1, c_2, \cdots, c_n\}$ *where* $c_i \in \{1, 2, \cdots, k\}$ *is an index specifying to*

which output manifold a point belongs, and $\{Y_1, Y_2, \cdots, Y_n\}$ *where* $Y \in |R^d$ *(for* $d < D$*) is the low dimensional embedding of a point on its associated manifold.*

Without any priors on the labels of the input data, we are left to simultaneously learn this labeling and estimate the parameters of the underlying manifolds. The natural solution to this class of problems is an Expectation-Maximization (EM) type algorithm. For this problem, we propose a variant of the well-known k-means [41] algorithm, which we call *k-manifolds*. In our case, the cluster representation is a mapping function from embedded coordinates to a high-dimensional representation as opposed to the centroid of the set. We now provide a sketch of the algorithm. The remainder of this section describes the problems that arise and their technical solutions.

Given: A set of data points $X = \{X_1, X_2, \cdots, X_n\}$, where $X_i \in \mathcal{R}^D$ and a set of embedding dimensions $\{d_1, d_2, \cdots, d_k\}$, where $d_i < D$:

1. Calculate $D_E(i,j) = ||X_i - X_j||_2$

2. Create a graph $G = (V, E)$ with a vertex for each point, and an edge between pairs of neighboring[a] points. For points, i, j, let the edge weight equal $D_E(i,j)$.

3. Compute all pairs shortest path distances on G. Let $D_G(i,j)$ be the length of the shortest path between node i and node j. Define a distance matrix D such that $D(i,j) = D_G(i,j)^2$.

4. **Initialization:** For each point, X_i and each cluster c, let w_{ci} be the probability of X_i arising from the manifold described by cluster c. By default, these values are initialized randomly, unless domain-specific priors are available.

5. **M-Step:** For each cluster, c, supply $\vec{w}_c$ and D and apply node-weighted MDS (described in section 5.1.2) to return the coordinates $\{Y_1^c, Y_2^c, \cdots, Y_n^c\}$ of the embedding in d_c dimensions. Learn the function $f_c : \mathcal{R}^{d_c} \rightarrow \mathcal{R}^D$ which generalizes the mapping $Y_i^c \rightarrow X_i$

6. **E-Step:** For each point, calculate the residual to the manifold, $X_i - f_c(Y_i^c)$, and re-estimate the cluster probability for each point accordingly.

7. Go to Step 5 until convergence.

[a] Two common methods for selecting neighboring points are ϵ-sphere and k-nearest neighbors.

The algorithm begins (in the same manner as the Isomap algorithm) by estimating geodesic distances between points. The goal is to partition the points so that within each partition, the geodesic distances are consistent with the Euclidean distances of the low dimensional embedding. This partitioning step is performed using an iterative approach. The classic k-means algorithm has two distinct steps: the assignment of data points to model(s) of best fit (E-Step) and the estimation the parameters of those models (M-Step). In most cases, the assignments of points to models are partial, or soft, assignments. In the M-Step, these assignments serve to weight the contribution of each data point in defining each model.

Since current manifold learning algorithms treat each data point equally, one challenge is to develop a manifold embedding technique for weighted point sets. This challenge is met with our algorithm by using *Node-Weighted Multidimensional Scaling*, which we introduce in Section 5.1.2. We then generalize the mapping from the embedded coordinates returned by node-weighted MDS to the original data points using a radial basis function network, which we describe in Section 5.1.2. This mapping function, whose domain is the coordinate space of the embedding, is used in the E-step to update the assignments of each point by determining how well they fit each model.

Node-Weighted MDS

In our approach, we require a weighted version of traditional multidimensional scaling (MDS). We call this procedure node-weighted MDS to distinguish it from traditional weighted multidimensional scaling, commonly referred to as individual differences scaling or INDSCAL [13], which considers the problem of balancing multiple distance (or similarity) matrices when each point may have different weights with respect to

different distance matrices. Below, we outline traditional multidimensional scaling.

Multidimensional Scaling. Given $n \times n$ matrix D, such that $D(i,j)$ is the desired squared distance from point i to point j:

1. Define $\tau = -HDH/2$, (H is the centering matrix: $H = I - \vec{e}\vec{e}^{\top}/n$, where $\vec{e} = [1, 1, \cdots, 1]^{\top}$).

2. Let $s_1, s_2, \ldots$ be the (sorted in decreasing order) eigenvalues of τ, and let $v_1, v_2, \ldots$ be the corresponding (column) eigenvectors. The matrix $Y = [\sqrt{s_1}v_1 | \sqrt{s_2}v_2 | \ldots \sqrt{s_k}v_k]$ has row vectors which are the coordinates of the best k-dimensional embedding.

The matrix $YY^{\top}$ is the best rank k approximation to τ (with respect to the L_2 matrix norm). The process [37] finds the k-dimensional coordinates that minimize

$$\sum_{ij} \left(|Y_i - Y_j|_2^2 - D(i,j) \right)^2 .$$

Node-Weighted Multidimensional Scaling. In contrast to traditional MDS, node-weighted MDS seeks to minimize the following:

$$\sum_{ij} w_i w_j \left(|Y_i - Y_j|_2^2 - D(i,j) \right)^2 .$$

The process starts by changing the initial centering matrix to be a weighted centering matrix:

$$H' = (I - \vec{e}\vec{w}^{\top})$$

and then defining the correlation matrix $\tau = -H'DH'/2$. We then seek τ_k, a rank-k approximation to τ, that minimizes the weighted L_2 matrix norm:

$$\sum_{ij} w_i w_j \left(\tau_k(i,j) - \tau(i,j)\right)^2 .$$

The weighted low rank approximation problem is formally:

> *Given a matrix τ and a weight matrix W of the same dimensions, find the matrix τ_k of rank k that minimizes the Frobenius norm of the weighted difference: $||W \otimes (A - M)||_F$, where the $\otimes$ operator indicates element-wise multiplication of matrix elements.*

This problem has been approached with both the weight matrices constrained to be $\{0, 1\}$ valued and the general case of real-valued weights. Recent work suggests an iterative solution to this problem.

However, our application has additional constraints on the matrices τ and W that allow a direct solution to this problem [58, 34]. In particular, our weight matrix is symmetric and of rank one, and can be expressed as the outer product of the node weights expressed as a column vectors $W = \vec{w}\vec{w}^\top$. If we define $\tilde{W} = diag(\vec{w})$, (a matrix of all zeros with the weights w along the diagonal), then this special case of weighted low-rank approximation simplifies as follows:

$$\begin{aligned} ||(\vec{w}\vec{w}^\top) \otimes (\tau - \tau_k)||_F &= ||\tilde{W}(\tau - \tau_k)\tilde{W}||_F \\ &= ||\tilde{W}\tau\tilde{W} - \tilde{W}\tau_k\tilde{W}||_F \\ &= ||\tilde{W}\tau\tilde{W} - R||_F, \end{aligned} \tag{5.1}$$

where R is the low rank (unweighted) approximation to $\tilde{W}\tau\tilde{W}$. R can be found with standard singular value decomposition, leaving:

$$\begin{aligned} \tilde{W}\tau_k\tilde{W} &= R \\ \tau_k &= \tilde{W}^{-1}R\tilde{W}^{-1}. \end{aligned} \tag{5.2}$$

Given τ_k, finding Y such that $YY^\top = \tau_k$, (using the same eigenvector decomposition as in MDS) gives the k-dimensional coordinates of the node-weighted MDS embedding.

Distances to Implied Manifolds

To implement the E-step of the algorithm, we need to re-weight the points based on how well they fit each of the k manifolds. This section details a method to estimate the distance of a point to a manifold which is defined only implicitly through the weighted embedding of points.

The result of node-weighted MDS and other manifold learning methods is a low dimensional embedding of the original data points, Y, where $Y \subseteq \mathcal{R}^d$ for some dimensionality, d. This implies a mapping of points in the embedded space to points in the original space of higher dimensionality, D:

$$M : \mathcal{R}^d \rightarrow \mathcal{R}^D.$$

We must now generalize this mapping into a function of the same form. One method for this generalization [24] employs generalized radial basis functions (GRBF) [47]

which estimate functions of the form:

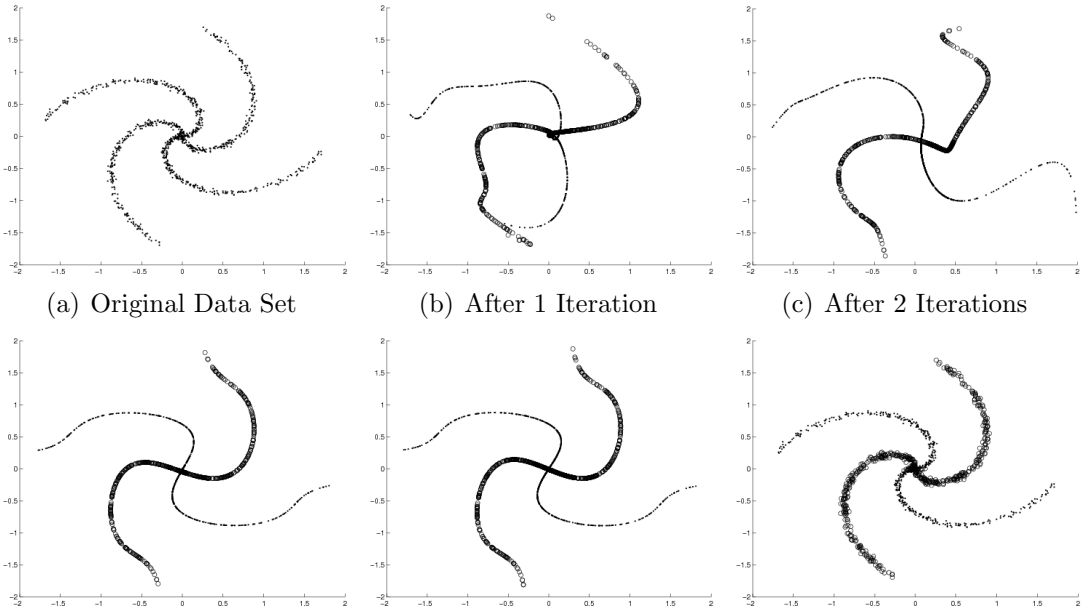
$$f_c(x) = \sum_{j=1}^{C} \lambda_j \phi(||x - z_j||_2) + b, \tag{5.3}$$

where $f_c : \mathcal{R}^{d_c} \rightarrow \mathcal{R}^D$ and d_c is the specified dimensionality of cluster c and D is the dimensionality of the original data points, b is a bias vector, λ_j is the real-valued weight of kernel center z_j, for $j \in 1, 2, ..., C$, and ϕ is a real-valued *basic function*. In our algorithm, we choose the Gaussian function for ϕ: $\phi(r) = e^{-\frac{r^2}{\sigma_w^2}}$, where σ_w is the average intra-center distance.

In addition to selecting the type of kernel, another free parameter for radial basis function networks is the number of kernel centers to use as hidden units in the network. This parameter can be tuned and optimized using cross-validation on the training set, however this operation is quite costly. In practice, a small number of kernel centers $C \approx .05n$, where n is the number of training set examples, produces reasonable results. We fit these C kernel centers to our input data $Y \subseteq \mathcal{R}^{d_c}$ by fitting a Gaussian mixture model with spherical covariances using the EM algorithm.

We now create the $n \times C$ activation matrix, R. Let R_{ic} be the distance from embedded point $Y_i \in Y$ to kernel center c. To learn the weight and bias vectors, λ and b (shown in Equation 5.3), respectively, we solve the following equation:

$$\begin{bmatrix} \lambda \\ b \end{bmatrix} = (W[R\ \vec{1}])^{-1}(WX),$$



(a) Original Data Set (b) After 1 Iteration (c) After 2 Iterations

(d) After 3 Iterations (e) Final Manifold Hypotheses (f) Final Assignment

Figure 5.4: (a) depicts the "4-arm spiral" data set of 1000 points in 2-dimensions. (b) through (e) show the hypothesized manifolds for both of the 1-dimensional clusters during the iterations of our algorithm. (f) shows the final assignment of each data point to its nearest hypothesized manifold. There were a total of 4 iterations in this trial.

where W is a diagonal matrix containing the values w_c which represent the weight of point i on cluster c, X contains the original data points, and $\vec{1}$ is a column vector of 1's.

Classifying Data Points

For each iteration of our algorithm, we calculate the likelihood of each point belonging to each of the k clusters, using the *softmin* function. That is, the likelihood of point X_i belonging to cluster c, given the embedding coordinates Y_i^c and the cluster manifold

prediction function, f_c, is:

$$w_i^c = \frac{e^{\frac{-(X_i - f_c(Y_i^c))^2}{\sigma^2}}}{\sum_{j=1}^{k} e^{\frac{-(X_i - f_j(Y_i^j))^2}{\sigma^2}}}$$

where σ^2 is simply the weighted variance for each cluster:

$$\hat{\sigma^c}^2 = \frac{\sum w_i^c}{(\sum w_i^c)^2 - \sum (w_i^{c2})} \sum w_i^c (X_i - f_c(Y_i^c))^2.$$

In practice, we apply a lower bound threshold to σ equal to the average inter-neighbor distance between the points in the original data set.

Figure 5.4 shows example manifold hypotheses for a synthetic data set and the final classification assignments for each data point using the k-manifolds scheme.

5.1.3 Applications of k-Manifolds

The extension of manifold learning to data sets that arise from multiple intersecting manifolds is an important step in applying these techniques more broadly. In this section, we present results showing manifold clustering results on a several examples including manifolds of different topology and dimension, and an application to a natural data set of human motion capture. In each case, the dimensionality of the component manifolds was provided to the algorithm.

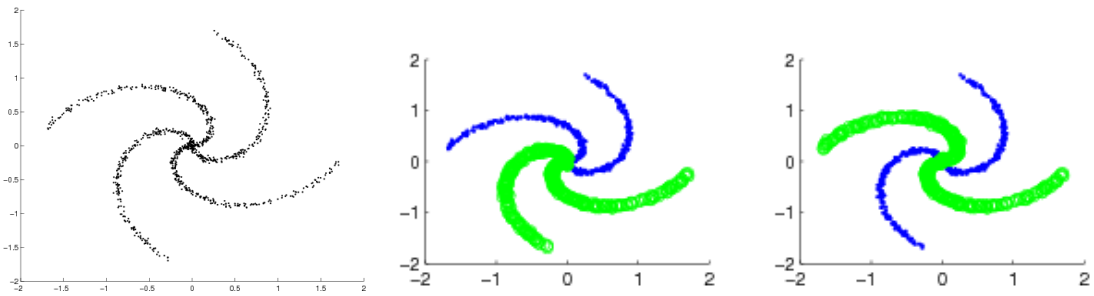


Figure 5.5: Two classification results from our algorithm on the "4-arm" spirals data set.

Locally Optimal Solutions

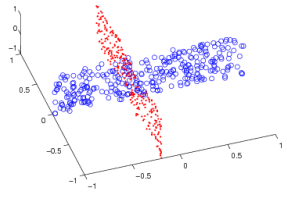
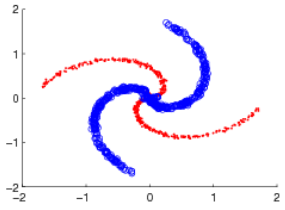
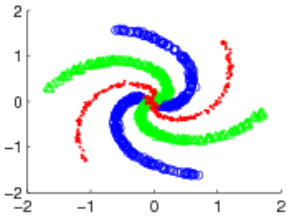
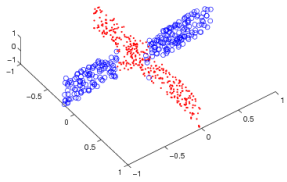
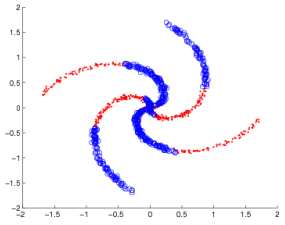
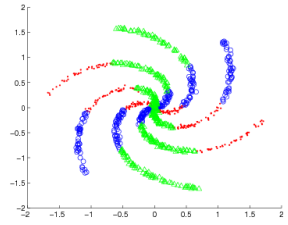
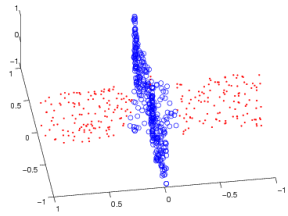
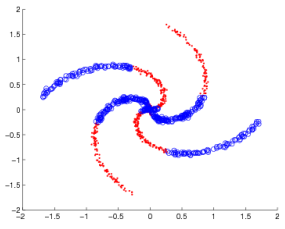
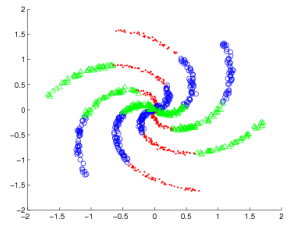
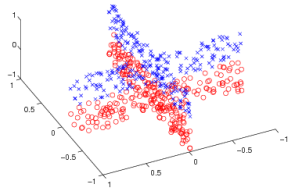
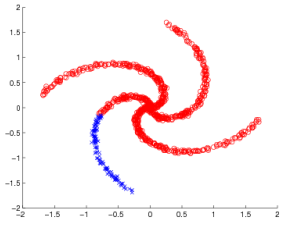
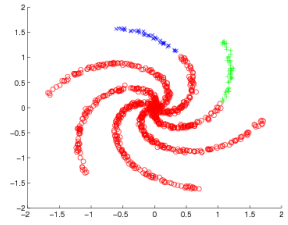
For certain data sets, multiple valid classifications are possible. Figure 5.5 shows two different results for our "4-arm spirals" data set. As with most unsupervised algorithms, depending on the random initialization of the cluster weights, our algorithm converges to either solution. The algorithm can be biased towards one solution or the other by specifying initial weights for the points in the original data set.

Comparison to Similar Methods

In this section, we compare our method to other, similar methods for unsupervised high-dimensional data clustering. We briefly describe each method and show the classification results in Table 5.1.

Spectral Clustering. Spectral Clustering is a recently popular technique for data clustering. Spectral clustering, which descends from graph partitioning, examines the eigenvectors of the Laplacian of a pairwise similarity, or affinity, matrix. In [9], an algorithm is presented which performs well for clustering data points which arise from disjoint manifolds. The affinity matrix, A is constructed using the pairwise Euclidean

Table 5.1: Comparison to other clustering methods. Each algorithm required specifying the number of clusters. For the first two examples, we specified 2 clusters, and for the "6-arm" spirals we specified 3 clusters.

k-manifolds			
GPCA			
k-subspaces			
Spectral Clustering			

distance between points, where $A_{ij} = e^{\frac{-||x_i - x_j||^2}{2\sigma^2}}$. This matrix is then normalized and the projected onto low-rank matrices. For the examples in Table 5.1, we used an implementation of the algorithm described by [9]. This competing algorithm is sensitive to the kernel size, σ, so we hand-tuned this free parameter to give the most reasonable results. This led to the choice of $\sigma = 10\sigma_s$, where σ_s is the average distance between a point and its closest neighbor, as this seemed to produce the best results.

GPCA. Generalized Principal Component Analysis (GPCA) extends PCA to the class of problems where the input data sets consist of points drawn from multiple, intersecting linear subspaces. GPCA applies methods from algebraic geometry to segment the data set in a non-iterative fashion. It is also important to note that GPCA can be extended to segment data drawn from intersecting nonlinear manifolds if the parametric form of the distribution is known *a priori*. GPCA has been used in computer vision for segmenting images, motions, and dynamic textures. The GPCA results in Table 5.1 were obtained using source code from [48] which is described in [73].

K-subspaces. K-subspaces is algorithm originally designed to cluster various 3D objects under varying illumination conditions. This algorithm, like our method, uses an iterative approach to learn the classification. However, unlike our algorithm, the input points must be derived from a linear subspace. The K-subspaces results were obtained using code from [72].

We compared our algorithm against implementations of spectral clustering, GPCA, and K-subspaces on three synthetic data sets:

- 3D points drawn off two perpendicular hyperplanes. Each of the hyperplanes is modulated by a sinusoid
- 2D points drawn off two intersecting "spirals" of the form:

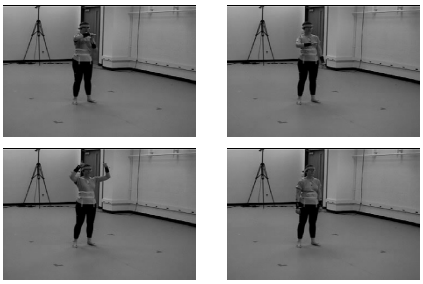
$$< e^s \cos(s + \theta_c), e^s \sin(s + \theta_c) >,$$

  where $s \in (0.0, 1.0)$ and θ_c a constant for all the points generated on each spiral
- 2D points drawn off three intersecting "spirals" of the same form as above.

Unsupervised Semantic Segmentation

We applied the k-manifolds algorithm to the analysis of human motion capture data of various simple activities from the CMU Motion Capture Database. Each data set contains examples of a single actor performing a series of simple motions. Each motion lies on a one-dimensional manifold in image space and the task is an unsupervised segmentation of the frames of the video into classes of different motions. Ground truth was obtained by manual annotation of the original video sequence.

One test was applied on a data set of an actor performing a series of basketball referee signals. The actor performed each of the 3 signals (technical foul, jump ball, and carrying) 3 times in the sequence. The data set consisted of 2212 frames of $x-$, $y-$, and $z-$coordinates for 175 markers. Each frame therefore represented a point in some 525-dimensional space (175 markers * 3 coordinates). We applied our method using $k = 3$ one-dimensional clusters to classify each frame and compared our results



(a) Example Frames

Cluster	1	2	3
Technical	514	20	0
Jump Ball	1	461	53
Carrying	17	0	672

(b) Confusion Matrix

Figure 5.6: Results on human motion capture data. The actor in the video performed a series of 3 basketball signals (technical foul, jump ball, and carrying) 3 times. (a) The images show examples of each signal plus the actor in the neutral position. (b) The confusion matrix describing the clustering results of our algorithm in terms of the number of frames classified as each signal.

to the ground truth. Figure 5.6 shows our results on this data set. In this case, our method performs with 94.8% accuracy.

Chapter 6

Conclusions and Future Work

The work of this dissertation makes the methods of manifold learning applicable to natural image sets and provides the framework for image manifold learning. Specifically, the major contributions include:

- Applying a statistical framework to quantify the differences between related images.
- Using the constraints implied by this framework to improve common vision tasks such as image registration, segmentation, and interpolation.
- Developing a novel algorithm for parameterizing complex topologies found in natural image sets.

In its current stage, this framework provides for generalized video analysis under known image transformation. In the future, this framework could be extended to automatically select which of set of metrics is most useful for a given data set. Consider the case where image variation is due to multiple causes, drawn from the set major modes of natural image transformations (rigid transform, non-rigid transform,

lighting variation, noise.) It would be useful to find a set of distance measures d_1, d_2, d_3, and d_4, such that two images A and B that vary due to only one cause (measured by d_1) have $d_1(A, B) >> 0$ and $d_x(A, B) < \epsilon$, for $x \in \{2, 3, 4\}$. Then, selecting a set of metrics for a particular problem can be posed as choosing among the different distance measures according to some criteria, such as minimizing embedding distortion.

These results have proven useful in the important domain of medical imaging and, in the future, could be used to improve current diagnostic techniques and develop new medical procedures.

References

[1] D. K. Agrafiotis and H. Xu. A self-organizing principle for learning nonlinear manifolds. *Proceedings of the National Academy of Sciences*, 99:15869–15872, 2002.

[2] Mukund Balasubramanian, Eric L. Schwartz, Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. The isomap algorithm and topological stability (technical comment). *Science*, 295(5552):7a, 2002.

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 585–591, Cambridge, MA, 2002. MIT Press.

[4] Yoshua Bengio and Martin Monperrus. Non-local manifold tangent learning. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 129–136. MIT Press, Cambridge, MA, 2005.

[5] Yoshua Bengio, Jean-François Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Le Roux, and Marie Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[6] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *PAMI*, 14(2):239–256, February 1992.

[7] Christopher M. Bishop, Markus Svensen, and Christopher K. I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

[8] Fred L. Bookstein. Principal warps: Thin plate splines and the decomposition of deformations. *Pattern Analysis and Machine Intelligence*, 11, June 1989.

[9] M. Brand and K. Huang. A unifying theorem for spectral embedding and clustering. In C. M. Bishop and B. J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

[10] Matthew Brand. Charting a manifold. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 961–968. MIT Press, Cambridge, MA, 2003.

[11] M.E. Brand. Continuous nonlinear dimensionality reduction by kernel eigenmaps. In *International Joint Conference on Artificial Intelligence (IJCAI)*, August 2003.

[12] Miguel Á. Carreira-Perpiñán and Richard S. Zemel. Proximity graphs for clustering and manifold learning. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 225–232. MIT Press, Cambridge, MA, 2005.

[13] J. Douglas Carroll and Jih-Jie Chang. Three-way scaling and clustering. *Psychometrika*, 35:238–319, 1970.

[14] H P Chang, P H Liu, H F Chang, and C H Chang. Thin-plate spine (tps) graphical analysis of the mandible on cephalometric tadiographs. *Dentomaxillofacial Radiology*, 31:137–141, 2002.

[15] H. Choi and S. Choi. Kernel isomap on noisy manifold. In *Proc. 4th IEEE Int. Conf. on Development and Learning (ICDL)*, Osaka, Japan, July 2005.

[16] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *PNAS*, 102(21):7426–7431, 2005.

[17] Andre Collignon, Frederik Maes, D. Delaere, Dirk Vandermeulen, P. Suetens, and G. Marchal. Automated multi-modality image registration based on information theory. *Information Processing in Medical Imaging*, pages 263–274, 1995.

[18] M. Corvi and G. Nicchiotti. Multiresolution image registration. In *Proc. International Conference on Image Processing (Vol. 3)-Volume 3*, page 3224, Washington, DC, USA, 1995. IEEE Computer Society.

[19] D de Ridder, O Kouropteva, O Okun, M Pietikainen, and R.P.W. Duin. Supervised locally linear embedding. In *Artificial Neural Networks and Neural Information Processing, ICANN/ICONIP 2003 Proceedings, Lecture Notes in Computer Science 2714*, pages 333–341. Springer, 2003.

[20] D. de Ridder, M. Loog, and M.J.T. Reinders. Local fisher embedding. In *Proc. of the 17th International Conference on Pattern Recognition, 2004. (ICPR 2004)*, volume 2, pages 295–298, August 2004.

[21] Vin de Silva and Joshua B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 705–712. MIT Press, Cambridge, MA, 2003.

[22] David Donoho and Carrie Grimes. When does isomap recover the natural parameterization of families of articulated images? Technical report, Stanford University, August 2002.

[23] David L. Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *PNAS*, 100(10):5591–5596, 2003.

[24] A. Elgammal and Chan-Su Lee. Separating style and content on a nonlinear manifold. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 478–485, June 2004.

[25] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24:381–395, 1981.

[26] U. Grenander. Elements of pattern theory. Johns Hopkins, Baltimore, 1996.

[27] Mark Griffin. Measuring cardiac strain using thin plate splines. In *Proc. Computational Techniques and Applications*, Brisbane, Australia, July 2001.

[28] Abdenour Hadid and Matti Pietikäinen. Efficient locally linear embeddings of imperfect manifolds. In Petra Perner and Azriel Rosenfeld, editors, *Machine Learning and Data Mining in Pattern Recognition, Third International Conference, MLDM 2003*, pages 188–201, 2003.

[29] Jihun Ham, Daniel D. Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 47, New York, NY, USA, 2004. ACM Press.

[30] Xiaofei He and Partha Niyogi. Locality preserving projections. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[31] M. Hein and Y. Audibert. Intrinsic dimensionality estimation of submanifolds in r^d. In L. De Raedt and S. Wrobel, editors, *Proceedings of the 22nd International Conference on Machine Learning*, pages 289 – 296, 2005.

[32] Geoffrey Hinton and Sam Roweis. Stochastic neighbor embedding. In *Neural Information Processing Systems*, 2002.

[33] Jeffrey Ho, Ming-Hsuan Yang, Jongwoo Lim, Kuang-Chih Lee, and David Kriegman. Clustering appearances of objects under varying illumination conditions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 11, 2003.

[34] Michal Irani and P. Anandan. Factorization with uncertainty. In *Proc. European Conference on Computer Vision*, pages 539–553, 2000.

[35] Odest Chadwicke Jenkins and Maja J. Mataric. A spatio-temporal extension to isomap nonlinear dimension reduction. In *ICML '04: Twenty-first international conference on Machine learning*, New York, NY, USA, 2004. ACM Press.

[36] I T Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.

[37] J. B Kruskal and M. Wish. Multidimensional scaling. *Sage University Paper series on Quantitative Application in the Social Sciences*, pages 07–011, 1978.

[38] Elizaveta Levina and Peter J. Bickel. Maximum likelihood estimation of intrinsic dimension. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 777–784. MIT Press, Cambridge, MA, 2005.

[39] I S Lim, P H Ciechomski, S Sarni, and D Thalmann. Planar arrangement of high-dimensional biomedical data sets by isomap coordinates. In *Proceedings of the 16th IEEE Symposium on Computer-Based Medical Systems (CBMS 2003)*, New York, 2003.

[40] Yi Ma, Stefano Soatto, Jana Kosecka, and Shankar Sastry. *An Invitation to 3D Vision*, chapter 11. Spinger-Verlag, November 2003.

[41] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *In Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[42] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, page 786, Washington, DC, USA, 1995. IEEE Computer Society.

[43] Hiroshi Murase and Shree K. Nayar. Visual learning and recognition of 3-d objects from appearance. *Int. J. Comput. Vision*, 14(1):5–24, 1995.

[44] J.H. Park, Z. Zhang, Hongyuan Zha, and R. Kasturi. Local smoothing for manifold learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 452–459, 2004.

[45] Robert Pless and Ian Simon. Using thousands of images of an object. In *CVPRIP*, 2002.

[46] Josien P. W. Pluim, J.B. Antoine Maintz, and Max A. Viergever. Mutual information based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004, August 2003.

[47] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78:113–125, 1990.

[48] Shankar Rao, Andrew Wagner, and Allen Yang. GPCA-V source code. `http://perception.csl.uiuc.edu/gpca/sample\_code/index.htm`, 2005.

[49] H. J. Ritter and T. Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61:241–254, 1989.

[50] Sam Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, December 2000.

[51] D. E. Rumelhart and D. Zipser. *Feature discovery by competitive learning*. MIT Press, Cambridge, MA, USA, 1986.

[52] Ferdinando Samaria and Andy Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, Sarasota, FL, December 1994.

[53] Gerardo I. Sanchez-Ortiz, Daniel Rueckert, and Peter Burger. Motion and deformation analysis of the heart using thin-plate splines and density and velocity encoded mr images. In *16th Statistical Workshop on Image Fusion and Shape Variability*, pages 71–78, Leeds, UK, July 1996.

[54] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, 1998.

[55] T. Sederberg and S. Parry. Free-form deformation of solid geometric models. In *Proceedings of SIGGRAPH '86*, pages 151–160, August 1996.

[56] R. Sibson. A brief description of natural neighbour interpolation. In Vic Barnet, editor, *Interpreting Multivariate Data*, pages 21–36. John Wiley and Sons, Chichester, 1981.

[57] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000.

[58] Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, 2003.

[59] C. Studholme, D. Hill, and D. Hawkes. An overlap invariant entropy measure of 3d medical image alignment. *Pattern Recognition*, 32:71–85, 1999.

[60] Joshua Tenebaum, Vin de Silva, and John Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, December 2000.

[61] D'Arcy Thompson. *On Growth and Form*. Cambridge Universtiy Press, 1917.

[62] A Trouve. Diffeomorphism groups and pattern matching in image analysis. *International Journal of Computer Vision*, 28:213–221, 1998.

[63] A Trouve and L Younes. Local analysis of a shape manifold. Technical Report 2002-03, Laboratoire d'Analyse, Geometrie et Applications, CNRS, Universite Paris, 2002.

[64] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Neuroscience*, 3(1):71–86, 1991.

[65] Jakob J. Verbeek, Sam T. Roweis, and Nikos Vlassis. Non-linear cca and pca by alignment of local models. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[66] J.J. Verbeek, N. Vlassis, and B. Krose. The generative self-organizing map: A probabilistic generalization of kohonen's SOM. Technical Report IAS-UVA-02-03, University of Amsterdam, The Netherlands, May 2002.

[67] R. Vidal, Yi Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.

[68] Paul A. Viola. *Alignment by maximization of mutual information*. PhD thesis, Massachusetts Institute of Technology, University of Maryland, 1995.

[69] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-04)*, volume II, pages 988–995, Washington D.C., 2004.

[70] William M. Wells, Paul A. Viola, and Ron Kikinis. Multi-modal volume registration by masimization of mutual information. *Medical Robotics and Computer Assisted Surgery*, pages 55–62, 1995.

[71] Yiming Wu and Kap Luk Chan. An extended isomap algorithm for learning multi-class manifold. In *Proceedings of IEEE International Conference on Machine Learning and Cybernetics (ICMLC2004)*, Shanghai, China, 2004.

[72] Allen Yang. K-Subspaces source code.

`http://perception.csl.uiuc.edu/gpca/sample\_code/index.htm`, 2005.

[73] A.Y. Yang, S. Rao, A. Wagner, Y. Ma, and R.M. Fossum. Hilbert functions and applications to the estimation of subspace arrangements. In *Proc. International Conference on Computer Vision*, volume 1, pages 158–165, 2005.

[74] Hongyuan Zha and Zhenyue Zhang. Isometric embedding and continuum ISOMAP. In *Proc. of the Twentieth International Conference on Machine Learning (ICML-2003)*, Washington, DC, August 2003.

[75] Qilong Zhang, Richard Souvenir, and Robert Pless. Segmentation informed by manifold learning. In Anand Rangarajan, Baba C. Vemuri, and Alan L. Yuille, editors, *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, volume 3757 of *Lecture Notes in Computer Science*, pages 398–413. Springer, 2005.

[76] Qilong Zhang, Richard Souvenir, and Robert Pless. On manifold structure of cardiac mri data: Application to segmentation. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1:1092–1098, 2006.

Vita

Richard M. Souvenir

Date of Birth September 3, 1979

Place of Birth Chicago, IL

Academic Degrees
B.S. Computer Science and Biology, August 2001
M.S. Computer Science, December 2003
D.Sc. Computer Science, August 2006

Professional Appointments
Media and Machines Laboratory
Washington University in Saint Louis
Graduate Research Assistant
Supervisor: Robert Pless

Computational Intelligence Center
Washington University in Saint Louis
Graduate Research Assistant
Supervisor: Weixiong Zhang

Publications

Book Chapters/Journal Publications

R. Souvenir and R. Pless, Image Distance Functions for Manifold Learning, Image & Vision Computing, accepted for publication January 2006.

R. Souvenir, J. Buhler, G. Stormo, and W. Zhang. An Iterative Method for Selecting Degenerate Multiplex PCR Primers, accepted for publication July 2006.

Refereed Conference/Workshop Publications

R. Souvenir, Q. Zhang, and R. Pless, Image Manifold Interpolation Using Free-Form Deformations, International Conference on Image Processing (ICIP 2006), to appear.

Q. Zhang, R. Souvenir, and R. Pless, On Manifold Structure of Cardiac MRI Data: Application to Segmentation, in Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR 2006), New York, NY, June 2006, pp. 1092-1098.

R. Souvenir and R. Pless, Manifold Clustering, in Proceedings of the 10th International Conference on Computer Vision (ICCV 2005), Beijing, China, October 2005, pp. 648-653.

Q. Zhang, R. Souvenir, and R. Pless, Segmentation Informed by Manifold Learning, Lecture Notes in Computer Science, volume 3757, October 2005, pp. 398-413.

R. Souvenir, J. Wright, and R. Pless, Spatio-Temporal Detection and Isolation: Results on the PETS2005 Datasets, IEEE Performance Evaluation and Tracking Systems (PETS 2005), Breckenridge, Colorado, January 2005.

R. Souvenir and R. Pless, Isomap and Nonparametric Models of Image Deformation, in Proceedings of 7th IEEE Workshop on Applications of Computer Vision / IEEE Workshop on Motion & Video Computing (WACV/MOTION 2005), Breckenridge, Colorado, January 2005, pp. 195-200.

J. Buhler, R. Souvenir, W. Zhang and R. Mitra, Design of a High-Throughput Assay for Alternative Splicing Using Polymerase Colonies, in Proceedings of the Pacific Symposium on Biocomputing (PSB-04), Hawaii, January 2004, pp. 5-16.

R. Souvenir, J. Buhler, G. Stormo, and W. Zhang, Selecting Degenerate Multiplex PCR Primers, in Proceedings of the 3rd Workshop on Algorithms in Bioinformatics (WABI-03), Budapest, Hungary, September 2003, pp. 512-526.

University Service

President, Washington University Graduate Professional Council
Delegate, National Conference on Graduate Student Leadership
President, CSE Graduate Student Association

Judge, Washington University Graduate Student Symposium
Panelist, Target Hope Conference

Honors

NSF Graduate Research Fellowship
NIH Computational Biology Training Grant

Teaching

Fundamentals of Computer Science (Summer 2003, 2005)
Seminar on Graphics, Robotics, and Vision (Spring 2004)

August 2006