Differential Structure in non-Linear Image Embedding Functions

Robert Pless
Department of Computer Science, Washington University in St. Louis
pless@cse.wustl.edu

Abstract

Many natural image sets are samples of a low dimensional manifold in the space of all possible images. When the image data set is not a linear combination of a small number of basis images, then linear dimensionality reduction techniques such as PCA and ICA fail, and non-linear dimensionality reduction techniques are required to automatically determine the intrinsic structure of the image set. Recent techniques such as ISOMAP and LLE provide a mapping between the images and a low dimensional parameterization of the images. In this paper we consider how choosing different image distance metrics affects the low-dimensional parameterization. For image sets that arise from non-rigid and human motion analysis, and MRI applications, differential motions in some directions of the low-dimensional space correspond to common transformations in the image domain. Defining distance measures that are invariant to these transformations makes Isomap a powerful tool for automatic registration of large image or video data sets.

1. Introduction

Faster computing power and cheap large scale memory has led to a surge in research in the machine learning community on the topic of dimensionality reduction: finding structure in a large set of points embedded in a very high dimensional space. Many problems in computer vision can be cast in this framework, as each image can be considered to be a point in a very high dimensional space (one dimension for each pixel). When an image data set is generated by varying just a few parameters, such as a combination of pose, lighting, or camera viewpoints, then the set of images can be described as sampling a continuous manifold of the space of all possible images.

Isomap [6] and Locally Linear Embedding (LLE) [4] typify a class of techniques that discover a low-dimensional parameterization of a point set. Given a set of images $\mathcal{I}$, these methods define a mapping:

$$f : \mathcal{I} \longrightarrow R^k$$

where k is usually a small number such as 2 or 3, and is ideally the number of free parameters that were varied in creating the image data set.

Significantly, the problem with these methods is that they define only a mapping of the original image set to a k-dimensional space. Unlike popular linear dimensionality reduction techniques (such as PCA or ICA) the function f defined by these methods is *not defined for any image not in the original image set*, and there is *no defined inverse mapping* that takes a new set of parameters (a point in R^k) and returns the corresponding image. Despite these drawbacks, this method has been effectively used in a variety of applications, by directly parameterizing image sets in the context of computing pose estimates in rigid body motions [8], visualization of biomedical image data sets [3], or by parameterizing a large set of filter responses to tune representations for visual tracking [9] or to learn and represent the space of Bidirectional Reflectance Distribution Functions (BRDFs) in graphics applications [7].

The contribution of this paper is a framework for the specialization of a wide class of non-linear dimensionality techniques for use in many computer vision problems. In particular we exploit the fact that the mapping function f is a diffeomorphism, and that for important application areas, the differential motion in the parameter space corresponds to specific image operators such as warping or contrast changes. These operators are incorporated into the local distance function. We illustrate this on deformable and human motion data sets and show how it gives rise to automatic registration algorithms to improve image de-noising for MRI data.

2. Projections and Inverse Projections

Given an input set $\mathcal{I}$, which is a finite subset of R^n, (where n is the number of pixels in an image), the dimensionality reduction techniques of Isomap and LLE produce a mapping function $f : \mathcal{I} \longrightarrow R^k$. Very briefly, Isomap begins by computing the distance between all pairs of images (using the square root of the sum of the squared pixel errors, which is the L_2 norm distance if the images are considered points in R^n). Then a graph is defined with each point as a node, and edges are created to the closest neighbors (usually choosing 5 to 15 neighbors). Then distances are computed between every pair of nodes in the graph using any all-pairs shortest path algorithm to give a complete distance matrix. Finally, this complete distance matrix is embedding into R^k, by solving an eigenvalue problem (a technique called multi-dimensional scaling). This mapping preserves the distance relationships defined by paths between nearest neighbors in the original data set. LLE is a method with similar aims that creates a mapping that preserves linear relationships between nearby points. The original papers for Isomap [6] and LLE [4] have pointers to online, free implementations of the algorithm, and several other papers have discussed appropriate distance measures in the direct application of Isomap to embedding image sets [8, 3]).

It is instructive to view PCA in the same light. Given an input data set $\mathcal{I}$ (also a finite subset of R^n), Principle Component Analysis computes a function f which projects each image (in our case) onto a set of basis images. The image set $\mathcal{I}$ defines a set of orthonormal basis images $\mathbf{b_1}, \mathbf{b_2}, \ldots \mathbf{b_k}$, and then the function f maps any image $\mathbf{x}$ in R^n to a set of coefficients that define a point in R^k:

$$f(\mathbf{x}) = (\mathbf{x}^\top \mathbf{b}_1, \mathbf{x}^\top \mathbf{b}_2, \ldots \mathbf{x}^\top \mathbf{b}_k) = (c_1, c_2, \ldots c_k)$$

Therefore, although the bases images, and therefore the function f are defined based upon an eigen-analysis of the image data set I, it actually gives a function f that is defined for all possible images of n pixels:

$$f_{\mathsf{PCA}} : \mathcal{R}^n \longrightarrow R^k$$

Furthermore, the inverse function is defined as well, so that any point in point in the coefficient space is mapped to a specific image by a linear combination of the basis images:

$$f^{-1}(c_1, c_2, \ldots c_k) = c_1\mathbf{b}_1 + c_2\mathbf{b}_2 + \ldots c_k\mathbf{b}_k \qquad (1)$$

So, for PCA, the inverse function is defined for all possible points in the coefficient space:

$$f^{-1}_{\mathsf{PCA}} : \mathcal{R}^k \longrightarrow R^n$$

These properties of PCA (a linear dimensionality reduction technique) highlight the benefits and drawbacks of non-linear dimensionality reduction. One major problem with techniques such as Isomap is that the mapping function f is only defined for the original input data set $\mathcal{I}$, and it is complicated to compute the "out of sample" projections for new images that are not in the set I [1]. In fact, that adding one new image to the input data set requires re-computing the mapping for all the images, and may make very large global changes in the projection of every image of the set. Even under the assumption that projection of the original images does not change, projecting a new image requires computing the distance to every original image.

To attack this problem, two new methods have been announced that offer continuous mappings between the coefficient space and the original (in our case image) space: Automatic Alignment [5] combines LLE with a set of pre-estimated local dimensionality reducers each of which is presumed to be fitted to a relatively flat subset of the manifold, and solves for a mixture of these projections that globally flattens the data while minimizing barycentric distortion in each point neighborhood. Charting [2] solves for a kernel-based mixture of projections that minimizes Euclidean distortion of local neighborhoods; it includes a solution for the local dimensionality reducers needed by automatic alignment.

While these methods define a smooth transformation between the image space and the coefficient space, they are ill suited for many image analysis applications because they still assume that the image manifold is locally *linear*. Locally, the inverse function has the form of Equation (1), (although the basis functions b_i may vary for different points in the coefficient space), so differential changes to the coefficients lead to changes in weights of the linear basis functions. Consider an image $\mathbf{x}$ with corresponding coefficients (c_1, c_2, … c_k). The partial derivative of the inverse mapping function (Equation 1) describes how the image varies when changing the c_1 coefficient:

$$\frac{\partial}{dc_1} f^{-1}_{\mathsf{PCA}}(c_1, c_2, \ldots c_k) = \mathbf{b}_1$$

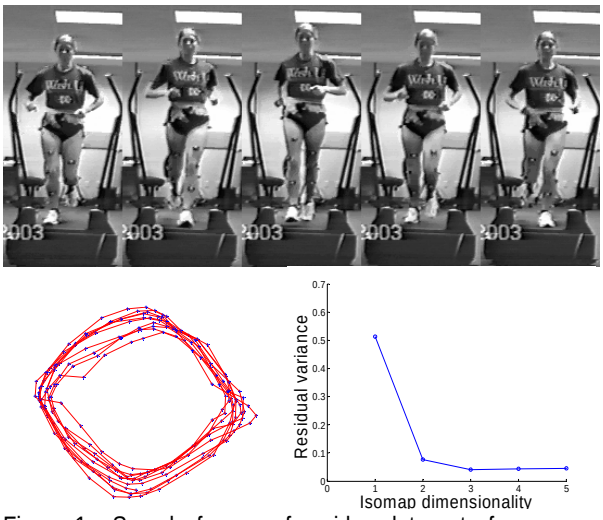

Figure 1. Sample frames of a video data set of a woman running on a treadmill. Bottom left, the two dimensional non-linear embedding of this data set (Using Isomap with distance defined by sum of the squared pixel intensity difference, and using 8 closest neighbors). Each blue dot is the (non-linear) projection of an original image, the red line connects the points from consecutive frames. The axes are irrelevant, as any Euclidean transformation of the points would have the same relative distances. Bottom right, plot of residual error shows that two dimensions capture almost all of the information in these local distance measurements.

Alternatively, moving through the coefficient space can be interpretted as an operator: changing coefficient c_1 by ε, changes the image **x** by the addition of part of the $\mathbf{b_1}$ basis image:

$$x' = x + \epsilon b_1$$

But this is not, usually, an interesting change. For image sets defined by images of objects taken from different poses, even locally images are not linear combinations of basis functions. Instead there are changes caused by non-rigid deformation of the object, and changes caused by relative motion of the object and camera, which may cause a family of relevant image warping functions, translations, rotations, homographies, and intensity variations such as gamma corrections, as well as actual deformations of the object itself.

Our goal, then, is to specialize Isomap, use the fact that the points in our high dimensional space are images, and find mapping functions f so that motion along an axis in the coefficient space is either an image warping transformation or a non-rigid deformation of the object.

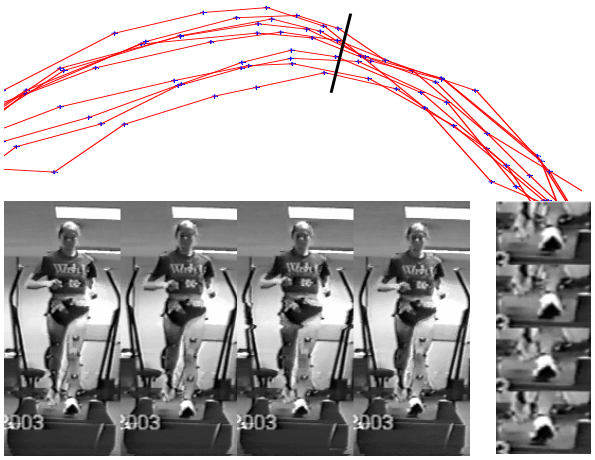

Figure 2. A expanded view of the top of the trajectory shown in Figure 1. The radial variation show images takes at the same part of the running cycle. The dominant variation here is translation to the left, and can be seen most clearly in the zoomed in view of the feet shown at the right.

3. Relevant Parameterizations

The intuition for this project is observed in the Isomap embedding of image set of a woman running on a treadmill. Figure 1 gives sample frames from the video, and the Isomap embedding. The cyclic nature of the running motion leads Isomap to embed the points in a circle. The "thickness" of the circle arises from the variation in the image appearance for images taken at the same part of the running cycle.

Note that the Isomap embedding has *already* separated the image variations into two components. Tangential motion in the coefficient space (moving around the circle) corresponds to changing what part of the running cycle the image depicts. Radial motion in the coefficient space by other changes --- variation in the stride --- in this sequence the dominant change is the left to right position of the runner on the treadmill. Figure 2 shows an expanded view of the Isomap embedding, and the image set generated by moving radially through the coefficient space. The displayed images are all original samples, and they related by translating the runner relative to the background. This visualization of the data set is, in itself, a useful diagnostic tool, but directly applying this operator to the image would require segmentation of the runner from the background.

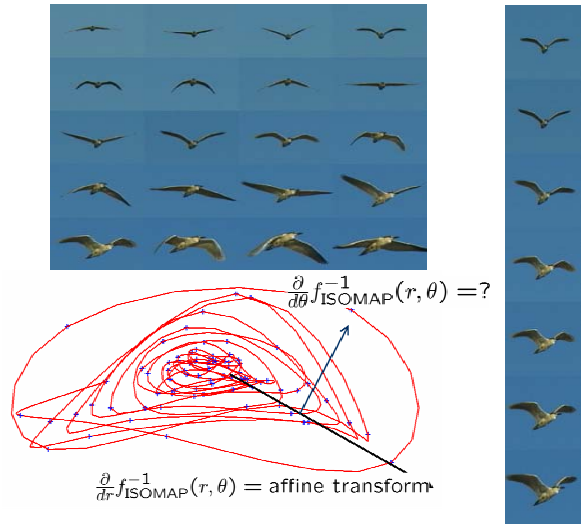

Figure 3. Every fourth image of a video sequence of a bird flying across the sky. The bottom left shows the Isomap embedding of this set of images. Moving radially in the embedding corresponds, locally, to an affine transformation of the image that depends only on the relative position of the bird from the camera. The transform required to move tangentially in the Isomap space would vary by location and require a motion model of the bird. At the right are the seven images closest to the dark radial arrow, Figure 4 and the text describe the process of finding the affine transformation.

Our second example uses a data set of a flying bird, taken against a featureless blue background. In this case, the data set is essentially a binary valued image, and the bird image can be represented as the set of image points that fall inside the silhouette. Isomap is performed on this data set using the symmetric Hausdorf distance and the eight nearest neighbors. This gives the embedding shown in Figure 3. As in the running video, there is a circular motion in the trajectory caused by the cyclic nature of the data. However, there is also a consistent radial motion, caused by image differences that arise from the approach of the bird toward the camera. That is, the image changes embedded as radial motions in the Isomap space results from *only* the approach of the bird to the camera, not the changing shape of the bird as it flies. Thus the Isomap embedding automatically de-couples the non-rigid component of the bird motion from the rigid component of the bird approaching the camera!

The right side of Figure 3 shows the images closest to a radial line in the Isomap embedding. These images appear to be related by a rigid transformation, but lacking a 3D model, a convenient set of transformations is linear coordinate transforms (image warping functions). In order to decouple the deformable motion of the bird from other variations in appearance, we will modify the distance function to ignore variation caused by anything other than the deformable motion. As before, we simplify this process by defining the image as a set of point coordinates that are not blue. Then the affine invariant distance function returns the following distance:

$$D(P,Q) = min_A \begin{matrix} \sum_i min_j ||P_i - AQ_j||_2^2 + \\ \sum_j min_i ||P_i - AQ_j||_2^2 \end{matrix}$$

A potential affine transformation A (a 3 by 2 matrix) warps the positions of the points in point set Q, this affine invariant distance measure returns the distance between P and the most similar affine transform of Q. Figure 4 shows that using this distance measure allows differentiation of deformable motion even in the more confused area at the beginning of the sequence. Futhermore, the solution for the best fitting affine matrix A between two images offers an image warping operator for interpolating between images that is an alternative to a weighted sum of these images.

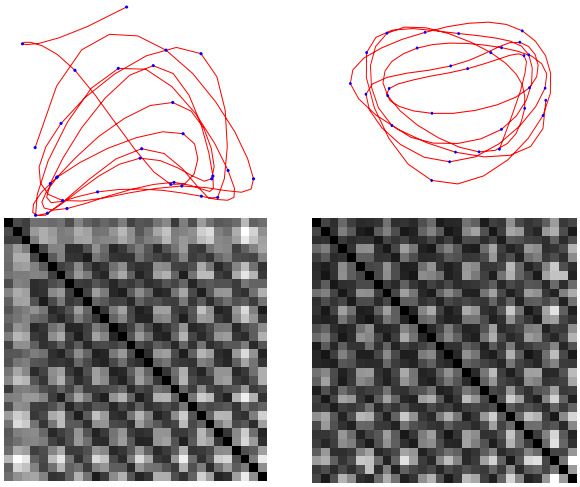

Figure 4. The affine invariant distance measure effectively decouples the image variation at the beginning of the bird video sequence. On the left is the Isomap embedding and the distance matrix for the first 30 frames of the sequence. One the right is the Isomap embedding using the affine invariant distance matrix. This distance matrix more closely resembles an exactly periodic function, the Isomap embedding more cleanly maps non-rigid deformations of the bird to tangential motions.

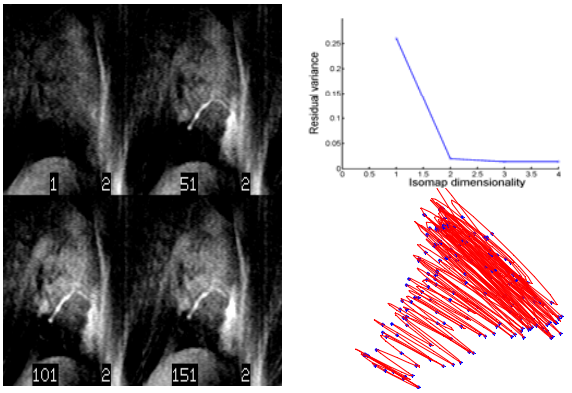

Figure 4. Four samples of a sequence of MRI images, and the associated Isomap embedding (using 8 neighbors, and sum of squared pixel intensity difference as a local distance measure). The plot of Isomap dimensional versus residual error indicates that 2 dimensions suffice to capture most of the distance information. The red line connects the image in order,

4. Application to MRI de-Noising

An increasingly important application domain is the analysis of MRI data. MRI data is typified by large data sets which are often noisy. An image of the same subject may vary for a number of reasons, including: including noise inherent in the sensor itself, motion of the subject during data capture, and time varying effect of contrast agents that are used to highlight particular types of tissue. The analysis of MRI data would be greatly improved with automatic techniques for image registration.

The direct application of Isomap to a particular MRI image set is shown in Figure 4. This image set is a "held breath" MRI of a heart. In this experimental design, the patient is asked to hold their breath, and the MRI pulses are triggered at the same point in consecutive heart beats until enough pulses are captured to reconstruct an image. Each image shown in Figure 4 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. First, the patient does not always hold their breath in exactly the same position, so between images there is variation in the position of the heart and liver (visible at the bottom of the images). Second, the contrast agent is slowly permeating through the tissues in view. Third, the MRI images themselves have noise.

The Isomap embedding of this data set is shown at the bottom right of Figure 4 (Isomap was run using sum of the squared pixel intensity differences, and 8 nearest images were used as neighbors for each image). The red line connects the images in the temporal order they were taken. Following the trend of this path roughly corresponds to the effect of the contrast agent permeating through the membrane. If we can ignore the effect of the contrast agent, then the remaining variability is due to the position of the organs in the image. We consider the local change in the effect of the contrast-agent to remap pixel intensity values that are expected to be in the range [0,1]: $I(x,y) = I^{\gamma}(x,y)$. Then, we can define the contrast-agent invariant function to be:

$$D(I_1, I_2) = min_{\gamma} \sum_{x,y} ||I_1(x,y) - I_2^{\gamma}(x,y)||_2^2$$

Although this is clearly a naïve function for many reasons (it is not based upon a physical model of the contrast agent dynamics, it does not account for the fact that only the parts of the image where the contrast agent has permeated should be affected and so on), it give a reasonable local approximation to the variation caused by change in the contrast agent. If we re-map the images using this distance function, the parameterization of the images will be independent of the *local* changes in contrast.

This Isomap embedding has been plotted at the top of Figure 5. The x-axis has been manually stretched out to improve the understandability of the figure, but no other changes are made. The embedding retains two degrees of freedom, the global permeation of the contrast agent through the heart region which is very directly encoded in the x-axis, and the changing position of the heart and liver as caused by the patient motion. If we project our data set onto the x and y axis, we can see the two dominant degrees of freedom. In particular, images whose projection onto the y-axis is similar taken when the positional changes are minimized. This allows us to average these images together with essentially *no* spatial blurring. An average of 10 images nearby on the y-axis is show at the bottom of figure 5.

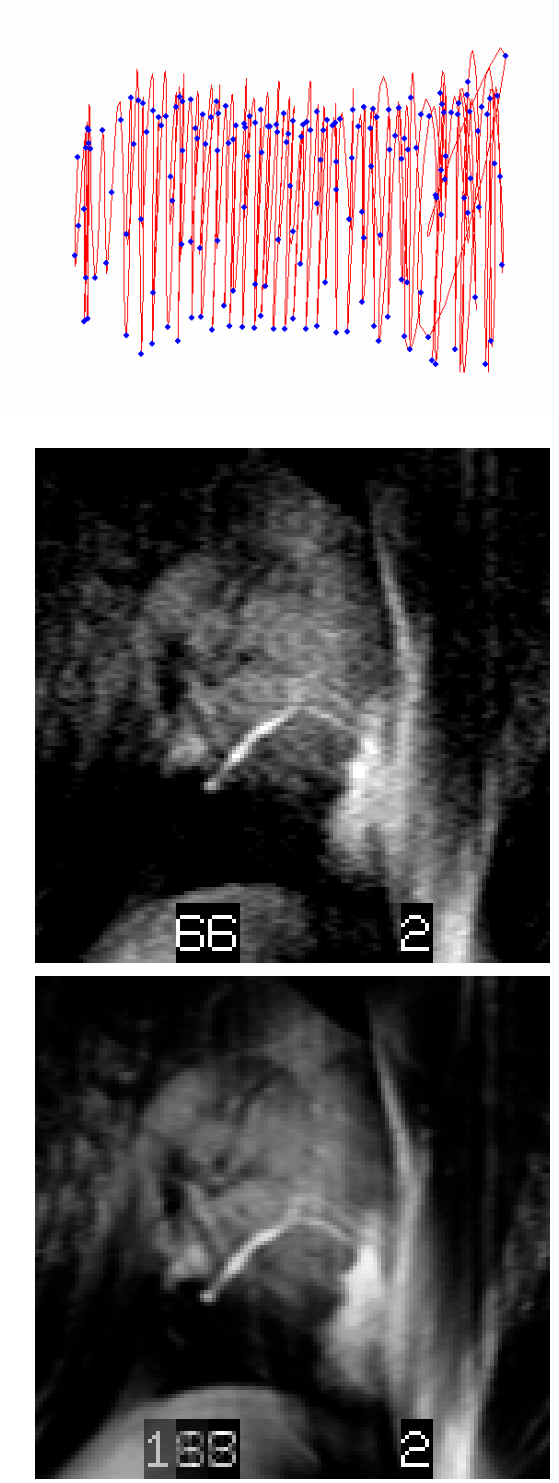

Figure 5. Using a gamma-invariant distance measure, the Isomap embedding aligns itself with two concrete degrees of freedom. Since the y-component encodes shifts in the image, averaging images with similar y-component does not result in spatial blurring, but does minimize pixel noise in individual images.

Several concluding thoughts are in order. First, is that 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 that images have, because image data sets are (even locally) almost never linear sums of other images. Finally, a small set of image transformation primitives gives powerful tools for registration of many different kinds of data sets.

10. References

[1] Y. Bengio, J-F. Paiement, and P. Vincent. ”Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering”, NIPS 2003.

[2] M. Brand, 2003. Charting a manifold. NIPS 2003.

[3] I. S. Lim, P. H. Ciechomski, S. Sarni, D. Thalmann, “Planar Arrangement of High-dimensional Biomedical Data Sets by Isomap Coordinates”, Proceedings of the 16th IEEE Symposium on Computer-Based Medical Systems (CBMS 2003), June 26--27, 2003, New York

[4] S. Roweis and L. Saul, “Nonlinear Dimensionality Reduction by Locally Linear Embedding”, *Science*, 290; 2323-2326, December 2000.

[5] Y. W. Teh,, AND S. Roweis, S. T. “Automatic alignment of hidden representations”. NIPS 2003.

[6] J. Tenebaum, V. Silva, and J. Langford, “A Global Geometric Framework for Nonlinear Dimensionality Reduction”, *Science*, 290; 2319-2323, December, 2000.

[7] W. Matusik, W., H. Pfister, M. Brand, and L McMillan, “A Data-Driven Reflectance Model.” In *Proceedings of SIGGRAPH 2003*.

[8] R. Pless and I. Simon, "Using Thousands of Images of an Object", *Computer Vision, Pattern Recognition and Image Processing*, 2002.

[9] Q. Wang, G. Xu, H. Ai, “Learning Object Intrinsic Structure for Robust Visual Tracking”, CVPR, 2003.