

LOCAL QUADRATURE RECONSTRUCTION ON SMOOTH MANIFOLDS

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Technology

by

Bhuwan Dhingra

Y8127167



to the

DEPARTMENT OF ELECTRICAL ENGINEERING

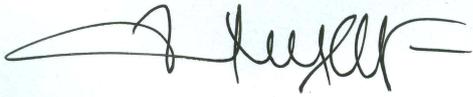
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

July 2013

CERTIFICATE

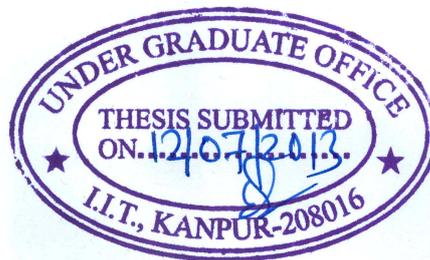
It is certified that the work contained in the thesis entitled "*Local Quadrature Reconstruction on Smooth Manifolds*" by *Bhuvan Dhingra* has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

July 2013



Prof. Amitabha Mukerjee
Department of Computer Science,
Indian Institute of Technology,
Kanpur-208016.

Prof. KS Venkatesh
Department of Electrical Engineering,
Indian Institute of Technology,
Kanpur-208016.



Abstract

Non-Linear Dimensionality Reduction (NLDR) techniques such as ISOMAP, LLE, Laplacian Eigenmaps etc. attempt to estimate low-dimensional latent descriptors for data assumed to be drawn from an m -dimensional manifold in an ambient n -dimensional space. Out-of-Sample Extension - the problem of estimating the latent vectors for novel data - has attracted considerable attention in the literature. In this thesis, we consider the opposite problem, that of reconstructing new high-dimensional points, given a novel latent-vector in a previously discovered embedding. Such a procedure finds relevance in applications such as video interpolation or robot motion planning.

Some global methods can be applied to the problem, but these are polynomials on the total number of data points N resulting in a complexity of $O(N^3)$, where N is often in the thousands. In contrast, we propose a Local Quadrature Reconstruction approach that looks at only the local k -neighbourhood for which the complexity reduces to $O(k^3)$ (k may be about 10). LQR achieves low error by estimating the second order error terms based on a second order differential geometric formulation for a small neighbourhood around the query point on the manifold. Main features of LQR include its fast reconstruction time and lack of a training phase, but since k increases as $O(m^2)$ it is currently limited to manifolds with low intrinsic dimension. Performance analysis of LQR on several point and image sets is presented, and a possible application for frame interpolation on videos is also studied.

*To my parents and sister for their love and
support.*

Acknowledgements

I would like to take this opportunity to thank several people for their invaluable help during the course of this thesis. First of all I would like to thank both my supervisors Prof Mukerjee and Prof Venkatesh for their unmitigated support for the last 1.5 years. They did a great job of motivating me towards both the current problem and research in general.

I would also like to thank Shish Basu Palit for insightful discussions on some of the more mathematical aspects of this thesis which escaped over my head. I thank my family for their constant love and support of my ambitions. Finally I would like to thank all my friends at IIT Kanpur who ensured an amazing stay for me here.

Contents

List of Figures	8
1 Manifolds and Dimensionality Reduction	1
1.1 Introduction	1
1.1.1 Notation	4
1.1.2 Organisation	4
1.2 Point Set Topology, Smooth Manifolds and Tangent Spaces	5
1.3 Dimension Reduction	8
1.4 Extension and Reconstruction	13
1.5 Quadrature Embeddings of Smooth Manifolds	16
2 Local Quadrature Reconstruction	20
2.1 Overview	20
2.2 The Algorithm	23
2.2.1 Motivation	23
2.2.2 Tangent space estimation	26
2.2.3 Linear Regression	27
2.2.4 Quadratic Regression	30
2.3 Complexity	34
2.4 Parameter Selection	35
3 Experiments and Results	37
3.1 Point Sets	38

3.2	Image Sets	38
3.2.1	Rotating Teapot	41
3.2.2	Disk-Shaped Planar Robot	45
3.2.3	Planar Articulated Robot Arm	51
3.3	Video Frame Reconstruction	52
4	Conclusions and Discussion	55
4.1	Future Work	57
	References	58

List of Figures

1.1	Non-Linear Dimensionality Reduction	9
1.2	Principal Curvatures of a surface in \mathbb{R}^3	17
2.1	m -dimensional manifold in \mathbb{R}^n	21
2.2	ϵ -neighbourhood of a point on the manifold \mathcal{M}	22
2.3	Tangent space on the manifold	23
2.4	Tangent Space Estimation	28
2.5	Linear Regression on the Tangent Space	29
2.6	Quadratic Regression	32
2.7	Local Quadrature v Linear Reconstruction on the Swiss Roll	34
3.1	LQR on a Spiral	39
3.2	Local Quadrature v Linear Reconstruction on the Spiral	40
3.3	Extrapolation v Interpolation on a 1-d manifold	40
3.4	ISOMAP on teapot dataset	41
3.5	1-D Isomap embedding for teapot dataset	42
3.6	Neighbourhood selection for teapot dataset	43
3.7	LQR on Teapot dataset	44
3.8	Local Quadrature v Linear Reconstruction on Teapot dataset	45
3.9	Squared Error v Angle of Rotation on Teapot dataset	46
3.10	2-dimensional disk dataset	47
3.11	LQR on the Disk dataset	48
3.12	Effect of Normal Space components	49

3.13 Local Quadrature v Linear Reconstruction on Disk dataset	49
3.14 LQR failures on Disk dataset	50
3.15 Error Variation on Disk Dataset	50
3.16 Reconstructions on Robot-Arm Dataset	51
3.17 Frame reconstruction on Foreman Sequence	53

Chapter 1

Manifolds and Dimensionality Reduction

1.1 Introduction

With the manufacture of exceedingly cheaper sensors in the last few decades, many real world applications work with high-dimensional data such as high-resolution images, speech signals, biomedical data etc. However, many of these sets have only a few underlying modes of variability or *degrees of freedom* and hence lie on or near a smooth low-dimensional manifold embedded in a high-dimensional ambient space. For example, consider a set of N images, each of size $p \times q$, of a disk in a plane with the disk taking an arbitrary position in each image (fig. 3.10). Here each image lies in \mathbb{R}^{pq} (since there are pq pixels), but the dataset as a whole has only 2 degrees of freedom - the image coordinates of the center of the disk. Hence, we say that these images are sampled (with noise) from a 2-dimensional manifold in \mathbb{R}^{pq} . *Manifold Learning* is the process of learning the structure of such manifolds from a set of sampled points. *Dimension Reduction* is also closely related to manifold learning and attempts to recover a low-dimensional representation of the data such that some geometrical or topological properties (such as pairwise distances) between the points are preserved. These low-dimensional codes are called the latent vectors of the high-dimensional data and can be used for data visualization and feature extraction, but as a tool for efficient data storage a

satisfactory reconstruction method is required which projects them back to the manifold in the ambient space. In this thesis we study the problem of *Out-of-Sample Reconstruction* - given a set of points sampled from an m -dimensional manifold \mathcal{M} in \mathbb{R}^n and their latent vectors in \mathbb{R}^m , find the inverse image \mathbf{x}_q on the manifold for a new out-of-sample latent vector $\mathbf{y}_q \in \mathbb{R}^m$. In simpler terms, we want to learn the non-linear function which takes points from the latent space to the manifold from a few known samples. Such a method can then be used for frame interpolation and reconstruction in the image domain, or for generating novel views of an object, or in robot motion planning for testing if a path is collision-free.

Principal Components Analysis (PCA) is the simplest dimension reduction technique which projects the data onto an m -dimensional linear subspace in the direction of maximum data variance. It provides a straightforward out-of-sample reconstruction method - since we have the basis vectors of the linear subspace we have the explicit mapping which takes latent vectors to the manifold. However, PCA fails to provide accurate embeddings/reconstruction for manifolds which are not isometric to a plane. Non-linear techniques are able to find good low-dimensional representations of arbitrary smooth manifolds, but don't provide a mapping between the two spaces for reconstruction. One fast and simple approach [1] for reconstruction is to express \mathbf{y}_q as the weighted combination of its known neighbours $\mathbf{y}_i \in \mathbb{R}^m$, and using the same weights to construct \mathbf{x}_q from the same points $\mathbf{x}_i \in \mathbb{R}^n$. This is equivalent to fitting a plane over a small neighbourhood on the manifold, and linearly interpolating the new point on it. Unless the sampling density is high or the manifold curvature low this assumption of linearity generally does not work well. Specifically, for image manifolds linearly interpolated reconstructed images have a high amount of blurring.

More sophisticated procedures have been suggested for general manifolds as well as image manifolds in [2] and [3]. However, both these methods rely on learning global functions in terms of the latent vectors using regression, and involve a computationally heavy training step. A natural way to reduce the computation time for these algorithms is to use only a small neighbourhood around the query point instead of the entire manifold. Specifically, the question we want to address is - can we improve over simple linear interpolation by

performing higher order regression on a local patch on the manifold?

The contribution of this work is an out-of-sample reconstruction technique called **Local Quadrature Reconstruction (LQR)**. We apply a differential geometric model, called a *Quadrature Embedding*, over a small neighbourhood on the manifold. Regressing over the known points in the neighbourhood we can generalize the model to unseen points and use it for reconstruction. A general second order model of the data has mn unknown parameters, which can be prohibitively high for some situations. These parameters represent the effect of curvature of the manifold in the neighbourhood under consideration. Instead of approximating all the curvatures along all dimensions, we find a basis set along which variance of the data is maximum using PCA and retain only $m + d$ components. If the sampling density is sufficiently high, we can assume that the first m components span the tangent space at \mathbf{p} and the next d are the normal to the tangent plane and describe the effect of curvature around \mathbf{p} . To summarize, suppose we are given a query point \mathbf{y}_q and let \mathbf{p} be its nearest neighbour's image on the manifold. We first find \mathbf{z}_q the projection of the desired reconstruction along the tangent space at \mathbf{p} using linear interpolation. Next, we estimate the principal curvatures $\kappa_1, \kappa_2, \dots, \kappa_m$ along each of the d normal components by regressing with the tangent space projections as the independent variables. We need a total of $k > 2m + \binom{m}{2} + 1$ points in the neighbourhood to solve the regression and computational complexity of the algorithm is $O(k^3 + kNm + n(m + d))$ which is comparable to simple linear interpolation.

The main feature of LQR is its computation speed, and the lack of the requirement of a separate training procedure. Its limitation is that the minimum number of neighbourhood points required increases as m^2 due to which it does not generalize well to manifolds with high intrinsic dimension. We show the performance of Local Quadrature Reconstruction on both point sets (Swiss Roll, Spiral) and image sets (Teapot, Disk, Robot-Arm). Most reconstructions show an improvement over simple linear interpolation, and in each dataset the Mean Squared Error (MSE) of LQR is lower than the MSE of linear reconstructions. At the same time, computation time remains comparable for both, and more importantly does not scale only linearly with number of points N , and ambient space dimension n . A natural applica-

tion for out-of-sample reconstruction is in video compression and frame interpolation, and we show the performance of LQR for interpolating the frames of the *Foreman* video sequence.

1.1.1 Notation

Vectors are denoted in bold-face alphabets \mathbf{x} , with subscripts \mathbf{x}_i indicating indexing and superscripts \mathbf{x}^j components. l_p -norm of a vector $\mathbf{x} \in \mathbb{R}^n$ is given by $\|\mathbf{x}\|_p = (\sum_{j=1}^n |x^j|^p)^{1/p}$. Capital alphabets X denote matrices or sets of points. Canonical vectors in \mathbb{R}^n are written as e_i for $i = 1, 2, \dots, n$, where e_i is 1 at the i^{th} position and 0 elsewhere.

We use \mathcal{M} to denote an m -dimensional manifold embedded in an n -dimensional euclidean space \mathbb{R}^n . Some manifolds can be endowed with a global m -dimensional coordinate system. These coordinates lie on a submanifold $\mathcal{S} \subset \mathbb{R}^m$. Throughout this work we use $N_{\mathbf{a}}^k$ to denote the k -nearest neighbours of \mathbf{a} . Unless otherwise specified, these neighbours are found from the set to which \mathbf{a} implicitly belongs. Gradient of a function is written as ∇f , and the hessian matrix as $\nabla^2 f$, whose elements are given by $[\nabla^2 f]_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$.

1.1.2 Organisation

The rest of this thesis is organised as follows. In the remaining part of this chapter we give an introduction to the theory of manifolds, and a brief literature review on dimensionality reduction and existing extension/reconstruction methods. We also introduce a differential geometric model on the manifold we use in the development of our algorithm. In chapter 2 we motivate and describe the main algorithm for out-of-sample reconstruction, and illustrate it on the swiss-roll manifold. Chapter 3 presents further experiments on point sets and image sets, and an example for video frame interpolation. We conclude with a discussion in chapter 4.

1.2 Point Set Topology, Smooth Manifolds and Tangent Spaces

We start with a brief overview of the theory of manifolds. For a more detailed study the reader is referred to [4] or [5]. Since we draw many concepts from point set topology we first introduce some definitions from there.

Definition 1. A **Topology** is a family of sets \mathcal{U} s.t they are closed under finite intersection and arbitrary unions.

The above definition implies that $X = \bigcup_{U \in \mathcal{U}} U$ and ϕ (empty set) are automatically members of \mathcal{U} , and we call \mathcal{U} a topology on X . Also the pair (X, \mathcal{U}) is called a *topological space*, and the members of \mathcal{U} are called the *open sets* of X w.r.t to the topology \mathcal{U} .

Definition 2. A **Continuous Map** between two topological spaces is a function $f : (X, \mathcal{U}) \rightarrow (Y, \mathcal{V})$ s.t. for every $V \in \mathcal{V}$ its inverse image $f^{-1}(V) = \{\mathbf{x} \in X | f(\mathbf{x}) \in V\}$ is in \mathcal{U} .

Basically, the map f is continuous if the inverse images of open sets are open. If a continuous map also has a continuous inverse $g : (Y, \mathcal{V}) \rightarrow (X, \mathcal{U})$ then it is called a *homeomorphism*. Finally, a *subspace* of (X, \mathcal{U}) is a subset $A \subset X$ whose topology is given by $\{A \cap U | U \in \mathcal{U}\}$.

The concept of manifolds is best motivated by the real world example of the earth we live on. Even though locally everywhere it seems to be flat, or in mathematical terms like a 2D plane, we know that as a whole it is round, or a 3D sphere. Also, we can represent the earth with a collection (an *atlas*) of several overlapping maps on paper each describing a local region, but there doesn't exist a single global 2D map which can represent it all together. Similarly, a manifold is also a hypersurface in a high-dimensional space which is locally homeomorphic to a lower-dimensional space.

Definition 3. An m -dimensional topological **Manifold** \mathcal{M} is defined as a Hausdorff topological space with a countable basis for the topology which is locally homeomorphic to \mathbb{R}^m .

The last condition above implies that for any point $\mathbf{p} \in \mathcal{M}$ there is an open set U containing \mathbf{p} which is homeomorphic to \mathbb{R}^m . The other two conditions, that M should be Hausdorff and have a countable basis are there to remove some special curves which are locally homeomorphic to \mathbb{R}^m but very wierd [4]. In this work we are concerned only with manifolds embedded in euclidean spaces which fortunately satisfy these conditions.

A chart on a manifold is conceptually the same as a local map of the earth - it defines a coordinate system on a local patch on the manifold. The concept of charts allow the generalization of many concepts from ordinary calculus to manifolds. More precisely,

Definition 4. A **Chart** (U, ϕ) on an m -dimensional manifold \mathcal{M} is a homeomorphism $\phi : U \rightarrow U'$ from an open subset U of M to an open subset U' in \mathbb{R}^m .

A collection of charts (U_α, ϕ_α) on \mathcal{M} s.t. $\bigcup U_\alpha = \mathcal{M}$ is called an *atlas* on \mathcal{M} . This is analogous to the the conventional atlas which has a collection of overlapping maps of all countries to give a representation of the entire globe.

Next we extend the notion of differentiability to manifolds. Suppose, $\phi_1 : U_1 \rightarrow U'_1$ and $\phi_2 : U_2 \rightarrow U'_2$ be two charts on \mathcal{M} and $U_{12} = U_1 \cap U_2$ is their intersection. Then a *chart transformation* $\phi_{12} : \phi_1(U_{12}) \rightarrow \phi_2(U_{12})$ is the mapping which takes points from the range of one chart to the range of the other chart.

Definition 5. A **Smooth Atlas** on a manifold \mathcal{M} is an atlas for which all the chart transformations are smooth.

Such an atlas on a manifold is also called a *smooth structure*, and a manifold equipped with a smooth structure is called a *smooth manifold*. We use $(\mathcal{M}, \mathcal{A})$ to denote a smooth manifold \mathcal{M} equipped with a smooth structure \mathcal{A} . We have already defined continuous maps between manifolds. Such maps are smooth as well if their composition with the charts on the manifolds is smooth:

Definition 6. A map $f : M \rightarrow N$ between two manifolds is called smooth at a point \mathbf{p} on \mathcal{M} if the composition $\varphi \circ f \circ \phi$, where φ and ϕ are charts around \mathbf{p} and $f(\mathbf{p})$, is smooth. If f is smooth at all points on \mathcal{M} then it is called a **Smooth Map**.

If the smooth map f has a smooth inverse as well, then it is called a *diffeomorphism*, and two manifolds are called *diffeomorphic* if there exists a diffeomorphism between them.

The idea behind tangent space at a point on a manifold is intuitive enough - it is the set of all possible directions along which one can move through that point tangentially. However, defining it is a little tricky since we have to do it intrinsically on the manifold itself. Unlike conventional definitions in geometry, here we cannot use the ambient space around. We have to define the tangent space in terms of the manifold itself and charts on it.

Definition 7. Consider two curves $\gamma_1 : \mathbb{R} \rightarrow \mathcal{M}$ and $\gamma_2 : \mathbb{R} \rightarrow \mathcal{M}$ s.t. $\gamma_1(0) = \gamma_2(0) = \mathbf{p}$ on the manifold, and let $\phi : U \rightarrow \mathbb{R}^m$ be a chart containing \mathbf{p} . Then γ_1 and γ_2 are called equivalent if the compositions $\phi \circ \gamma_1$ and $\phi \circ \gamma_2$ are differentiable at 0 and their derivatives at 0 coincide. The equivalence classes defined by such an equivalence relation are called the **Tangent Vectors** at \mathbf{p} and the set of all the tangent vectors is called the **Tangent Space** at \mathbf{p} denoted by $T_{\mathbf{p}}\mathcal{M}$.

The above definition seems to be dependent on the particular choice of the chart ϕ , but it is in fact not and the tangent spaces defined by different charts at the same point coincide. A formal definition of the tangent space in terms of function germs can be found in [4]. Furthermore, $T_{\mathbf{p}}\mathcal{M}$ has a vector space structure and maps between two tangent spaces are linear.

Tangent spaces help us *linearize* a manifold, i.e. to each point on the manifold it associates a linear subspace. However, for a proper extension of some geometric concepts to manifolds we still need the concept of a metric on the manifold. This is achieved by defining **Riemannian Manifolds**. A Riemannian manifold is a smooth manifold \mathcal{M} equipped with an inner product $g_{\mathbf{p}}$ over the tangent space $T_{\mathbf{p}}\mathcal{M}$ at each point \mathbf{p} s.t. this inner product varies smoothly from point to point. The family of inner products $g_{\mathbf{p}}$ is called a **Riemannian**

Metric. Submanifolds of euclidean spaces are automatically riemannian with the riemannian metric here being the scalar/dot product of vectors.

1.3 Dimension Reduction

There exist several real-world datasets where the data lies in a high-dimensional space - speech signals, image sets, fMRI data, videos. However many of these lie on or near a manifold with intrinsic dimensionality much less than the dimension of the space in which they are embedded. *Dimensionality Reduction* is the process of recovering a new representation of the high-dimensional data in a euclidean space whose dimension is equal to the intrinsic dimension of the manifold. The new representation can then be used for visualization or as features for further processing. A closely related problem is that of manifold learning which attempts to learn the structure of the manifold explicitly (in terms of tangent spaces or other manifold properties) [6]. Several methods have been proposed, both linear and non-linear, and [7], [8] give a comparative review detailing their performance and applicability on several real and artificial datasets. Here, we give a short description of some of the popular methods.

Fig. 1.1 gives an example of applying a Non-Linear Dimensionality Reduction (NLDR) method called ISOMAP on an image manifold. The images show a disk-shaped robot free to move about in a 2-d plane. The manifold has intrinsic dimension 2 and the 2-D embedding recovered from ISOMAP is shown. It is important to note that the relationship between the latent vectors and the images is highly non-linear. Observing the position of the disk in the image along the two axes shows the faithfulness of the recovered embedding.

Before we begin with a description of dimensionality reduction methods we would like to discuss the condition for which globally consistent sets of m -dimensional latent vectors of manifolds exist. We already know that an m -dimensional manifold is *locally* homeomorphic to an open subset of \mathbb{R}^m . If in addition it is also *globally* homeomorphic, then we can define a continuous mapping $\mathbf{c} : \mathcal{M} \rightarrow \mathbb{R}^m$, which gives the coordinates $\mathbf{c}(\mathbf{x})$ of any point on the

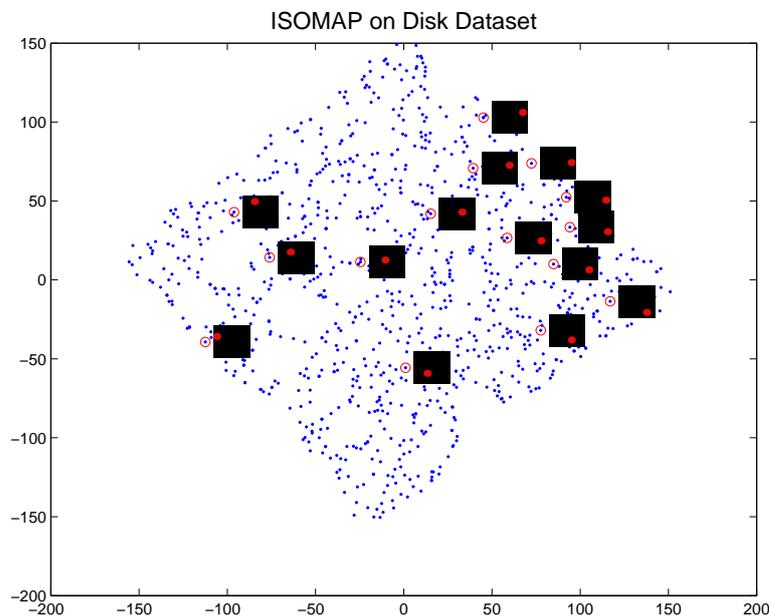


Figure 1.1: **Non-Linear Dimensionality Reduction (NLDR)**, Recovered 2-D ISOMAP embedding of *Disk* dataset consisting of images of a disk-shaped robot with 2 degrees of freedom. High-dimensional images are overlaid on top. Neighbourhood size $k = 8$

manifold. An example where this condition fails is that of a 3D sphere - it is a 2D manifold embedded in \mathbb{R}^3 , and any local patch on it is homeomorphic to \mathbb{R}^2 . But its not globally homeomorphic to \mathbb{R}^2 , and hence no continuous global 2-dimensional coordinate system can be assigned to points on the sphere. In this case dimensionality reduction methods which generally preserve inter-point distances also fail. A general work-around this problem is to cut-off a section of the manifold such that the resulting surface is homeomorphic to \mathbb{R}^m .

Broadly, dimensionality reduction methods can be classified as either linear or non-linear. Amongst linear methods, Principal Components Analysis (PCA) is the most popular, and if the data is actually sampled from a plane in \mathbb{R}^n then it can recover the plane optimally. It is a projective method, which means it finds a linear subspace on which the data is projected to give the low-dimensional representation. The subspace is found by an SVD on the centered covariance matrix of the data, and amounts to finding directions along which data variance is maximum. PCA has many desirable properties - simplicity of implementation, computation

speed, and a mapping between both the spaces which allows for easy extension and reconstruction. However, for many datasets no m -dimensional linear subspace exists which can describe all the variance in the data, and PCA ends up predicting a much higher value for the intrinsic dimensionality of the manifold than is actually the case. Other linear methods include - Independent Components Analysis (ICA), Canonical Components Analysis (CCA), Factor Analysis and Linear Discriminant Analysis (LDA), but all these fail for data which is not isometric to an m -dimensional hyperplane in \mathbb{R}^n .

Kernel-PCA (KPCA) [9] is a natural non-linear extension to PCA. It applies the kernel trick to create a non-linear version of PCA, i.e the data is first projected to a high-dimensional space where PCA is applied on it. The projection is achieved by implicitly mapping to a high dimensional feature space \mathcal{F} using $\phi : \mathbb{R}^N \rightarrow \mathcal{F}$ through a kernel function $K(\mathbf{x}_1, \mathbf{x}_2) = \langle \phi(\mathbf{x}_1) \cdot \phi(\mathbf{x}_2) \rangle$. The PCA method is modified so that it only requires inner products and then K is used to perform PCA in \mathcal{F} . This reduces to diagonalizing $\hat{K} = -PKP$ where $P = I_m - \frac{1}{m}\mathbf{1}_m$, normalizing the eigenvectors so that $\alpha_k \cdot \alpha_k = \frac{1}{\lambda_k}$ and extract principal components of any point \mathbf{x} by computing its projection onto the eigenvector. Using kernel PCA, it is possible to work in the space of all possible d^{th} order products between the data.

While PCA explicitly provides the mapping from the high-dimensional to the low-dimensional space and vice-versa, there is no such mapping for kernel PCA. Hence, to find the embedding of a new point on the manifold, we have to recompute all the eigenvectors, which can be computationally expensive. The Nystrom method gives a way of approximating the eigenvectors of a matrix $K \in M_n$, with $rank(K) = r \ll n$, from the eigenvectors of the smaller submatrix of K . As a result we can perform eigendecomposition of a $r \times r$ matrix rather than an $n \times n$ matrix. For an even faster out-of-sample computation, the approximate Nystrom method can be used in which no eigendecomposition needs to be performed in the testing phase (see [7] for details).

An important class of non-linear dimensionality reduction methods falls under the category of *spectral embedding* techniques. These methods compute the low-dimensional em-

bedding by finding eigenvectors of a specially constructed matrix. Examples include Multidimensional Scaling (MDS), ISometric MAPping (ISOMAP), Locally Linear Embedding (LLE), hessian LLE, Laplacian Eigenmaps etc. Some of these are discussed in detail below.

MDS builds a low-dimensional embedding of points such that their pairwise distances correspond to the pairwise dissimilarities in the high-dimensional data. The centering matrix P is used to convert the dissimilarity matrix A into a matrix of inner products $\hat{A} = -PAP$, whose gram vectors are the embeddings of the high dimensional points. These gram vectors can be found through an eigendecomposition of \hat{A} , and the results are the same as those found through PCA on the original coordinates of the data. Similar to KPCA, MDS also diagonalizes an inner product matrix, and hence the Nystrom method can be used to extend it to out-of-sample points and also to speed up the process (Landmark MDS). If euclidean distances are used as the pairwise dissimilarities between the high-dimensional points then the procedure is called Classical MDS.

ISOMAP [10] builds on classical MDS and attempts to preserve the intrinsic geometric structure of the data. It has been shown to give excellent performance on even highly non-linear manifolds and proceeds in three steps. In the first step the nearest neighbours of each point are determined, and a weighted graph G with points as the nodes and edges between neighbours is constructed. Second, the geodesic distances D_G between all pairs of points are computed using the shortest path metric between two points in the constructed graph. These set of distances are converted to dot-product form by applying an operator τ which double centres D_G^2 . Let λ_p be the p^{th} eigenvalue (in decreasing order) of the matrix $\tau(D_G)$, and \mathbf{v}_p^i be the i^{th} component of the p^{th} eigenvector. Then the p^{th} component of the embedding \mathbf{y}_i is equal to $\sqrt{\lambda_p} \mathbf{v}_p^i$. If the sampling on the Manifold is dense enough, the ISOMAP algorithm is asymptotically guaranteed to recover the true dimensionality and data structure of the Manifold. The biggest limitation of ISOMAP is that it cannot handle out of sample data for both the forward and reverse mappings.

LLE [1] is an unsupervised learning algorithm that computes low dimensional, neighbour-

hood preserving embeddings of high dimensional data. Each point is expressed as a weighted combination of its nearest neighbours, and these relationships are preserved in a global low dimensional coordinate system. The reconstruction weights can be computed in a closed form by solving a constrained least squares problem, and the embeddings are computed by an eigendecomposition. The weights can also be easily computed from the matrix of pairwise distances. Like ISOMAP, LLE is also able to recover good representations of data sampled from non-linear manifolds.

In [11] LLE, ISOMAP, Laplacian Eigenmaps and MDS are cast into a unified framework in which they are seen as learning the eigenfunctions of a kernel. The common algorithm for dimensionality reduction starts with an (optionally) transformed similarity matrix M . This similarity matrix is constructed from the data set D by $M_{ij} = K_D(\mathbf{x}_i, \mathbf{x}_j)$ where K_D is the kernel function which depends on the method of reduction and sometimes on the data set D . Then the embedding of each example \mathbf{x}_i is the vector \mathbf{y}_i with y_{ij} the i -th element of the j -th principal eigenvector \mathbf{v}_j of M or, in the case MDS and ISOMAP, the embedding is e_i with $e_{ij} = \sqrt{\lambda_j} y_{ij}$. The form of the kernel function K_D is also provided for each of the mentioned techniques. Hence, all spectral embedding methods can be seen as special cases of Kernel-PCA and Bengio et al. in [12] generalize the Nystrom method to these for an out-of-sample extension method.

Recently, deep learning has attracted a lot of interest from the machine learning community for its potential as an unsupervised learning tool. In [13] a nonlinear dimensionality reduction method based on multilayer encoder and decoder networks is presented. The combined system is called an autoencoder and facilitates both dimension reduction and reconstruction of the data. The required weights are adjusted by gradients obtained by backpropagating the errors through the two networks. However, unless the number of hidden layers is very small, a very good initial estimate of the weights is required for the iteration to converge to the global optimum. Hence, 'Restricted Boltzmann Machine (RBM)' is introduced which is a two-layer network whose input is the pixel data of the image, and output the first layer of features in the autoencoder. Features h_j are binary values set to 1 with probability

$\sigma(b_j + \sum_i v_i w_{ij})$, and pixels v_i are sampled from a gaussian with unit variance and mean $b_i + \sum_j h_j w_{ij}$. This network is trained independently of others by constructing the features and reconstructing the pixels from those features keeping the weights fixed. The final converged weights are the initial estimates for the autoencoder. Such weights are found for each layer of the autoencoder, and then fine-tuned using error backpropagation. Experimental results on image sets show its effectiveness for finding the embeddings as well reconstructing the high-dimensional data.

1.4 Extension and Reconstruction

Definition 8. *Given the high-dimensional data X and their corresponding latent vectors Y **Out-of-Sample Extension** is defined as finding the latent vector \mathbf{y}_q for a new point \mathbf{x}_q on the manifold without repeating the NLDR process. Similarly, **Out-of-Sample Reconstruction** is defined as finding the high dimensional counterpart \mathbf{x}_q for a new point \mathbf{y}_q in the latent space.*

As an example, suppose we are given the high-dimensional images and their latent vectors in fig. 1.1. Then given a new image sampled from the manifold, out-of-sample extension finds its 2-D latent vectors in the space shown, and given a new latent vector in the 2-D latent space, out-of-sample reconstruction finds the corresponding image in the image space.

Most non-linear manifold learning algorithms suffer from the limitation that they only find an embedding for the training data and not the explicit mappings between the two spaces in consideration, and hence do not allow extension or reconstruction of new points. Linear approaches, on the other hand, fail to preserve the intrinsic non-linear nature of many datasets. The nystrom method provides an extension for the spectral embedding techniques, but suffers from the limitation of being computationally expensive. The *approximate* nystrom method provides a step-around the computationally heavy eigendecomposition, but as a result the final embedding is not exact. Several other methods have since been proposed in the literature.

In [14] Trosset et al. provide an out-of-sample extension algorithm formulated as an unconstrained nonlinear least-squares problem. This allows an iterative computation of the exact out-of-sample embedding. Another iterative algorithm for out-of-sample extension of multiple points is presented in [15]. It starts with initial estimates for new points found through spline fitting, followed by computing the actual coordinates through an update rule. The estimates are recomputed using the updated coordinates and the whole procedure repeated. In [16], the authors develop a manifold learning algorithm which minimises an objective function with two terms - one for preserving the local structure of each neighbourhood in data as found by a simple linear PCA, and a second term for finding the global mapping to the low-dimensional space in a manner analogous to KPCA. Their construction also provides an explicit mapping for out-of-sample extension.

Our work shares some of its inspiration with [17], which computes tangent spaces locally to find extensions. *Generalised Out-Of-Sample Extension (GooSE)* is presented which computes a mapping between local neighbourhoods in the higher and lower dimensional spaces. For a new test point $\mathbf{x} \in \mathcal{M}$, its k -nearest neighbours $N_{\mathbf{x}}^k$ are found and PCA is performed on $N_{\mathbf{x}}^k \cup \mathbf{x}$ to find their tangent space projections Y' . Y' is assumed to be a scaled and rotated version of Y (i.e. $Y = BRY'$), and B and R are found using a least squares estimation on the k -nearest neighbours. These can be used to project \mathbf{x} to the target embedding. The parameter which needs to be selected is k the neighbourhood size, and this is determined empirically by finding the local minima as the algorithm is repeated for different values of k . GooSE is used only for extension however, and the authors don't talk about the reverse problem of reconstruction anywhere.

While there is an extensive literature on out-of-sample extension, very few algorithms exist which can reconstruct high-dimensional data from low-dimensional embeddings. This is expected since out-of-sample reconstruction is in general a hard problem if the higher dimension n is much greater than the lower dimension m . The simplest of these is presented in [18], which implements linear reconstruction of high-dimensional face images, by expressing points in the lower dimension as a weighted combination of their k nearest neighbours, and

forming the image from its neighbouring images using the same weights. Suppose, for a query point \mathbf{y}_q its k -nearest neighbours in Y are given by $N_{\mathbf{y}_q}^k$. Then we can find optimal weights w_i^* s.t. the following error is minimised [1]:

$$\mathcal{E}(W) = \|\mathbf{y}_q - \sum_{\mathbf{y}_i \in N_{\mathbf{y}_q}^k} w_i \mathbf{y}_i\|^2 \quad (1.1)$$

The optimal weights can be found by solving a least squares problem, and regularization is used to avoid unstable solutions. Once we have w_i^* the linear reconstruction is simply,

$$\mathbf{x}_q^{lin} = \sum_{i=1}^k w_i^* \mathbf{x}_i \quad (1.2)$$

For the densely sampled set, results are accurate, but as the sampling decreases the reconstructions become more blurred. Linear reconstruction has the desirable property of being restricted to a small neighbourhood on the manifold, which allows fast computation even at very high dimensions. Much better reconstructions can be obtained if this property is relaxed and all the points on the manifold are used. Linear reconstructions have two main steps - nearest neighbour search and inversion for regression, and hence its complexity is $O(kNm + k^3)$.

In [3], Souvenir et al. present a method for reconstructing images from their latent vectors, by learning a transformation for all points on the manifold. The Free-Form Deformation (FFD) model expresses the transformation from one image to another in terms of a lattice of control points and standard one-dimensional cubic splines. To facilitate compression and reconstruction these lattice control points are expressed in terms of the manifold coordinates, and linear interpolation is used to extend a reference image \mathbf{I}_{ref} to generate the image \mathbf{I}_q at unknown coordinates. Deformation fields are found by a global optimization, making the method computationally expensive.

Dollar, Rabaud and Belongie present a manifold learning technique *Locally Smooth Manifold Learning (LSML)* in [2] which learns a warping function from any point \mathbf{x} on the manifold to any point in a small neighbourhood around it. They make a first order approximation and

write $\mathcal{W}(\mathbf{x}, \boldsymbol{\epsilon}) = \mathbf{x} + \mathcal{H}(\mathbf{x})\boldsymbol{\epsilon}$, where \mathcal{W} is the warping, $\boldsymbol{\epsilon}$ is the m -dimensional displacement on the manifold, and the function \mathcal{H} captures the m modes of variability at \mathbf{x} . This is similar to finding the tangent vectors at \mathbf{x} , except here they need not be orthogonal. To solve for \mathcal{H} , first an estimate of the tangent space at each point on the manifold is found using PCA on k -nearest neighbours. Then linear regression is performed to combine these into a single global function. Both these steps require global computations making LSML computationally expensive. To ensure smoothness, radial basis features are computed over the data before regression. An image specific version is also given since the algorithm becomes intractable at very large n . The learned function \mathcal{H} can be used for dimensionality reduction with straightforward out-of-sample extension and reconstruction. Experiments on both point sets and image sets show LSML's ability to find accurate reconstructions in sparsely populated regions and even beyond the support of training data.

Besides the above methods, autoencoder networks also allow out-of-sample reconstruction, since during training a decoder network is also learned which propagates the latent vectors layer by layer to get the high-dimensional data. However, these networks have high training as well as storage complexities. In this work our aim is to use only local data and learn the local structure of the manifold to obtain accurate reconstructions. We build on the linear interpolation technique and give a second order approximation, which is empirically shown to reduce blurring.

1.5 Quadrature Embeddings of Smooth Manifolds

Let $N_{\mathbf{p}}(\epsilon)$ denote the ϵ -neighbourhood of a point \mathbf{p} on a manifold. We can write any point \mathbf{x} in this neighbourhood in terms of the principal curvatures at \mathbf{p} . For better elucidation, first consider a simple m -dimensional manifold \mathcal{M} embedded in \mathbb{R}^{m+1} . Then a neighbourhood $N_{\mathbf{p}}(\epsilon)$ on this manifold can be represented by a hypersurface of the form $S = \{[z_1 \ z_2 \ \dots \ z_m \ h(z_1, z_2, \dots, z_m)] : [z_1 \ z_2 \ \dots \ z_m] \in T_{\mathbf{p}}\mathcal{M}\} \subset \mathbb{R}^{m+1}$. Here $[z_1 \ z_2 \ \dots \ z_m]$ is the projection of \mathbf{x} onto the tangent plane $T_{\mathbf{p}}\mathcal{M}$, and h is in general a non-linear function

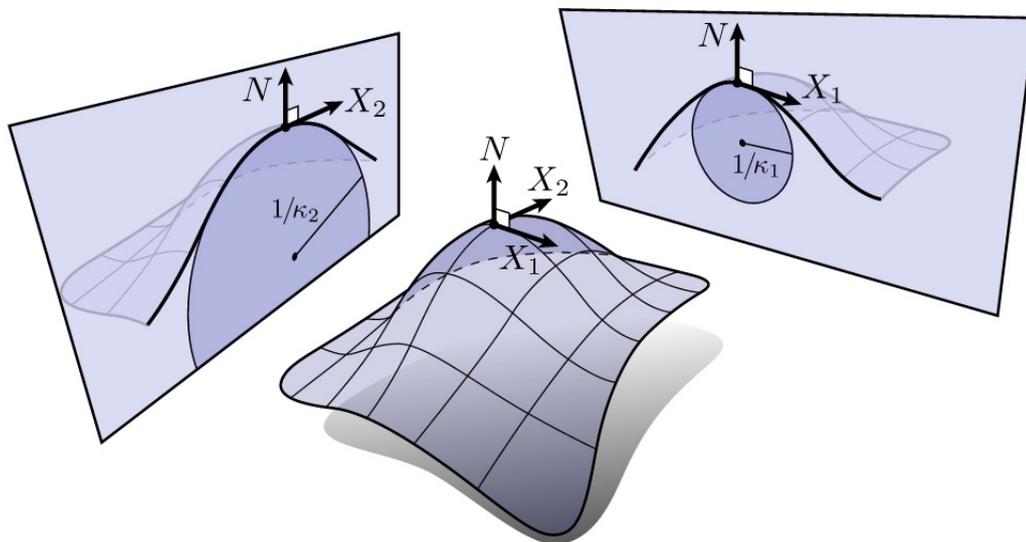


Figure 1.2: **Principal Curvatures of a surface in \mathbb{R}^3** , The normal vector N , and principal directions X_1 and X_2 at a point on the surface. κ_1 and κ_2 are the principal curvatures. Figure obtained from <http://brickisland.net/cs177/?p=144>.

describing the effect of curvature around \mathbf{p} . For a smooth manifold h is also smooth and we can write its Taylor series expansion ($\mathbf{z} = [z_1 \ z_2 \ \dots \ z_m]$):

$$h(\mathbf{z}) = h(\mathbf{p}) + \nabla h(\mathbf{p})^T(\mathbf{z} - \mathbf{p}) + \frac{1}{2}(\mathbf{z} - \mathbf{p})^T \nabla^2 h(\mathbf{p})(\mathbf{z} - \mathbf{p}) + O(\|\mathbf{z} - \mathbf{p}\|^3) \quad (1.3)$$

$O(\|\mathbf{z} - \mathbf{p}\|^3)$ represents cubic and higher order terms and the hessian $\nabla^2 h(\mathbf{p})$ has the eigendecomposition,

$$\nabla^2 h(\mathbf{p}) = V \Lambda V^T, \quad \text{where } V = [\mathbf{v}_1, \dots, \mathbf{v}_m], \quad \text{and } \Lambda = [\kappa_1, \kappa_2, \dots, \kappa_m] \quad (1.4)$$

Here V, Λ are the eigenvector and eigenvalue matrices, $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$ are called the **Principal Directions** and $\kappa_1, \dots, \kappa_m$ the **Principal Curvatures** of the hypersurface S at \mathbf{p} . Principal curvatures represent the second order effects of manifold curvature in the neighbourhood $N_{\mathbf{p}}(\epsilon)$. They can be best visualized as shown in fig. 1.2 - as the curvatures of the intersection of a normal plane (along N in figure) with the manifold at \mathbf{p} . The direction along which this is maximum gives the first principal curvature, direction of maximum curvature

orthogonal to it the second principal curvature, and so on. Principal directions are basically directions normal to the normal plane along the tangent space at these locations of maximum curvature. Suppose we assume \mathbf{p} to be the origin of our coordinate system in \mathbb{R}^{m+1} then the principal directions $\mathbf{v}_1, \dots, \mathbf{v}_m$ are orthogonal and always span the tangent space at \mathbf{p} . We further assume that the tangent space is aligned with the first m canonical vectors $\mathbf{e}_1, \dots, \mathbf{e}_m$, such that each $\mathbf{v}_i = \mathbf{e}_i$. In this case, $h(\mathbf{p}) = \nabla h(\mathbf{p}) = 0$, and $\langle \mathbf{z}, \mathbf{v}_i \rangle = z_i$, implying:

$$h(\mathbf{z}) = \frac{1}{2} \sum_{i=1}^m \kappa_i \langle \mathbf{z}, \mathbf{v}_i \rangle^2 + O(\|\mathbf{z}\|^3) \quad (1.5)$$

$$= \frac{1}{2} \sum_{i=1}^m \kappa_i z_i^2 + O(\|\mathbf{z}\|^3) \quad (1.6)$$

Furthermore, if the neighbourhood size ϵ is small enough, the effect of the higher order terms $O(\|\mathbf{z}\|^3)$ can be ignored to give a second order approximation for $N_{\mathbf{p}}(\epsilon)$, called its *Quadrature Embedding*. Generalizing it to an arbitrary m -dimensional manifold in \mathbb{R}^n , we have the following tangent space parametrization for $\mathbf{x} \in N_{\mathbf{p}}(\epsilon)$ [19] [20],

$$\mathbf{x} = [z_1 \quad z_2 \quad \dots \quad z_m \quad h^1(z_1, \dots, z_m) \quad h^2(z_1, \dots, z_m) \quad \dots \quad h^{n-m}(z_1, \dots, z_m)] \quad (1.7)$$

Again if we assume a coordinate system for each h^i s.t. principal directions are along the canonical vectors and ignore the effect of higher order terms, then:

$$h^i(z_1, \dots, z_m) = \frac{1}{2} \sum_{j=1}^m \kappa_j^i z_j^2 \quad i = 1, 2, \dots, (n - m) \quad (1.8)$$

Here we have m principal curvatures along $n - m$ dimensions, giving rise to $m \times (n - m)$ parameters. We call z_1, z_2, \dots, z_m the **Tangent Space Components** of \mathbf{x} and h^1, h^2, \dots, h^{n-m} the **Normal Space Components** of \mathbf{x} . Such a parametrization of $\mathbf{x} \in \mathcal{M}$ which retains only upto the second order terms is called its **Quadrature Embedding**.

This differential geometric model for a riemannian manifold is useful for cases where n is not much greater than m , but in general we have too many parameters for regression. Instead, to make the problem more tractable, we extract d principal components which are

linear combinations of the normal space components along directions of maximum variance in the neighbourhood. Regressing along only these $d \ll n - m$ we can obtain much better reconstructions than simple linear interpolation.

Chapter 2

Local Quadrature Reconstruction

Next we present the main contribution of this thesis - Local Quadrature Reconstruction, which estimates curvatures of high-dimensional data in a small neighbourhood for out-of-sample reconstruction. Our algorithm is motivated by a tangent space parametrization given in [19] of the points in a neighbourhood on the manifold. Smoothness of the manifold allows us to expand this parametrization in terms of its Taylor series and if we ignore higher order terms assuming a high sampling density, we get the quadrature embedding of the manifold characterized by principal curvatures along each of the $n - m$ higher dimensions. We estimate curvatures along directions of maximum data variance in the neighbourhood using quadratic regression and project the new point to the manifold using the fitted model of the manifold. First, we give an overview of the problem and our solution in section 2.1, followed by the algorithm in detail in section 2.2, and an overview of the free parameters involved in section 2.4.

2.1 Overview

Consider a smooth m -dimensional manifold \mathcal{M} embedded in \mathbb{R}^n where $n \geq m + 1$. Any point \mathbf{x} has only m degrees of freedom, and assuming the manifold is globally homeomorphic to $\mathcal{S} \subseteq \mathbb{R}^m$ it can be written as $\mathbf{x} = f_{\mathcal{M}}(\mathbf{y})$, where $\mathbf{y} \in \mathcal{S}$ are called the latent variables of \mathbf{x} and $f_{\mathcal{M}}$ is a differentiable and invertible non-linear function. The set of all possible latent variables \mathcal{S} is a submanifold of \mathbb{R}^m and may not be unique (different dimensionality reduction

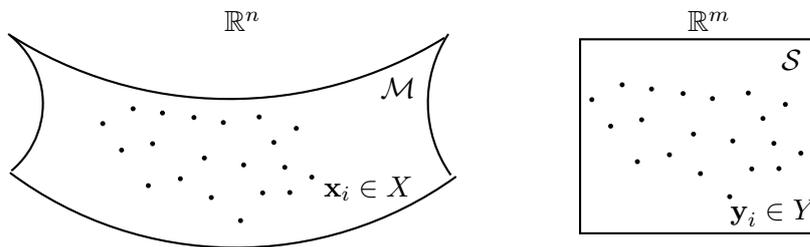


Figure 2.1: m -dimensional manifold in \mathbb{R}^n Points sampled from the manifold \mathcal{M} and their latent variables in \mathcal{S} .

algorithms may extract different latent variables). The inverse of $f_{\mathcal{M}}$ is denoted by $f_{\mathcal{S}}$, and any point $\mathbf{y} \in \mathcal{S}$ can be written as $\mathbf{y} = f_{\mathcal{S}}(\mathbf{x})$ for some $\mathbf{x} \in \mathcal{M}$. An example of such manifolds are image sets, such as the disk dataset where \mathcal{M} is the space of all images \mathbf{x} in $\mathbb{R}^{100 \times 100 \times 3}$ and one possible latent variable space is $\mathcal{S} = [10, 90) \times [10, 90)$. In robotics this particular latent space is called a *configuration space* since it gives the image coordinates of the center of the disk. The function $f_{\mathcal{M}}$ constructs an image \mathbf{x} given the pair of coordinates (x_c, y_c) . We use the term *manifold coordinates* interchangeably with *latent variables* without ambiguity.

Now suppose we are given N points $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ sampled from a hypothetical manifold \mathcal{M} and their latent vectors $Y = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$ (Fig. 2.1). Then given a new latent vector $\mathbf{y}_q \in \mathcal{S} \subseteq \mathbb{R}^m$ our goal is to compute its high dimensional counterpart $\mathbf{x}_q = f_{\mathcal{M}}(\mathbf{y}_q) \in \mathcal{M} \subseteq \mathbb{R}^n$. This is called *Out-of-Sample Reconstruction*. Note that the implicit manifold \mathcal{S} may not be known or available (or even exist) in some situations, but it is always possible to recover some embedding in \mathbb{R}^m for a high-dimensional dataset using NLDR.

Recall that an m -dimensional manifold is locally homeomorphic to the euclidean space \mathbb{R}^m , i.e. for each $\mathbf{p} \in \mathcal{M}$ there is an open set U containing \mathbf{p} which is homeomorphic to an open ball in \mathbb{R}^m . Our analysis is also restricted to a small neighbourhood $N_{\mathbf{p}}(\epsilon)$ on \mathcal{M} containing \mathbf{x}_q (Fig. 2.2). The mapping itself will be non-linear in general, but instead we fit an m -dimensional linear plane through the neighbourhood, and assume a second order form over d normal space components over which data variance is maximum. This reduces to finding an $(m + d)$ -dimensional linear basis set for the elements of $N_{\mathbf{p}}(\epsilon)$ using Principal

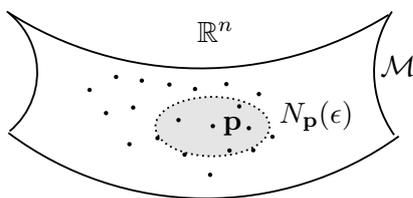


Figure 2.2: ϵ -neighbourhood of a point on the manifold \mathcal{M} , $N_{\mathbf{p}}(\epsilon) = \{\mathbf{x} \in \mathcal{M} : \|\mathbf{x} - \mathbf{p}\|_2 \leq \epsilon\}$

Components Analysis (PCA), and interpolating on this basis set. The first m components represent directions of the maximum spread in the neighbourhood and correspond to the m tangent vectors at p . The remaining d components lie along the normal space at \mathbf{p} represent the effect of curvature and are non-zero due to the non-linear nature of the manifold in $N_{\mathbf{p}}(\epsilon)$. If the manifold was isometric to a plane, then we would have only m Tangent Space components and no Normal Space components. The reconstruction procedure can be divided into three parts:

1. *Subspace Projection:* The local neighbourhood $N_{\mathbf{p}}(\epsilon)$ is projected to an $m+d$ -dimensional linear subspace using Principal Components Analysis (PCA). This is equivalent to projecting the data to the tangent space $T_p\mathcal{M}$ at \mathbf{p} , and then applying a d -dimensional PCA on the remaining residuals.
2. *Linear Regression:* Least squares linear regression is used to find a transformation between $N_{\mathbf{y}_q}^k$, the given latent variables of the k -nearest neighbours of the query point, and the projections of the same points onto $T_p\mathcal{M}$. Then it is straightforward to apply this transform on \mathbf{y}_q and obtain its tangent space projection \mathbf{z}_q . Note that both the tangent space and the latent space is m -dimensional.
3. *Quadratic Regression:* Least squares quadratic regression is performed to find the curvatures along the remaining d Normal Space principal components, with \mathbf{z}_q obtained in step 2 as the independent variable. Once we have the curvatures we can find the projection of the query point along these components. The reconstruction is obtained by reprojecting the interpolated projections back to \mathbb{R}^n .

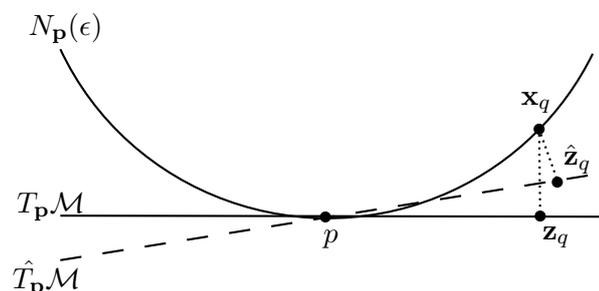


Figure 2.3: **Tangent space on the manifold** Neighbourhood of \mathbf{p} on \mathcal{M} (for visualization the n -dimensional manifold is shown as a curve in \mathbb{R}^2 , and the tangent as a line). Tangent lines spanning the canonical tangent space $T_{\mathbf{p}}\mathcal{M}$ and the estimated tangent space $\hat{T}_{\mathbf{p}}\mathcal{M}$ are also shown.

In our case we have access to only a sampling of $N_{\mathbf{p}}(\epsilon)$ which is denoted as $N_{\mathbf{p}}^k$ - the k -nearest neighbours of \mathbf{p} . As the sampling density on the manifold increases the k -nearest neighbours of \mathbf{p} move closer to it and we can take a smaller and smaller region around \mathbf{p} to sample them, implying that $\epsilon \rightarrow 0$. From here on we shall denote the k -nearest neighbours of \mathbf{y}_q in \mathcal{S} as $N_{\mathbf{y}_q}^k$ and their corresponding manifold points as $f_{\mathcal{M}}(N_{\mathbf{y}_q}^k)$.

2.2 The Algorithm

2.2.1 Motivation

Fig. 2.3 shows the ϵ -neighbourhood of a point \mathbf{p} on the manifold. For the current analysis we can take \mathbf{p} as the reference point or origin of this local neighbourhood. Recall that the tangent space at any point on an m -dimensional manifold also has dimension m . Hence, without loss of generality a rotation can be applied to the manifold so that the first m canonical vectors of \mathbb{R}^n span the tangent space $T_{\mathbf{p}}\mathcal{M}$ at \mathbf{p} :

$$T_{\mathbf{p}}\mathcal{M} = \text{span}(\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m) \quad (2.1)$$

Here \mathbf{e}_i is the i -th canonical vector. The *span* of a basis set is the set of all finite linear combinations of elements of the set, i.e $\text{span}(\mathbf{v}_1, \mathbf{v}_2) = \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2$, where $\lambda_1 + \lambda_2 = 1$. In the latent space \mathcal{S} , or specifically in the neighbourhood $N_{\mathbf{y}_p}^k$, we take \mathbf{y}_p as the origin.

We use the tangent space parametrization given in [19] for points in $N_{\mathbf{p}}(\epsilon)$. Let $h^{(l)} : T_{\mathbf{p}}\mathcal{M} \rightarrow \mathbb{R} \quad l = 1, 2, \dots, (n - m)$ be smooth functions defined on the tangent space at \mathbf{p} such that any point $\mathbf{x} \in N_{\mathbf{p}}(\epsilon)$ can be written as:

$$\mathbf{x} = [\mathbf{z} \quad h^{(1)}(\mathbf{z}) \quad h^{(2)}(\mathbf{z}) \quad \dots \quad h^{(n-m)}(\mathbf{z})]^T \quad (2.2)$$

Here \mathbf{z} is the component of \mathbf{x} along $T_{\mathbf{p}}\mathcal{M}$, and $h^{(l)}(\mathbf{z})$ describes the structure of the manifold around \mathbf{p} along dimension $l + m$. Such a decomposition of \mathbf{x} , in terms of its tangent space components is possible because $T_{\mathbf{p}}\mathcal{M}$ is spanned by the first m canonical vectors in \mathbb{R}^n . Denote $E^{(m)}$ as the matrix which projects points in $N_{\mathbf{p}}(\epsilon)$ to the tangent space at \mathbf{p} ($E^{(m)} = [\mathbf{e}_1 \quad \dots \quad \mathbf{e}_m]$). Then the collection of all $\mathbf{z} = E^{(m)T}\mathbf{x}$ defines an m -dimensional coordinate system on $N_{\mathbf{p}}(\epsilon)$. Recall that \mathbf{x} can also be written in terms of the global latent variables as $\mathbf{x} = f_{\mathcal{M}}(\mathbf{y})$, where $f_{\mathcal{M}}$ is a non-linear function. Most NLDR algorithms such as ISOMAP and LLE assume a locally linear structure of $f_{\mathcal{M}}$ when computing the latent variables. For our purposes we also use local linearity, i.e. $\forall \mathbf{x} \in N_{\mathbf{p}}(\epsilon)$, there is some $\mathbf{y} \in \mathcal{S}$ s.t $\mathbf{x} \simeq \mathbf{x}_p + J(\mathbf{0})(\mathbf{y} - \mathbf{y}_p)$. Here J is the $n \times m$ jacobian matrix given by $J_{ij}(\mathbf{0}) = \frac{\partial f_{\mathcal{M}}^{(i)}(\mathbf{0})}{\partial y_j}$. Projecting both sides onto $E^{(m)}$, we get:

$$E^{(m)T}\mathbf{x} = E^{(m)T}\mathbf{x}_p + E^{(m)T}J(\mathbf{0})(\mathbf{y} - \mathbf{y}_p) \quad (2.3)$$

$$\Rightarrow \mathbf{z} = A(\mathbf{y}) \quad (2.4)$$

Where A is an $m \times m$ linear transformation. It can be solved for using least squares linear regression on the neighbourhood, and applied to the latent variables of the query point \mathbf{y}_q to find its local tangent space coordinates \mathbf{z}_q .

Next we consider the components of \mathbf{x} orthogonal to $T_{\mathbf{p}}\mathcal{M}$ described by the non-linear functions $h^{(l)}$. Since the manifold is differentiable we can write the Taylor series expansion of each $h^{(l)}$ around \mathbf{p} . In the basis space of the principal directions of the hessian $\nabla^2 h^{(l)}$, we have (see section 1.5):

$$h^{(l)}(\mathbf{z}) = \frac{1}{2}(\kappa_1^{(l)} z_1^2 + \dots + \kappa_m^{(l)} z_m^2) + O(\|\mathbf{z}\|_2^3) \quad l = 1, 2, \dots, (n - m) \quad (2.5)$$

Here $\kappa_1^{(l)}, \kappa_2^{(l)} \dots \kappa_m^{(l)}$ are the principal curvatures of the hypersurface defined by h_l and are equal to the m eigenvalues of the hessian $\nabla^2 h_l(\mathbf{0})$ and $O(\|\mathbf{z}\|_2^3)$ is the contribution of higher order terms. Note that $h^{(l)}(\mathbf{0}) = 0$ and $h'^{(l)}(\mathbf{0}) = 0$ by definition. Since $\mathbf{x} \in N_{\mathbf{p}}(\epsilon)$ and $\mathbf{p} = \mathbf{0}$, we have $\|\mathbf{x}\| \leq \epsilon$ which also implies that \mathbf{z} is bounded as $\|\mathbf{z}\| \leq \epsilon$. From here on, we assume that the sampling density is high enough (or ϵ small enough) that the higher order terms $O(\|\mathbf{z}\|_2^3)$ contribute little to eq. 2.5, and assume a quadratic form on $h^{(l)}(\mathbf{z})$:

$$h^{(l)}(\mathbf{z}) = \frac{1}{2} \sum_{j=1}^m \kappa_j^{(l)} z_j^2 \quad l = 1, 2, \dots, (n - m) \quad (2.6)$$

Given $k > m$ points in $N_{\mathbf{p}}(\epsilon)$ we can perform a least squares regression with $z_1^2, z_2^2 \dots z_m^2$ as the dependent variables to find $\kappa_1^{(l)}, \dots, \kappa_m^{(l)}$ for all $l = 1, 2, \dots, (n - m)$. This may seem ambitious as $n \gg m$ and we would have to perform far too many regressions using just a small set of k points. One of the main insights of our work however, is that we limit our computation $d \ll (n - m)$ principal components normal to the tangent space $T_{\mathbf{p}}\mathcal{M}$ along which the effect of curvature is maximum. Quadratic regression along these directions seems to account for a high fraction of the residual error left after linear interpolation on the tangent space. Extracting these principal directions is also particularly simple using PCA, and can be combined with tangent space estimation. In general we can expect with a high probability that the principal components will lie along the directions of maximum curvature.

Summary

The above analysis suggests a natural method for out-of-sample reconstruction. Given a query point \mathbf{y}_q we find a small neighbourhood on the manifold which contains \mathbf{x}_q . This is done by first finding \mathbf{y}_p , the nearest neighbour of \mathbf{y}_q , and then looking for the k -nearest neighbours of $\mathbf{p} = f_{\mathcal{M}}(\mathbf{y}_p)$ on the manifold \mathcal{M} . The tangent space at \mathbf{p} is estimated by performing PCA on this set, and all the points in the neighbourhood are projected to the m -dimensional tangent space. Assuming the linear relationship given in eq. 2.3, we perform

an LLE like interpolation [1] and reconstruct \mathbf{z}_q linearly from its k -nearest neighbours. During PCA, we also extract an additional d orthogonal components along which we perform quadratic regression of the form of eq. 2.6. The found curvatures are then used to find the d NL components of \mathbf{x}_q from the its tangent space projection \mathbf{z}_q . These steps are discussed in greater detail in the following sections.

For visualization, we demonstrate the reconstruction procedure on a well-studied manifold in literature - the **Swiss-Roll**. It is a 2-dimensional manifold embedded in \mathbb{R}^3 ($n = 3, m = 2$) which allows us to visualize the tangent spaces on the manifold. The swiss roll is diffeomorphic to \mathbb{R}^2 , if $\mathbf{y} = [y_1 \ y_2] \in [\frac{3\pi}{2}, \frac{9\pi}{2}] \times [0, 5]$ are the latent variables then the corresponding point on the swiss roll is given by $\mathbf{x} = [y_1 \cos y_1 \ y_1 \sin y_1 \ y_2]$. Fig. 2.4a shows a swiss-roll with 1000 points sampled from it without noise, and Fig. 2.4b shows the 2d coordinates of the same points. Dimension of the ambient space is only 3 which makes reconstruction on it simple, but it serves the purpose of providing a visualization of the procedure and its not surprising to note how the quadratic assumption works fine in local neighbourhoods on the manifold despite the fact that the actual relation between X and Y is in terms of *cos* and *sin*.

2.2.2 Tangent space estimation

We use Principal Components Analysis (PCA) to find a basis set for the tangent space $T_{\mathbf{p}}\mathcal{M}$ at \mathbf{p} . Suppose $N_{\mathbf{p}}^k$ is the set of the k -nearest neighbours of \mathbf{p} in X , then the covariance matrix M_k of this set is given by,

$$M_k = \frac{1}{k} \sum_{\mathbf{x}_i \in N_{\mathbf{p}}^k} \mathbf{x}_i \mathbf{x}_i^T \quad (2.7)$$

PCA performs eigendecomposition on this matrix, which has at most k -nonzero eigenvalues since $\text{rank}(M_k) \leq k$. In the following we assume that the k points are independent and hence the covariance matrix has full rank. If $U = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_k]$ and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_k)$ are the eigenvector and eigenvalue matrices of M_k respectively such that $\lambda_1 \geq \lambda_2 \geq \dots, \lambda_k$, then the estimated tangent space at \mathbf{p} is:

$$\hat{T}_{\mathbf{p}}\mathcal{M} = \text{span}(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m) \quad (2.8)$$

Note that we need $k > m$ points in the neighbourhood to estimate the tangent space. Let $U^{(m)} = [\mathbf{u}_1 \ \dots \ \mathbf{u}_m]$, then the projection of a point $\mathbf{x} \in N_{\mathbf{p}}^k$ on $\hat{T}_{\mathbf{p}}\mathcal{M}$ is given by $\hat{\mathbf{z}} = U^{(m)T} \mathbf{x}$. Denote $N_{\mathbf{z}_p}^k$ as the set of the projections of $N_{\mathbf{p}}^k$ onto $\hat{T}_{\mathbf{p}}\mathcal{M}$.

In [19], Tyagi et al. show that as the sampling density on the manifold increases the angle between the estimated tangent plane $\hat{T}_{\mathbf{p}}\mathcal{M}$ and the true tangent plane $T_{\mathbf{p}}\mathcal{M}$ goes to zero, or equivalently $\|E^{(m)}E^{(m)T} - U^{(m)}U^{(m)T}\|_{F \rightarrow 0}$, where $E^{(m)}$ is the matrix with the first m canonical vectors as its columns. Their main result is presented in Theorem 2 and shows that with a sufficient number of samples the principal components found by PCA reliably span the canonical tangent space $T_{\mathbf{p}}\mathcal{M}$.

For image manifolds, $n \gg m$ and if the sampling density is not very high, then using $N_{\mathbf{p}}^k$ to find the tangent space is not optimal. Here removing a single point from the neighbourhood (in our case the test point being reconstructed) skews the tangent vectors (see section 3.2 for details). Instead we use $f_{\mathcal{M}}(N_{\mathbf{y}_q}^k)$, the manifold points of k -nearest neighbours of \mathbf{y}_q in \mathcal{S} , for finding the tangent space. This neighbourhood has uniform variability in all the directions, and the estimated tangent vectors are more accurate.

For the swiss-roll dataset, we use $(m + d) = 3$ principal components (the third one is for quadratic regression), out of which the first two constitute the Tangent Space $T_p\mathcal{M}$, and the third one the orthogonal component. An overview of the tangent space estimation procedure is shown in Fig. 2.4. The neighbourhood size parameter for finding the tangent space around a point was set to $k_1 = 18$.

2.2.3 Linear Regression

Next we describe the method for finding \mathbf{z}_q given the projections $N_{\mathbf{z}_q}^k$, their latent variables $N_{\mathbf{y}_q}^k$ and \mathbf{y}_q . Assuming the linear form given in eq. 2.3, we wish to find weights w_i which

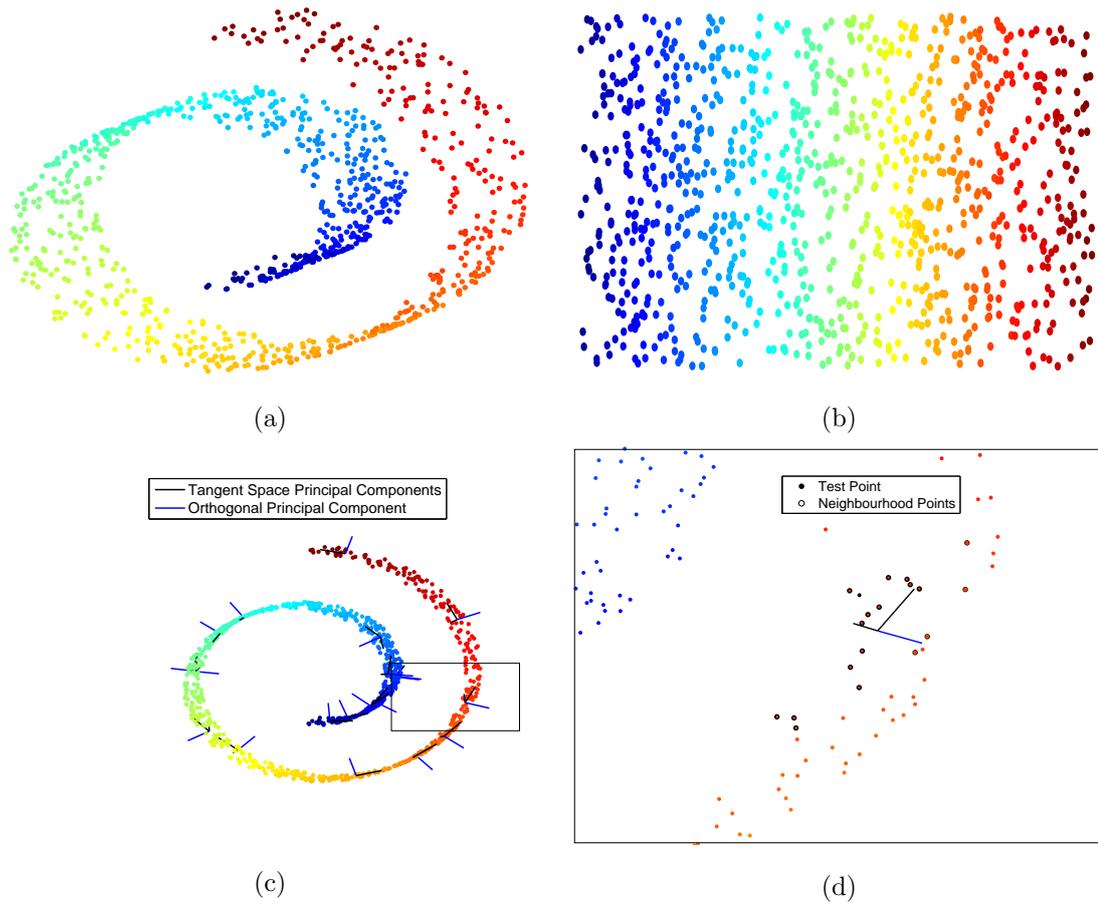


Figure 2.4: **Tangent Space Estimation**, A visualization of tangent space estimation on the swiss roll manifold. (a) Swiss Roll in \mathbb{R}^3 , (b) latent variables in \mathbb{R}^2 , (c) Estimated Tangent Spaces for 50 random neighbourhoods, (d) Close view of a point with its neighbours and tangent space. PCA neighbourhood size $k_1 = 18$.

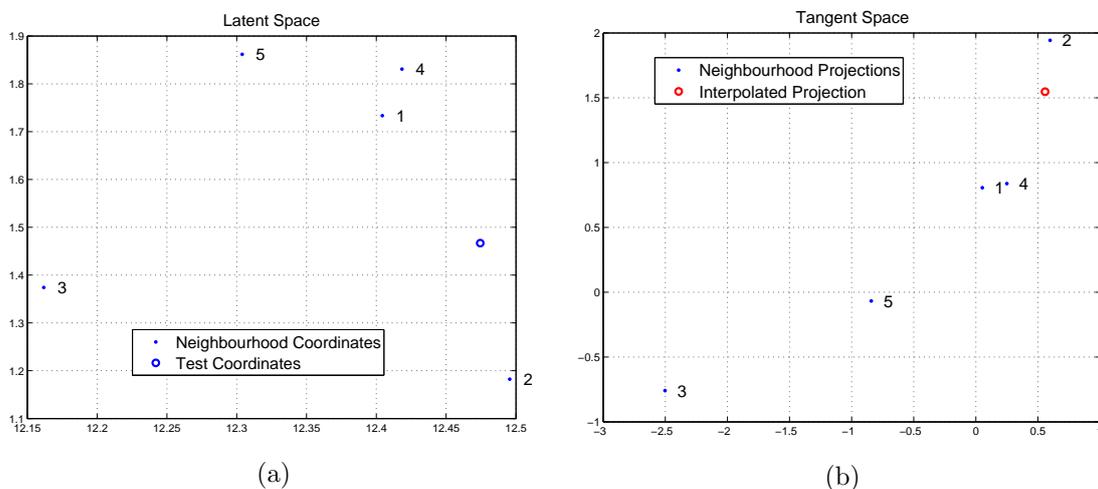


Figure 2.5: **Linear Regression on the Tangent Space**, Linear regression on the Tangent Space for the test point shown in Fig. 2.4d. Neighbourhood size for interpolation $k = 5$. (a) $N_{\mathbf{y}_q}^k$ in \mathcal{S} , (b) $N_{\mathbf{z}_q}^k$ on the tangent space $\hat{T}_p\mathcal{M}$ with the interpolated reconstruction. Regularization parameter $\lambda_L = 10^{-3}$.

minimize the cost function $\mathcal{E}(W) = \|\mathbf{y}_q - \sum_i w_i \mathbf{y}_i\|^2$. We follow the procedure given in [1] and perform a least squares regression to find the optimal weights. Let C denote the local covariance matrix - $C_{ij} = (\mathbf{y}_q - \mathbf{y}_i) \cdot (\mathbf{y}_q - \mathbf{y}_j)$, then optimal weights are given by:

$$w_i^* = \frac{\sum_j C_{ij}^{-1}}{\sum_{lm} C_{lm}^{-1}} \quad (2.9)$$

Using these weights, we interpolate on the tangent plane to obtain the projection of the reconstruction $\hat{\mathbf{z}}_q$ on it:

$$\hat{\mathbf{z}}_q = \sum_{\hat{\mathbf{z}}_i \in N_{\mathbf{z}_q}^k} w_i^* \hat{\mathbf{z}}_i \quad (2.10)$$

To avoid overfitted solutions we regularize the covariance matrix before computing the weights:

$$C_{ij} = C_{ij} + \lambda_L \text{trace}(C) \mathbf{I}_{k_2} \quad (2.11)$$

For the swiss-roll dataset the neighbourhood size parameter k_2 was set to 5, and the regularization $\lambda_L = 10^{-3}$. Fig. 2.5 shows the coordinate space and the tangent space for the

point shown in fig. 2.4d. The red circle shows the interpolated result on tangent space $T_{\mathbf{p}}\mathcal{M}$. The two spaces \mathcal{S} and $\hat{T}_{\mathbf{p}}\mathcal{M}$ look visually very different because the relative scaling between the axes is different for them.

2.2.4 Quadratic Regression

We already have one reconstruction from $\hat{\mathbf{z}}_q$ by projecting it on the estimated tangent space $\hat{T}_{\mathbf{p}}\mathcal{M}$:

$$\hat{\mathbf{x}}_q^m = U^{(m)}\hat{\mathbf{z}}_q \quad (2.12)$$

Here m represents the number of principal components we have regressed on. While this in itself is a good reconstruction, for very high-dimensional data such as image sets, a high amount of blurring is observed (fig. 3.12a). In principle it is simply a linear interpolation on a linear subspace:

$$\hat{\mathbf{x}}_q^m = U^{(m)}\hat{\mathbf{z}}_q \quad (2.13)$$

$$= U^{(m)} \left[\sum_{i=1}^k w_i^* \hat{\mathbf{z}}_i \right] \quad (2.14)$$

$$= U^{(m)}U^{(m)T} \left[\sum_{i=1}^k w_i^* \mathbf{x}_i \right] \quad (2.15)$$

$U^{(m)}U^{(m)T}$ is the projection matrix which retains the first m principal components of the data, and the above sum is taken over $N_{\mathbf{y}_p}^k$. For more accurate reconstructions we use eq. 2.6 and perform regression on d more principal components (this is possible since PCA decorrelates the data [7]). Let $\mathbf{u}_{m+1}, \mathbf{u}_{m+2}, \dots, \mathbf{u}_{m+d}$ be the next d components normal to $\hat{T}_{\mathbf{p}}\mathcal{M} = \text{span}(\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m)$. These normal components point along directions of maximum data variance in the normal space, and hence we can also expect the curvature to be maximum along these principal components.

Ideally we would need $d = n - m$ components to completely describe the data in $N_{\mathbf{p}}(\epsilon)$, but since we have only k points in the neighbourhood we can extract at most $d = k - m$ components. Along these directions the variance of neighbourhood points is maximum. Then, the projection of the any point $\mathbf{x} \in N_{\mathbf{p}}(\epsilon)$ onto \mathbf{u}_{m+i} is given by:

$$\hat{z}_{m+i} = \mathbf{u}_{m+i} \cdot \mathbf{x} = \sum_{j=1}^n u_{m+i}^j x^j \quad i = 1, 2, \dots, d \quad (2.16)$$

Assuming a low error in the predicted tangent space, we ignore $u_{m+i}^1, u_{m+i}^2, \dots, u_{m+i}^m$ since \mathbf{u}_{m+i} is normal to $\hat{T}_{\mathbf{p}}\mathcal{M}$. Then the orthogonal component becomes a linear combination of the quadratic forms in eq. 2.6:

$$\hat{z}_{m+i} = \sum_{j=m+1}^n u_{m+i}^j h^{(j)}(\mathbf{z}) \quad i = 1, 2, \dots, d \quad (2.17)$$

If our coordinate system is aligned to the principal directions of $h^{(j)}$, then it has the simple form $h^{(j)}(\mathbf{z}) = \frac{1}{2} \sum_{i=1}^m \kappa_i^{(j)} z_i^2$. These principal directions span the tangent space and we have one basis set on the tangent space from PCA. We can substitute the projection $\hat{\mathbf{z}}$ onto $\hat{T}_{\mathbf{p}}\mathcal{M}$ into the equation for $h^{(j)}$ by adding cross-terms of the form $\hat{z}_i \hat{z}_j$. Until now we have also assumed the recovered principal components perfectly span the tangent space, however this may not always be the case. To make the regression more robust in these cases we also add linear terms $\hat{z}_1, \hat{z}_2, \dots, \hat{z}_m$ to $h^{(j)}$ and a constant term μ :

$$\hat{z}_{m+i} = \sum_{i=1}^m a_i \hat{z}_i + \sum_{i=1}^m b_i \hat{z}_i^2 + \sum_{i=1}^{m-1} \sum_{j=i+1}^m C_{ij} \hat{z}_i \hat{z}_j + \mu \quad i = 1, 2, \dots, d \quad (2.18)$$

The constant term μ is added to allow for the displacement of the principal components from \mathbf{p} , since PCA finds principal components about the mean of $N_{\mathbf{p}}^k$ which may not lie on the manifold. The parameters a_i, b_i, c_{ij}, μ are found by solving an overdetermined system of equations on the k neighbourhood points. The dependent variables are the PCA projections \hat{z}_j , and the least squares solution is found which minimizes the following error:

$$\mathcal{E}(\mathbf{a}, \mathbf{b}, \mathbf{C}, \mu) = \sum_{\mathbf{z} \in N_{\mathbf{p}}^k} \|\hat{z}_{m+i} - \sum a_i \hat{z}_i - \sum b_i \hat{z}_i^2 - \sum C_{ij} \hat{z}_i \hat{z}_j - \mu\|^2 + \lambda_Q [\mathbf{a} \cdot \mathbf{a} + \mathbf{b} \cdot \mathbf{b} + \mathbf{c} \cdot \mathbf{c} + \mu^2] \quad (2.19)$$

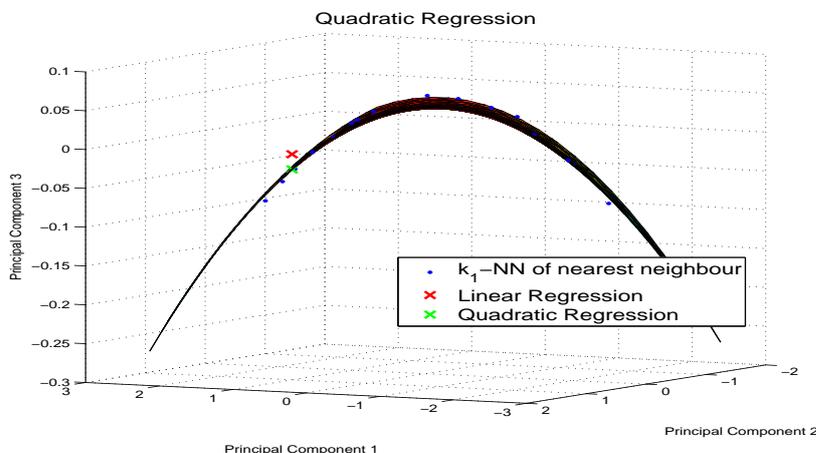


Figure 2.6: **Quadratic Regression**, Regression with the independent variable as the first two principal components and the dependent variable as the third component using $N_{\mathbf{p}}^{k_1}$. Regularization parameter $\lambda_Q = 1$.

The last term is for regularization, which is important since its easy to overfit the data in high-dimensions. Note that we need at least $k > (2m + \binom{m}{2} + 1)$ (and in practice more) nearest neighbours to solve the equations. Fig. 2.6 shows this procedure for the swiss roll. For comparison we also show the linearly interpolated result in red.

Once we have the parameters $\mathbf{a}, \mathbf{b}, C, \mu$ for each of the d extra components, we can find the projections of the query point onto them in a straight forward manner. The final quadrature reconstruction is given by projecting the m tangent space components linearly interpolated and d normal space components quadratically interpolated onto the basis set $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{m+d}$. We denote the whole process of reconstruction on the tangent space by \hat{f} , i.e. $\hat{\mathbf{x}}_q = \hat{f}(\mathbf{y}_q)$. The entire procedure is listed in algorithmic form in Algorithm 1.

Regularization is an important step during both the steps above. For linear regression we use $\lambda_L = 10^{-3}$, and for quadratic regression we use $\lambda_Q = 1$. The high value of λ_Q also ensures that we dont get solutions with very high errors when the test point lies outside the support of $N_{\mathbf{p}}^{k_1}$.

Data: X, Y, \mathbf{y}_q

Parameters: $k_1, k_2, d, \lambda_L, \lambda_Q$

Result: $\hat{\mathbf{x}}_q$

// Find Neighbourhood for PCA

1. $\mathbf{y}_p \leftarrow nn(Y, \mathbf{y}_q, 1)$

2. $\mathbf{p} = f\mathcal{M}(\mathbf{y}_p)$

3. $N_p^{k_1} \leftarrow nn(X, \mathbf{p}, k_1)$

// Eigendecomposition of covariance matrix

4. $M_{k_1} \leftarrow cov(N_p^{k_1})$

5. $\Lambda V = M_{k_1} V$

6. $Z_{lin} = f\mathcal{M}(N_{\mathbf{y}_q}^{k_2}) V_{1\dots m}$

// Linear Regression on Tangent Space

7. $C_{ij} \leftarrow (\mathbf{y}_q - \mathbf{y}_i) \cdot (\mathbf{y}_q - \mathbf{y}_j), \quad i, j = 1, 2 \dots k_2, \quad \mathbf{y}_i \in N_{\mathbf{y}_q}^{k_2}$

8. $C = C + \lambda_L trace(C) I_{k_2}$

9. $W^* \leftarrow C^{-1} \mathbf{1}_{k_2}$

10. $W^* \leftarrow W^* / sum(W^*)$

11. $\mathbf{z}_q \leftarrow W Z_{lin}$

12. $\hat{\mathbf{x}}_q \leftarrow V_{1\dots m}^T \mathbf{z}_q$

// Quadratic Regression on d orthogonal principal components

13. $Z_{quad} = N_p^{k_1} V_{1\dots m}$

14. $A \leftarrow [Z_{quad}^2 \quad Z_{quad}^1 Z_{quad}^2 \quad \dots \quad Z_{quad}^{m-1} Z_{quad}^m \quad Z_{quad} \quad \mathbf{1}]$

15. $B = A' A + \lambda_Q I_{2m + \binom{m}{2} + 2}$

16. **for** $i = m + 1 \rightarrow m + d$ **do**

17. $par \leftarrow B^{-1} A N_p^{k_1} V_i$
18. $pr_i = [z_q^2 \quad z_q^1 z_q^2 \quad \dots \quad z_q^{m-1} z_q^m \quad z_q \quad \mathbf{1}] par$
19. $\hat{\mathbf{x}}_q \leftarrow \hat{\mathbf{x}}_q + V_i^T pr_i$

end

20. **return** $\hat{\mathbf{x}}_q$

Algorithm 1: Local Quadrature Reconstruction

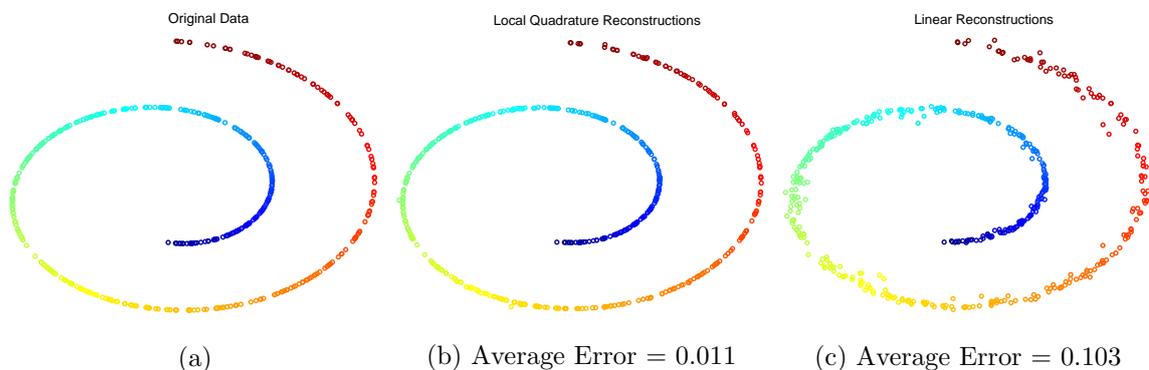


Figure 2.7: **Local Quadrature v Linear Reconstruction on the Swiss Roll**, Side-view of test point reconstructions on the swiss-roll. (a) Original Data. (b) Local Quadrature reconstructions. (c) Linear Reconstructions. $k_1 = 18$, $k_2 = 5$, $\lambda_L = 10^{-3}$, $\lambda_Q = 1$

To get a quantitative measure of the two reconstruction procedures (quadratic and linear), we divide the 1000 points randomly into two sets of 500 each. Each point in the second set was reconstructed from its neighbours in the first set. Fig. 2.7 shows a side-view of the test points reconstructed on the Swiss Roll. Linear reconstructions may lie at high distances from the manifold due to the effect of curvature. This effect is not seen in LQR which maintains the shape of swiss-roll (*cf* Fig. 2.7b). Error is measured as the squared distance between the original point and its reconstruction in \mathbb{R}^3 . The average linear error across the 500 points reconstructed was 0.103 and for LQR it was 0.011.

2.3 Complexity

We report the computational complexity of our algorithm in terms of $k = \max(k_1, k_2)$, m , n , d . There are five main steps involved in LQR. First, we need to compute the k -nearest neighbours of either p or y_q or both, for which simple linear search has the complexity $O(kNn)$ or $O(kNm)$, depending on which space we compute the neighbours in. Next is the eigendecomposition of M_{k_1} which is a $k_1 \times k_1$ matrix. This step has the computation time $O(k^3)$. The main steps during both the regressions is inversion, and hence for linear regression the

complexity is $O(k^3)$, and for quadratic regression along each component is $O(m^6)$. Since $k \geq 2m + \binom{m}{2} + 2 > m^2$, regression along each component is $O(k^3)$. Lastly, we project the found reconstruction onto the $m + d$ principal components, each of which is n dimensional, to find the reconstruction. The total complexity of the algorithm then becomes $O(kNn + dk^3 + n(m + d))$ where nearest neighbour search is performed on the manifold and $O(kNm + dk^3 + n(m + d))$ where nearest neighbour search is performed in the latent space. As explained in section 2.2.2, for image manifolds where $n \gg m$ we use $N_{\mathbf{y}_q}^k$ for finding the tangent space and hence the complexity varies only linearly with n here which may be very large.

2.4 Parameter Selection

Free parameters in the algorithm which need to be set include -

- k_1 - PCA Neighbourhood Size
- k_2 - Linear Interpolation Neighbourhood Size
- d - Number of Normal Space Principal Components
- λ_L - Regularization Parameter for linear regression
- λ_Q - Regularization Parameter for quadratic regression

The PCA neighbourhood size k_1 denotes the number of neighbours of \mathbf{p} to use for finding the principal components, while k_2 is the number of nearest neighbours of \mathbf{y}_q through which a plane is fit for interpolating \mathbf{z}_q . Higher values result in the poor tangent space estimation and lower values overfit to the neighbourhood. They can be set using standard techniques like cross-validation on the training set. Global minima for the parameters existed for all the datasets we tested on.

Next, d is the number of components extracted normal to the tangent space. Since these are found by an SVD of the covariance matrix of the k_1 -nearest neighbours of \mathbf{p} , which has

rank k_1 , the maximum value d can take is $k_1 - m$. In practice, using the same value of d for all test points does not give satisfactory results. Instead we use a simple rule to decide the number of principal components to extract from a neighbourhood $N_{\mathbf{p}}(\epsilon)$. If $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{k_1}$ are the k_1 -nonzero eigenvalues of the covariance matrix of neighbours, then $(m + d)$ is set to the minimum value such that,

$$\frac{\sum_{i=1}^{m+d} \lambda_i}{\sum_{i=1}^{k_1} \lambda_i} > t \quad (2.20)$$

Here t is the threshold of percentage of energy we want to consider. The LHS above is a measure of the variance in the data accounted for by the first $(m + d)$ eigenvectors of PCA. Setting high values of t may cause overfitting of the normal principal components to the data samples, but lower values may cause loss of information. Again this can be optimized using cross-validation.

Regularization is important during regression to avoid over-fitting to the data. Specially during quadratic regression the number of coefficients is given by $(2m + \binom{m}{2} + 2)$ for $m \geq 2$, and the number of known points is k_1 which may not be much greater. Another effect observed without regularization is that the query point might be projected far outside the range of training data. Regularizing penalizes such cases and leads to stable results. λ_L was set to 10^{-3} in all our experiments and λ_Q optimum values of λ_Q varied from 0.1-10.

Chapter 3

Experiments and Results

In this chapter we explore the performance of the out-of-sample reconstruction method described in chapter 2. Consider a set of points $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, $\mathbf{x}_i \in \mathbb{R}^n$ sampled from a differentiable manifold \mathcal{M} whose intrinsic dimension is m . We assume that the manifold is homeomorphic to \mathbb{R}^m , meaning that it can be endowed with a global coordinate system. For datasets where this is not true, such as the teapot dataset, we remove sections of the manifold to make it homeomorphic to \mathbb{R}^m (see section 1.3 for details). Let $Y = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$ be these global latent vectors or latent variables of X . These may be known explicitly from the function generating the data, or may be found using NLDR techniques such as ISOMAP, LLE etc. In all our experiments the dataset is randomly split into training and testing sets and each point in the test set is reconstructed from its latent vector using neighbouring training set examples. Quality of reconstruction is measured using the squared euclidean error between the true high-dimensional data and the found reconstruction, and we report the mean squared error (MSE) over all the test points.

We have already shown reconstructions for the swiss-roll and we start with another point set manifold in section 3.1. While point sets are good for visualization, practical applications for the proposed algorithm include image reconstruction and interpolation which lie in much higher dimensional spaces. Section 3.2 gives the results of LQR applied on image sets such as the teapot dataset and the disk dataset. Many videos can be interpreted as 1-dimensional submanifolds of some unknown image manifold and we show a preliminary analysis of using

LQR for frame interpolation, details for which are given in section 3.3.

3.1 Point Sets

We have already discussed the performance of LQR on one point set - the swiss-roll, in chapter 2. For the swiss-roll the embedding dimension n is only 1 more than its manifold dimension m , and hence the number of components we can regress on is also $m + d = n$. In general we would want that when $n \gg m$, the number of normal space components required d would be small, i.e. we can get good reconstructions by only interpolating along a few normal principal components of the data. The next dataset we use - a **Spiral** in \mathbb{R}^3 emphasizes this point better. It is a 1-dimensional manifold, if $y \in [0, 10\pi]$ is the latent vector then the manifold point in \mathbb{R}^3 is given by $\mathbf{x} = [\cos y, \sin y, y]$. Note that $n = 3$, $m = 1$ and we extract $d = 2$ principal components for reconstruction. Fig. 3.1a shows 400 points sampled from the spiral manifold. These are randomly divided into training and testing sets of equal size ($N = 200$), and the test points are reconstructed from their latent vectors.

Fig. 3.1 gives an overview of the reconstruction on this spiral. The neighbourhood sizes were set to $k_1 = 8$ and $k_2 = 4$, and the regularization parameters were set to $\lambda_L = 10^{-3}$ and $\lambda_Q = 10^{-2}$. The tangent vectors are accurately extracted even in places with low sampling density (Fig. 3.1b), which is expected due to the low value of m . Fig. 3.1d shows the quadratic curve fit on the tangent space of $N_{\mathbf{z}_p}^{k_1}$ the neighbourhood shown in 3.1c. For comparison the linear reconstruction (interpolated using $N_{y_q}^{k_2}$) is also shown. Finally, we show reconstructions of all 200 test points in Fig. 3.2. LQR is able to preserve the manifold shape effectively throughout except at the boundaries. At the boundary the test point is extrapolated instead of being interpolated which causes the increased error (see fig. 3.3 for example).

3.2 Image Sets

The point sets discussed above demonstrate the effectiveness of LQR on arbitrary non-linear differentiable manifolds, but practical applications of reconstruction on manifolds typically

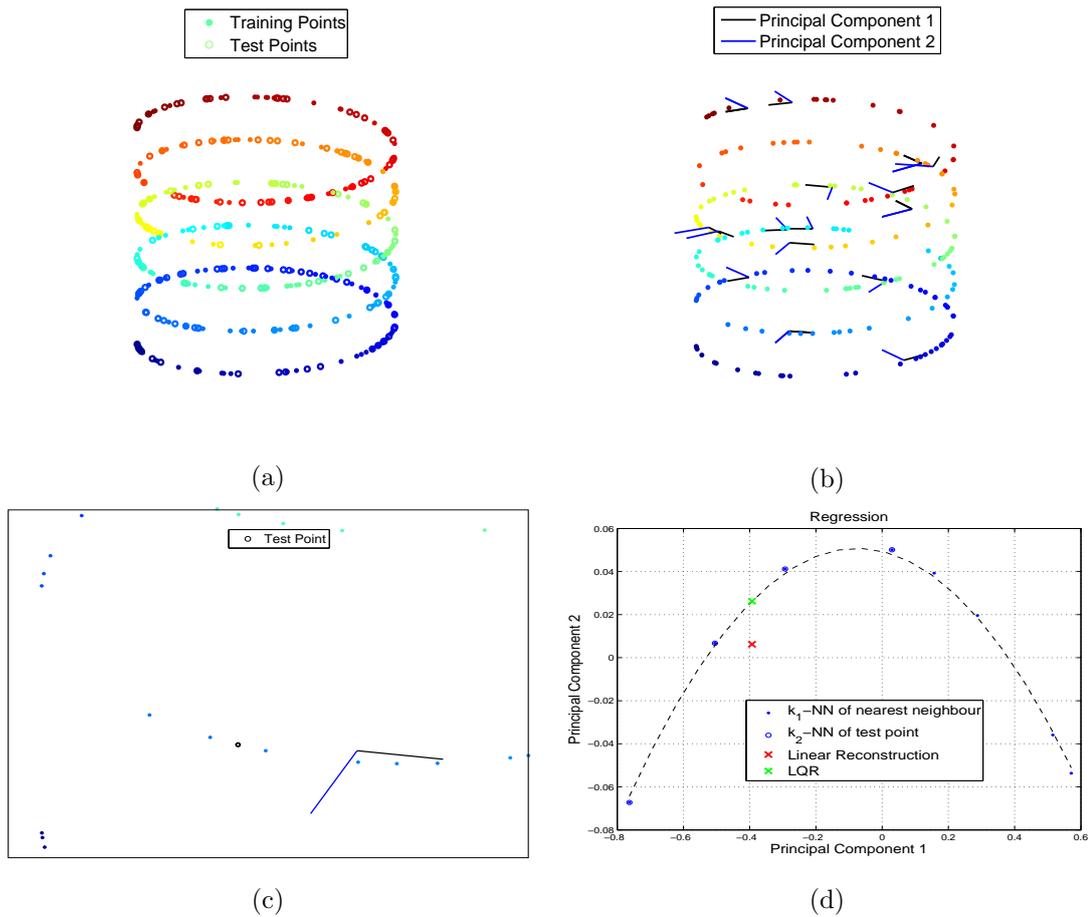


Figure 3.1: **LQR on a Spiral** A visualization of the reconstruction procedure on the spiral dataset. (a) Points sampled from the 1d spiral. (b) First two principal components computed for 20 neighbourhoods ($k_1 = 8$). (c) Close view of a test point and the principal components computed in its neighbourhood. (d) Linear and Quadratic regression along first and second components, respectively ($\lambda_L = 10^{-3}$, $\lambda_Q = 10^{-2}$). Neighbourhood size parameter $k_2 = 4$.

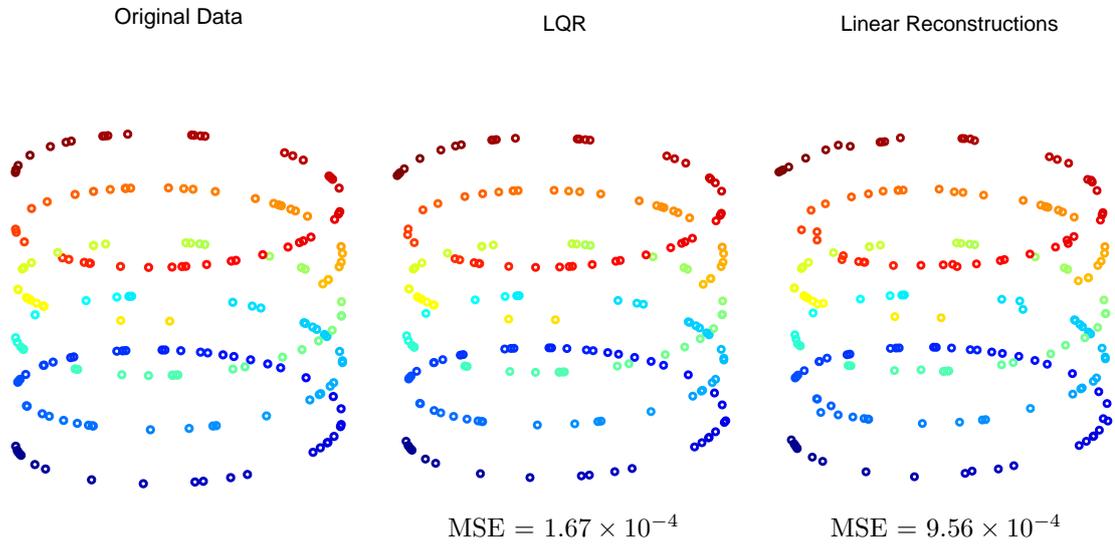


Figure 3.2: **Local Quadrature v Linear Reconstruction on the Spiral** Comparison of test point reconstructions on the 1d Spiral manifold. (a) Original Data. (b) Local Quadrature reconstructions. (c) Linear Reconstructions.

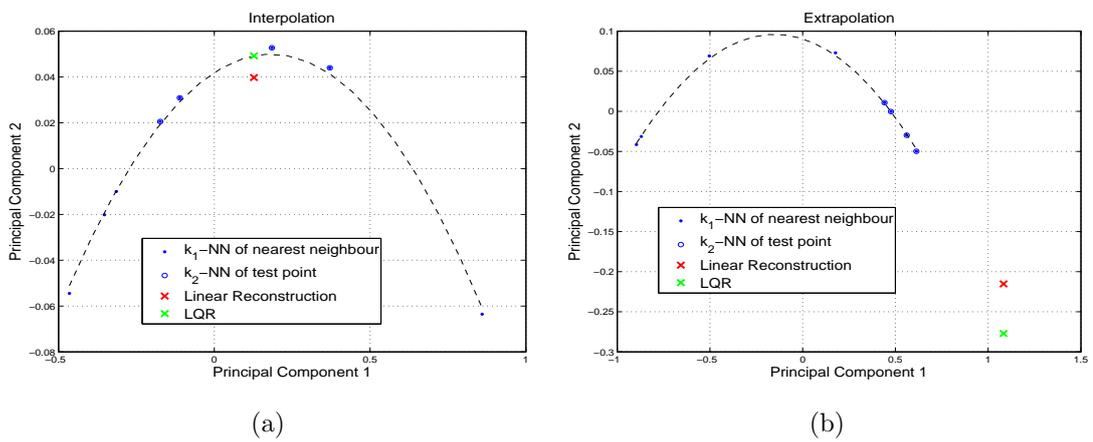


Figure 3.3: **Extrapolation v Interpolation on a 1-d manifold** A comparison of regression for (a) a boundary point and (b) an interior point on the 1-d spiral. At the boundary point the reconstruction is extrapolated far beyond the support of the training data, and hence both reconstructions are inaccurate.

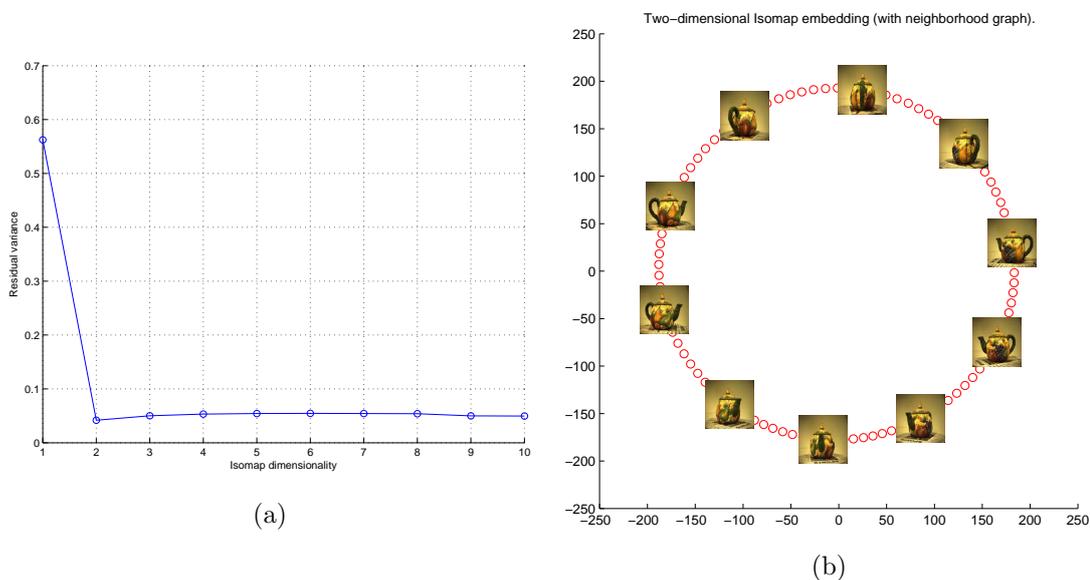


Figure 3.4: **ISOMAP on teapot dataset** (a) Residual variance at different ISOMAP dimensionalities. (b) Embedding in \mathbb{R}^2 . Residual variance at $m = 1$ is 0.57 and at $m = 2$ it is 0.04 since the manifold is globally homeomorphic to a circle in \mathbb{R}^2 . ISOMAP neighbourhood size parameter $k = 4$.

involve much larger values of n . Image sets are an important class of manifolds for which n is the number of pixels in each image, and m depends on the variability of the data. Linear reconstruction is particularly simple on these image manifolds, but unless the sampling density is high it gives heavily blurred results. In this section we study whether LQR is able to provide better solutions. Theoretically, the analysis presented in chapter 2 cannot be extended to image manifolds since spatial aliasing prevents them from being differentiable, unless there are an infinite number of pixels in the image or it is analog. However, practically we find that LQR works well on image sets and quadratic regression improves reconstruction over linear regression.

3.2.1 Rotating Teapot

Our first dataset is the **Teapot** set from [21] which has $N = 100$ images of a teapot undergoing a 360 degree rotation. Each image is $76 \times 101 \times 3$ in size and lies in \mathbb{R}^{23028} , but the intrinsic dimensionality of the manifold is only $m = 1$ since there is only 1 free parameter in



Figure 3.5: **1-D Isomap embedding for teapot dataset**, ISOMAP embedding of the teapot dataset after removing a portion of the manifold. Residual variance for $m = 1$ is only 10^{-4} . ISOMAP neighbourhood size parameter $k = 4$.

generating the images which is $\theta \in [0, 2\pi]$ (the angle of rotation of the teapot). In this case we only have the high-dimensional data without any knowledge of the latent vectors, hence we use ISOMAP to learn the global latent vectors of each image.

The teapot manifold has an S_1 topology and is not homeomorphic to \mathbb{R} , but instead to circle in \mathbb{R}^2 . Due to this no global 1-d coordinate system exists for it in \mathbb{R} . This is also evident from the residual variance between ISOMAP embeddings at dimensions 1 and 2, and the recovered embedding in \mathbb{R}^2 (see Fig. 3.4). Even though the intrinsic dimensionality of the manifold is 1, no set of 1-d latent vectors can be assigned to the points on it which have low error. For LQR we want 1-d latent vectors for the points on the teapot manifold since the tangent space at each point is only 1-dimensional. Here we can calculate a set of 1-d polar coordinates from the 2-d coordinates given by ISOMAP, but in general this is not possible for arbitrary manifolds with the same problem. To overcome this, a section of the manifold is removed, in this case all images with $\theta > \frac{9\pi}{5}$. Once the neighbourhood connectivity is lost, and the first point is no longer connected to the last point the remaining set becomes homeomorphic to \mathbb{R} . The ISOMAP embedding of this new manifold in 1 dimension is shown in Fig. 3.5.

The new dataset for $\theta \in [0, \frac{9\pi}{5}]$ is then used for testing the LQR. The latent vectors are set to the ISOMAP embedding shown in Fig. 3.5b. We separate 18 uniformly spaced frames

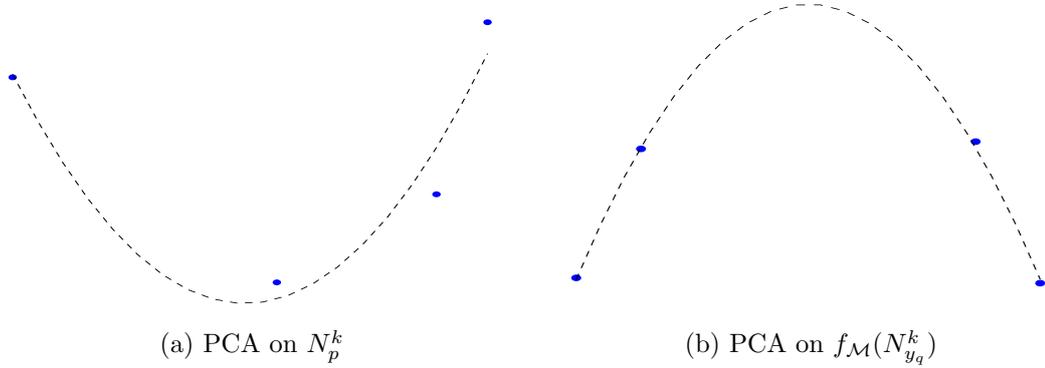


Figure 3.6: **Neighbourhood selection for teapot dataset** Projections of neighbourhood points onto the tangent space found by PCA performed on N_p^k and $f_{\mathcal{M}}(N_{y_q}^k)$. In the latter, the tangent vectors are rotated leading to poor quadratic regression.

as test images and keep the remaining as training images. Both the neighbourhood size parameters were set to $k = k_1 = k_2 = 4$ (the low value of k_1 is due to the small training set size). In fact we use $f_{\mathcal{M}}(N_{y_q}^k)$ for both tangent space estimation and interpolation, unlike the other datasets discussed till now where N_p^k is used to find the tangent space and $f_{\mathcal{M}}(N_{y_q}^k)$ for interpolation. The reason for this is explained in Fig. 3.6. Basically, the sampling density for the teapot manifold is so low that removing just one point from a neighbourhood skews the tangent in the other direction. In this case the assumption that $z_2 = az_1^2 + bz_1 + c$ (z_1, z_2 are the projections along the first two principal components) fails to hold, instead we have $az_1^2 + bz_2^2 + cz_1 + dz_2 + e = 0$ which is a much more difficult regression to solve, and doesn't work well in general. In the neighbourhood around p we have already removed the test point while forming the training and test sets, hence the found tangent vectors are rotated in the other direction (*cf* Fig. 3.6a), causing a poor quadratic fit along \mathbf{u}_2 . Using $f_{\mathcal{M}}(N_{y_q}^k)$ solves this problem (Fig. 3.6b), and also transfers the nearest neighbour search step to the latent space where it is much quicker. During reconstruction we extract a total of $m + d = 2$ principal components to regress along (higher components are mostly noise and do not tell us much about the manifold).

An example of LQR on this set is shown in fig. 3.7. The tangent vectors found by PCA on $f_{\mathcal{M}}(N_{y_q}^k)$ are shown in fig. 3.7b and projections onto them of neighbouring points and

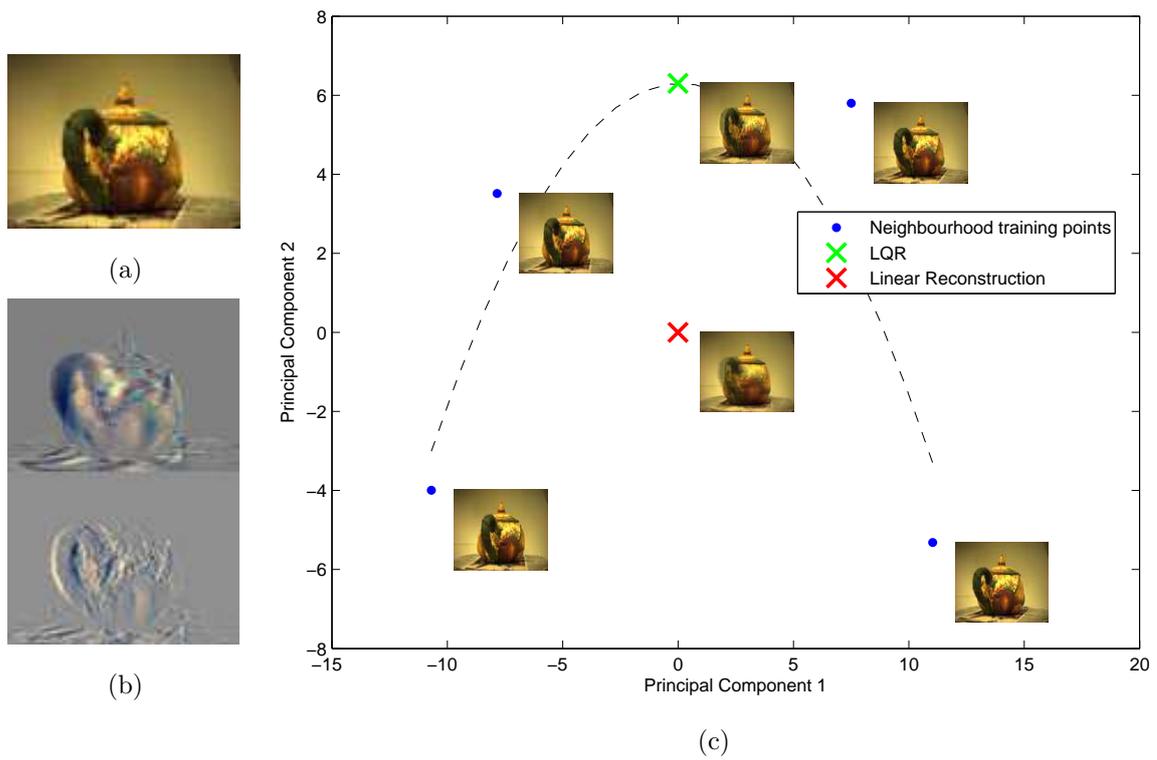


Figure 3.7: **LQR on Teapot dataset**, An overview of the reconstruction procedure on the teapot dataset. (a) Test Image, (b) Tangents found, (c) Regression. Neighbourhood size parameters $k_1 = k_2 = k = 4$, and $N_{y_q}^k$ used both for finding principal components and interpolation. $\lambda_L = 10^{-3}$ and $\lambda_Q = 0.5$. Quadratic interpolation along \mathbf{u}_2 gives reconstruction with lesser blur than linear interpolation.



Figure 3.8: **Local Quadrature v Linear Reconstruction on Teapot dataset**, Example reconstructions on the teapot dataset. Top row - Original Data, Second Row - LQR, Bottom row - Linear reconstruction.

found reconstructions are shown in fig. 3.7c. $\lambda_L = 10^{-3}$ as before but $\lambda_Q = 0.5$ which is high considering the curve fit, but it is necessary since lower values project the test point much farther out in the direction of \mathbf{u}_2 than the support of the training points used to find it, leading to poorer results. Linear reconstruction is also included for comparison. Observing the handles and design on the teapot, one can clearly see that the LQR produces results with lower blur. More examples of reconstruction on the teapot set are shown in fig. 3.8, and variation in squared error as the angle of rotation changes is shown in fig. 3.9. On all the images tested for reconstruction, LQR outperforms the linear method in terms of the squared error between true data and found reconstruction.

3.2.2 Disk-Shaped Planar Robot

The **Disk** dataset consists of images of a circular robot translating in a 2-D plane. Images lie in $\mathbb{R}^{100 \times 100 \times 3}$ with two modes of variation (translation along x and y). Hence, its a 2-dimensional manifold with one set of latent vectors given by the image coordinates of the center of the disk. We use this latent space $\mathbf{y} = [y_1 y_2] \in [10, 90] \times [10, 90]$ which was also used while generating the images. The neighbourhood size parameters are set to $k_1 = 14$ and $k_2 = 7$ and $m = 2$. Here we use the method described in section 2.4 to set the number of normal space components d , with the value of energy threshold t to 0.99. As was the case

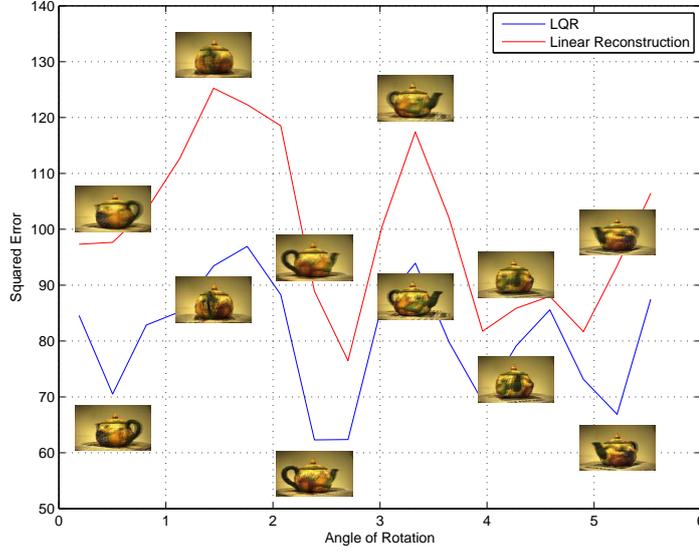


Figure 3.9: **Squared Error v Angle of Rotation on Teapot dataset**, Variation in squared error between original image and reconstruction for both linear reconstruction and LQR. Average LQR Error = 80.44, Average Linear Error = 99.97.

for the teapot set, we again use $f_{\mathcal{M}}(N_{y_q}^{k_1})$ for finding the tangent space since otherwise the tangent vectors found are inaccurate (see fig. 3.6).

Fig. 3.10 shows 1000 samples from the disk dataset out of which $N = 800$ are randomly set as training points and the remaining as test points. Reconstruction for one of these is shown in fig. 3.11. Regularization parameters are set to $\lambda_L = 10^{-3}$ and $\lambda_Q = 1$. Using a PCA eigenenergy threshold value of $t = 0.99$, we extract $(m + d) = 10$ principal components (fig. 3.11a), 2 tangent space and 8 normal space components. Linear regression on the tangent space using $k_2 = 7$ neighbours is shown in figs. 3.11b and 3.11c, and quadratic regression along $d = 1$ is shown in fig. 3.11d. The surface found from quadratic regression fits well to the points in the neighbourhood, with a visible improvement from the linear reconstruction. Finally, we compare the images reconstructed from LQR and linear reconstruction in fig. 3.11e.

Fig. 3.12 illustrates the effect of adding normal space components to the reconstruction. We perform quadratic regression along the $(m + d)^{th}$ component to obtain the projection of

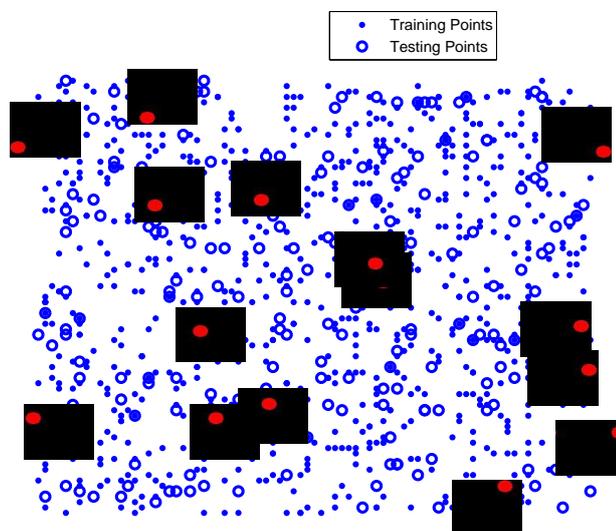


Figure 3.10: **2-dimensional disk dataset**, Latent vectors of the images in the 2-dimensional disk dataset, with some images overlaid on top.

the query point in it. Each added component leads to decrease in the blur in the images, however this effect is less pronounced for higher values of d which is expected since here the effect of noise becomes pronounced. Looking at the regression surfaces, we can see that the second order assumption falls short at high d .

Out of the 200 test images, LQR improved upon linear reconstruction in 183. Some examples from these along with their reconstructions is shown in fig. 3.13. LQR is able to successfully reduce the blurring in linear reconstruction in most cases, and images 2,3 and 5 show significant improvements. The 17 images where LQR failed were mostly points on the boundary of the manifold (where the disk is at the edge of the image), fig. 3.14. Here the tangent space is not computed accurately and the query point is also extrapolated beyond the support of training data. This is similar to the effect observed for swiss-roll and spiral manifolds and is expected. A visualization of how the error varies on the manifold is given in fig. 3.15 which shows the latent vectors in \mathbb{R}^2 of both the training and test points. For the test point, the size of the dot is proportional to the reconstruction error.

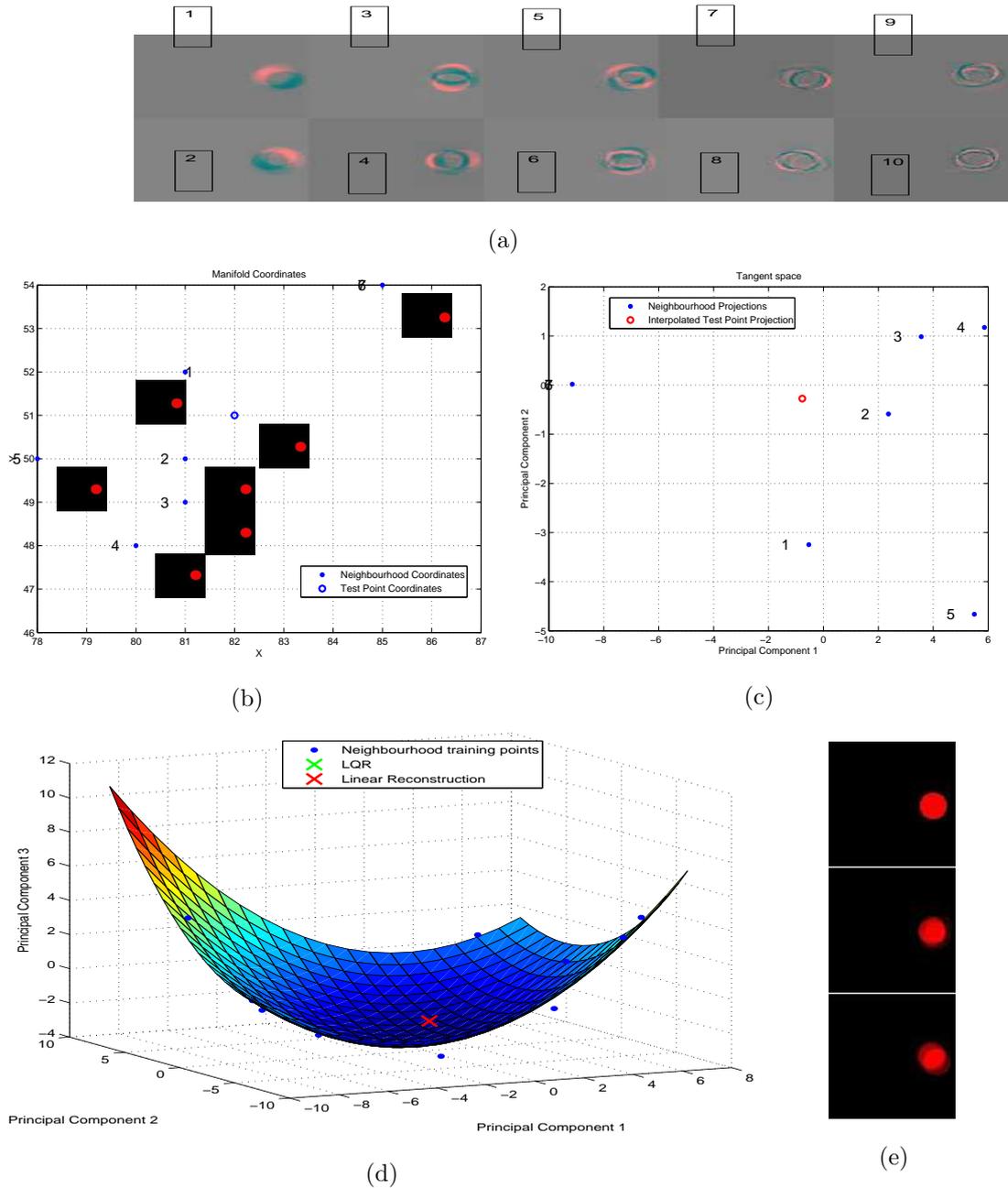


Figure 3.11: **LQR on the Disk dataset**, Step-by-step LQR procedure on the disk dataset. (a) $(m + d) = 10$ principal components, (b) Latent vectors $N_{\mathbf{y}_q}^{k_2}$, (c) Their projections onto $\hat{T}_{\mathbf{x}_q}\mathcal{M}$, (d) Quadratic regression along \mathbf{u}_3 ($d = 1$), (e) Found reconstructions (top-original image, middle - LQR, bottom - linear reconstruction). $k_1 = 14$, $k_2 = 7$, $\lambda_L = 10^{-3}$, $\lambda_Q = 1$.

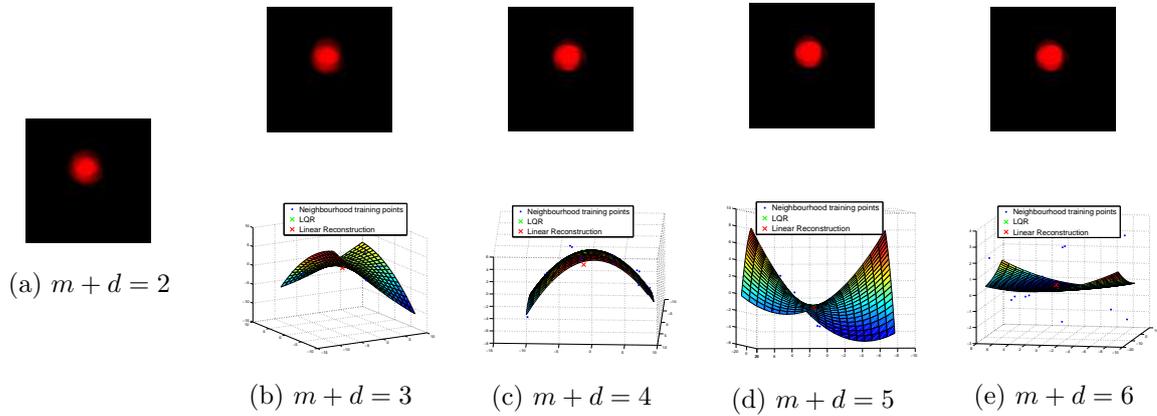


Figure 3.12: **Effect of Normal Space components**, The change in a reconstructed image as each higher order principal component is added (top row), and the quadratic regression along these components (bottom row). The first image on far left is the reconstruction with just linear interpolation on $\hat{T}_{\mathbf{p}}\mathcal{M}$.

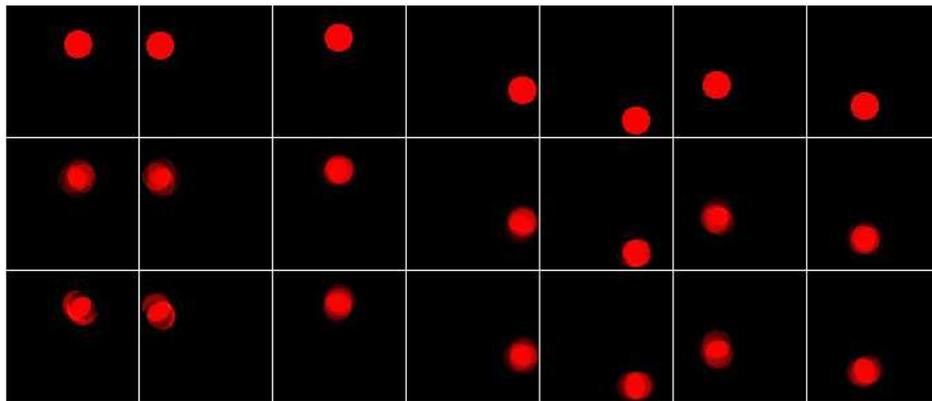


Figure 3.13: **Local Quadrature v Linear Reconstruction on Disk dataset**, Top row - Original test images, Middle Row - LQR, Bottom Row - Linear Reconstructions. $k_1 = 14$, $k_2 = 7$, $\lambda_L = 10^{-3}$, $\lambda_Q = 1$.

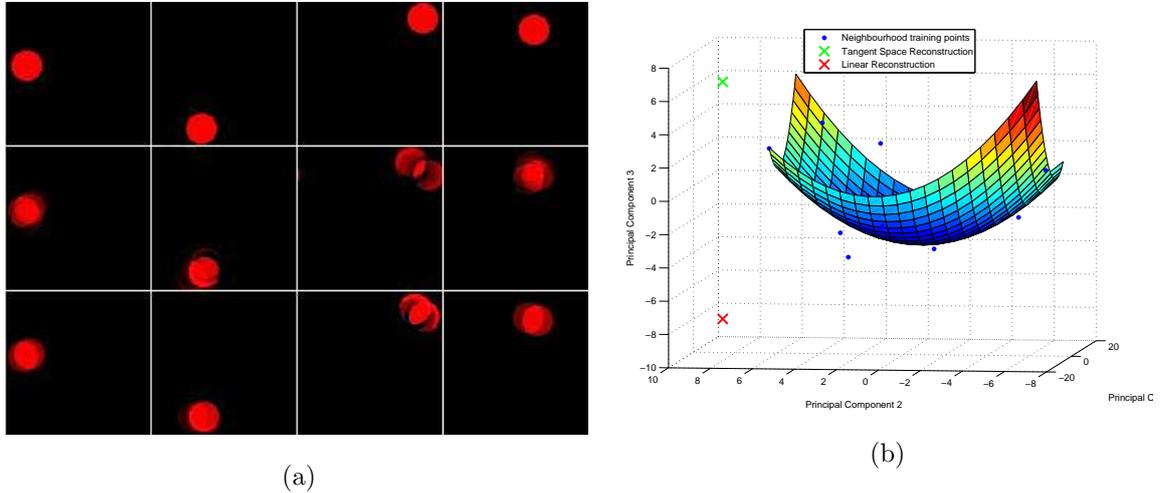


Figure 3.14: **LQR failures on Disk dataset**, (a) Example images where LQR performs worse than linear reconstruction (mostly boundary images). (b) Quadratic regression along first orthogonal component for image 3. Query point is extrapolated far beyond the support of training data. $k_1 = 14$, $k_2 = 7$, $\lambda_L = 10^{-3}$, $\lambda_Q = 1$.

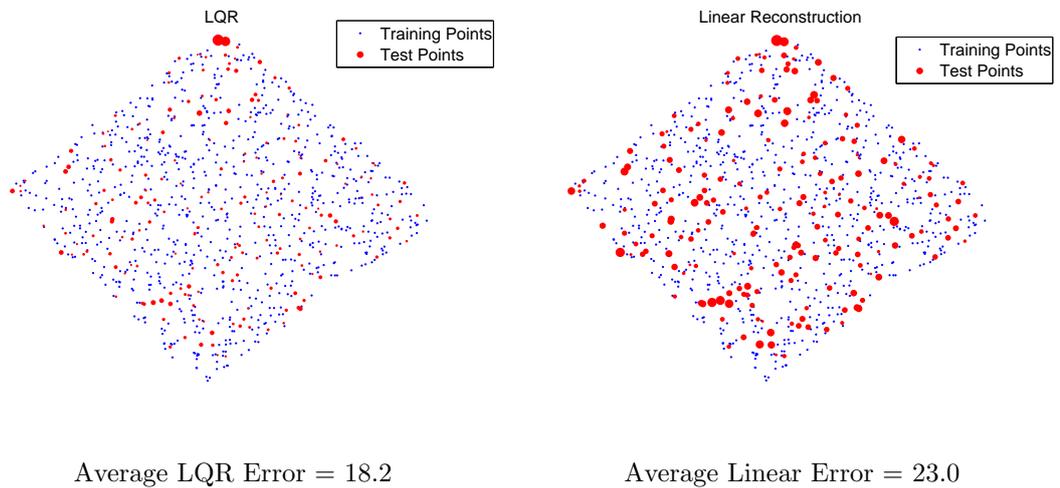


Figure 3.15: **Error Variation on Disk Dataset**, A visualization of the squared error between original image and found reconstruction on the disk dataset for both the linear and quadratic methods. Size of the marker for test data is proportional to the squared error.

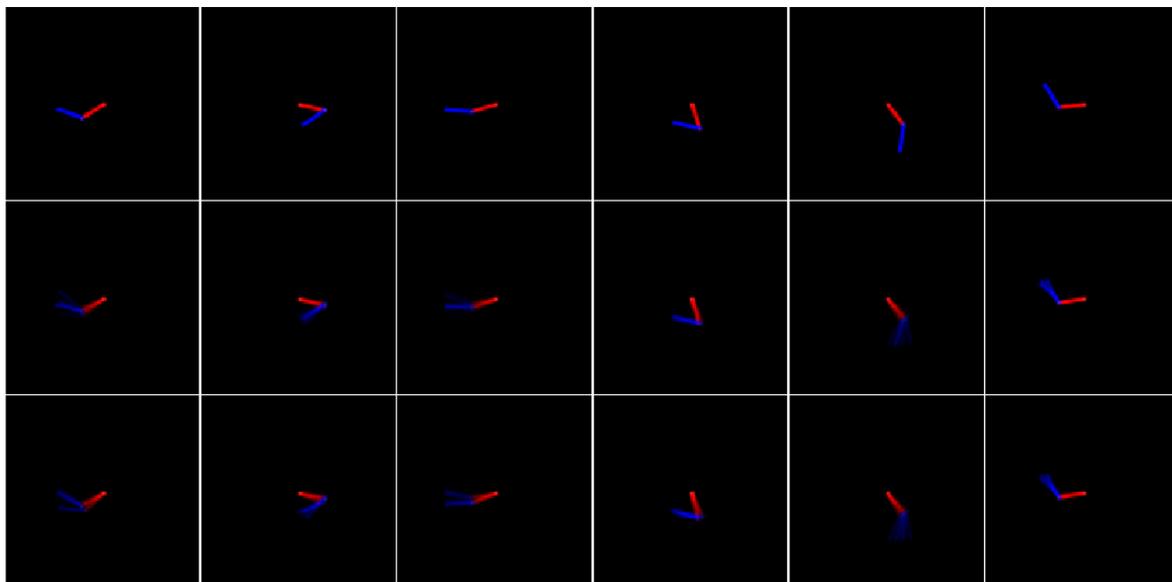


Figure 3.16: **Reconstructions on Robot-Arm Dataset**, Reconstruction results on the robot-arm test images. Top - Original Data, Middle - LQR (MSE = 8.74), Bottom - Linear Reconstruction (MSE = 9.63). $k_1 = 14$, $k_2 = 7$, $\lambda_L = 10^{-3}$, $\lambda_Q = 1$.

3.2.3 Planar Articulated Robot Arm

Local planning in the Probabilistic Roadmap algorithm for robot-motion planning requires reconstructing high-dimensional workspace data from the robot's configuration space (C -space). LQR can be used here, and to demonstrate the application we show its performance on the **Robot Arm** dataset. This set contains 735 images of a planar articulated robot arm with two links free to move about. It has 2 degrees of freedom - the angles made by the links with the horizontal. Images lie in $\mathbb{R}^{100 \times 100 \times 3}$, and the configuration space is given by $[0, \pi] \times [0, \pi]$. Out of the 735 total images we randomly selected 135 for testing and reconstructed them from their known neighbours in the remaining $N = 600$ images. Fig. 3.16 shows a few example reconstructions on this dataset. The neighbourhood sizes were set to $k_1 = 14$ and $k_2 = 7$, and the tangent space was computed from $f_{\mathcal{M}}(N_{\mathbf{y}_q}^{k_1})$. Number of normal space components d was set the same way as for the disk dataset before with a threshold of $t = 0.99$. Mean Squared Error over all the 135 test points for LQR was 9.3% lower than the MSE for Linear Reconstruction.

3.3 Video Frame Reconstruction

One possible application of any reconstruction procedure in image processing is frame interpolation. Many video sequences have only a few degrees of freedom and can be viewed as a set of points near a low-dimensional manifold embedded in a high-dimensional space [22]. Even where the sequence does not actually lie on a manifold, the video can still be seen as a 1-dimensional trajectory in the space of all images and NLDR methods can be used to assign latent vectors to each frame [23]. Motivated by this, we explore whether we can use LQR for video compression and frame interpolation. For example, a typical compression scheme would involve finding out latent vectors for the frames using NLDR and uniformly downsampling the video by removing certain frames at the transmitter side. Then at the receiver side, LQR can be used to find the missing frames from their latent vectors and the remaining 'train' frames.

The foreman sequence consists of 300 frames of a construction worker first talking into the camera, and then pointing towards a site. Initial part of the video shows the worker moving his head about after which there is abrupt motion and the camera view shifts to a construction site. After this the last few frames are an almost static view of a construction site. ISOMAP with $k = 4$ returns an embedding for this sequence in \mathbb{R} with a residual error of 0.0003. Fig. 3.17 shows reconstruction on 5 uniformly selected frames from their neighbouring frames. Both the neighbourhood size parameters were set to $k = k_1 = k_2 = 4$, and the regularization parameters were set to $\lambda_L = 10^{-3}$ and $\lambda_Q = 1$. The number of normal components d for quadratic regression was set in the same way as for the disk dataset by choosing a threshold energy we want to retain in the principal components (see eq. 2.20). $t = 0.99$ was used which in practice, for a neighbourhood size $k_1 = 4$ always yielded $(m + d) = 3$.

The initial part of the sequence has the worker moving his head about in a few directions against a fixed background. Since such a motion has only a few degrees of freedom, this part 'conforms' well to a manifold and our quadratic regression works well. The first three frames shown in fig. 3.17 exemplify this, and LQRs have much lesser blur than corresponding linear

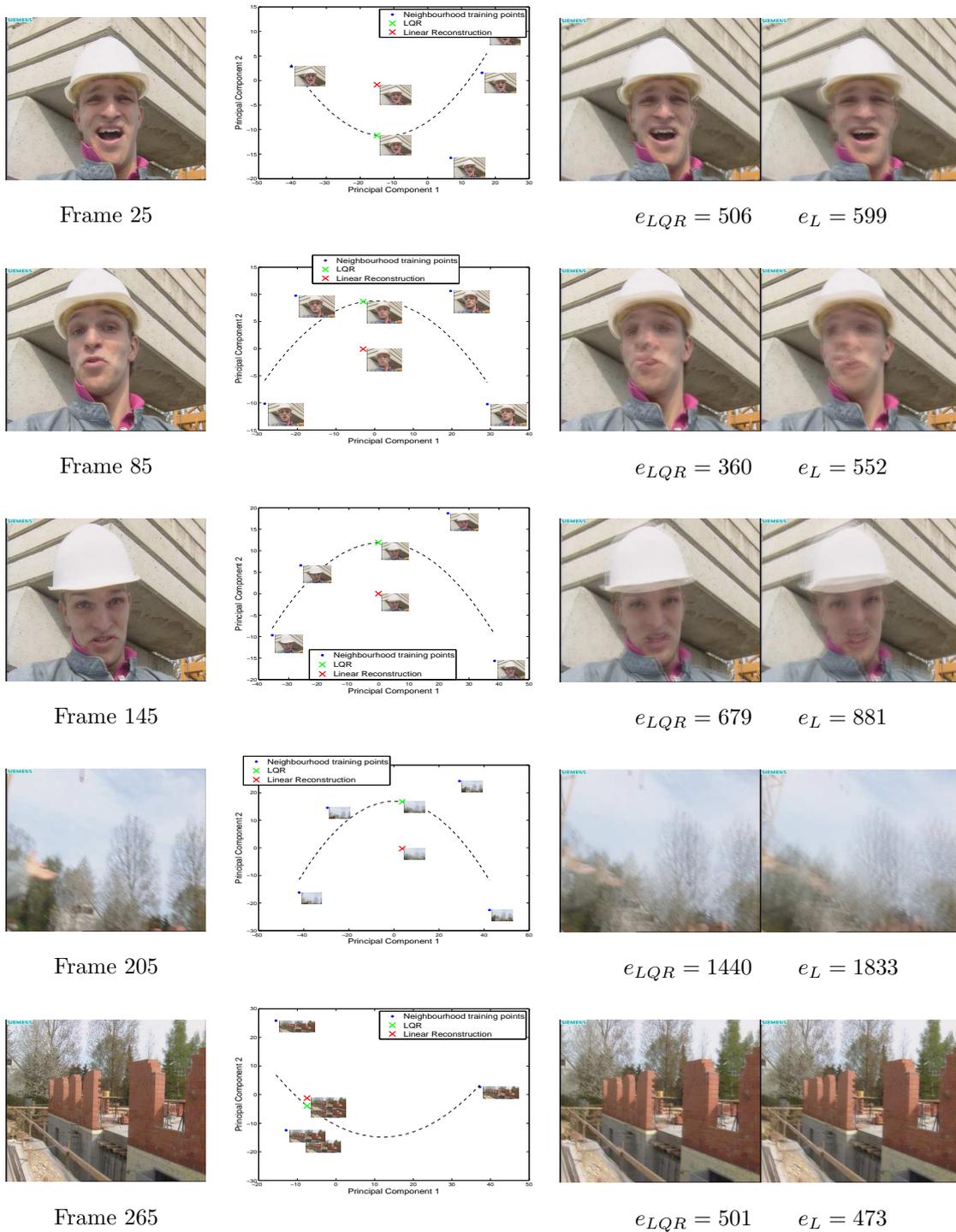


Figure 3.17: **Frame reconstruction on Foreman Sequence**, Uniformly selected frames were reconstructed using both quadratic and linear reconstruction. First column - Original frame. Second Column - Regression for $d = 1$ on estimated tangent space at that frame. Third column - Quadratic and Linear reconstructions. Squared errors are written below the reconstructions. $k = k_1 = k_2 = 4$, $\lambda_L = 10^{-3}$, $\lambda_Q = 1$.

ones. Regression along $d = 1$ orthogonal component is also shown in the second column along with the linearly interpolated result. Frame 4 belongs to the part of the sequence with abrupt motion where background also changes and both the reconstructions have a high amount of blur and a large squared error. Due to the abrupt motion neighbouring frames are spaced far apart in the image space, and interpolation naturally becomes a harder problem, however LQR is still able to significantly improve the linear reconstruction and has a lower squared error with lower blur. The last frame shown is from the part of the video with an almost static shot of a construction site. At this part of the video there is not much change from one frame to the next and both reconstructions are almost identical.

Chapter 4

Conclusions and Discussion

In this work we have presented an algorithm *Local Quadrature Reconstruction (LQR)* for out-of-sample reconstruction, which estimates a non-linear function $f_{\mathcal{M}} : \mathcal{S} \rightarrow \mathcal{M}$ for reconstructing high-dimensional data from its low-dimensional latent variables. Our procedure is based on fitting a differential geometric model to a local neighbourhood on the manifold such that each point in the neighbourhood can be expressed in terms of its tangent space components and a small number of normal space components. We assume a linear form over the tangent space and a quadratic form over the normal space. Tangent and Normal vectors along directions of maximum data variance are extracted using Principal Components Analysis on a set of k points sampled from the neighbourhood, and parameters of the geometric model are found using least squares regression. Reconstructions are found by linearly interpolating between the latent space and tangent space.

Important features of LQR include its low computational complexity of $O(kNm + k^3 + n(m + d))$ which ensures the reconstruction time does not blow up with total number of points on the manifold or dimension of the ambient space. We note that the time complexity is comparable to simple linear interpolation from k -nearest neighbours [1], with better quality of reconstructions observed on all datasets tested (with n upto 30000, but intrinsic dimension $m \leq 2$). Also, there is no training time as such for LQR, since no global optimization or learning needs to be performed. Though we restrict our attention to the latent spaces generated by ISOMAP or known from the functions generating the data, we expect LQR

to generalize well to other latent spaces recovered by other NLDR algorithms such as LLE, MVU etc.

A significant limitation of the LQR procedure arises since the principal components are not well-aligned with the principal directions (eigenvectors of the hessian matrix, along which principal curvatures are defined). This means that the quadratic reconstruction needs to consider also the off-diagonal terms, so that the number of neighbourhood points needed grows as $O(m^2)$. This limits the effectiveness of LQR to situations with low intrinsic dimension, since increasing neighbourhood size k causes several problems. First, we need an exponentially greater number of samples on the manifold to ensure all the k neighbours are on a small patch of the manifold. Secondly, the computation time of the algorithm varies as k^3 or m^6 and any advantage we have of remaining in a local neighbourhood vanishes as m becomes large. Finally, the number of parameters we need to estimate also increases as m^2 along each normal space component. At large values of m the curves are overfit due to the large number of parameters and hence the quality of reconstructions also decreases. For these reasons we limit our analysis to manifolds with low intrinsic dimension, which still covers a wide variety of practical datasets.

In chapter 3 we analyse the performance of LQR on several point sets and image sets sampled from hypothetical manifolds in high-dimensional spaces. Results on point sets such as the 2-d swiss-roll and 1-d spiral in \mathbb{R}^3 show how the second order assumption is successful in preserving the shape of the reconstructed manifold while linear reconstruction is not. Image sets used for demonstration include a rotating teapot, 2-d translation of disk robot, and 2-d planar robotic arm all of which are synthetically generated. We refer to the problem of blurring, where linearly reconstructed images do not have sharp boundaries, and give several examples where LQR is able to significantly reduce the blurring over linear reconstructions. Mean Squared Error (MSE) across all test points is also computed and LQR has lower MSE than linear reconstruction across all datasets studied. Cases where LQR fails, or has a higher error compared to linear reconstruction, are mostly boundary points where the query point is extrapolated far beyond the support of neighbouring samples. However at these points even

linear reconstruction has a very high error.

Potential applications of LQR lie in the image domain and include compression and interpolation. Compression can be achieved by retaining only low-dimensional latent vectors instead of high-dimensional images. Interpolation can be used for frame-rate upconversion or reconstruction of dropped frames of videos. We demonstrate this application on the *foreman* video sequence. It is interesting to note how LQR provides significant improvements on portions where the video has only a few modes of variability, which is expected since these portions satisfy the assumptions we make about the structure of the data. LQR can also be applied for generating novel views of an object from a set of known views for analysis in computer vision (ex: rotating teapot). In Robot motion planning, the method can be used to generate interim points between two "nearby" poses that are both collision free. Such points are useful in planning a local path between these points - if collision-free, such paths can be added to a graph of free paths called the roadmap.

4.1 Future Work

We have already discussed one limitation of LQR - inability to generalize to manifolds with a large number of degrees of freedom or m . For these manifolds we need a large number of neighboring samples and regression does not work due to overfitting. However, many practical datasets such as the MNIST handwritten digits, or several *Face* image datasets assume an intrinsic dimensionality as large as $m = 30$. It is still not clear to the authors how the effect of curvature can be exploited for such cases.

Another aspect of LQR which requires further investigation is its applicability. The manifolds studied in this paper are all 'clean' manifolds in the sense we assume no noise during sampling and that the manifolds have a uniform structure throughout and a constant well defined intrinsic dimension. This may not be the case for high-dimensional data in general and it remains to be seen how well LQR performs when these conditions do not hold.

References

- [1] L. Saul and S. Roweis, “An introduction to locally linear embedding,” *unpublished*. Available at: <http://www.cs.toronto.edu/~roweis/lle/publications.html>, 2000.
- [2] P. Dollár, V. Rabaud, and S. Belongie, “Learning to traverse image manifolds,” *Advances in neural information processing systems*, vol. 19, p. 361, 2007.
- [3] R. Souvenir, Q. Zhang, and R. Pless, “Image manifold interpolation using free-form deformations,” in *Image Processing, 2006 IEEE International Conference on*, pp. 1437–1440, IEEE, 2006.
- [4] B. I. Dundas, “Differential topology,” *Johns Hopkins University*, 2002.
- [5] L. W. Tu, *An introduction to manifolds*. Springer Science+ Business Media, 2011.
- [6] L. Cayton, “Algorithms for manifold learning,” *University of California, San Diego, Tech. Rep. CS2008-0923*, 2005.
- [7] C. J. Burges, “Dimension reduction: A guided tour,” *Foundations and Trends® in Machine Learning*, vol. 2, no. 4, pp. 275–364, 2009.
- [8] L. Van der Maaten, E. Postma, and H. Van den Herik, “Dimensionality reduction: A comparative review,” *Journal of Machine Learning Research*, vol. 10, pp. 1–41, 2009.
- [9] B. Scholkopf, A. Smola, and K.-R. Müller, “Kernel principal component analysis,” in *Advances in kernel methods-support vector learning*, Citeseer, 1999.
- [10] V. d. S. Joshua B. Tenenbaum and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, pp. 2319–2323, December 2000.

-
- [11] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf, “A kernel view of the dimensionality reduction of manifolds,” in *Proceedings of the twenty-first international conference on Machine learning*, p. 47, ACM, 2004.
- [12] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet, “Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering,” *Advances in neural information processing systems*, vol. 16, pp. 177–184, 2004.
- [13] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [14] M. W. Trosset and C. E. Priebe, “The out-of-sample problem for classical multidimensional scaling,” *Computational statistics & data analysis*, vol. 52, no. 10, pp. 4635–4642, 2008.
- [15] S. Xiang, F. Nie, Y. Song, C. Zhang, and C. Zhang, “Embedding new data points for manifold learning via coordinate propagation,” *Knowledge and Information Systems*, vol. 19, no. 2, pp. 159–184, 2009.
- [16] Y. Yang, F. Nie, S. Xiang, Y. Zhuang, and W. Wang, “Local and global regressive mapping for manifold learning with out-of-sample extrapolation,” in *Proceedings of AAAI*, pp. 649–654, 2010.
- [17] H. Strange and R. Zwigelaar, “A generalised solution to the out-of-sample extension problem in manifold learning,” in *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.
- [18] C. Zhang, J. Wang, N. Zhao, and D. Zhang, “Reconstruction and analysis of multi-pose face images based on nonlinear dimensionality reduction,” *Pattern Recognition*, vol. 37, no. 2, pp. 325–336, 2004.
- [19] H. Tyagi, E. Vural, and P. Frossard, “Tangent space estimation for smooth embeddings of riemannian manifolds,” *arXiv preprint arXiv:1208.1065*, 2012.

-
- [20] D. N. Kaslovsky and F. G. Meyer, “Overcoming noise, avoiding curvature: Optimal scale selection for tangent plane recovery,” in *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pp. 892–895, IEEE, 2012.
- [21] K. Q. Weinberger and L. K. Saul, “Unsupervised learning of image manifolds by semidefinite programming,” in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II–988, IEEE, 2004.
- [22] R. Pless and R. Souvenir, “A survey of manifold learning for images,” *IPSJ Transactions on Computer Vision and Applications*, vol. 1, no. 0, pp. 83–94, 2009.
- [23] R. Pless, “Image spaces and video trajectories: Using isomap to explore video sequences,” in *Proceedings of the Ninth IEEE International Conference on Computer Vision*, vol. 2, p. 1433, 2003.