# Similarity Preserving Dimensionality Reduction for Image Data

**by**

**Gitam Shikkenawis**

**201221004**

A thesis submitted in the partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY
in
Information and Communication Technology
to

**Dhirubhai Ambani Institute of Information and Communication Technology**
**Gandhinagar, India**

**August 2016**

# Declaration

This is to certify that

i) the thesis comprises my original work towards the degree of Doctor of Philosophy in Information and Communication Technology at DA-IICT and has not been submitted elsewhere for a degree.

ii) due acknowledgment has been made in the text to all other material used.

Signature of Student
Gitam Shikkenawis

# Certificate

This is to certify that the thesis work entitled *Similarity Preserving Dimensionality Reduction for Image Data* has been carried out by *Gitam Shikkenawis (201221004)* for the degree of Doctor of Philosophy in Information and Communication Technology at this Institute under my supervision.

Thesis Supervisor
Prof. Suman K. Mitra

# Acknowledgments

I am already emotional as I am writing this. The journey that I started not so long ago as it seems, have so many wonderful memories, experiences and there are so many people who helped me to get through this. A note of gratitude is just not enough, I owe a lot more to them.

The first one has to be my guide, Prof. Suman K mitra. He might not remember but he was the first person who interacted with me on the day of my M. Tech admission, and has guided me through all the years I have spent at DAIICT. His constant support, motivation and encouragement kept me going during both M. Tech and PhD. There were times when things were not going well, it was sir's belief and patience which helped me pass through that phase. He provided me with the necessary direction and encouragement that any student may hope for from an adviser. Being Dean of Academic Programs at DAIICT, he always had a lot of administrative responsibilities, but he never let it affect the research work, and took part in all the discussions with enthusiasm. I admire his dedication and understanding and wish to inherit the same. Though this journey is coming to an end, my bonding with sir will be the same throughout my life. I would also like to thank his wife, Swati Ma'am, for all the love and care she has always bestowed upon me.

I would like to extend my gratitude towards Prof. S. C. Sahasrabudhe, ex-director, DAIICT for always taking interest in my work, motivating and supporting me. It was Prof. Sahasrabudhe who introduced me to Locality Preserving Projection, on which I worked during both M. Tech and PhD. Even on his last day at DAIICT, he spared his time for me and had a detailed discussion on my work. I will always be thankful to him for constant involvement, suggestions and persistence on quality work.

This thesis is shaped by comments and suggestions by my Research Progress Seminar (RPS) committee members, Prof. Manjunath V. Joshi, DAIICT and Prof. V. Sunitha, DAIICT. It was a great pleasure to present parts of this thesis in every semester where the work was evaluated by Prof. Joshi's experience and Prof. Sunitha's enthusiasm and encouragement. I would also like to thank Prof. Ajit Rajwade, ex-DAIICT faculty and currently a faculty at Dept. of Computer Science & Engineering at Indian Institute of Technology, Bombay (IIT-B). He introduced me to the world of image denoising and transformed domain techniques for image restoration, basis of second part of my thesis. I would also like to express my gratitude to all the faculty and staff members at DAIICT, a special thanks to Mr. Soman Nair, registrar, DAIICT, for his kind support and encouragement. My sincere appreciation to Tata Consultancy Services (TCS) for providing me the financial assistance

through my dissertation. They have been very involved and active during the whole process. Interactions with researchers from TCS research labs were very encouraging. I would also show my gratitude towards thesis reviewers Prof. Ajay Kumar, Associate Professor, Department of Computing, The Hong Kong Polytechnic University, Hong Kong and Prof. Malay Kumar Kundu, Professor, Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India for spending their valuable time in reviewing and giving suggestions to further improve the thesis.

Apart from the research work, I had a lot of memorable moments, fun and outings. I have made a few bonds which will remain the same forever. Ashish Bhaiya (Ashish Phophalia), is the first in the list, with whom I have shared a great friendship during last six years. I will never forget our all time favorite campus walks, intense discussions on all sorts happy and sad moments and fights for no reason. Later, we were joined by Minakshi, his wife, and I carried the same bond with her. A heartfelt thanks to both of them for always supporting and helping me. It was a great experience to work with other students, especially all the office mates in Prof. Mitra's office. I had great time discussing technical and non-technical aspects with Purvi, Deshna, Twinkle, Ravi, Radhika, Prashant, Rupsa, Bhavesh sir, Mukesh sir, Milind and Shrishail. It is hard to believe that six years of stay at DAIICT is coming to an end. With all this, I can not miss out to thank my friends Heta, Yasmin, Komal and Prakruti, who were always a call away.

I owe deepest gratitude to my parents Vandana and Chandrahas Shikkenawis. Without them, this journey would have never been accomplished. Apart from their continuous support, motivation and faith in me, they have sacrificed a lot.. My mother has always handled all my frustrations and anger with patience. On the other hand, my father stayed alone and travelled so much for my mother to be with me. It must have been very difficult for them but they never made me feel it. Hopefully, this moment will be joyous, cheering and satisfying for them. I would like to thank all my family members for the support and encouragement throughout this journey. Thank you Piyush, for always being my support system and being there virtually through all the ups and downs. A note of thanks to Salil dada, for always guiding me and being my big, caring brother. This can not be complete without thanking my partners in crime Ayush, Amey and Radhika. I received a very warm welcome in my new family post my wedding, and they all have been very kind. My in-laws, Jyotshana and Bipinchandra Joshi, have been very encouraging and supportive of my work, and have always kept my best interests in mind. Last, and most definitely not the least, my better half, Brijesh, a simple thanks would not be enough for him. He has lived this dream with me, supported me at all the stages and motivated me during low times. He never showed displeasure over my disappearance for so many months and kept me going. I am really fortunate to have him as my best friend and life partner, he has made my life even more beautiful.

Gitam Shikkenawis

# Abstract

Data collection and storage capabilities have increased manifold in last few decades, leading to information overload. Number of variables used to represent each data observation is called dimension of the data and dealing with large dimensions is a challenging task. Images have become a source of such large data which is increasing day by day with advances in image capturing devices and demand of high resolution images. Images typically consist of large dimensions and processing that becomes very difficult even for machines. Dimensionality reduction techniques learn a compact representation of such data by exploring the properties such as correlation, pairwise distances, neighborhood structure etc. The idea is to retain these properties in lower dimensional representation as well, inducing minimum information loss. Early age techniques of dimensionality reduction preserve the global structure of the data, but, many a times, local manifold structure is more important than the global Euclidean structure. This thesis is an attempt to develop robust and powerful dimensionality reduction technique based on similarity preservation for image data. In particular, the thesis emphasizes on the dimensionality reduction techniques those are linear in nature and are based on preserving the local relationship of the image data.

In this work, Locality Preserving Projection (LPP), that preserves the local structure of data is studied and its various extensions are proposed. LPP works on the concept that neighboring data points in the high dimensional space should remain neighbors in the low dimensional space as well. Ambiguities in regions having data points from different classes close by, less reducibility capacity, data dependent parameters, ignorance of discriminant information, non-orthogonality of the basis, vectorized processing are some of the issues with conventional LPP. Some of the variants of LPP have been introduced that try to resolve these problems. Discriminant information, if considered, can play vital role in obtaining separation between different classes. Variants of LPP, considering not only the local structure, but also the dissimilarity between the data points are proposed in the first part of the thesis. Data representation, face and facial expression recognition experiments are performed using the proposed dimensionality reduction frameworks.

Though, conventional LPP and its variants well preserve the local manifold structure, the basis vectors learnt are not orthogonal. Having orthonormal basis is advantageous in many applications and makes data reconstruction much simpler. Orthogonal LPP (OLPP), orthogonalizes the basis of LPP in an iterative manner by selecting basis vector orthogonal to all the previously learnt basis vectors. Apart from dimensionality reduction and data representation, a new application of OLPP is explored in the form of Image Denoising. Recent state of the art approaches for image denoising work on two major hypotheses i.e. non-local self-similarity and sparse linear approximations of the data. Locality preserving nature of the proposed approach automatically takes care of self-similarity present in the image and clusters similar patches in the projection domain. A framework for inferring an orthonormal set of dictionary vectors using OLPP from the noisy data is presented and denoising is performed in the transformed domain. OLPP appears to be a natural and suitable mechanism for the task of image denoising.

Vectorized nature of OLPP requires high dimensional data to be converted to vector format, hence may lose spatial neighborhood information of raw data. On the other hand, processing two dimensional data (mainly images) directly, not only preserves spatial information, but also improves the computational efficiency considerably. Two Dimensional Orthogonal Locality Preserving Projection (2D-OLPP) is expected to learn the transformation from two dimensional data itself. Like OLPP, a global image denoising framework using 2D-OLPP is suggested. As the performance of image denoising is highly dependent on clustering of similar patches, discriminant information is also considered in the formulation of Two Dimensional Orthogonal LPDP (2D-OLPDP). This technique is expected to achieve superior clustering of image patches and thereby enhance the quality of denoising.

Various techniques based on similarity preserving dimensionality reduction proposed during the course of this thesis are applied on face recognition and image denoising tasks. Extensive experiments for both the applications have been carried on benchmark databases and show promising results. The face recognition accuracies seem to improve over the existing locality preserving dimensionality reduction approaches, and on the other end, image denoising performance using the proposed frameworks is at par with or surpasses the state of the art image denoising approaches with much less computational complexity.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Advances in data collection and storage capabilities during recent decades have led to information overload. Traditional statistical methods break down partly because of the increase in the number of observations, but mostly because of the increase in the number of variables associated with each observation. Dimension of the data is the number of variables that are measured for each observation [11]. Analyzing the data in high dimensional space is a challenging task, generally referred to as the curse of dimensionality [12]. It becomes difficult to deal with very high dimensional data, be it classification, recognition or analysis task [13]. Usually, in such a high dimensional space, only a few data points are meaningful. Also, in many cases, not all the measured variables are important for understanding the underlying phenomena of interest [12]. In other words, the distribution of natural/meaningful data points in high dimensional space is non-uniform and is generally concentrated around some kind of low dimensional structures.

In recent times, research is going on in this direction to handle to issue of large dimensional data. The present work is an attempt towards the same direction. One of the widely used such large data comes from the image databases that are increasing day by day. A study of handling large data such as images is of demand of recent days. The thesis addresses a few of such issues by proposing some of the techniques of image data dimensionality reduction. In particular, this thesis emphasizes on the dimensionality reduction techniques those are linear in nature and are based on preserving the local relationship of the data and there by addressing the global manifold of the image data.

## 1.1 Overview of Dimensionality Reduction

Dimensionality reduction mitigates the curse of dimensionality and other undesired properties of high dimensional spaces [14]. As a result, dimensionality reduction facilitates, among others, classification, visualization, and compression of high-dimensional data [15]. In literature, many linear and non-linear dimensionality reduction techniques have been proposed [16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]. These techniques try to represent the original high dimensional data in a compact manner keeping the information content intact. A brief overview of some dimensionality reduction techniques is presented in this chapter.

### 1.1.1 Linear Dimensionality Reduction Techniques

All the linear dimensionality reduction approaches learn a transformation matrix *i.e.* a set of basis vectors, that are used to map the data from original space to the newly found subspace, hence these approaches are also called domain transformation techniques. Linear approaches work on the assumption that the data lies on a linear manifold and hence linear in nature, thus a linear matrix $\mathbf{V}$, that is used to transform the data into the learned space. The set of $N$ training data points of $m \times n$ dimension is represented as $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_N}]$, $\mathbf{X}$ being the data matrix. Most of the techniques can not process data in matrix format, hence require the data to be converted in vector format. General framework of linear dimensionality reduction techniques is shown in Figure 1.1. The basis matrix of size $N \times (mn)$ is obtained, strongest $d$ $(d << mn)$ column vectors of which can be selected to project the data in lower dimensional subspace. Projection of $i^{th}$ data point $\mathbf{x_i}$ is computed as : $\mathbf{y_i} = \mathbf{V}^T\mathbf{x_i}$.

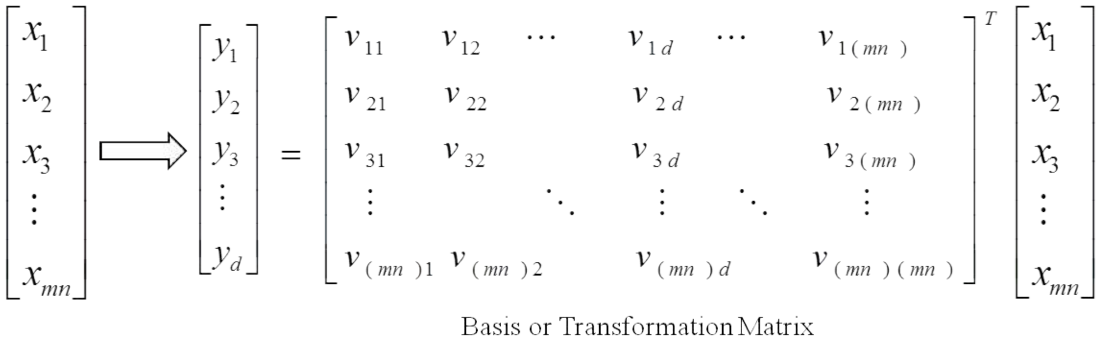

$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{mn} \end{bmatrix} \Longrightarrow \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_d \end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1d} & \cdots & v_{1(mn)} \\ v_{21} & v_{22} & & v_{2d} & & v_{2(mn)} \\ v_{31} & v_{32} & & v_{3d} & & v_{3(mn)} \\ \vdots & & \ddots & \vdots & \ddots & \vdots \\ v_{(mn)1} & v_{(mn)2} & & v_{(mn)d} & & v_{(mn)(mn)} \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{mn} \end{bmatrix}$
Basis or Transformation Matrix

Figure 1.1: General framework of linear dimensionality reduction techniques

Principal Component Analysis [16, 17] is one of the most popular linear dimensionality reduction methods. It which finds subspace from the data covariance information and preserves the directions of maximum variance. Linear Discriminant Analysis (LDA) [18], a supervised linear dimensionality reduction approach, maximizes inter class variability and minimizes intra-class variability in order to have better separation between different classes. Another popular linear approach is Independent Component Analysis (ICA) [19] that aims at making the components as independent as possible. A brief overview of these linear dimensionality reduction techniques is discussed in this section.

### Principal Component Analysis (PCA)

Principal Component Analysis (PCA) [16], [17] is one of the most widely used linear dimensionality reduction method which tries to find the basis vectors in the direction of maximum variance of the given data. To project the set of training images to the subspace using PCA, the transformation matrix $\mathbf{V}$ has to be found. The transformation can be represented by $\mathbf{Y} = \mathbf{V}^T\mathbf{X}$ where $\mathbf{Y}$ is the mapping of $\mathbf{X}$ in the subspace. The objective function for that is as follows:

$$max\sum_{i=1}^{m}\left(\mathbf{y_i}-\overline{\mathbf{y}}\right)^2;\overline{\mathbf{y}}=\frac{1}{m}\sum_{i=1}^{m}\mathbf{y_i} \tag{1.1}$$

The vectors that satisfy this objective function are the set of orthonormal vectors found by considering the eigenvectors of the covariance matrix $\mathbf{C}$ associated with $K << N^2$ largest eigenvalues. The projection is found by solving the eigenvalue problem: $\mathbf{CV} = \lambda\mathbf{V}$.

### Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) [18] aims at maximizing the separability between the classes i.e. to find the directions that are efficient for discrimination. The aim is achieved by minimizing the within class scatter and maximizing the between class scatter of the data points. The objective function of LDA is given below:

$$max\frac{\mathbf{V}^T\mathbf{S_B}\mathbf{V}}{\mathbf{V}^T\mathbf{S_W}\mathbf{V}}; \tag{1.2}$$

$$\mathbf{S_B}=\sum_{i=1}^{l}n_i\left(\mathbf{d^i}-\mathbf{d}\right)\left(\mathbf{d^i}-\mathbf{d}\right)^T;\mathbf{S_W}=\sum_{i=1}^{l}\left(\sum_{j=1}^{n_i}\left(\mathbf{x_j}^i-\mathbf{d^i}\right)\left(\mathbf{x_j}^i-\mathbf{d^i}\right)^T\right) \tag{1.3}$$

here, $\mathbf{S_B}$ is the between class scatter matrix, $\mathbf{S_W}$ is the within class scatter matrix, $l$ is the number of classes in the data set, $n_i$ - the number of samples in the $i^{th}$ class,

$\mathbf{d}$ - the mean vector of all the samples, $\mathbf{d^i}$ - mean vector of $i^{th}$ class and $\mathbf{x_j}^i$ is the $j^{th}$ sample of $i^{th}$ class. The LDA basis are obtained by solving generalized eigenvalue problem: $\mathbf{S_B V} = \lambda \mathbf{S_W V}$.

### Independent Component Analysis (ICA)

Another much used linear transformation method is Independent Component Analysis (ICA) [19]. Objective of this dimensionality reduction approach is to minimize the statistical dependence of components of the representation and capture the essential structure of data. The representation is found so as to make the components as independent as possible. The maximally independent components $\mathbf{S}$ are obtained as follows:

$$\mathbf{S} = \mathbf{VX} \tag{1.4}$$

where, $\mathbf{X}$ is the observed data vector $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_m})$, $\mathbf{V}$ is the transformation matrix and $\mathbf{S}$, the maximally independent components $\mathbf{S} = (\mathbf{s_1}, \mathbf{s_2}, \cdots, \mathbf{s_n})$. Computational complexity of obtaining these independent components is much higher than PCA.

I addition to these three benchmark dimensionality reduction approaches, various other dimensionality reduction approaches have also been proposed in literature. Recently, dimensionality reduction approach which works on the idea of preserving independence between multiple subspaces is proposed in [28]. Furthermore, this independent subspace structure preservation based approach also proves that for supervised learning, with data having $K$ classes, only $2K$ dimensions are required for structure preservation. A unified framework for subspace analysis, the approach, specially designed for face images, uses a face difference model by decomposing face into three different components, intrinsic difference, transformation difference and noise [29]. Unified subspace analysis using some of the most popular subspace methods i.e. PCA, LDA and Bayesian algorithm is carried out on these face difference components.

The approaches mentioned above, work on the assumption that the data is linear in nature. Data may not always lie on a linear manifold; many a times, the manifold on which data lies happens to be non-linear. In such cases, linear dimensionality reduction methods fail to discover the non-linearity present in the data. Non-linear dimensionality reduction approaches have been proposed in literature, that try to unveil the underlying non-linear manifold structure of the data. A brief overview of these approaches is presented in the next section.

### 1.1.2 Non-linear Dimensionality Reduction Techniques

There are many cases where the high dimensional data has perceptually very less independent degrees of freedom. Also, many a times, the data does not lie on or near a linear subspace. Given the high dimensional data that actually lies on very low dimensional manifold; the goal of nonlinear dimensionality reduction is to find meaningful low dimensional representation that captures the degrees of freedom of the data. In such cases, linear dimensionality reduction methods may fail because of the non-linearity present in the actual data. To show the non-linearity of the data, some data points from Swiss roll and S-shaped surfaces are sampled and shown in Figure 1.2.

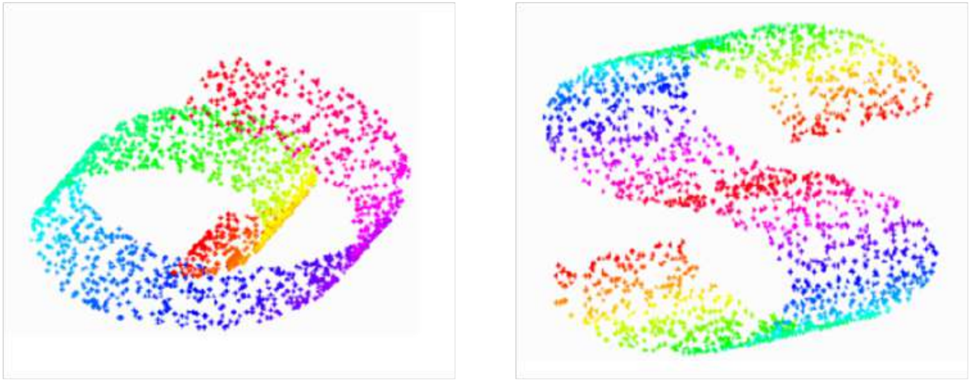


Figure 1.2: Examples of data originally embedded in non-linear subspace : Swiss Roll (left) and S-shaped (right) dataset

In such cases, to find the underlying low dimensional manifold of the data, non-linear dimensionality reduction methods have been proposed. Isomap [20] is a non-linear dimensionality reduction technique that preserves the intrinsic geometry of the data using geodesic distances between the data points. Like Isomap, Locally Linear Embedding (LLE) [21] finds a non-linear manifold by stitching small linear neighborhoods. It finds a set of weights that perform local linear interpolations to closely approximate the data. Laplacian Eigenmaps [22, 23], obtain lower dimensional embedding by drawing correspondence between the graph laplacian, the Laplace Beltrami operator, and connections to the heat equations.

#### Isometric Map (ISOMAP)

Isometric Map [20] is a nonlinear dimensionality reduction technique which preserves the intrinsic geometry of the data using the geodesic or shortest path distances between the data points. For the data lying on a nonlinear manifold, the straight line Euclidean distance may appear to be very less in cases where the actual geodesic distances are very high as shown in Figure 1.3 [1].

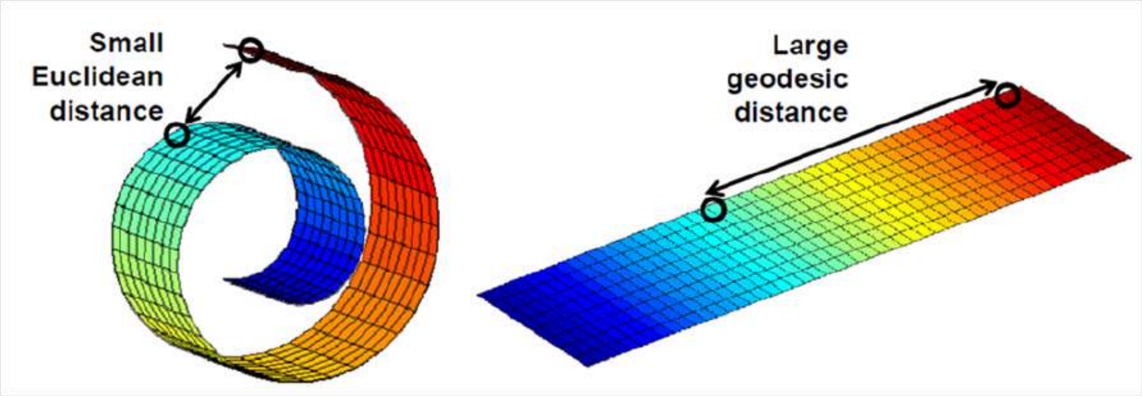


Figure 1.3: Example of Euclidean and Geodesic distances in case of Swiss Roll dataset

Hence, to find the actual geometry of the data, geodesic distances between the pair of points are found. The objective function of ISOMAP is:

$$min \left\| \tau \left( \mathbf{D_G} \right) - \tau \left( \mathbf{D_y} \right) \right\|_{L^2} \tag{1.5}$$

here, $\mathbf{D_G}$ is the matrix containing the shortest path distances between all pairs of points and $\mathbf{D_y}$ is the matrix of Euclidian distances between the projected points. The shortest path distances give the geodesic distance between the points which help to find the underlying manifold structure. Multidimensional Scaling (MDS) is used to find the basis functions of the newly found subspace and out of these, the $k$ eigenvectors corresponding to the lowest eigenvalues are chosen for reducing the dimensions. Isomap is locally linear and provides nonlinear aspects when stitched together.

### Locally Linear Embedding (LLE)

Locally Linear Embedding (LLE) [21] finds nonlinear manifold by stitching small linear neighborhood. The basic idea of LLE is to approximate each data points by a weighted linear combination of its neighbors and to find a low dimensional configuration of the data points so that the linear approximations are best preserved [10]. There are three basic steps of LLE algorithm.

[1] K. Chen, Isometric Feature Mapping, Modelling and Visualization of high dimensional data, University of Manchester

**Step 1: Defining the neighbors:**
Neighbors of all the data points combination of which will provide approximation to that data point are defined by either $K$-nearest neighbor approach or using the $\epsilon$-neighborhood based on Euclidean distance for each data point.
**Step 2: Finding reconstruction weights:**
Optimal weights that reconstruct each data point from its neighbors are obtained by minimizing the reconstruction error:

$$min \left\| \mathbf{x_i} - \sum_{j=1}^{k} W_{ij}\mathbf{x_{n_{ij}}} \right\|^2 s.t. \sum_{j=1}^{k} W_{ij} = 1 \tag{1.6}$$

where, $\mathbf{x_i}$ is a data point, $N_i = \mathbf{x_{n_{i1}}}, \mathbf{x_{n_{ik}}}$ is the neighbor set of $\mathbf{x_i}$ and $W_{ij}$ is the weight corresponding to each neighbor of $\mathbf{x_i}$.
**Step 3: Representation in low dimensional (embedding) space:**
Once the weights are found, the task is to map the high dimensional observation $\mathbf{x_i}$ to a low dimensional observation $\mathbf{y_i}$. This is done by choosing $\mathbf{y_i}$ that minimizes the embedding function subject two constraints as given below:

$$min \left\| \mathbf{y_i} - \sum_{j=1}^{k} W_{ij}\mathbf{y_{n_{ij}}} \right\|^2 s.t. \sum_{j=1}^{k} \mathbf{y_i} = 0, \frac{1}{N}\sum_{i=1}^{N} \mathbf{y_i}\mathbf{y_i}^T = \mathbf{I} \tag{1.7}$$

As opposed to Step 2, here the weights are known and optimum coordinates are to be calculated. Eigenvectors of $\mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$. Eigenvectors corresponding to the lowest eigenvalues are selected. The algorithm finds global minima of the reconstruction and embedding weights.

### Laplacian Eigenmaps

Laplacian Eigenmaps [22], [23] is another non-linear dimensionality reduction approach that has locality preserving properties and a natural connection to embedding. The algorithm for finding the low dimensional embedding of the high dimensional is proposed by drawing the correspondence between the graph laplacian, the Laplace Beltrami operator and the connections to the heat equations. Also, the locality preserving property makes it relatively insensitive to outliers and noise. Lower dimensional embeddings of data points $\mathbf{x_i}$, $\mathbf{y_i}$ are obtained by solving the following generalized eigenvalue solution:

$$\mathbf{Ly} = \lambda\mathbf{Dy} \tag{1.8}$$

here, $D_{ii} = \sum_i S_{ij}$ is a diagonal weight matrix, $\mathbf{S}$ is the weight matrix that is obtained by weighing the neighbors of each data point using the Heat Kernel function i.e. $S_{ij} = e^{\frac{-\|\mathbf{x_i}-\mathbf{x_j}\|^2}{t}}$ and $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is the laplacian matrix. Let $\mathbf{y_0}, \cdots, \mathbf{y_{m-1}}$ be the solutions to Equation 1.8, ordered according to their eigenvalues with $\mathbf{y_0}$ having the smallest eigenvalue. The image of $\mathbf{x_i}$ under the embedding into the lower dimensional space $R^k$ is given by $(\mathbf{y_1}(\mathbf{i}), \cdots, \mathbf{y_m}(\mathbf{i}))$.

Non-linear dimensionality reduction methods do yield impressive results on some benchmark artificial data sets, as well as on real world data sets. However, their non-linear property makes them computationally expensive. Moreover, they yield maps that are defined only on the training data points and do not provide any mechanism to project new data point in the lower dimensional subspace. The embeddings learnt using these non-linear approaches are data dependent and change with inclusion or exclusion of data points. Thus, new or unknown data point can not be projected on the existing lower dimensional representation of the data in subspace. Hence, non-linear dimensionality reduction approaches are not suitable for recognition related applications. To overcome these issues as well as explore the non-linearities of the data, similarity preserving linear dimensionality reduction approaches have been proposed.

### 1.1.3 Similarity Preserving Dimensionality Reduction

Similarity preserving dimensionality reduction approaches try to capture non-linearities present in the data by preserving neighborhood information of the data points yet learn a transformation matrix that can be used to map unknown data points in the projection space, thus linear in nature. Neighborhood Preserving Embedding (NPE) [24], is a linear approximation of LLE that aims to discover the local structure of the data manifold. Locality Preserving Projections (LPP) [26, 30], a linear approximation of Laplacian Eigenmaps, evolves around the idea that, data points that are similar to each other in the original space, should remain close in the newly found projection space as well. Thus, the aim is to obtain a subspace that detects the data manifold structure in the best possible way.

#### Neighborhood Preserving Embedding (NPE)

Neighborhood Preserving Embedding (NPE) [24] and Orthogonal Neighborhood Preserving Projection (ONPP) [25] are linear extensions of LLE. A linear transformation matrix is obtained which can now be used to project the data in the lower dimensional subspace. The procedure of learning the transformation matrix is similar to that of

LLE as explained in Section 1.1.2.

Neighbors of all the data points, combination of which will provide approximation to that data point, are defined by either $K$-nearest neighbor approach or using the $\epsilon$-neighborhood based on Euclidean distance for each data point. Optimal weights that reconstruct each data point from its neighbors are obtained by minimizing the reconstruction error:

$$min \left\| \mathbf{x_i} - \sum_{j=1}^{k} W_{ij} \mathbf{x_{n_{ij}}} \right\|^2 \; s.t. \sum_{j=1}^{k} W_{ij} = 1 \tag{1.9}$$

where, $\mathbf{x_i}$ is a data point, $N_i = \{\mathbf{x_{n_{i1}}}, \mathbf{x_{n_{ik}}}\}$ is the neighbor set of $\mathbf{x_i}$ and $W_{ij}$ is the weight corresponding to each neighbor of $\mathbf{x_i}$.

In case of NPE, the transformation matrix $\mathbf{V}$ is computed by solving the following minimization problem:

$$min \left\| \mathbf{y_i} - \sum_{j=1}^{k} W_{ij} \mathbf{y_{n_{ij}}} \right\|^2 \; s.t. \sum_{j=1}^{k} \mathbf{y_i} = 0, \frac{1}{N} \sum_{i=1}^{N} \mathbf{y_i} \mathbf{y_i}^T = \mathbf{I} \tag{1.10}$$

The constraints imposed here are normalization constrains on the projection data. The optimization problem results in solving the following generalized eigenvalue problem:

$$\mathbf{XMX}^T\mathbf{V} = \lambda \mathbf{XX}^T\mathbf{V} \tag{1.11}$$

where, $\mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$. Eigenvectors corresponding to the smallest non-zero eigenvalues are the strongest. The basis i.e. the transformation matrix obtained following this procedure is not orthogonal. Many applications require the data to be converted back in the original space and this becomes easy with orthogonal basis. ONPP seeks orthogonal transformation matrix, excludes the normalization constraint imposed in NPE, thus minimizing $min \left\| \mathbf{y_i} - \sum_{j=1}^{k} W_{ij} \mathbf{y_{n_{ij}}} \right\|^2$. Solution of the optimization problem results in following eigenvalue solution:

$$\mathbf{XMX}^T\mathbf{V} = \lambda \mathbf{V} \tag{1.12}$$

In case of ONPP, the eigenvalue solution of Equation 1.12 realizes the transformation matrix with orthogonal basis vectors.

### Locality Preserving Projection (LPP)

Locality Preserving Projection [26], linear extension of Laplacian Eigenmaps discussed Section 1.1.2, explores the data non-linearity by preserving the local i.e. neighborhood of the data. The objective function of LPP aims at projecting similar data point close by in the projection space thus exploiting the non-linear manifold structure. The objective function of LPP is as follows:

$$min \sum_{ij} (\mathbf{y_i} - \mathbf{y_j})^2 S_{ij} \tag{1.13}$$

here, $\mathbf{y_i}$s are projections of the data points $\mathbf{x_i}$s in the LPP domain using the learned transformation i.e. basis matrix $\mathbf{V}$ i.e. $\mathbf{y_i} = \mathbf{V}^T\mathbf{x_i}$. $\mathbf{S}$ is the similarity matrix, weights in which are assigned using the Heat Kernel function i.e. $S_{ij} = e^{\frac{-\left\|\mathbf{x_i}-\mathbf{x_j}\right\|^2}{t}}$.

Further simplifying, the objective function turns out to be $\mathbf{V}^T\mathbf{XLX}^T\mathbf{V}$. Here, $\mathbf{V}$ is the transformation matrix, $\mathbf{L} = \mathbf{M} - \mathbf{S}$ is the Laplacian matrix [31] and $\mathbf{M_{ii}} = \sum_i S_{ij}$. A constraint $\mathbf{V^TXMX^TV} = \mathbf{I}$ is imposed on the objective function which incorporates normalization on the data points. The transformation matrix $\mathbf{V}$ that minimizes the objective function under the constraint is given by solution to the generalized eigenvalue problem[26]:

$$\mathbf{XLX^TV} = \lambda\mathbf{XMX^TV} \tag{1.14}$$

Thus, various dimensionality reduction approaches aim at representing the high dimensional data in much compact way without losing the information content. In this thesis, similarity preserving dimensionality reduction approaches have been studied for image data. The main motivation of the the thesis is detailed study of locality preserving projection that preserves the neighborhood similarity in the projection space. Extensions of LPP have been suggested for robust and powerful performance. Dimensionality reduction and domain transformation has been widely used in applications such as object recognition, face and facial expression recognition [19, 16, 26, 6, 32, 9], image restoration like denoising, deblurring, inpainting, compressive sensing etc. [33, 34, 35, 36, 37]. Different applications of dimensionality reduction based on similarity preservation have been explored over the course of the thesis.

## 1.2 Motivation of the Work

An observed object is often represented by a high dimensional real-valued vector and each object in this high dimensional space is called a data point. Generally, distribution of such high dimensional data points in the original space is not uniform and may seek representation in lower dimensional space. In applications, such as object recognition, bio-informatics and data mining, high data dimensionality imposes great burdens on the robust and accurate recognition due to insufficient knowledge about the data population and limited number of training samples [38]. Dimensionality reduction thus becomes the most critical module of such recognition systems. The objective of dimensionality reduction is retaining maximum information required for classification using a compact representation in the lower dimensional subspace, thus reducing the computational complexity of the subsequent classification.

A straightforward way to achieve dimensionality reduction is by maximizing the information carried by data in lower-dimensional subspace as suggested in Principal Component Analysis (PCA) [17], but it is the discriminative information that plays role in recognition task. Thus, discriminant analysis such as Linear Discriminant Analysis (LDA) [18] is preferred over PCA especially for recognition related applications. These linear approaches work on the assumption that the data is linear in original space which may not always be the case with high dimensional data. Neigbhorhood similarity preserving dimensionality reduction techniques such as Locality Preserving Projection (LPP) [26] and Neighborhood Preserving Embedding (NPE) [24] try to deal with the non-linearity present in the data. For tasks such as recognition, dependence of such approaches only on a few neighboring data points of the point of interest may lead to ambiguity as they may consider data points from different classes as neighbors. Also, it is important to enhance the reducibility capacity of such similarity preserving approaches and at the same time retain as much information as possible to achieve faster, accurate and robust performance.

As discussed earlier, both capturing non-linearity by similarity preservation and discriminant information are equally important for data classification and recognition. There is a strong need to evolve dimensionality reduction techniques that circumvent the issues, enhance subsequent data classification and make the system more robust. Various recognition problems such as face recognition can be addressed using these approaches.

Face images are represented as high-dimensional pixel arrays. Due to high correlation between the neighboring pixel values, they often belong to intrinsically lower dimensional manifold. Face recognition is one of the most widely used applications where dimensionality reduction is applied prior to the recognition task [19, 16, 26, 6, 32, 9]. Also, because of high variations in face images such as pose, expressions, illumination conditions, change in appearance etc, they may lie on a non-linear manifold. In this work, we are trying to device a dimensionality reduction scheme, that captures non-linearities present in the data and achieves maximum separability between data points from different classes simultaneously.

Apart from classification and recognition tasks, the noise removal aspect of dimensionality reduction has been explored in this work. Most of the information is carried by the coefficients/variable retained in the lower dimensional subspace. All the other variables with very small coefficient values carry noise or undesired information and hence can be eliminated. This phenomenon is known as noise removal effect of dimensionality reduction [38]. Thus, dimensionality reduction also helps in eliminating noise from the data.

In early 90s, transformed domain noise elimination approaches (in Fourier, Discrete Cosine and Wavelet domains) [33, 39, 40] became highly popular. These approaches showed that the information content of the image is mostly present in the low frequency components whereas noise is carried in the high frequency components. Thus, by thresholding (making the high frequency components zero), noise can be suppressed in the transform domain as well.

'Non-local self similarity' introduced the concept of using structural similarities between fixed sized patches from different spatial locations of the image [41, 42] for eliminating noise. Recent state of the art techniques for image denoising rely on two statistics of natural images: (1) There exists self-similarity between the patches from different locations of the same image [41], (2) Image patches can be sparsely represented by linear transforms i.e. using the linear combinations of the basis vectors also known as 'dictionary'. The dictionary could be comprised of the fixed universal basis [35] or can be adaptively learned from the image patches. State of the art approaches cluster similar patches for a given reference patch and learn patch/cluster specific dictionary [34, 43, 37, 44]. However, both the processes are independent and repeated for each patch/cluster.

The aim of local information preserving schemes is learning a transformation that projects the similar data points close to each other in the transform/projection domain. The basis are adaptively learned by taking care of similarity information for each data point. Thus, the main aspects of the recent image denoising approaches, i.e. similarity within the image patches and noise removal in the transform domain can be handled together. Another motivation of the present work is to build similarity based dimensionality reduction that can be suitable for robust and accurate noise elimination procedure.

## 1.3 Scope and Accomplishments of the Thesis

The scope of this thesis is study and development of dimensionality reduction techniques based on similarity preservation. Locality Preserving Projection (LPP) [26] is a linear dimensionality reduction technique that tries to explore non-linearities present in the data by preserving the neighborhood information *i.e.* similarity between the data points in the transformed domain as well. Conventional LPP is sensitive to noise, outliers and depends highly on the parameters for constructing the neighborhood graph and the weight matrix. A few extensions [45, 46, 47, 48] of LPP have been proposed in literature. This thesis is an attempt to overcome the issues of the conventional LPP approach towards making it more robust, achieve better discrimination capability, reduce the computational complexity and utilize it in various pattern recognition and image processing applications.

LPP and its variants, so far have been used in object recognition and data clustering related applications, most celebrated being face recognition. Face recognition experiments on various benchmark face databases have been conducted. Orthogonal and two dimensional variants of LPP have been formulated for faster and easier data processing. Orthogonal property of the basis helps in efficient data reconstruction, hence apart from recognition, applicability of the proposals can be extended towards image restoration processes, particularly image denoising. Similarity preservation property takes care of the non-local self similarity within the image by treating image patches as data points in the process of learning the basis. The basis vector thus learned, are tuned as per the clusters of similar patches and can be effectively utilized for eliminating noise in the transform domain. Hence, both the major hypotheses of recent state of the art image denoising, discussed in Section 1.2, are consolidated in similarity preserving transformations. An attempt has been made to establish similarity preserving projection as a natural choice for image denoising. Chapter wise major accomplishments of the thesis are listed down:

(1) Neighborhood of the data points is extended to a moderate distance in order to enhance the reducibility capacity of LPP as well as to resolve the ambiguities present in the regions where data points from different classes are close to each other. Also, based on the class label information, inter-class distance is enhanced whereas intra-class distance is decreased to achieve better class discrimination in the projection domain of Extended Supervised Locality Preserving Projection with Modified Distance (ESLPP-MD). Kernel based variants of LPP try to capture non-linearity present in the face images in a much better way.

(2) A dimensionality reduction technique, Locality Preserving Discriminant Projec-

tion (LPDP), is proposed which not only preserves local structure of the data but also tries to discriminate different classes when projected in the transformed domain. Improved face recognition performances with much less dimensions on some of the benchmark face databases have been observed. Kernelization of LPDP (K-LPDP) shows further improvements by exploring the non-linearities present in the data.

(3) The basis obtained using LPP and its variants are non-orthogonal. Orthogonalization of extended LPP (OLPP) and its suitability for image denoising are explored. Similarity preserving property of OLPP automatically takes care of non-local self similarity while inferring the basis (transformation matrix) and hence only one global basis is sufficient for entire or a large portion of the image. Proposed amalgamation of sparsity and global dictionary make the current approach more suitable for image denoising task with reduced computational complexity.

(4) Vectorized nature of OLPP requires high dimensional data to be converted to vector format, hence may lose spatial neighborhood information of the raw data. On the other hand, processing two dimensional data directly, not only preserves the spatial information, but also improves the computational efficiency considerably. Two Dimensional Orthogonal Locality Preserving Projection (2D-OLPP) learns the transformation from two dimensional data itself and is used for image denoising task. A global basis is adequate for the entire image. The proposed approach outperforms several state of the art image denoising approaches for gray-scale, color and texture images.

(5) Basis obtained using LPDP are non-orthogonal, also images in matrix format are required to be converted to vector format before processing. Two Dimensional Orthogonal Locality Preserving Discriminant Projection (2D-OLPDP), an orthogonal variant of LPDP that directly processes the data in matrix format has been utilized for image denoising.

## 1.4 Organization of Thesis Chapters

Image data typically consists of very high dimensions and processing it in its raw format with such large dimensions is a challenge especially for machines. Looking at the image, human visual system does not process individual pixels but observe the image as a whole entity, understanding the correlations present in scene and extracting the information. As opposed to this, the machines look at the image just as a collection of numbers and processing these huge piles of numbers is a difficult task especially in the era of huge data collection. Correlation within the image, between nearby pixels or self similar portions of the image, can be explored to represent high dimensional images with only a few significant coefficients by domain transformation, generally known as dimensionality reduction. An overview of dimensionality reduction is presented in Section 1.1.

Out of all, similarity preserving dimensionality techniques try to attain balance between linear and non-linear dimensionality reduction. Locality preserving projection (LPP) preserves neighborhood information of each training data point in the transformed domain thereby giving emphasis on similarity between them. This work is an attempt to develop robust and powerful dimensionality reduction technique based on similarity preservation. LPP has been studied and analyzed thoroughly and based on the observations, in each contributory chapter, similarity preserving dimensionality reduction techniques are proposed and various applications are explored. Brief overview of the chapters is covered below.

Locality preserving projection and some of its extensions are first discussed in Chapter 2. These approaches do not have control over projection of non-neighboring data points in the subspace. Intra-class and inter-class distances can be utilized to achieve enhanced class discrimination. Extended supervised LPP with modified distance (ESLPP-MD) is proposed in this chapter which increases or decreases distance between data points depending on the class labels to achieve more discrimination between different classes. The performance of such similarity preserving dimensionality reduction techniques do not guarantee to capture complex non-linearities present in the data. Kernel based variants are proposed to address this issue, that map the data to feature space which explores non-linearity. Data representation, face and facial expression recognition performances are reported on a large set of databases and show superior performance as compared to other variants of LPP.

ESLPP-MD manipulates distance between data points to achieve better class discrimination, which in a sense gives false impression of the data points to the basis learning mechanism. Instead of changing the distances, the objective function itself can be tuned to learn a transformation which projects the neighboring points from the same class close to each other and those from different classes far apart in the subspace. Locality Preserving Discriminant Projection and its kernel based variant are proposed in Chapter 3. Performance of LPDP is compared with a few other contemporary approaches on some benchmark databases for face recognition. The current method seems to perform significantly better.

Chapter 4 deals with Orthogonal property of LPP. The transformation matrix aka. basis learnt using LPP and Extended LPP are not orthogonal which is a desired property in many applications. Orthogonality of the basis especially makes the reconstruction of data faster and easier. Hence, orthogonalization of ELPP is suggested in this chapter and applied for an application which requires to represent the data back in the original domain i.e. image denoising. Recent state of art image denoising techniques work on the concept of exploring self similarity within the image in the form of image patches and noise elimination in transformed domain. Orthogonal LPP takes care of both the aspects which is well supported by comparable denoising performance of OLPP with several state of the art methods for gray-scale and color images.

All the dimensionality reduction approaches discussed so far process the data in vector format, thus image matrices are required to be converted in vectors first, causing loss of spatial neighborhood information. Two Dimensional Orthogonal Locality Preserving Projection (2D-OLPP) is formulated in Chapter 5 which processes images in matrix format directly. This not only preserves spatial information, but also improves the computational efficiency considerably. This approach inherits all the properties of OLPP, thus suitable for image denoising. Image parches are processes directly and similarity preserving property groups them in the projection space automatically and 2D data processing helps in improving the performance of image denoising outperforming state of the art approaches.

As discussed in Chapter 3, discriminant information, if considered can play pivotal role in clustering the data. Two Dimensional Orthogonal extension of LPDP is proposed in Chapter 6. While learning the basis, along with similarity, this approach also takes into account the dissimilarity information as well. 2D-OLPDP is also applied for image denoising in the same way as that of OLPP and 2D-OLPP. Incorporation of discriminant information while learning the basis clusters the noisy image patches in a better way thereby expected to enhance the quality of denoising. The results obtained are very encouraging and appeared to be comparable with the forerunner approaches of image denoising.

Variants of linear dimensionality reduction based on local similarity are proposed with varied applications such as face recognition and image denoising. As overall conclusion of the entire work is presented in Chapter 7 with some of the possible directions of future work.

# Chapter 2

# Locality Preserving Projection and its variants

As discussed in Chapter 1, there is a need for a linear dimensionality reduction approach, that takes care of the similarity present in the data by preserving closeness of the data points in the transform domain. Locality Preserving Projection (LPP) captures the local structure of data with the goal that data points that are similar to each other in the original domain should remain close in the projection domain as well. In this chapter, LPP, some observations about LPP and its extensions are discussed in detail.

## 2.1 Locality Preserving Projection (LPP)

The non-linear dimensionality reduction methods [20], [21] do yield impressive results on some benchmark artificial data-sets, as well as on real world data sets. However, their non-linear property makes them computationally expensive. Moreover, they yield maps that are defined only on the training data points and how to evaluate the map on novel test points remains unclear [26]. In many real world problems, local manifold structure is more important than the global Euclidean structure [30].

Locality reserving Projection (LPP) [26] is a recently proposed linear approach for dimensionality reduction that tries to capture the non-linear manifold structure of the data. It finds an embedding that preserves local information and obtains a subspace that best detects the essential data manifold structure. This is achieved by projecting similar data points close by in the LPP subspace. In LPP, neighborhood information is stored in a graph and basis vectors are found using the notion of Laplacian of a graph. A weighing function assigns weights to the edges of the graph that incurs heavy penalty if data points are mapped far apart hence giving more emphasis to nearest neighbors.

LPP is obtained by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the manifold [26]. The objective function is:

$$min \sum_{ij} (\mathbf{y_i} - \mathbf{y_j})^2 S_{ij} \tag{2.1}$$

here, $\mathbf{y_i}$ is the projection of $i^{th}$ data point $\mathbf{x_i}$ in the LPP domain using learned transformation matrix $\mathbf{V}$ i.e. $\mathbf{y_i} = \mathbf{V}^T\mathbf{x_i}$. $\mathbf{S}$ is the similarity matrix. It is a symmetric matrix representing the weights of edges of the adjacency graph. The data points are considered as the nodes of the graph while existence of edges depends on whether two nodes are considered as neighbors or not. The procedure of calculating $S_{ij}$, as suggested in He et al. [26] consists of two steps:

**Constructing the adjacency graph:** Let $G$ be the graph having the images in the training data-set as its nodes. An edge is present between nodes $i$ and $j$ if $\mathbf{x_i}$ and $\mathbf{x_j}$ are neighbors i.e. close to each other. The closeness can be determined in two different ways.

- $\epsilon$ **- neighborhood:** The nodes $i$ and $j$ are connected by an edge if $\|\mathbf{x_i} - \mathbf{x_j}\|^2 < \epsilon \in R$. Here the norm is the usual Euclidean norm in $R^n$.

- $k$ **- nearest neighbors:** The nodes $i$ and $j$ are connected by an edge if $i$ is among $k$-nearest neighbors of $j$ or vice-versa, $k \in N$. Once the adjacency graph is obtained, LPP will try to optimally preserve the local structure defined by the adjacency graph.

  In case of LPP, for constructing the adjacency graph, $K$-NN is widely used over the $\epsilon$- neighborhood method because of its simplicity and ease of implementation. Choosing the value of $K$ properly is difficult because of the non-linearity and variety of the high dimensional data.

**Estimation of weights:** $\mathbf{S}$ is a sparse symmetric matrix of size $N \times N$, where number of training data point is represented by $N$, with $S_{ij}$ having the weight of the edge connecting vertices $i$-$j$ and 0 if no such edge is present. Again, there are two variations for weighing the edges.

- **No parameter:** $S_{ij} = 1$ if nodes $i$ and $j$ are connected by an edge.

- **Heat kernel:** $S_{ij} = e^{\frac{-\|\mathbf{x_i}-\mathbf{x_j}\|^2}{t}}$ if nodes $i$ and $j$ are connected by an edge, $t \in R$.

The objective function with this choice of symmetric weights incurs a heavy penalty if neighboring points $\mathbf{x_i}$ and $\mathbf{x_j}$ are mapped far apart. This is an attempt to ensure that if $\mathbf{x_i}$ and $\mathbf{x_j}$ are close then their mappings in the projection space are close as well.

**Transformation matrix computation:** In order to compute the transformation matrix $\mathbf{V}$, the objective function can be reduced to the matrix form.

$$\begin{aligned}
&\tfrac{1}{2}\textstyle\sum_{ij}\left(\mathbf{y_i}-\mathbf{y_j}\right)^2 S_{ij}\\
&=\tfrac{1}{2}\textstyle\sum_{ij}\left(\mathbf{V}^T\mathbf{x_i}-\mathbf{V}^T\mathbf{x_j}\right)^2 S_{ij}\\
&=\tfrac{1}{2}\textstyle\sum_{ij}\left(\mathbf{V}^T\mathbf{x_i}-\mathbf{V}^T\mathbf{x_j}\right)^T\left(\mathbf{V}^T\mathbf{x_i}-\mathbf{V}^T\mathbf{x_j}\right) S_{ij}\\
&=\tfrac{1}{2}\textstyle\sum_{ij}\left(2\mathbf{V}^T\mathbf{x_i}S_{ij}\mathbf{x_i}^T\mathbf{V}-2\mathbf{V}^T\mathbf{x_i}S_{ij}\mathbf{x_j}^T\mathbf{V}\right)\\
&=\textstyle\sum_{i}\mathbf{V}^T\mathbf{x_i}M_{ii}\mathbf{x_i}^T\mathbf{V}-\mathbf{V}^T\mathbf{XSX}^T\mathbf{V}\\
&=\mathbf{V}^T\mathbf{XMX}^T\mathbf{V}-\mathbf{V}^T\mathbf{XSX}^T\mathbf{V}\\
&=\mathbf{V}^T\mathbf{X}\left(\mathbf{M}-\mathbf{S}\right)\mathbf{X}^T\mathbf{V}\\
&=\mathbf{V}^T\mathbf{XLX}^T\mathbf{V}
\end{aligned} \tag{2.2}$$

here, $\mathbf{V}$ is the transformation matrix, $\mathbf{L} = \mathbf{M} - \mathbf{S}$ is the Laplacian matrix [31] and $\mathbf{M_{ii}} = \sum_i S_{ij}$, a diagonal matrix which provides a natural measure on the data points. Here, each data point $\mathbf{x_i}$ is of dimension $m \times n$. A constraint $\mathbf{V^TXMX^TV} = \mathbf{I}$ is imposed on the objective function which incorporates normalization on the data points using the volume of the graph $G$ [49], [22]. The transformation matrix $\mathbf{V}$ that minimizes the objective function under the constraint is given by solution to the generalized eigenvalue problem[26]:

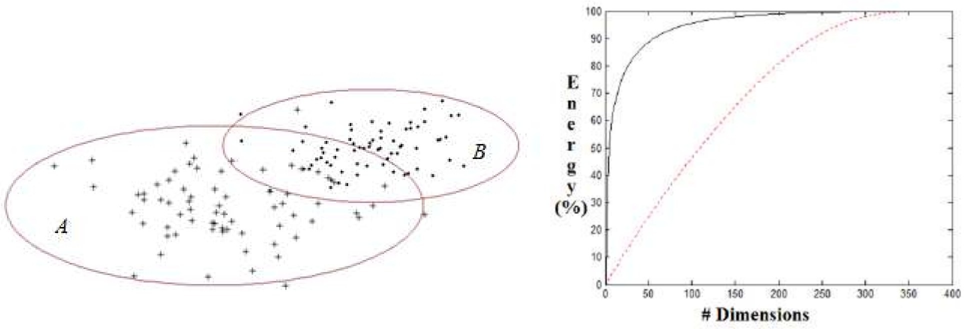
$$\mathbf{XLX^TV} = \lambda\mathbf{XMX^TV} \tag{2.3}$$

This transformation matrix is then used to project the high dimensional data in the lower dimensional subspace.

### 2.1.1 Observations From LPP

Manifold obtained using LPP highly depends on the construction of the similarity matrix. In case of conventional LPP, in order to obtain the similarity matrix, setting the values of parameters plays a vital role. First thing to be fixed is the neighborhood selection approach, as the graph structure is built-up using this information. Another parameter is the width of the Gaussian kernel that is controlled by $t$ in the heat kernel approach, which weighs the edges of the graph constructed in the previous step. Generally, mean of the pair wise distances is used as $t$ but this value does not necessarily find the optimal underlying manifold for all the data sets.

The local structure of the data is well preserved by LPP but it pays little or no attention to the overlapping regions of two or more classes. Many a times it happens that nearest neighbor of a data point is a data point belonging to the other class. In such cases, though the points belong to different classes, they could be connected because of their closeness. One such example is shown in Figure 2.1(a). Here, two classes are denoted by A and B, a region is shown where the boundaries of two classes overlap. In this region, data points from both the classes are neighbors.



(a)

(b)

Figure 2.1: (a) An example where data points from class A and B are mapped close by (b) Energy curve for PCA (solid) and LPP (dotted)

In LPP, similarity function $e^{\frac{-\left\|\mathbf{x_i}-\mathbf{x_j}\right\|^2}{t}}$ is applied to few nearest neighbors of the data point of interest. Considering only the nearest neighbors may lead to wrong classification in the region of overlap. LPP preserves local structure of data in the projection domain thus trying to handle data non-linearity, but may fail if the data lies on a complex non-linear manifold. LPP suggests project of similar data points close by but how the dissimilar data points should be mapped in the projection space is not defined. Discriminant information plays important role in recognition tasks which gets de-emphasized because of the locality preserving property of LPP.

Another observation is on energy preservation capacity of conventional LPP. Higher energy preservation ensures better reducibility capacity. Figure 2.1(b) shows energy curves of PCA and LPP for an experiment. Experiments over a range of data indicate that considering only half of the dimensions in the PCA space preserves more than 90% energy whereas the same in the LPP space comes out to be only 60%. Though the figures may vary in accordance with the data, in general, the energy preservation capacity of LPP remains much less than that of PCA and hence, the reducibility capacity of LPP needs to be increased.

LPP processes the data in vector format, hence all the data points are required to be converted into vectors. Due to vectorization, spatial neighborhood information gets lost. The vectorization procedure leads to high dimensional data vectors resulting in high dimensional basis matrix and increase in computational time. Also, most of the linear dimensionality reduction approaches such as LDA, LPP, NPP often suffer from small sample size problem i.e. number of data samples is much less than the dimension of the data [50, 51, 9]. In such cases, the matrices become singular, hence cannot be processed [9]. The learned LPP basis are obtained by solving generalized eigenvalue problem and hence the basis are not orthogonal in nature. Non-orthogonality makes its use difficult not only for reconstruction but also for many other applications. Several variants of LPP have been proposed in literature to overcome the issues and device a more robust and parameterless dimensionality reduction approach.

## 2.2 Existing Variants of LPP

### 2.2.1 Extended Locality Preserving Projection (ELPP)

Extended Locality Preserving Projection (ELPP) [48] aims at improving the reducibility capacity as well as resolving ambiguities in the overlapping region. A weighing scheme has been used that extends the neighborhood of data by considering the data points that are at a moderate distance from the point of interest along with the nearest neighbors. Weighing function automatically discards the data points far apart as it works based on the distance information. In order to exploit the natural grouping of the data, $k$-means algorithm is used based on which the neighbors are decided. The value of $k$ is assumed to be available as a prior information.

Conventional LPP uses $k$ - nearest neighbor or $\epsilon$ neighborhood approach to consider the data points as neighbors. For both the approaches, $k$ and $\epsilon$ remain constant for all the data points and it is hard to fix these parameters. The training set first undergoes $k$-means classification and then if class assigned to the two points is same, they are connected by an edge. This is an adaptive strategy for the data points to select the neighbors. Weight is assigned in the similarity matrix $\tilde{S_{ij}}$ according to the newly proposed z-shaped weighing function. Based on the range of values given as input, weights are assigned to the distances over the complete scale as per Equation 2.4. Plot of the function using different parameters is shown in Figure 2.2. As the distance between the data points *i.e.* $x$ increases, weight at that point decreases. The z-shaped weighing function decreases gradually and its slope can be controlled as compared to the Heat Kernel function used in conventional LPP which shows rapid fall as the distance increases.

$$\mathbf{S}_{ij} = \left\{ \begin{array}{ll} 1; & \text{if } x_{ij} \leq a \\ 1 - 2\left(\frac{x_{ij}-a}{b-a}\right)^2; & \text{if } a \leq x_{ij} \leq \frac{a+b}{2} \\ 2\left(\frac{x_{ij}-b}{b-a}\right)^2; & \text{if } \frac{a+b}{2} \leq x_{ij} \leq b \\ 0; \text{otherwise} & \end{array} \right\} \tag{2.4}$$

here, $a$ and $b$ specify range of values along which the function changes its values and can be controlled. Between $a$ and $b$, two functions are used to make the final output function the z-shaped one. The slope of the function is dependent on the parameters $a$ and $b$. $a$ is set to be a very small value and to set the value of $b$, the natural clusters found using $k$-means approach are used. After the $k$-means clustering step, data clusters are formed and labels to all the data points are assigned. For each cluster, maximum pairwise distance between all the data points belonging to that cluster (referred to as the radius of the cluster) is set to be $b$. Hence, for each cluster, $b$ has a unique value depending upon the pairwise distances between the data points belonging to it. This makes the procedure adaptive according to the data. Thus, both the processes, selection of neighbors and weight assignment, become adaptive and data dependent. Parameters are not to be set explicitly. As opposed to the conventional LPP where only few neighbors are considered, here the data points that are at a moderate distance from the point of interest are also taken into consideration and weighed accordingly.

Thus, the Laplacian matrix turns out to be $\tilde{\mathbf{L}} = \tilde{\mathbf{M}} - \tilde{\mathbf{S}}$ [31] where $\tilde{\mathbf{M}} = \sum_i \tilde{S_{ij}}$. The objective function now turns out to be $min\mathbf{V^TX\tilde{L}X^TV}$ subject to the constraint $\mathbf{V^TX\tilde{M}X^TV} = 1$. Eigenvectors of the generalized eigenvalue solution of $\mathbf{X\tilde{L}X^TV} = \lambda\mathbf{X\tilde{M}X^TV}$ for the basis vectors of the newly found subspace of ELPP.

### 2.2.2 Supervised Locality Preserving Projection (SLPP)

As described in Section 2.1, LPP is an unsupervised learning method which does not take into account the class membership information. In practical scenarios, many a times neighboring data points belong to different classes. According to the objective function of LPP, if the data points are neighbors i.e. close to each other in the original space, they should remain close in the reduced space as well even if they do not belong to same class. In such cases, LPP may lead to wrong classification as it does not incorporate the class label information. This information can be utilized to enhance the discriminant analysis as proposed in LDA [18]. It can be said that the locality preserving property and discriminant ability are significant in learning a

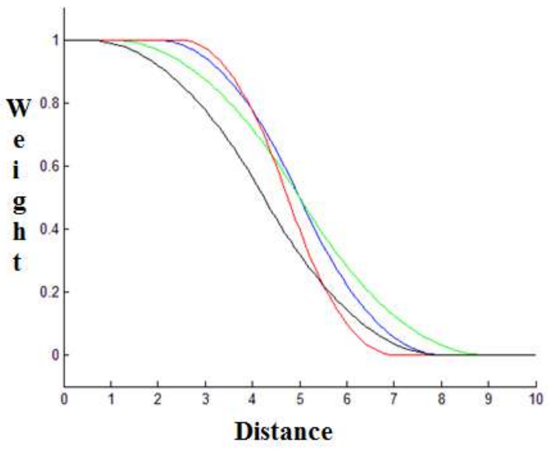


Figure 2.2: Plot of the z-shaped function with different parameters

new feature subspace [52]. The class information can be combined with the locality preserving property of LPP to enhance the performance.

Supervised Locality Preserving Projection (SLPP) [52] is a variant of LPP, where known class labels of data points are used. The neighbors of a point are decided based on already known class labels i.e. prior information about the original data set is used to learn the feature subspace. Choosing neighbors in this manner will prevent the points from two different classes to be projected close by. The objective function of SLPP is same as LPP only the computation of weight matrix **S** is different. In case of LPP, the similarity matrix is computed based only on the nearest neighbors and is independent of the class information. Whereas, in case of SLPP, two points are considered to be neighbors only if they belong to same class.

To make the computations simpler, data samples are arranged compactly i.e. all the samples from same class are arranged together in **X** which simplifies the computation of **S**. As a result, **X** is changed to be the orderly matrix which is composed of sub matrices $\mathbf{X_{subi}}$ of size $n \times M$, $i = 1, 2, ...c$, where $c$ is number of classes. In this way, nearest neighbors for each $\mathbf{x_j} \in \mathbf{X_{sub1}}$ are sought in $\mathbf{X_{sub1}}$ only. For each class, matrix $\mathbf{S_{subi}}$ is calculated which will then be used for constructing **S**. Here, the weight is set to be 1 if two data points belong to the same class otherwise weight is made 0. Hence, $\mathbf{S_{subi}}$ turns out to be :

$$\mathbf{S_{subi}} = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & 1 & \vdots \\ \vdots & 1 & \ddots & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix}$$

$\mathbf{S}$ is constructed by arranging the $\mathbf{S_{subi}}$s for all the classes diagonally i.e.

$$\mathbf{S} = \begin{bmatrix} \mathbf{S_{sub1}} & & & \\ & \mathbf{S_{sub2}} & & \\ & & \ddots & \\ & & & \mathbf{S_{subc}} \end{bmatrix}$$

All other computations for finding the transformation matrix are carried out in the same manner as that in LPP i.e. the transformation matrix $\mathbf{V}$ is found by solving the generalized eigenvalue problem: $\mathbf{XLX}^T\mathbf{V} = \lambda\mathbf{XMX}^T\mathbf{V}$.

### 2.2.3 Other variants

Locality Preserving Projection (LPP) [26], [30] a linear dimensionality reduction approach, tries to capture the non-linearity present in the data using neighborhood information. Non-orthonormal basis vectors are obtained using notion of Laplacian of graph constructed by considering the data points i.e. images as nodes. Orthogonal variant of LPP, Orthogonal Locality Preserving Projection (OLPP) [32] produces orthogonal basis functions with the aim of having more locality preserving power. The basis in this case are obtained iteratively, satisfying the orthogonality constraint. Conventional LPP is sensitive to noise and outliers and depends highly on the parameters for constructing the neighborhood graph and weight matrix. A few extensions [47], [46], [45], [53], [54] have been proposed to overcome these issues.

Robust path based similarity is used in Enhanced Locality Preserving Projection [46] to obtain robustness against noise and outliers. This semi-supervised dimensionality reduction approach uses pairwise constraints to construct the underlying graph, consequently resulting in enhanced performance. Parameter free LPP [47] is developed using Pearson correlation and adaptive neighborhood information. In this variant of LPP, the parameters are adaptive according to the data and it imposes the constrain on the projected data to be uncorrelated. Solution schemes have been devised in [45] to overcome the issues occurring with the conventional LPP because of small sample size of the data and generalized eigenvalue problem. Many a times, the smallest eigenvalues of the generalized eigenvalue solution of conventional LPP framework turnout to be zero. Selection of eigenvectors corresponding to zero eigenvalues may result in same representation of more than one samples in the projection space [45]. The proposed formulation represents the LPP formulation as a maximization problem in order to avoid this issue which now selects the eigenvectors corresponding to the highest eigenvalues as the strongest basis vectors.

## 2.3 Extended Supervised LPP with Modified Distance (ESLPP-MD)

In the formulation of SLPP, all the data points belonging to same class are treated in the same manner i.e. in the similarity matrix, weight 1 is assigned if the data points belong to the same class otherwise weight is made 0 without considering the distance between data points. If the distance information is incorporated along with the class information, the underlying manifold can be revealed in a better way. The methods discussed so far concentrate on the neighboring data points coming from same class, but nothing has been suggested about the neighboring data points from different classes. An improvement over SLPP is suggested in this section, that tries to focus these issues. Distance between data points from different classes must be relatively larger than their actual Euclidean distance as they belong to different classes. The class information can be incorporated so as to make the distance between two points belonging to different classes relatively larger than their Euclidean distance. This could be achieved by shrinking the intra-class distance while expanding the inter-class distance which will result in better separability between classes. The data points can be made strong neighbors by shrinking the distance if they belong to same class, otherwise the distance can be increased so as to make them weak neighbors. To manipulate the distance according to this, Equation 2.5 is as follows:

$$dist\left(x_i, x_j\right) = \left\{ \begin{array}{ll} \sqrt{1 - e^{-\frac{d(i,j)^2}{\beta}}} & ;c_i = c_j \\ \sqrt{e^{\frac{d(i,j)^2}{\beta}}} - \alpha & ;c_i \neq c_j \end{array} \right\} \tag{2.5}$$

here, $c_i$ is the class label of $x_i$ whereas $c_j$ is the class label of $x_j$. Also, $d\left(i, j\right) = \|x_i - x_j\|$ is the Euclidean distance between the data points $x_i$ and $x_j$. Constant $\alpha$ is used to adjust the similarity between the points from different classes. It takes the values between $[0, 1]$. Parameter $\beta$ is used to prevent the distance from increasing too fast when $d\left(i, j\right)$ is relatively large. Usually, the average distance between all pairs of data points is taken as $\beta$. The plot of the distance functions defined in Equation 2.5 is shown in Figure 2.3. Here, the dotted line shows the behavior of the function if the data points belong to different classes and the solid line indicates the same for data points from the same class. Similar functions have been used to have better separability in case of Locally Linear Embedding (LLE) [10].

Modifying distances in this manner adds more discriminating power for different classes. This new distance matrix is used for constructing the weight matrix $\tilde{S}_{ij}$. In addition to the class information, if the weights assigned in $\tilde{\mathbf{S}}$ are based on the

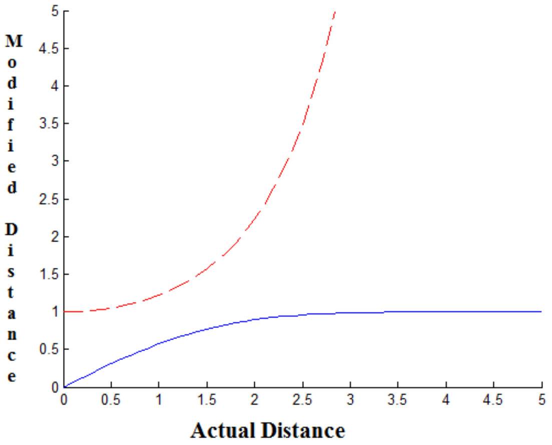


Figure 2.3: Plot of the distance function for the data points belonging to same class (solid) and different classes (dotted) [10]

distance as opposed to SLPP where a fixed value is assigned to all the neighboring data points, it may be more helpful in revealing the underlying manifold. One way to do so is by using a weighing function to compute the matrix $\tilde{\mathbf{S}}$ as in case of LPP. But, as observed from LPP, to overcome the ambiguity occurring in the overlapping regions of two or more classes as well as to increase the reducibility capacity, the same z-shaped weighing function (Equation 2.4) is used which automatically gives emphasis to the data points at a moderate distance from the point of interest in addition to the nearest neighbors. The transformation matrix then can be found as in case of ELPP [48]. This scheme is called Extended Supervised Locality Preserving Projection with modified distance (ESLPP-MD). Extension in the sense that it tries to assign weights to the data points which are at a moderate distance from the point of interest, in addition to the closest neighbors as in the case of LPP. Modified in the sense that it uses a modified distance between two data points as opposed to Euclidean distance used in LPP. Note that distance $x$ defined in the function to compute $\widetilde{\mathbf{S}}$ is actually "*dist*" that is defined earlier in Equation 2.5.

The dimensionality reduction approaches discussed so far are vectorized in nature i.e. they process the data in the vector format only. This requires the images of size $m \times n$ in matrix format to be converted into vectors of size $mn \times 1$. The final data matrix containing all $N$ training images in vector format turns out to be of size $mn \times N$. Generally, $N << mn$, which makes the matrix singular in many cases. To overcome the problem of singularity, Principle Component Analysis (PCA) is applied on the data. As suggested in Turk et al. [17], only first $N$ eigenvectors of the covariance matrix carry the meaningful information rather than $mn$. Hence, $N$

most significant components are chosen to be processed in the LPP formulation, thus tackling the problem of singularity.

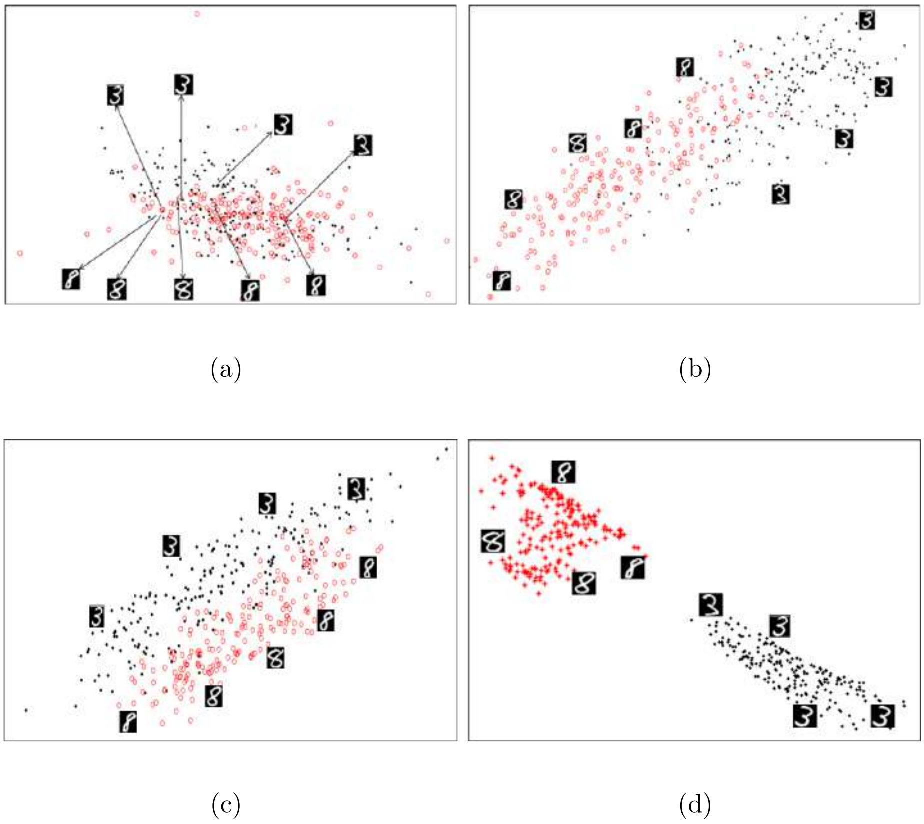
### 2.3.1 Projection of the data in the similarity preserving subspace

In order to visualize the data projected in the subspace of similarity preserving dimensionality reduction approaches discussed in previous section, strongest two and three dimensional projections of various types of image data are reported. The strongest dimensions are those corresponding to the smallest non-zero eigenvalues of the generalized eigenvalue solutions. The projection results on MNIST data set of handwritten digits [55] and video data sets are shown.

#### MNIST database of hand written digits

The MNIST database of handwritten digits [55] has a training set of $60,000$ examples, and a test set of $10,000$ examples. It was constructed from NIST's Special Database 3 and Special Database 1 which contain images of handwritten digits. The digits have been size-normalized and centered in fixed-size images of $20 \times 20$ pixels. For experimentation purpose, 200 images per digit were selected randomly from the database.

Digit pairs are chosen to observe the behavior of the approach when there is some similarity between two digits. As all the images are of handwritten digits, sometimes because of the way of writing, there may be a lot of similarity between 3 and 8. In case of 1 and 7 also, structural similarity can be observed. The results of projection of different digits are shown in Figure 2.4 and Figure 2.5. $X$-axis (horizontal axis) represents the direction of the strongest component of the transformation matrix while $Y$-axis (vertical axis) represents the $2^{nd}$ strongest component. The same convention is used to show the results throughout the thesis. It is to be noted that the digit data is of 400 dimensions out of which only 2 strongest dimensions are used for projection. In case of both the digit pairs, LPP seems to be merging the different digits, showing no separation between different clusters. ELPP isolates the clusters of digits 1 and 7 quite well, but projection of digits 3 and 8 is mixed up. In the 2 dimensional space, SLPP is also separating the digits but still some ambiguity is present at the boundary of two classes. ESLPP-MD is able to separate the structurally similar digit pairs in a much better way. Clear separation between clusters of digits can be observed using only 2 dimensions. Also, in the projection space, the intrinsic characteristics of a digit are revealed i.e. thickness and slope of the digit are changing as we move forward on the horizontal axis.



(a)

(b)

(c)

(d)

Figure 2.4: Strongest 2D projection of digits 3 and 8 using (a)LPP (b)ELPP (c)SLPP (d)ESLPP-MD. Horizontal axis represents the direction of the strongest component of basis $\mathbf{V}$ while vertical axis represents the $2^{nd}$ strongest component.

### Video Database

To evaluate how the algorithm behaves on face data and a person's varying expressions, the tests are performed on two video datasets. First one is the Interview database [56] that contains face images of a single person having mainly 3 expressions i.e. Normal, Laughing and Open Mouth. Strongest 2 dimensional projection of the interview data using LPP, ELPP, SLPP and ESLPP-MD is show in Figure 2.6. The expression sub-manifold is revealed separating the images having different expressions in a much better way using the current proposal.

A video data-set (DA-IICT video data-set) has also been created in which videos of around 30 seconds of 11 subjects are recorded. Single video contains four different expressions of a subject i.e. Angry, Normal, Smiling and Open mouth as shown in

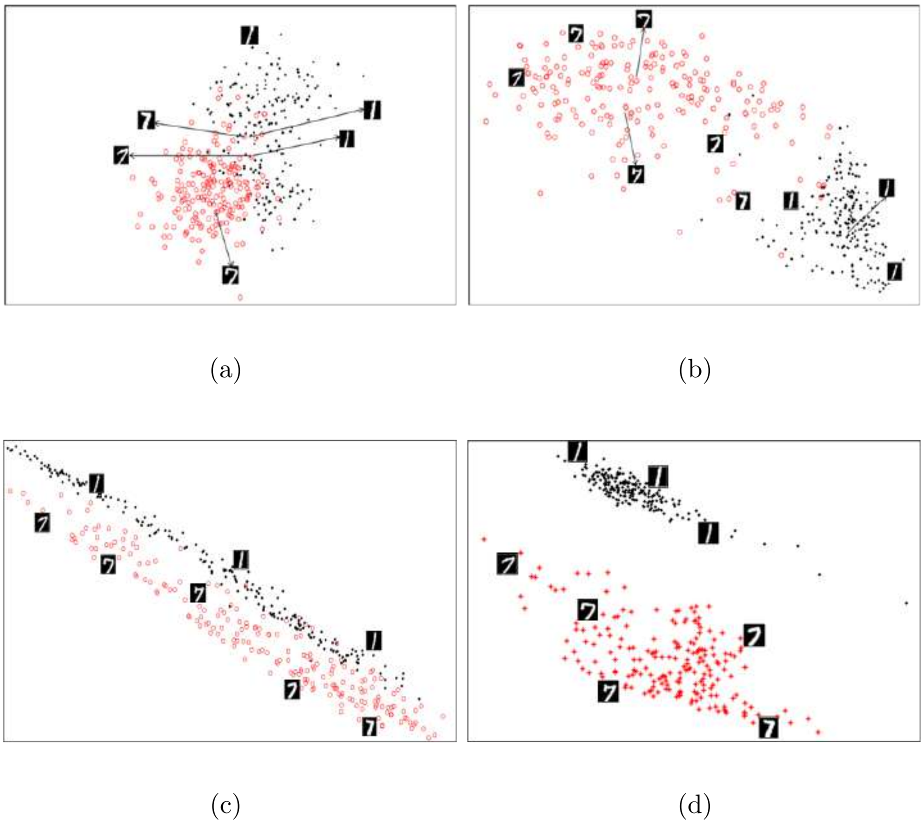


Figure 2.5: Strongest 2D projection of digits 1 and 7 using (a)LPP (b)ELPP (c)SLPP (d)ESLPP-MD. Horizontal axis represents the direction of the strongest component of basis **V** while vertical axis represents the $2^{nd}$ strongest component.

Figure 2.7. The strongest 3 dimensional projection of the whole Video data-set (i.e. 11 subjects, 4 expressions) using ELPP and ESLPP-MD is shown in Figure 2.8 and Figure 2.9 respectively. Clear discrimination between projections of face images of different persons can be observed in the projection space of ESLPP-MD. Also, to check how different expressions of a person are being projected, projection of some of the subjects has been enlarged and each expression is represented by a different color. It can be clearly observed that, the sub manifolds of expression are also significantly discriminated.

In addition to the projection results, face and facial expression recognition experiments on the the video database are carried out by randomly sampling 50% face images for training data set and out of the rest of the samples, 3 distinct test data sets are generated. Results in terms of error rates in percentage for these 3 different test runs are reported in Table 2.1 for face recognition. Each run is performed on dis-

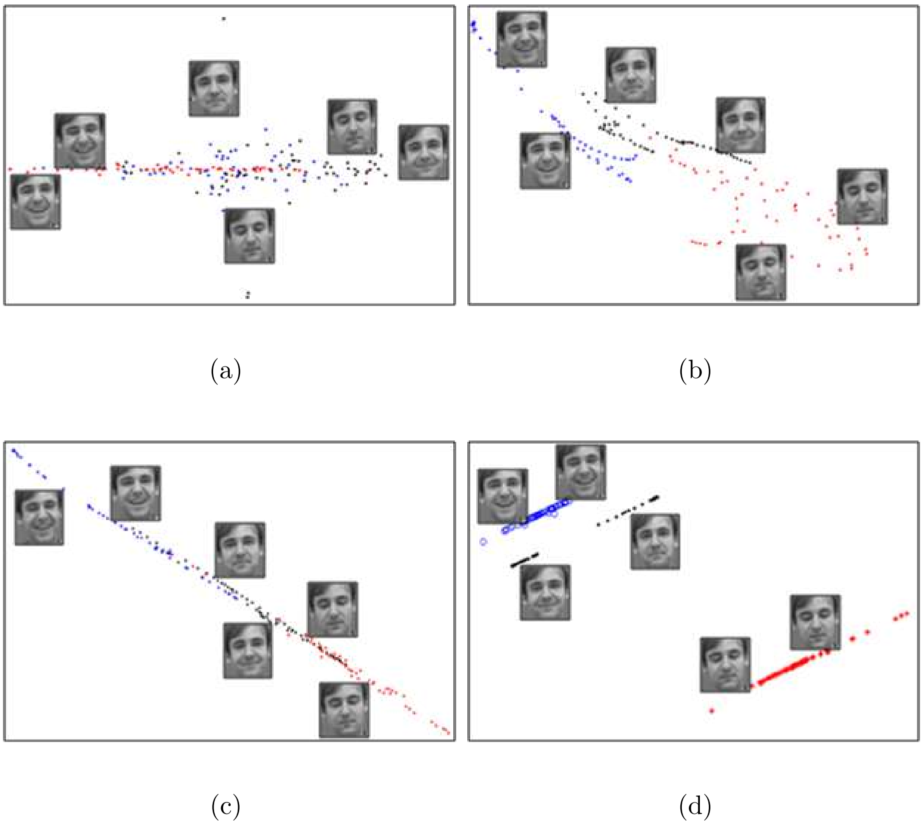


Figure 2.6: Strongest 2D projection of face images from Interview data set having three expressions (a)LPP (b)ELPP (c)SLPP (d)ESLPP-MD. Horizontal axis represents the direction of the strongest component of basis **V** while vertical axis represents the $2^{nd}$ strongest component.

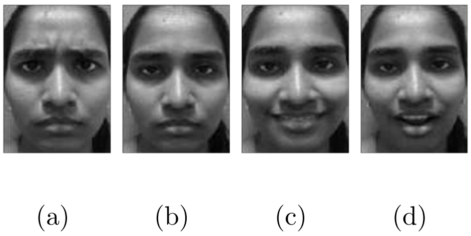


Figure 2.7: Examples of facial expressions from the Video dataset : (a) angry, (b) normal, (c) smiling, (d) open mouth respectively.

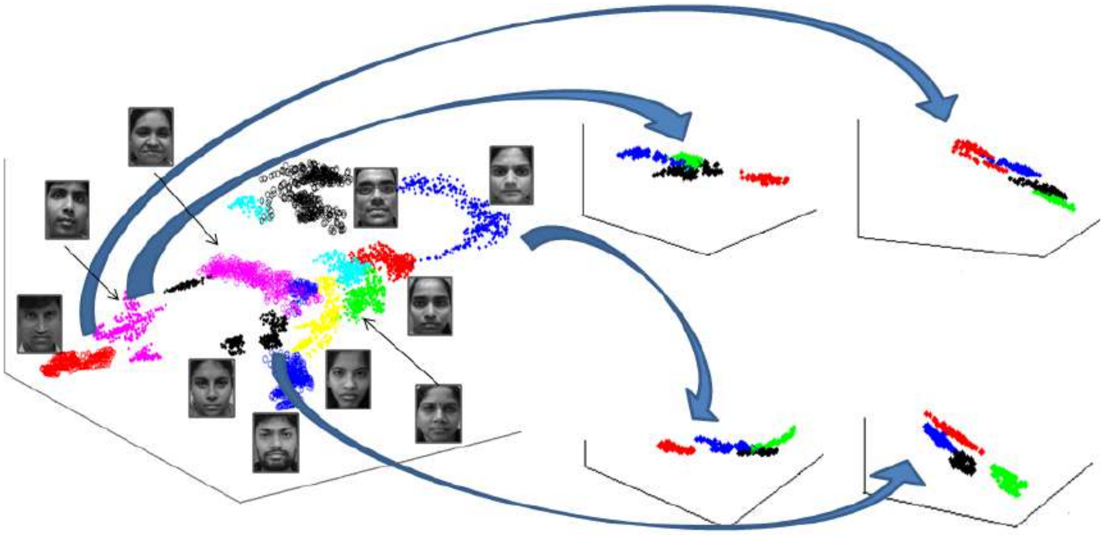


Figure 2.8: Strongest 3 dimensional projection using ELPP of all the subjects from video data-set and some examples of expression discrimination in a person's manifold. Each person is represented by a different sign or color in the figure on left whereas for some persons in the enlarged projection, each expression is represented by different color. $x$, $y$ and $z$ axes represent the direction of first three strongest components of basis $\mathbf{V}$ respectively.

tinct test data-set. ESLPP-MD produces 100% face recognition accuracy similar to all the other competitive approaches. It can be observed that as opposed to the other competing approaches, ESLPP-MD achieves more than 99% recognition rate using only three strongest dimensions. For each subject, four different facial expressions are recorded in the database, hence, facial expression recognition is also performed to analyze performance of the approaches in recognizing the expression sub-manifold. The error rates are reported in Table 2.2 with various dimensions for the same runs as used for face recognition. 85% recognition rate is obtained using the proposed ESLPP-MD which is comparable to ELPP and SLPP.

## 2.4 Kernelization of LPP

The data, in its raw form, may not necessarily lie on a linear manifold. The dimensionality reduction approaches that work on the idea of retaining local information try to capture the non-linearity present in the data by modeling the local neighborhood information, but may fail if the underlying structure contains complex non-linearities. One way to capture these non-linearities is by mapping the data in a non-linear feature space and then apply the dimensionality reduction algorithm on the data [57].

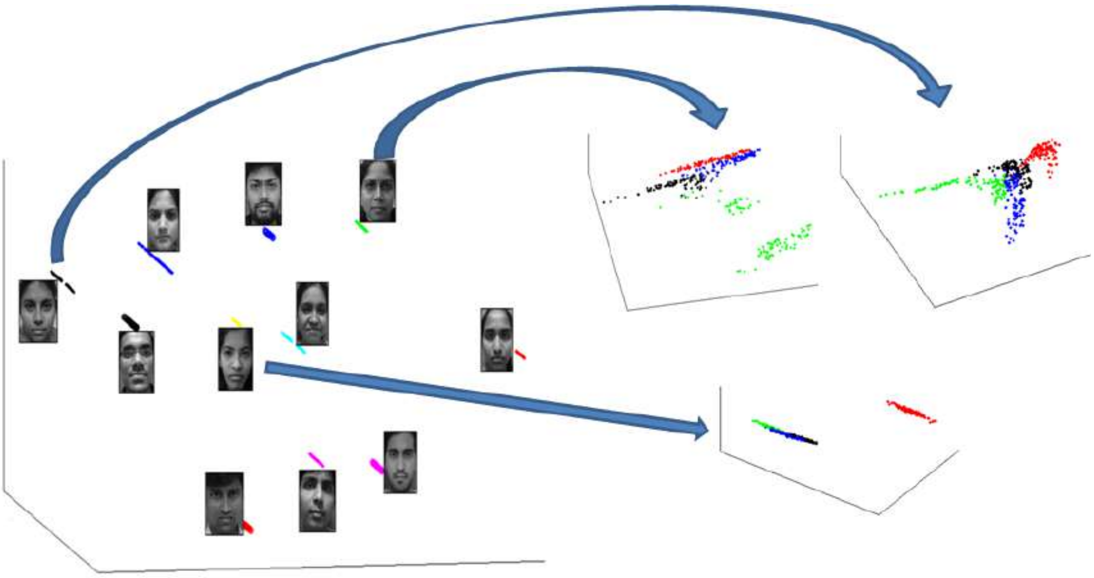


Figure 2.9: Strongest 3 dimensional projection using ESLPP-MD of all the subjects from video data-set and some examples of expression discrimination in a person's manifold. Each person is represented by a different sign or color (1 sample face image for corresponding person is shown) in the figure on left whereas for some persons in the enlarged projection, each expression is represented by different color. $x$, $y$ and $z$ axes represent the direction of first three strongest components of basis $\mathbf{V}$ respectively.

Table 2.1: Errors (in %) of face recognition on various test runs from the video data set using nearest neighbor approach. Actual dimensions of the raw data were 10000.

| **Run** | **#Dime-** | **Approach** | | | |
|---|---|---|---|---|---|
| | **nsions** | LPP | ELPP | SLPP | ESLPP-MD |
| #1 | 3 | 85.22 | 3.86 | 15.22 | 0.45 |
| | 50 | 88.18 | 0.22 | 0 | 0.22 |
| | 500 | 86.59 | 0 | 0 | 0.22 |
| | 1000 | 15.68 | 0 | 0 | 0.22 |
| #2 | 3 | 83.63 | 5 | 7.27 | 0.45 |
| | 50 | 84.31 | 0.22 | 0 | 0.22 |
| | 500 | 83.86 | 0 | 0 | 0.22 |
| | 1000 | 14.77 | 0 | 0 | 0.22 |
| #3 | 3 | 83.86 | 5.22 | 7.5 | 0.90 |
| | 50 | 83.40 | 0.68 | 0.45 | 0.68 |
| | 500 | 81.81 | 0.22 | 0.22 | 0.68 |
| | 1000 | 14.77 | 0.22 | 0.22 | 0.68 |

Table 2.2: Errors (in %) of facial expression recognition on various test runs from the video data set using nearest neighbor approach. Actual dimensions of the raw data were 10000.

| **Run** | **#Dimensions** | **Approach** | | | |
|---|---|---|---|---|---|
| | | LPP | ELPP | SLPP | ESLPP-MD |
| #1 | 3 | 93.18 | 33.86 | 32.95 | 26.18 |
| | 50 | 94.31 | 16.59 | 21.36 | 18.86 |
| | 500 | 94.31 | 15.90 | 21.13 | 19.77 |
| | 1000 | 28.40 | 15.90 | 20.68 | 15.22 |
| #2 | 3 | 91.36 | 34.09 | 22.27 | 33.40 |
| | 50 | 91.81 | 15.90 | 17.27 | 21.36 |
| | 500 | 90.68 | 15.90 | 17.27 | 18.18 |
| | 1000 | 27.5 | 15.22 | 16.36 | 17.72 |
| #3 | 3 | 90.22 | 33.63 | 32.27 | 25 |
| | 50 | 90.22 | 17.5 | 17.27 | 18.86 |
| | 500 | 90 | 17.72 | 17.04 | 15.22 |
| | 1000 | 28.40 | 15.90 | 17.04 | 15 |

Kernel trick, using the kernel functions have been widely used in literature to map the data in a new feature space such that the data non-linearities can be addressed in a much better way. Many a times linear separability between multiple classes can be achieved by kernel mapping. Examples to show projection of the data in the feature space are reported in Figure 2.10 and Figure 2.11. It can be observed that, in original space, linear separation between two classes is not possible, but after the mapping in feature space, two classes are separable using a linear plane.

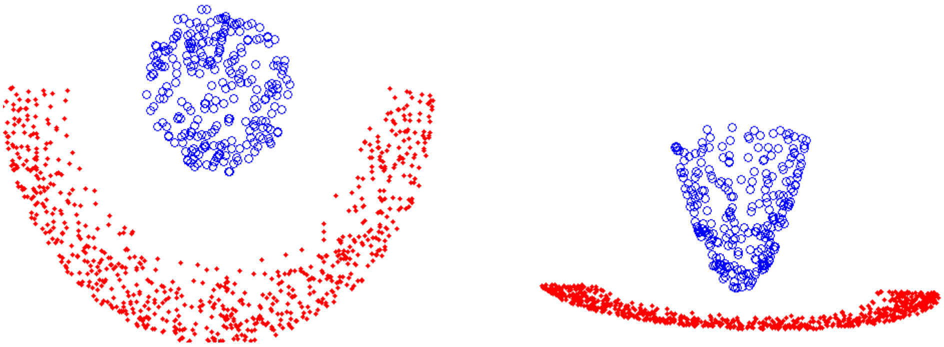


Figure 2.10: Data points in the original space (left) and representation of the data points in feature space where two classes can be separated using a hyper-plane (right).

Thus, function $\phi : \mathbb{R}^n \rightarrow \mathcal{F}$ is used to map the data from the original $n$-dimensional space to a non-linear feature space $\mathcal{F}$. Kernelized variants of PCA (K-PCA) [58, 57],

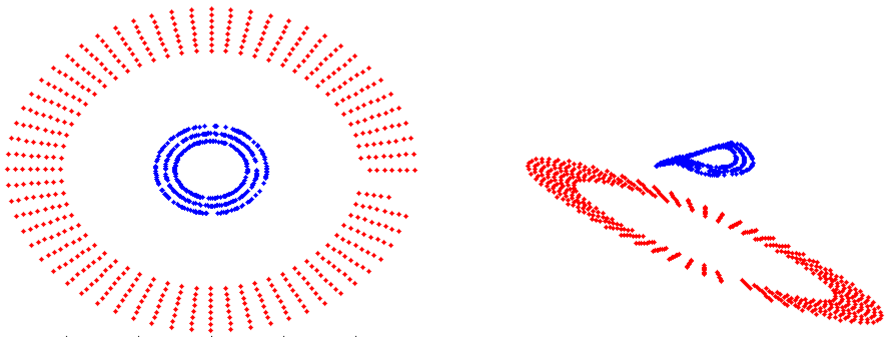


Figure 2.11: Data points in the original space (left) and representation of the data points in feature space where two classes can be separated using a hyper-plane (right).

LDA (K-LDA) [59, 60] and LPP (K-LPP) [61] have already been proposed. The general framework for the kernel based dimensionality reduction approaches is shown in Figure 2.12. The data is first mapped to a feature space using the kernel trick, which is nothing but the inner product between the data points in the non-linear feature space, using the kernel function. Dimensionality reduction procedure, i.e. learning the transformation matrix that transforms the data from the feature space to the lower dimensional subspace, is carried out on this kernel matrix. Projected data points thus tend to have better representation of the non-linear structure.

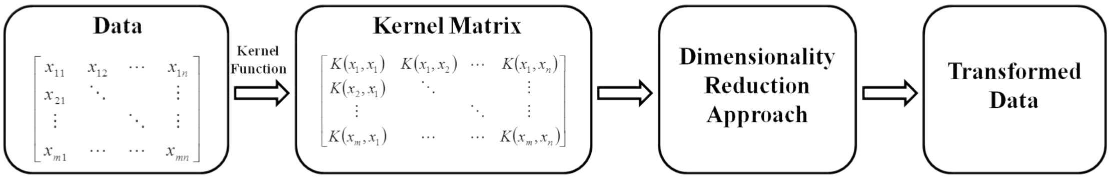

Data
$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ x_{m1} & \cdots & \cdots & x_{mn} \end{bmatrix}$
Kernel Function
Kernel Matrix
$\begin{bmatrix} K(x_1,x_1) & K(x_1,x_2) & \cdots & K(x_1,x_n) \\ K(x_2,x_1) & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ K(x_m,x_1) & \cdots & \cdots & K(x_m,x_n) \end{bmatrix}$
Dimensionality Reduction Approach
Transformed Data

Figure 2.12: General framework of Kernel based methods

Let the data in the original space be denoted by $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}$, then the data mapped in the feature space is denoted as $\phi\left(\mathbf{x_1}\right), \phi\left(\mathbf{x_2}\right), \ldots \phi\left(\mathbf{x_N}\right)$.

$$k\left(\mathbf{x_i}, \mathbf{x_j}\right) = \left\langle \phi\left(\mathbf{x_i}\right), \phi\left(\mathbf{x_j}\right)\right\rangle \tag{2.6}$$

$\phi^T\left(\mathbf{X}\right)\phi\left(\mathbf{X}\right)$ can be represented by a kernel matrix $\mathbf{K}$ which is inner product between the data points in feature space i.e. $\mathbf{k}\left(\mathbf{x_i}, \mathbf{x_j}\right) = \left\langle \phi\left(\mathbf{x_i}\right), \phi\left(\mathbf{x_j}\right)\right\rangle$. Hence, it can be said that kernels are nothing but the inner product between the data points in some space, and even without knowing the non-linear mapping function $\phi$ explicitly, the relationship between two data points in the feature space can be directly determined using $k$ [61]. Therefore, we are able to use kernel functions for computing these dot products without actually performing the map [62, 63] known as the kernel trick. Thus, mapping the data in the feature space $\mathcal{F}$ using function $\phi$ is equivalent to choosing $k$ (the kernel). For some choices of kernels $k\left(\mathbf{x_i},\mathbf{x_j}\right)$, it can be shown by methods of functional analysis that there exists a map $\phi$ into some dot product space $\mathcal{F}$ (possibly of infinite dimension) such that $k$ computes the dot product in $\mathcal{F}$ [58].

Some of the very popular kernels are as linear kernel, polynomial kernel, Gaussian kernel etc. [64] mathematical forms of which are as follows:

- Linear Kernel: $k\left(\mathbf{x_i},\mathbf{x_j}\right) = \mathbf{x_i}' * \mathbf{x_j}$
- Polynomial Kernel: $k\left(\mathbf{x_i},\mathbf{x_j}\right) = \left(\mathbf{x_i}' * \mathbf{x_j}\right)^d$
- Gaussian Kernel : $k\left(\mathbf{x_i},\mathbf{x_j}\right) = e^{\left(-\frac{\left\|\mathbf{x_i}-\mathbf{x_j}\right\|^2}{2\sigma^2}\right)}$

Using one of these or any other kernel, the data can be mapped to feature space and then to transformed space using the dimensionality reduction approaches. A brief overview of kernel based variants of PCA, LDA and LPP is presented before discussing about the kernelization of ESLPP-MD.

### 2.4.1 Overview of existing kernel based approaches

**Kernel PCA [58, 57]**

Principle Component Analysis (PCA) is a linear dimensionality reduction technique which works on the assumption that the data is also linear in nature. This linearity assumption may not always be true for the real world databases. Hence, as discussed earlier in this chapter, to exploit the non-linearity, the data is first mapped to a non-linear feature space. This technique, known as Kernel PCA (K-PCA), was first introduced in Schölkopf et al. Assume data mapped in the feature space is denoted as $\phi\left(\mathbf{x_1}\right), \phi\left(\mathbf{x_2}\right), \ldots \phi\left(\mathbf{x_N}\right)$ ans centered i.e. $\sum_{i=1}^{N} \phi\left(\mathbf{x_1}\right) = 0$. The covariance matrix in the feature space can be found as follows:

$$\mathbf{C_F} = \frac{1}{N}\phi\left(\mathbf{x_i}\right)\phi\left(\mathbf{x_j}\right)^T \tag{2.7}$$

The eigenvectors of the covariance matrix constitute PCA basis for the training data. In this case, the basis turn out to be of the feature space in which the data was mapped i.e. $\mathbf{V_F} = \phi\left(\mathbf{x_i}\right)\mathbf{V}\forall i \in 1, 2, \cdots, N$. Derivation for obtaining K-PCA basis is reported as follows:

$$\begin{aligned}
&\mathbf{C_F V_F} = \lambda \mathbf{V_F} \\
&\equiv \phi\left(\mathbf{x_i}\right) \mathbf{C_F V_F} = \lambda \phi\left(\mathbf{x_i}\right) \mathbf{V_F}; \forall i \in 1, 2, \cdots, N \\
&\equiv \mathbf{K}^2 \mathbf{V} = N\lambda \mathbf{KV}; k\left(\mathbf{x_i}, \mathbf{x_j}\right) = \left\langle \phi\left(\mathbf{x_i}\right), \phi\left(\mathbf{x_j}\right) \right\rangle \\
&\equiv \mathbf{KV} = N\lambda \mathbf{V}
\end{aligned} \tag{2.8}$$

Thus, using the kernel trick, the computations become much simpler and eigenvectors of $\mathbf{KV} = N\lambda\mathbf{V}$constitute the K-PCA basis.

### Kernel LDA [59, 60]

Similar to Kernel PCA, a non-linear generalization of Linear Discriminant Analysis is proposed [60]. As per the convention used earlier, function $\phi : \mathbb{R}^n \to \mathcal{F}$ is used to map data in feature space $\mathcal{F}$. The between scatter matrix $\mathbf{S_{B_F}}$ and within scatter matrix $\mathbf{S_{W_F}}$ for the data mapped in feature space are represented as follows:

$$\begin{aligned}
&\mathbf{S_{B_F}} = \textstyle\sum_{i=1}^{c} n_i \left(\mathbf{d^i}_\mathbf{F} - \mathbf{d_F}\right) \left(\mathbf{d^i}_\mathbf{F} - \mathbf{d_F}\right)^T \\
&\mathbf{S_{W_F}} = \textstyle\sum_{i=1}^{c} \left( \sum_{j=1}^{n_i} \left(\phi\left(\mathbf{x^i_j}\right) - \mathbf{d^i}_\mathbf{F}\right) \left(\phi\left(\mathbf{x^i_j}\right) - \mathbf{d^i}_\mathbf{F}\right)^T \right)
\end{aligned} \tag{2.9}$$

here, $c$ is the number of classes in the data set, $n_i$ the number of samples in the $i^{th}$ class, $\mathbf{d_F}$ the mean vector of all the samples, $\mathbf{d^i}_\mathbf{F}$ mean vector of $i^{th}$ class and $\phi\left(\mathbf{x^i_j}\right)$ is the $j^{th}$ sample of $i^{th}$ class in the feature space. The Fisher discriminant criterion to be maximized in the feature space $\mathcal{F}$ is:

$$max \frac{\mathbf{V_F}^T \mathbf{S_{B_F}} \mathbf{V_F}}{\mathbf{V_F}^T \mathbf{S_{W_F}} \mathbf{V_F}} \tag{2.10}$$

Replacing $\mathbf{V_F} = \phi\left(\mathbf{x_i}\right) \mathbf{V}$ and simplifying using the kernel trick, the maximization function can be rewritten as shown in Equation 2.11. It can be observed that, in case of K-LDA also, explicit mapping of the data in the feature space is not required.

$$max \frac{\mathbf{V}^T \mathbf{S_{B_{new}}} \mathbf{V}}{\mathbf{V}^T \mathbf{S_{W_{new}}} \mathbf{V}} \tag{2.11}$$

here, $\mathbf{S_{B_{new}}} = \left(\mathbf{D_i} - \mathbf{D}\right)\left(\mathbf{D_i} - \mathbf{D}\right)^T$, $\left(\mathbf{D_i}\right)_j = \frac{1}{n_i} \sum_{k=1}^{n_i} k\left(\mathbf{x_j}, \mathbf{x^i_k}\right)$, $\left(\mathbf{K_j}\right)_{nm} = k\left(\mathbf{x_n}, \mathbf{x^j_m}\right)$ is the kernel matrix of class $j$, $\mathbf{S_{W_{new}}} = \sum_{j=1}^{c} \mathbf{K_j}\left(\mathbf{I} - \mathbf{1_{n_i}}\right)\mathbf{K_j}^T$, $\mathbf{I}$ is the identity matrix and $\mathbf{1_{n_i}}$ is the matrix with all the entries having value $1/n_i$.

The K-LDA basis are obtained by solving the generalized eigenvalue problem: $\mathbf{S_{B_{new}}} \mathbf{V} = \lambda \mathbf{S_{W_{new}}} V$.

Kernel weighted nonlinear discriminant analysis (KWNDA), a variant of K-LDA, first maps data in a feature space using kernels [65]. To achieve improved class discrimination, weighted between class scatter matrix is introduced. This approach outperforms existing kernel discriminant analysis methods in terms of the classification accuracy.

**Kernel LPP [61]**

As discussed for kernel based PCA and LDA, for kernel LPP, the basis in the feature space can be obtained by solving the following generalized eigenvalue problem:

$$\phi^T(\mathbf{X})\,\mathbf{L}\phi(\mathbf{X})\,\mathbf{V_F} = \lambda\phi^T(\mathbf{X})\,\mathbf{M}\phi(\mathbf{X})\,\mathbf{V_F} \tag{2.12}$$

To generalize LPP to the nonlinear case, we formulate it in a way that uses dot product exclusively [26]. The dot product in the feature space can be represented using the kernel function: $k(\mathbf{x_i}, \mathbf{x_j}) = \langle\phi(\mathbf{x_i}), \phi(\mathbf{x_j})\rangle = \phi^T(\mathbf{x_i})\,\phi(\mathbf{x_j})$.

Also, the LPP basis in the feature space can be represented as a linear combination $\phi(\mathbf{x_i})$s as : $\mathbf{V_F} = \phi(\mathbf{X})\,\mathbf{V}$. Hence, after simplifying, the eigenvalue problem of Equation 2.12 turns out to be:

$$\mathbf{KLKV} = \lambda\mathbf{KMKV} \tag{2.13}$$

### 2.4.2 Kernel ESLPP-MD

As suggested in [61], kernelized version of the proposed ESLPP-MD is developed. In the objective function of ESLPP-MD, we have, $min\sum_{i,j}\|\mathbf{y_i}-\mathbf{y_j}\|^2\tilde{S}_{ij}$, here as data is first mapped to the feature space, $\mathbf{y_i} = \mathbf{V_F}^T\phi(\mathbf{x_i})$. The transformation can be represented as $\mathbf{V_F} = \phi(\mathbf{X})\,\mathbf{V}$ hence, the objective function turns out to be:

$$\begin{aligned}&\tfrac{1}{2}min\textstyle\sum_{i,j}\|\mathbf{y_i}-\mathbf{y_j}\|^2\tilde{S}_{ij}\\ &= \mathbf{V_F}^T\phi(\mathbf{X})\left(\tilde{\mathbf{M}}-\tilde{\mathbf{S}}\right)\phi^T(\mathbf{X})\,\mathbf{V_F}\\ &= \mathbf{V}^T\phi^T(\mathbf{X})\,\phi(\mathbf{X})\left(\tilde{\mathbf{M}}-\tilde{\mathbf{S}}\right)\phi^T(\mathbf{X})\,\phi(\mathbf{X})\,\mathbf{V}\\ &= \mathbf{V}^T\mathbf{K}\tilde{\mathbf{L}}\mathbf{KV}\end{aligned} \tag{2.14}$$

here, $\phi(\mathbf{X})^T\phi(\mathbf{X})$ is nothing but the inner product between the data points in the feature space $F$ and hence can be represented using $k(\mathbf{x_i}, \mathbf{x_j}) = \langle\phi(\mathbf{x_i}), \phi(\mathbf{x_j})\rangle$. $k$ can be selected from any of the Kernels discussed above. Note that, here $\tilde{S}$ is computed in the same way as ESLPP-MD. The normalization constraint in the feature space can be written as $\mathbf{V}^T\mathbf{K}\tilde{\mathbf{M}}\mathbf{KV} = 1$. Thus, the basis can be computed by solving the generalized eigenvalue solution as follows:

$$\mathbf{K}\tilde{\mathbf{L}}\mathbf{KV} = \lambda\mathbf{K}\tilde{\mathbf{M}}\mathbf{KV} \tag{2.15}$$

## 2.5 Face Recognition

The human face is undoubtedly the most common characteristic used by humans to recognize other people. Even though extensive research efforts have been made for Face Recognition in past few years, a system that can be deployed effectively in an unconstrained environment is yet to be seen. The only system that does seem to work well in almost all the conditions is the Human Visual System and to replicate that on a machine is still an open research problem. The current state of art in the Face Recognition is not yet sufficient for more demanding applications [66]. There are other biometric technologies such as fingerprint and iris that currently offer better accuracy but at the same time they require much greater explicit cooperation from the user.

Face Recognition can be divided in two basic applications: identification and verification. In a recognition scenario, the matching is one-to-many, i.e. the face to be identified or recognized is unknown and it is matched against faces of the database containing known individuals [67]. In the verification (authentication) problem, the system confirms or rejects the claimed identity of the input face making the mapping one to one. In both the cases, a database containing face images of known persons is created which can be called training set. Images of these or other persons are used as the test images to match against the images in the already existing database.

High end applications of Face Recognition in human computer interaction, law enforcement and surveillance, information security etc. have made it more appealing area among researchers. It can be applied without active participation from the subject which is an added benefit as far as the biometrics are concerned. Face can be used as a security measure and could replace the identity card or personal identification number or passwords. Automated face recognition can be applied live to search for a watch-list of people, or by using surveillance video of a crime to search from the database of suspects. Another application is Expression Recognition system. It can be used for medical treatment i.e. to monitor the expressions of patients, emotion recognition and animations.

The challenges such as pose and expression variation, vastly changing illumination conditions, change in the appearance of a person over a period of time, scale variability are still very far from being solved. The approaches of Face Recognition can be broadly classified into geometric or feature based techniques and template matching or appearance based techniques [67].

Various methods have been proposed for Face Recognition in literature. There are computational approaches based on deformable templates which characterize the human face [68] or facial features such as eyes, nose, mouth [69], [70], [71], [72],

[73]. Methods based on the search of a significant group of features for example a couple of eyes and a mouth suitably located constitute a significant group in the context of a face [74], [72], [75], [76], [77]. Apart from the feature based approaches, many appearance based approaches have been proposed which work by reducing the dimensionality of the data and finding the subspace of the manifold [16], [18], [19], [20], [21], [26], [24], [30] etc.

Face Recognition is one of the most widely used applications where dimensionality reduction is used and then the desired recognition task is performed. Face images are represented as high-dimensional pixel arrays and due to high correlation between the neighboring pixel values; they often belong to an intrinsically low dimensional manifold. The distribution of data in a high dimensional space is non-uniform and is generally concentrated around some kind of low dimensional structures [22]. Machine learning algorithms find it difficult to classify the given data in such a high dimensional space. Hence the representation of data in a lower dimensional space is highly sought. Locality Preserving Projection is widely used for face recognition tasks and performs better than some of the traditional dimensionality reduction approaches such as PCA and LDA.

In this section, performance of all the variants of LPP discussed in this chapter are evaluated on some of the widely used benchmark face data-sets. Performances of ELPP, ESLPP-MD and their respective Kernel based versions K-ELPP and K-ESLPP-MD are compared with LPP [26, 30], SLPP [52], K-LPP [26] and K-SLPP [61]. Polynomial kernel with degree 2 is used for all the experiments involving kernel mapping before applying the dimensionality reduction techniques.

It is to be noted that we have not compared the proposed scheme with feature based face recognition techniques or any other state of the art face recognition approaches as the aim of this thesis is to propose a robust and efficient similarity preserving dimensionality reduction technique. Face recognition is taken up as an application to showcase the efficiency of the proposal and hence comparisons are reported with other variants of LPP. Also, a complete face recognition system consists of a combination of various pre-processing stages to take care of illumination, pose and other external environmental changes as well as more sophisticated classifiers which boost the performance resulting in higher accuracy. In this thesis, dimensionality reduction is directly applied on the raw face data without any pre-processing and the simplest nearest neighbor classifier, also known as rank one accuracy, is applied for recognition of new test face. It is a common practice in face recognition area to represent the recognition performance as a function of rank. Rank shows among how many top matches is the correct answer [78]. In thesis thesis, all face recognition results are reported in terms of rank one error rate (%).

Three database are used for testing i.e. the ORL database [79], the YALE face database [80] and the AR database [81]. The databases are divided into distinct training and testing sets and nearest neighbor classifier is used to classify the test samples in the projection space. Training samples are used to calculate the transformation matrix that maps the data into the face subspace, test samples are then projected in the lower dimensional projection space using the same transformation matrix and test samples are identified with nearest neighbor classifier. In case of kernel based methods, samples are mapped to the feature space before finding the face subspace.

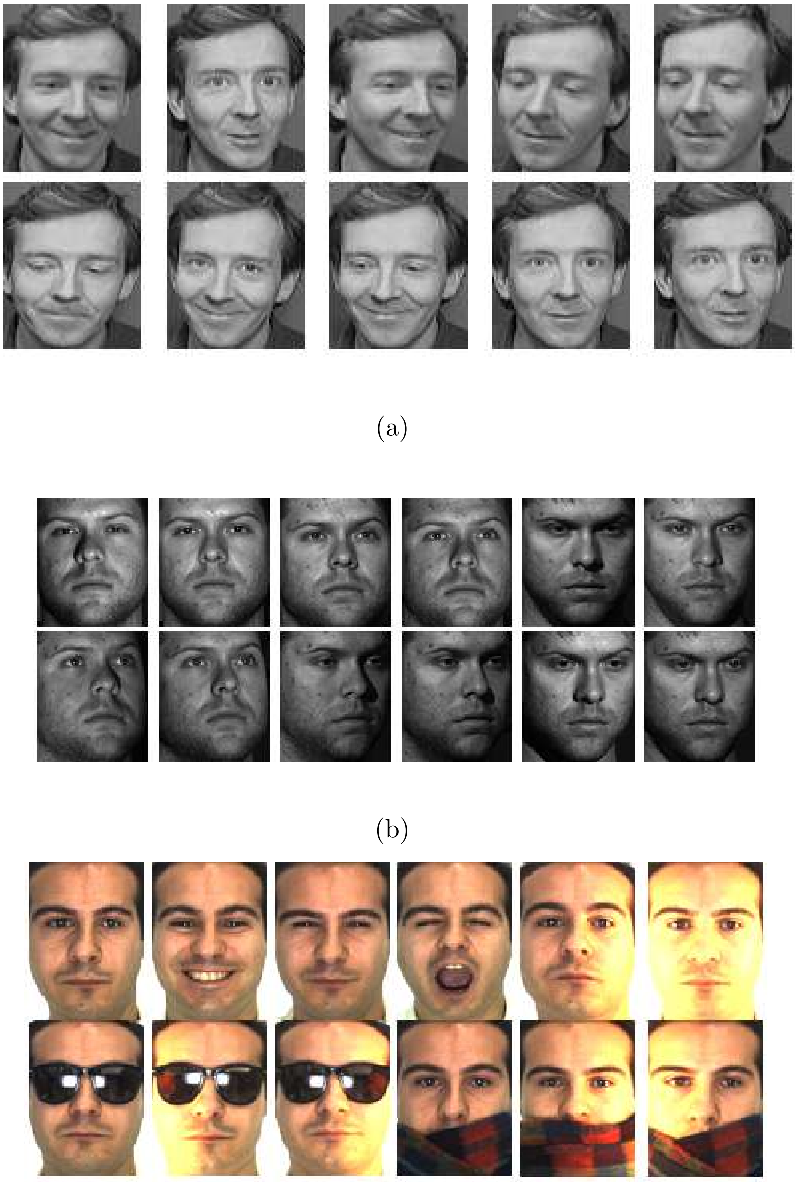
### 2.5.1 ORL Database

The ORL database [79] of faces contains 10 different images of 40 subjects. The face images were captured varying illumination conditions, facial expressions such as open or closed eyes, smiling or not smiling and facial details such as glasses or no glasses. The images were cropped and re-sized 64 × 64 with 256 gray levels per pixels. Some of the sample images from the ORL database are included in Figure 2.13(a). The images are randomly divided into distinct training and testing subsets. 60% samples are used for training whereas 40% samples are used for testing. Recognition results in terms of error rates for LPP, ELPP, SLPP and ESLPP-MD are presented in graphical format in Figure 2.14 and face recognition errors for their respective Kernelized variants i.e. K-LPP, K-ELPP, K-SLPP and K-ESLPP-MD are reported in Figure 2.15.

The plot represents error rates for all the competitive methods with varying number of strongest dimensions used. It can be observed that kernelized versions of all the methods outperform respective non-kernelized methods at higher dimensions with less error rate. Also, K-ESLPP-MD outperforms all other approaches across all the dimensions.

### 2.5.2 YALE Face Database B

The Yale face database B [80] was constructed at the Yale Center for Computational Vision and Control. The database contains 5850 images of 10 subjects each seen under 576 viewing conditions (9 poses × 64 illumination conditions). Again the images were cropped and re-sized to 64 × 64 with 256 gray levels per pixel, some of which are illustrated in Figure 2.13(b). 60% samples from the dataset are randomly selected for training and rest of the samples form test set. Recognition results with varied number of dimensions for ESLPP-MD and other LPP based approaches are reported in Figure 2.16 and Figure 2.17 shows comparison of kernel methods.



(a)

(b)

(c)

Figure 2.13: Sample face images from (a) ORL database (b) YALE face database B (c) AR database

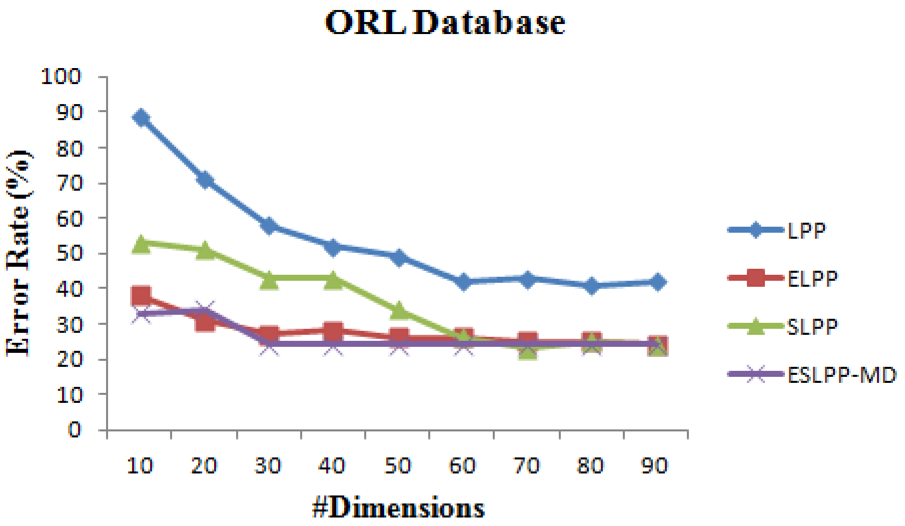


Figure 2.14: Error Rate (%) vs. Dimensionality Reduction on the ORL database (Comparison between variants of LPP)

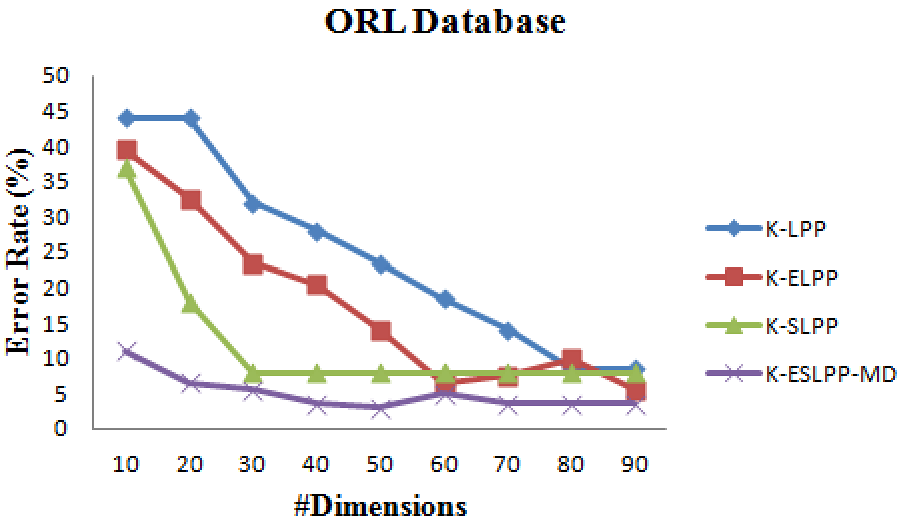


Figure 2.15: Error Rate (%) vs. Dimensionality Reduction on the ORL database (Comparison between kernelized variants of LPP)

The error rate for face recognition using is less in case of Extended Supervised LPP with modified distance (ESLPP-MD) than LPP, ELPP, SLPP. Not only that, ESLPP-MD performs better than K-LPP and is comparable to K-ELPP. The kernelized version of ESLPP-MD beats all the approaches with 98% recognition accuracy with only 30 dimensions.

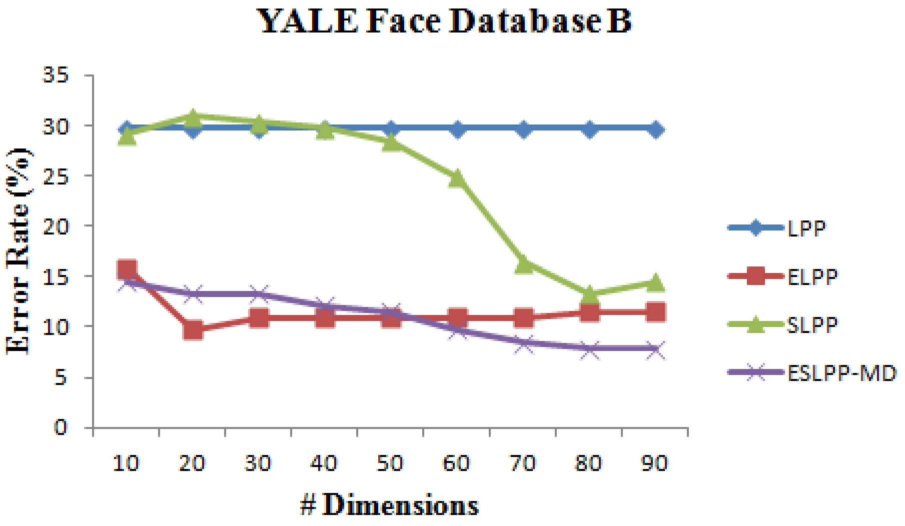


Figure 2.16: Error Rate (%) vs. Dimensionality Reduction on the YALE face database B (Comparison between variants of LPP)

### 2.5.3 AR Database

The AR face database [81] contains 4000 frontal view color images of 100 subjects with different illumination conditions, facial expressions and occlusions such as scarf and sunglasses as shown in Figure 2.13(c). These complex changes make the database more challenging for face recognition problems. Each image of size 40 × 55 is converted to gray scale with 256 gray levels per pixel. 50% samples for each subject are used for training and rest of them are used for testing from a subset of 60 subjects from the database. Results of face recognition showing error rates corresponding to various number of dimensions are illustrated in Figure 2.18 and Figure 2.19.

As can be observed from the plot, ESLPP-MD shows significant improvement in error rates as compared to LPP, ELPP and SLPP. In case of kernelized versions of the approaches, K-ELPP, K-SLPP and K-ESLPP-MD produce less than 10% recognition errors, however performance of K-ESLPP-MD is superior to all the other approaches with more than 99% accuracy.

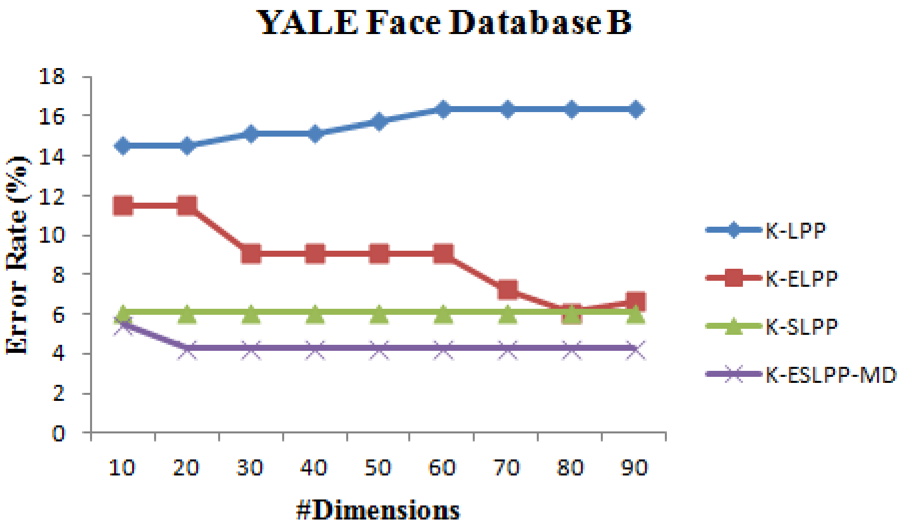


Figure 2.17: Error Rate (%) vs. Dimensionality Reduction on the YALE face database B (Comparison between kernelized variants of LPP)

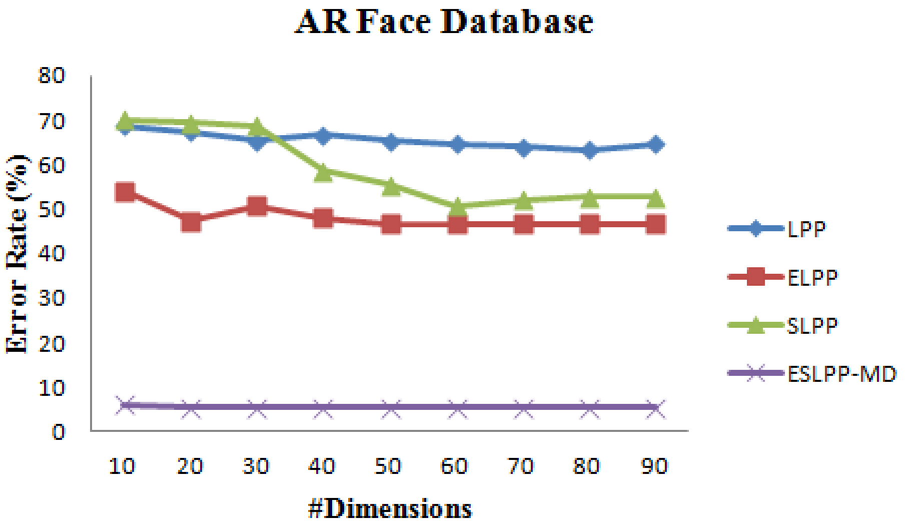


Figure 2.18: Error Rate (%) vs. Dimensionality Reduction on the AR face database (Comparison between variants of LPP)

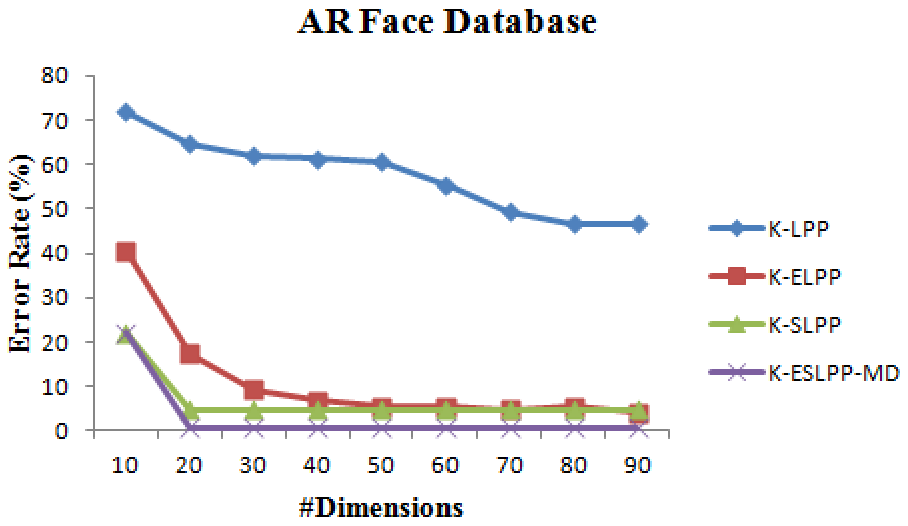


Figure 2.19: Error Rate (%) vs. Dimensionality Reduction on the AR face database (Comparison between kernelized variants of LPP)

## 2.6 Conclusion

Some variants of LPP are suggested in this chapter to have a more robust projection of the data with much less dimensions resulting in better data discrimination. Extending neighborhood by considering moderately distant data points and weighing in a z-shaped, monotonically decreasing fashion not only resolves problems in the overlapping regions of two or more classes to a certain extent, but also enhances the reducibility power of Extended LPP (ELPP). In case of recognition scenarios, class labels of the training data points are already available (supervised) that are incorporated to enhance class discrimination. This information is used to make data points from same class as strong neighbors whereas mapping data points from different classes far apart by reducing and increasing the distance between them respectively. Projection and recognition results using ESLPP with modified distance (ESLPP-MD) are encouraging and show improved class discrimination ability with only a few strongest dimensions. The approach used here is also revealing the expression sub-manifold of a person, boosting the facial expression recognition performance. Significant improvement in the recognition rate on complex database such as AR shows ability of ESLPP-MD to capture illumination and appearance changes as well as occlusions such as faces occluded by scarf and eyes occluded by glasses. Use of kernels to map the data in feature space before applying dimensionality reduction helps in capturing the non-linearity present in the data. Usually, kernel based versions of

LPP perform better than the non-kernelized ones but ESLPP-MD is comparable to Kernel-SLPP. Moreover, the proposed kernelized version of ESLPP-MD outperforms all other techniques. Less than 8% errors across all the test databases are achieved with K-ESLPP-MD with only 20 most significant dimensions.

ESLPP-MD manipulates distance between data points to achieve better class discrimination, which in a sense gives false impression of the data points to the basis learning mechanism. However, shrinking and diverging the distance according to class labels of data points shows better class discrimination of training data points, unknown data points may face difficulty in classification when projected on the learned basis. Thus, evolving the objective function itself to achieve better class discrimination may turn out to be a better solution.

Dimensionality reduction techniques do not always guarantee enhanced performance with increase in number of dimensions used to represent the data. This may be due to redundant information getting added with more number of dimensions. However, in this thesis, we have not studied the redundancy pattern of the dimensions and experiments have also not been carried out for the same. Hence, number of dimensions that produce the best results may depend on the databases used and their complexity.

Also, it is to be noted that, the emphasis of this thesis is on dimensionality reduction approaches and their usability for various challenges. For face recognition, dimensionality reduction techniques are directly applied on the raw face images from different databases and nearest neighbor (rank one) classifier, which is the simplest classifier, is used to classify the test image in the reduced dimensions. Most of the benchmark face recognition techniques are designed with various pre-processing stages along with more robust classifiers to deal with the face images captured in unconstrained environment. Hence, experiments of such databases have not been performed. Also, as main scope of the thesis is not to resolve face recognition issues, but to devise more robust dimensionality reduction approaches, study of benchmark face recognition systems has not been carried out.

# Chapter 3

# Locality Preserving Discriminant Projection

Locality Preserving Projection (LPP) and its variants try to capture the non-linearity present in the data by preserving the neighborhood information in the transformed domain as well. In case of the supervised variants of LPP [52], class information is incorporated while learning the projection matrix. All these techniques aim at projecting the data points having same class labels nearby, without paying attention to the data points belonging to different classes. As discussed in Section 1.2, it is the discriminant property of the data that plays major role in recognition problems. To achieve better class discrimination, data points from different classes should be mapped far apart. We tried to achieve this by shrinking or diverging the distance between data pints depending on the class labels making them strong and week neighbors respectively in the previous proposal ESLPP-MD. Instead of changing the distances, the objective function itself can be tuned to learn a transformation which projects the neighboring points from the same class close to each other and those from different classes far apart in the subspace. In this chapter, an approach that incorporates the discriminant information as well, along with the local structure is proposed. The new objective function is a combination of minimization and maximization functions with the goal of projecting similar data points belonging to same class close by, and dissimilar data points from different classes far apart, achieving better discrimination. Before discussing the proposed approach, an overview of some of the approaches that utilize discriminant information while learning the basis for dimensionality reduction is presented.

## 3.1 Overview of Discriminant Approaches

One of the first works in dimensionality reduction that considered the discriminant information to achieve better separation between classes is Linear Discriminant Analysis (LDA) [18]. LDA takes into consideration the class labels of the training data points. It tries to maximize between class scatter and minimize within class scatter, thus achieving more discrimination between data points from different classes. Formulation of LDA has been discussed in detail in Chapter 1. LDA is a global dimensionality reduction approach and does not pay attention to local structure of the data. Various dimensionality reduction approaches have been proposed in literature that use the discriminant information yet are local in nature [1, 2, 3, 4, 5, 82]. A brief overview of some such approaches is discussed here.

### 3.1.1 Discriminant LPP [1]

As discussed earlier, LPP well preserves the local structure of the data but it seems to de-emphasize the discriminant information which may play vital role in recognition tasks. Discriminant LPP (DLPP) [1] is proposed by adding the discrimination information in LPP with an aim to improve the recognition performance. The objective function of DLPP is as follows:

$$min\frac{\sum_{k=1}^{c}\sum_{i,j=1}^{n_i}\left(\mathbf{y_i^k}-\mathbf{y_j^k}\right)^2 W_{ij}^k}{\sum_{i,j=1}^{c}\left(\mathbf{m_i}-\mathbf{m_j}\right)^2 B_{ij}} \tag{3.1}$$

here, $c$ is number of classes in the data, $\mathbf{y_i} = \mathbf{V^T x_i}$ is projection of data point $\mathbf{x_i}$ using the learned basis $\mathbf{V}$, $\mathbf{m_i} = \mathbf{V^T d_i}$ is mean vector of each class in the projection domain, $W_{ij}^k$ and $B_{ij}$ are the weight matrices weight in which are assigned using the Heat Kernel function as suggested in formulation of LPP [26].

The problem now reduces to solving a generalized eigenvalue problem as follows:

$$\mathbf{XLX^TV} = \lambda\mathbf{DHD^TV} \tag{3.2}$$

Laplacian matrices $\mathbf{L}$ and $\mathbf{H}$ are found in the same manner as that of LPP. Computation of $\mathbf{H}$ is carried out from the mean vectors of each class $\mathbf{D} = [\mathbf{d_1}, \mathbf{d_2}, \cdots, \mathbf{d_c}]$.

### 3.1.2 Discriminant LPP with maximum margin criterion [2, 3, 4]

Various approaches have been proposed in literature that combine the ideas of locality preserving projection (LPP) and maximum margin criterion (MMC) [83]. It is well known that MMC is a method proposed to maximize trace of difference between within class and between class scatter matrices from which LDA can be derived by incorporating some constraints. However, MMC has the same weakness as LDA that both of them neglect the locality information [4]. Thus, the objective function LPP is combined with MCC to achieve better class discrimination. Discriminant LPP with MMC (DLPP/MMC) on this idea has been proposed in [2, 3, 4]. To formulate DLPP/MMC, in addition to the minimization criterion of LPP, a maximization criterion as per MMC has also been added, thus the objective function now turns out to be:

$$\begin{aligned} &min \mathbf{V^T X L X^T V} \,\&\, max \mathbf{V^T} \left(\mathbf{S_B} - \alpha \mathbf{S_W}\right) \mathbf{V} \\ &such\, that\, \mathbf{V^T X M X^T V} = \mathbf{I} \end{aligned} \tag{3.3}$$

here, $\mathbf{M}$ and $\mathbf{L}$ are defined in the same manner as that of LPP [26], $\alpha$ is rescaling factor which is generally set to be 1, $\mathbf{S_B} = \sum_{i=1}^{c} n_i \left(\mathbf{d^i} - \mathbf{d}\right) \left(\mathbf{d^i} - \mathbf{d}\right)^T$ and $\mathbf{S_W} = \sum_{i=1}^{c} \left( \sum_{j=1}^{n_i} \left(\mathbf{x^i_j} - \mathbf{d^i}\right) \left(\mathbf{x^i_j} - \mathbf{d^i}\right)^T \right)$ are between class and within class scatter matrices respectively, $c$ is number of classes in the data set, $n_i$ - number of samples in the $i^{th}$ class, $\mathbf{d}$ - mean vector of all the samples, $\mathbf{d^i}$ - mean vector of $i^{th}$ class and $\mathbf{x^i_j}$ is the $j^{th}$ sample of $i^{th}$ class. After simplifying, the problem reduces to a generalized eigenvalue problem solving which gives DLPP/MMC basis:

$$\left(\mathbf{XLX^T} - \left(\mathbf{S_B} - \alpha \mathbf{S_W}\right)\right) \mathbf{V} = \lambda \mathbf{XMX^T V} \tag{3.4}$$

### 3.1.3 Locality Preserved Maximum Information Projection [5]

Locality Preserved Maximum Information Projection (LPMIP) works on the idea of preserving the local information while maximizing the out-of-locality information of the samples simultaneously [5]. A trade of between local and global information is achieved by using a regularization factor $\alpha$. The objective function of LPMIP is defined as follows:

$$max\, \alpha \sum_{i=1}^{n} \sum_{j \notin \mathcal{O}(i;\epsilon)} \left(\mathbf{y_i} - \mathbf{y_j}\right)^2 W_{ij} - (1-\alpha) \sum_{i=1}^{n} \sum_{j \in \mathcal{O}(i;\epsilon)} \left(\mathbf{y_i} - \mathbf{y_j}\right)^2 W_{ij} \tag{3.5}$$

here, $W_{ij}$ is the weight between $i^{th}$ and $j^{th}$ data point which is obtained using the Heat Kernel function as that of LPP [26], $\mathcal{O}\left(i;\epsilon\right)$ denotes collection of data points falling withing the the $\epsilon$ neighborhood of the $i^{th}$ data points $\mathbf{x_i}$.

LPMIP basis can be obtained directly by solving an eigenvalue problem as the is no constraint involved.

$$\mathbf{XMX^TV} = \lambda\mathbf{V} \tag{3.6}$$

here, $\mathbf{M} = \alpha\tilde{\mathbf{L}} - \mathbf{L}$. $\tilde{\mathbf{L}}$ and $\mathbf{L}$ are the Laplacian matrices obtained from the two parts of the objective function in Equation 3.5 respectively on the lines of LPP [26]. Detailed derivation can be traces from [5]. As opposed to the locality preserving approaches, the eigenvectors corresponding to the largest eigenvalues of the solution are the strongest.

## 3.2 Locality Preserving Discriminant Projection

Locality Preserving Discriminant Projection (LPDP) is formulated, that not only preserves the local structure of the data, but also tries to separate distinct classes in the projection domain. Instead of manipulating the distances between data points, an objective function which is a combination of minimization and maximization problems suggested. The function to minimize the distance between two data points in the projection space if they belong to same class, and maximize it if they come from different classes is proposed. First part of the objective function as shown in Equation 3.7, tries to minimize the distance between two data points belonging to same class in the projection space by assigning higher weights in the similarity matrix whereas heavy penalty in the form of lower weights is assigned if they are from different classes. Note that same objective function is used in all variants of LPP.

$$argmin \sum_{ij} (\mathbf{y_i} - \mathbf{y_j})^2 S_{ij} \tag{3.7}$$

here, $\mathbf{y_i}$ is projection of the data point $\mathbf{x_i}$ in the newly found LPDP projection space using the transformation matrix $\mathbf{V}$ i.e. $\mathbf{y_i} = \mathbf{V}^T\mathbf{x_i}$. $S_{ij}$ is the similarity matrix in which the weight according to the z-shaped weighing function [48] is assigned based on the Euclidean distance between them. The weight is assigned only if the data points belong to same class. Plot of the z-shaped weighing function with different parameters is shown in Figure 3.1.

$$\mathbf{S}_{ij} = \left\{ \begin{array}{ll} 1; & \text{if } d_{ij} \leq a \\ 1 - 2\left(\frac{d_{ij}-a}{b-a}\right)^2; & \text{if } a \leq d_{ij} \leq \frac{a+b}{2} \\ 2\left(\frac{d_{ij}-b}{b-a}\right)^2; & \text{if } \frac{a+b}{2} \leq d_{ij} \leq b \\ 0; \text{otherwise} & \end{array} \right\} \tag{3.8}$$

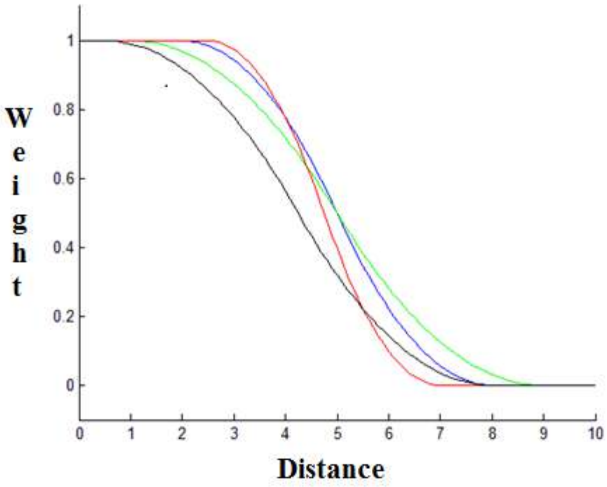


Figure 3.1: Plot Z-shaped function with different parameters

Here, $d_{ij}$ represents the Euclidean distance $\|\mathbf{x_i} - \mathbf{x_j}\|^2$ between $\mathbf{x_i}$ and $\mathbf{x_j}$. $a$ and $b$ specify the range along which the function changes its values and can be controlled. Between $a$ and $b$ two functions are used to make the final output function the z-shaped one. The slope of the function is dependent on the parameters $a$ and $b$. $a$ is set to be a very small value and to set the value of $b$, for each class, maximum pairwise distance of all the data points belonging to that class is used. Hence, for each class, $b$ has a unique value depending upon pairwise distance between the data points belonging to it. All the data points belonging to a particular class have same value of $b$. This makes the procedure adaptive according to the data. Thus, selection of neighbors and weight, both the processes become adaptive and data dependent. Parameters are not to be set explicitly.

Replacing $\mathbf{y_i}$ with $\mathbf{V}^T\mathbf{x_i}$ in Equation 3.7, the minimization problem turns out to be:

$$\begin{aligned}
&\tfrac{1}{2}\textstyle\sum_{ij}(\mathbf{y_i} - \mathbf{y_j})^2 S_{ij} \\
&= \tfrac{1}{2}\textstyle\sum_{ij}(\mathbf{V^T x_i} - \mathbf{V^T x_j})^2 S_{ij} \\
&= \tfrac{1}{2}\textstyle\sum_{ij}(\mathbf{V^T x_i} - \mathbf{V^T x_j})(\mathbf{V^T x_i} - \mathbf{V^T x_j})^T S_{ij} \\
&= \tfrac{1}{2}\textstyle\sum_{ij}(2\mathbf{V^T x_i} S_{ij} \mathbf{x_i V} - 2\mathbf{V^T x_i} S_{ij} \mathbf{x_j V}) \\
&= \mathbf{V^T X M_S X^T V} - \mathbf{V^T X S X^T V} \\
&= \mathbf{V^T X (M_S - S) X^T V} = \mathbf{V^T X L_S X^T V}
\end{aligned} \tag{3.9}$$

$\mathbf{M_S} = \sum_i S_{ij}$ is a diagonal matrix and $\mathbf{L_S} = \mathbf{M_S} - \mathbf{S}$ is the Laplacian matrix.

The main contribution of the present work is the second part as mentioned below. This maximization problem is combined with the first part to make the objective function for locality preserving discriminant approach. Second part of the objective function in Equation 3.10 involves maximization problem in order to ensure that the data points from different classes should be mapped apart from each other. Here, mirror image of the z-shaped weighing function i.e. s-shaped function is used for weighing. Plot of the s-shaped weighing function is shown in Figure 3.2. If the data points belong to different classes, only then the weight according to the distance between them is assigned. The farther the points are in the original space, the higher is the weight assigned. As mentioned earlier, this is to ensure the separability between distinct classes.

$$argmax \sum_{ij} (\mathbf{y_i} - \mathbf{y_j})^2 D_{ij} \tag{3.10}$$

Here, $D_{ij}$ is the dis-similarity matrix, weight in which is assigned according to the s-shaped weighing function (Equation 3.11).

$$\mathbf{D}_{ij} = \left\{ \begin{array}{ll} 0; & \text{if } d_{ij} \leq a \\ 2\left(\frac{d_{ij}-b}{b-a}\right)^2; & \text{if } a \leq d_{ij} \leq \frac{a+b}{2} \\ 1 - 2\left(\frac{d_{ij}-a}{b-a}\right)^2; & \text{if } \frac{a+b}{2} \leq d_{ij} \leq b \\ 1; \text{otherwise} & \end{array} \right\} \tag{3.11}$$

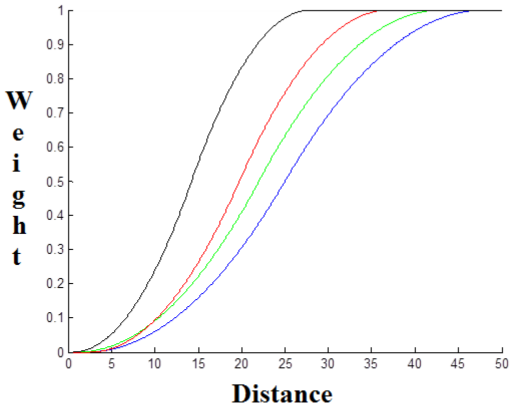


Figure 3.2: Plot S-shaped function with different parameters

As shown in Equation 3.9, the maximization problem in matrix format can be represented as:

$$\begin{aligned}
&\tfrac{1}{2}\textstyle\sum_{ij}(\mathbf{y_i}-\mathbf{y_j})^2 D_{ij}\\
&=\tfrac{1}{2}\textstyle\sum_{ij}(\mathbf{V^T x_i}-\mathbf{V^T x_j})^2 D_{ij}\\
&=\tfrac{1}{2}\textstyle\sum_{ij}(\mathbf{V^T x_i}-\mathbf{V^T x_j})(\mathbf{V^T x_i}-\mathbf{V^T x_j})^T D_{ij}\\
&=\tfrac{1}{2}\textstyle\sum_{ij}(2\mathbf{V^T x_i}D_{ij}\mathbf{x_i V}-2\mathbf{V^T x_i}D_{ij}\mathbf{x_j V})\\
&=\mathbf{V^T X M_D X^T V}-\mathbf{V^T X D X^T V}\\
&=\mathbf{V^T X}(\mathbf{M_D}-\mathbf{D})\mathbf{X^T V}=\mathbf{V^T X L_D X^T V}
\end{aligned}\tag{3.12}$$

here, $\mathbf{L_D}=\mathbf{M_D}-\mathbf{D}$ and $\mathbf{M_D}=\sum_{ij}D_{ij}$. The maximization problem can be converted to a minimization problem in order to express both the objective functions together. Experiments were carried out taking convex combination of both the objective functions to be minimized i.e. $\lambda\left[\left(\mathbf{V^T X L_S X^T V}\right)\right]-(1-\lambda)\left[\mathbf{V^T X L_D X^T V}\right]$. It has been observed that assigning equal weights to both the parts produces the best results. Thus, the final optimization problem used is as follows:

$$argmin\,\mathbf{V^T X L_S X^T V}-\mathbf{V^T X L_D X^T V}\tag{3.13}$$

In order to negate the arbitrary scaling, data normalization constraint is imposed as follows:

$$\mathbf{YY^T}=1\Rightarrow\mathbf{V^T X X^T V}=1\tag{3.14}$$

Hence, the final optimization problem turns out to be:

$$argmin_{\mathbf{V^T X X^T V}=1}\,\mathbf{V^T X L_S X^T V}-\mathbf{V^T X L_D X^T V}\tag{3.15}$$

The optimized solution is obtained by using the Lagrange's multiplier approach.

$$\mathcal{L}=\mathbf{V^T X L_S X^T V}-\mathbf{V^T X L_D X^T V}-\lambda\left(\mathbf{V^T X X^T V}-1\right)$$

Equating the partial derivatives of $\mathcal{L}$ with respect to $\lambda$ and $\mathbf{V}$ to zero, we obtain:

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial\mathbf{V}}=0&\Rightarrow 2\mathbf{X L_S X^T V}-2\mathbf{X L_D X^T V}-2\lambda\mathbf{X X^T V}=0\\
&\Rightarrow\mathbf{X L_S X^T V}-\mathbf{X L_D X^T V}=\lambda\mathbf{X X^T V}
\end{aligned}$$

Thus, the transformation matrix $\mathbf{V}$ that solves the optimization problem is obtained from the following generalized eigenvalue problem:

$$(\mathbf{X L_S X^T}-\mathbf{X L_D X^T})\mathbf{V}=\lambda\mathbf{X X^T V}\tag{3.16}$$

The data matrix $\mathbf{X}$ thus transformed (Equation 3.16) ensures that points which are neighbors and belong to the same class would also be neighbors where as neighboring points not belonging to same class would be projected apart.

### 3.2.1 Difference between LPDP and other Discriminant approaches

Though, the proposed dimensionality reduction approach LPDP works on the similar concept of preserving the local structure and separating different classes as that of the discriminant approaches discussed earlier, there are considerable differences between the existing ones and the current proposal. The locality preserving property that all these techniques have in common is the locality preserving property inherited from LPP [26] i.e. similar data points should be projected close by in the projection space. It is the way of adding the discriminant information that is different in all the competitive approaches.

DLPP [1] discussed in Section 3.1.1, aims at achieving maximum class separation by maximizing the distance between means of different classes. Based on the mean vectors of respective classes, discrimination in the projection space is attained. DLPP/MMC [2, 3, 4] explained in Section 3.1.2 uses maximum margin criterion by maximizing the difference the between class and withing class variances. The between class and within class matrices are obtained in the same manner as that of LDA [18].

The dimensionality reduction approach that is closest to the current proposal is LPMIP [5] described in Section 3.1.3. This method tries to preserve maximum information in the projection space by maximizing the distance between data points that do not belong within $\epsilon$ neighborhood while minimizig the distance between data points which are neighbors. Weighing is performed using the Heat kernel function as done for LPP. Though LPDP shares similar objective like LPMIP, there are considerable differences between both the approaches. LPDP imposes a normalization constraint on the data points in the projection domain which is ignored by LPMIP. It has been reported in [49, 22, 8] that ignoring the normalization constraint results in decrease in the performance of the dimensionality reduction techniques. Also, this constraint is very stringent, relaxing which affects the basis computation and hence it is very important to impose it while learning the basis. LPMIP uses Heat kernel function to weight the data points in both the parts. It has been discussed in Section 2 that in the Heat kernel function, width of the kernel is controlled by $t$. Generally, mean of the pair wise distances is used as $t$ but this value does not necessarily find the optimal underlying manifold for all data sets. The proposed LPDP formulation uses z-shaped (Equation 3.8) and s-shaped (Equation 3.11) weighing functions which automatically tune the parameters based on the data and hence are more suitable. It has been reported in [5] that, LPMIP pays very little emphasis (between 0.001 to 0.0001) to non-neighbor data points, giving more emphasis to local structure [5]. In case of LPDP, equal emphasis is given to both local structure preservation and discrimination criterion which has been established experimentally as it produces the best results.

In addition to comparing these techniques theoretically, extensive face recognition experiments using DLPP/MMC, LPMIP and LPDP along with some locality preserving dimensionality reduction techniques have been reported later in this chapter in Figure 3.11.

### 3.2.2 Projection of the video database using LPDP

As discussed in Section 2.3.1, strongest two and three dimensional projections of image data are shown to visualize effectiveness of the proposed dimensionality reduction schemes. Video database consisting face images of subjects with four different facial expressions (angry, normal, smile and open mouth) is used to report the projection results in Section 2.3.1. Frames of the face images are extracted and re-sized to 100 × 100 thus having 10,000 dimensions. To show the class discriminating ability of the approach, strongest three dimensional projection of individuals from the video data set are shown in Figure 3.3. Face images of different persons are denoted by different color or sign along with the corresponding sample face image. As the data contains four different facial expressions, expression sub-manifolds of the projected data are enlarged for some persons. It can be observed that using only three strongest dimensions, the approach not only discriminates different persons, but also discriminates various expressions of a person.

In addition to the projection results, face and facial expression recognition experiments on the the video database are carried out by randomly sampling 50% face images for training data set and out of the rest of the samples, 3 distinct test data sets are generated. Results in terms on error rates in percentage for 3 different test runs are reported in Table 3.1 for face recognition. Each run is performed on a distinct test data-set. It can be observed that, the performance of LPDP is comparable to the other approaches with all the approaches giving almost 100% accuracy. As the database contains various expressions of the subject, facial expression recognition is also performed in order to analyze the performance of the approaches in recognizing the expression sub-manifold. The error rates are reported in Table 3.2 with various dimensions for the same runs as used for face recognition. LPDP clearly outperforms the other approaches with the average recognition rate of 87% whereas the average recognition rates for LPP, ELPP, SLPP and ESLPP-MD are 71.9%, 84.3%, 84.9% and 84% respectively. This experiment clearly shows the discrimination power of the proposed LPDP scheme.

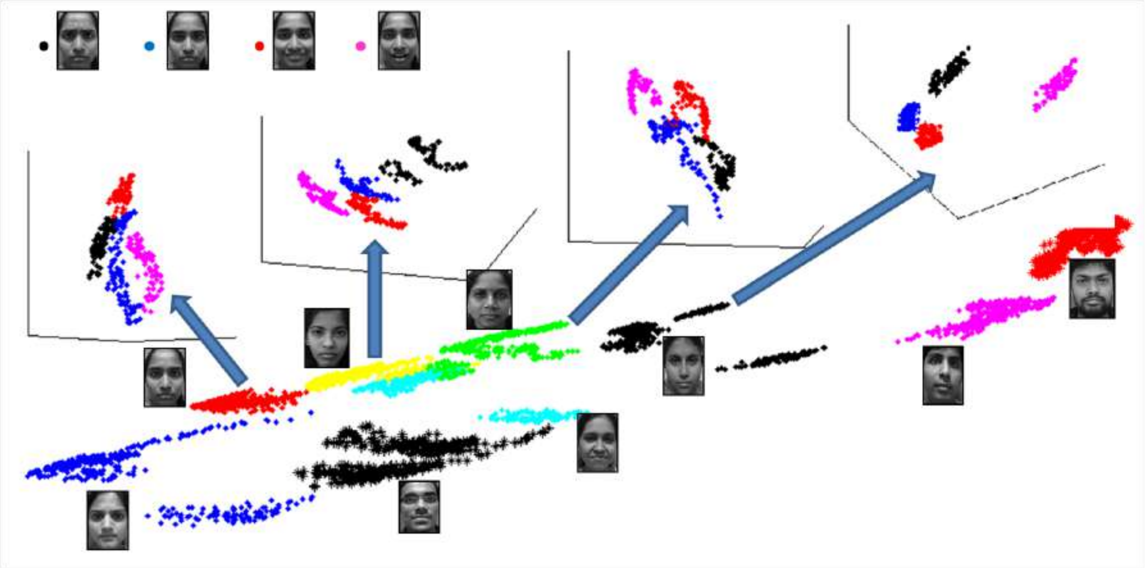


Figure 3.3: Strongest 3 dimensional projection using LPDP of all the subjects from video data-set and some examples of expression discrimination in a person's manifold. Each person is represented by a different sign or color (1 sample face image for corresponding person is shown) in the figure on left whereas for some persons in the enlarged projection, each expression is represented by different color. $x$, $y$ and $z$ axes represent the direction of first three strongest components of basis $\mathbf{V}$ respectively.

Table 3.1: Errors (in %) of face recognition on various test runs from the video data set using nearest neighbor approach. Actual dimensions of the raw data were 10000.

| **Run** | **#Dimensions** | **Approach** | | | | |
|---|---|---|---|---|---|---|
| | | LPP | ELPP | SLPP | ESLPP-MD | LPDP |
| #1 | 3 | 85.22 | 3.86 | 15.22 | 0.45 | 4.5 |
| | 50 | 88.18 | 0.22 | 0 | 0.22 | 0.2 |
| | 500 | 86.59 | 0 | 0 | 0.22 | 0 |
| | 1000 | 15.68 | 0 | 0 | 0.22 | 0 |
| #2 | 3 | 83.63 | 5 | 7.27 | 0.45 | 4 |
| | 50 | 84.31 | 0.22 | 0 | 0.22 | 0 |
| | 500 | 83.86 | 0 | 0 | 0.22 | 0 |
| | 1000 | 14.77 | 0 | 0 | 0.22 | 0 |
| #3 | 3 | 83.86 | 5.22 | 7.5 | 0.90 | 4.5 |
| | 50 | 83.40 | 0.68 | 0.45 | 0.68 | 0.9 |
| | 500 | 81.81 | 0.22 | 0.22 | 0.68 | 0.68 |
| | 1000 | 14.77 | 0.22 | 0.22 | 0.68 | 0.45 |

Table 3.2: Errors (in %) of facial expression recognition on various test runs from the video data set using nearest neighbor approach. Actual dimensions of the raw data were 10000.

| **Run** | **#Dimensions** | **Approach** | | | | |
|---|---|---|---|---|---|---|
| | | LPP | ELPP | SLPP | ESLPP-MD | LPDP |
| #1 | 3 | 93.18 | 33.86 | 32.95 | 26.18 | 28 |
| | 50 | 94.31 | 16.59 | 21.36 | 18.86 | 16.75 |
| | 500 | 94.31 | 15.90 | 21.13 | 19.77 | 15.25 |
| | 1000 | 28.40 | 15.90 | 20.68 | 15.22 | 13 |
| #2 | 3 | 91.36 | 34.09 | 22.27 | 33.40 | 25.25 |
| | 50 | 91.81 | 15.90 | 17.27 | 21.36 | 15.25 |
| | 500 | 90.68 | 15.90 | 17.27 | 18.18 | 13.75 |
| | 1000 | 27.5 | 15.22 | 16.36 | 17.72 | 12.75 |
| #3 | 3 | 90.22 | 33.63 | 32.27 | 25 | 27.25 |
| | 50 | 90.22 | 17.5 | 17.27 | 18.86 | 18.75 |
| | 500 | 90 | 17.72 | 17.04 | 15.22 | 16 |
| | 1000 | 28.40 | 15.90 | 17.04 | 15 | 13.5 |

## 3.3 Kernelization of LPDP

As discussed in Section 2.4, data in its raw form may not lie on a linear manifold and hence linear dimensionality reduction approaches may fail to reveal the data non-linearity. To exploit the non-linearity of the data and achieve better discrimination, data points are first transformed in a non-linear feature space and then the dimensionality reduction approach is applied. Extensions of PCA [58, 57], LDA [59, 60], LPP [61] for the kernel based variants have been proposed in literature. Details about the process of kernelization, its advantages and kernel based variants of PCA, LDA, LPP and ESLPP-MD are discussed in Section 2.4. As the current proposal LPDP is also a linear dimensionality reduction approach, complex non-linear structures may be difficult to discriminate. Hence, kernelized extension of Locality Preserving Discriminant Projection (K-LPDP) is proposed. Formulation and derivation of the kernelization of locality preserving discriminant projection (K-LPDP) are discussed here. The mapping of a data point from $mn$-dimensional space to a non-linear feature space $\mathcal{F}$ is denoted as $\phi : \mathbb{R}^n \rightarrow \mathcal{F}$ where $\phi$ is the mapping function. Hence, the data points in original space $\{\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_N}\}$ get mapped to $\{\phi(\mathbf{x_1}), \phi(\mathbf{x_2}), \cdots, \phi(\mathbf{x_N})\}$. The dimensionality reduction approaches are applied on these mappings as opposed to the data in the original space.

Before applying dimensionality reduction, the kernel based approaches first map the data in a feature space as discussed in Section 2.4. For kernelized version of

LPDP, the objective function and constraint can be rewritten as:

$$min\, \mathbf{V_F}^T \phi\left(\mathbf{X}\right)\left[\mathbf{L_S}-\mathbf{L_D}\right]\phi^T\left(\mathbf{X}\right)\mathbf{V_F} \qquad \text{subject to } \mathbf{V_F}^T \phi\left(\mathbf{X}\right)\phi^T\left(\mathbf{X}\right)\mathbf{V_F}=\mathbf{I} \tag{3.17}$$

here, $\mathbf{V_F}$ is the transformation matrix learned from the data represented in the feature space $\mathcal{F}$, thus, $\mathbf{V_F}=\phi\left(\mathbf{X}\right)\mathbf{V}$. Replacing $\mathbf{V_F}$ in the form of $\mathbf{V}$, optimization problem becomes:

$$min\, \mathbf{V}^T\phi^T\left(\mathbf{X}\right)\phi\left(\mathbf{X}\right)\left[\mathbf{L_S}-\mathbf{L_D}\right]\phi^T\left(\mathbf{X}\right)\phi\left(\mathbf{X}\right)\mathbf{V} \qquad \text{subject to } \mathbf{V}^T\phi^T\left(\mathbf{X}\right)\phi\left(\mathbf{X}\right)\phi^T\left(\mathbf{X}\right)\phi\left(\mathbf{X}\right)\mathbf{V}=\mathbf{I} \tag{3.18}$$

$\phi^T\left(\mathbf{X}\right)\phi\left(\mathbf{X}\right)$ can be represented by a kernel matrix $\mathbf{K}$ which is inner product between the data points in feature space i.e. $\mathbf{k}\left(\mathbf{x_i},\mathbf{x_j}\right)=\left\langle\phi\left(\mathbf{x_i}\right),\phi\left(\mathbf{x_j}\right)\right\rangle$. Thus, relationship of the data points in the feature space can be obtained without knowledge about the mapping function $\phi$, using the kernel trick as discussed in Section 2.4. The optimization problem in Equation 3.18 can be reformulated using the Kernel matrix $\mathbf{K}$. Thus, the transformation matrix can be obtained from the solution of the following generalized eigen value problem:

$$\mathbf{K}\left(\mathbf{L_S}-\mathbf{L_D}\right)\mathbf{KV}=\lambda\mathbf{KKV} \tag{3.19}$$

Some of the very popular kernels are as linear kernel, polynomial kernel, Gaussian kernel etc. which have been listed in Section 2.4. One of these or any other kernel can be used to map the data in kernel space. Comparison of the proposed approach with some of the other kernel based dimensionality reduction approaches is carried out for face recognition in the next section.

## 3.4 Experimental Results

As reported earlier, face recognition is the most widely used application of dimensionality reduction. Also, as the class labels of the training face samples are generally known, better discrimination between face images of different persons can be achieved by using the discriminant information along with the local neighborhood information. Face recognition using LPDP and K-LPDP has been performed on some benchmark face databases and the results are compared with LPP, ELPP, SLPP, ESLPP-MD and their respective kernel based variants. Out of different kernel functions used for the experimentation purpose, Gaussian kernel produced the best results, hence all the experiments of the kernel based variants reported in this thesis are carried out with the Gaussian kernel. The face recognition experiments are performed using nearest neighbor classifier and reported in terms of rank one error rate. In addition to the local approaches, LPDP is also compared with other discriminant approaches DLPP/MMC [2, 3, 4] and LPMIP [5].

### 3.4.1 Face database description

Face recognition experiments are performed on some of the benchmark face databases. Before reporting the face recognition performance, a brief overview of the databases used in presented.

#### ORL Database

The ORL face database [79] [1] contains face images with different illumination and facial expressions of 40 subjects, samples of which are shown in Figure 2.13. Face images are cropped and re-sized to 64 × 64, thus the data lies in 4096 dimensional space. The 60% training and 40% testing samples selected randomly from the ORL database for face recognition experiments.

#### The YALE face database B

The Yale face database B [80] contains 5850 images of 10 subjects each seen under 576 viewing conditions (9 poses × 64 illumination conditions). The changes in pose and illumination conditions makes it a difficult data-set for face recognition tasks. Some sample images from the database are shown in Figure 2.13. Again, the database has been randomly partitioned in 60% training and 40% training samples, each lying in 4096 dimensional space.

#### Japanese Female Facial Expression Database (JAFFE)

Japanese Female Facial Expression Database (JAFFE) [84] contains 213 images of 7 facial expressions (6 basic facial expressions + 1 neutral) posed by 10 Japanese female models. Few images from the JAFFE database are shown in Figure 3.4. 50% samples are randomly sampled for training and the rest are used for testing.

[1] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

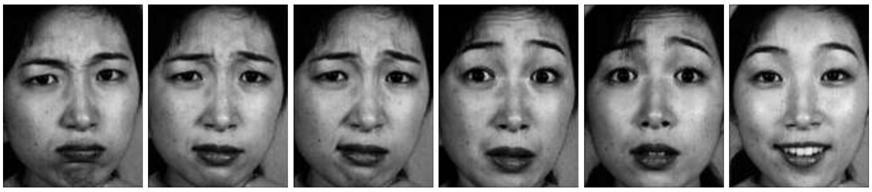


Figure 3.4: Sample face images from JAFFE database

### 3.4.2 LPDP vs. K-LPDP

As it is reported that kernelization procedure helps in exploiting the non-linear manifold of the data and hence it is expected to enhance the recognition accuracy, kernelized version of LPDP is also compared with conventional LPDP approach to show the advantage of using the kernel mapping.

Face recognition results using LPDP and K-LPDP on both ORL and YALE face database B are reported in Table 3.3 for dimensions ranging from 10 to 90. It can be observed that the kernelized version outperforms conventional LPDP across almost all the dimensions. 97.5% face recognition accuracy for the ORL face database and 96.97% for the YALE face database B using the kernelized LPDP approach is achieved as opposed to 95% and 92.12% using the conventional LPDP approach respectively. Improved performance after mapping the data in the kernel space shows the significance of kernelization procedure in unveiling the non-linearity of the data.

Table 3.3: Results (Errors in %) of face recognition using LPDP and K-LPDP on the ORL database and YALE face database B with varying dimensions.

| **# Dimensions** | **Error Rate %** | | | |
|---|---|---|---|---|
| | **ORL Face database** | | **YALE face database B** | |
| | **LPDP** | **K-LPDP** | **LPDP** | **K-LPDP** |
| 10 | 12 | 12 | 13.94 | **3.03** |
| 20 | 10 | 12.5 | 11.52 | 4.24 |
| 30 | 10.5 | 12.5 | 9.7 | 4.24 |
| 40 | 7.5 | 13 | 9.7 | 3.64 |
| 50 | 7 | 12 | 8.48 | 3.64 |
| 60 | 6 | 5 | 8.48 | 3.64 |
| 70 | 6 | 3 | 7.88 | 3.64 |
| 80 | 5 | **2.5** | 7.88 | 3.64 |
| 90 | 5 | 2.5 | 7.88 | 3.64 |

### 3.4.3 Performance analysis of LPDP and other locality preserving techniques

Face recognition performance of the proposed dimensionality reduction technique is compared with LPP[26, 30], SLPP [52], ELPP [48] and ESLPP-MD (proposed in Chapter 2) on all three face databases. The kernel based variant of LPDP i.e. K-LPDP is also compared with K-LPP [26], K-SLPP [61], K-ELPP and K-ESLPP-MD.

The error rates for LPP, ELPP, SLPP, ESLPP-MD and LPDP using nearest neighbor classifier on ORL database are reported in Figure 3.5. Significant improvement in the performance can be observed for LPDP with least error rates across all the dimensions. Moreover, the kernelized variants of these dimensionality reduction approaches are also compared and the error rates for the ORL database are reported in Figure 3.6. Performance of K-ESLPP-MD discussed in Section 2.3 is on par with that of K-LPDP. However, it can be observed that, for both the databases, the proposed approach achieves more than 97% face recognition accuracy, out performing other competing approaches.

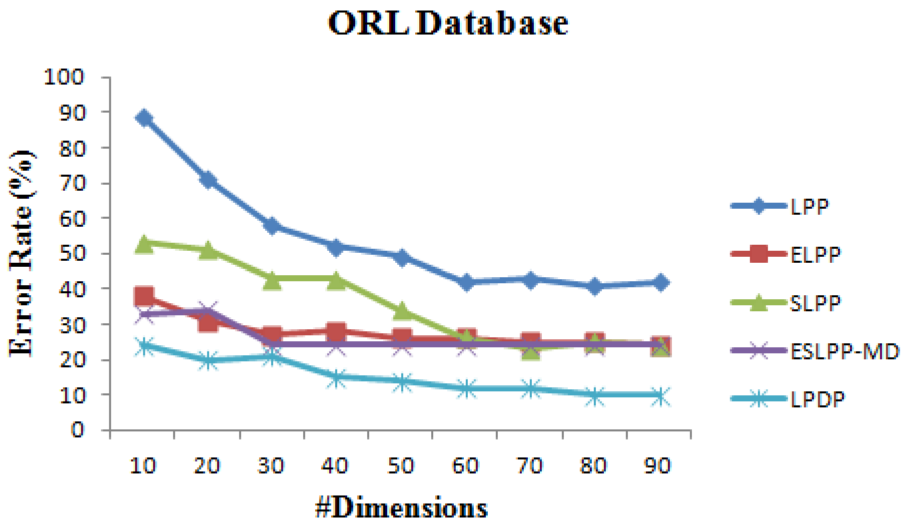


Figure 3.5: Error Rate (%) vs. Number of dimensions for the ORL database (Comparison with variants of LPP)

To show the discriminant ability of the proposed approach, strongest two dimensional projections of similar-looking faces of two persons from the YALE face database using all the competitive approaches are shown in Figure 3.7. Horizontal axis represents the strongest component of the projection matrix whereas the vertical axis

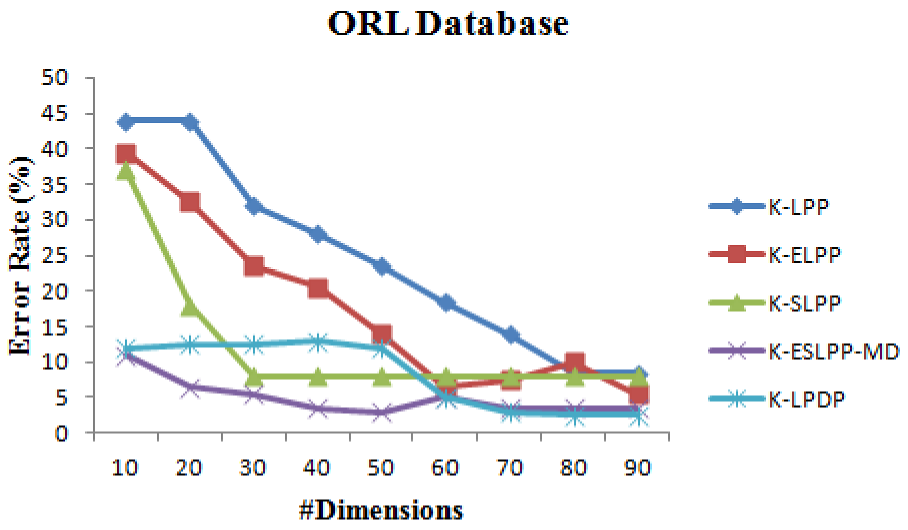
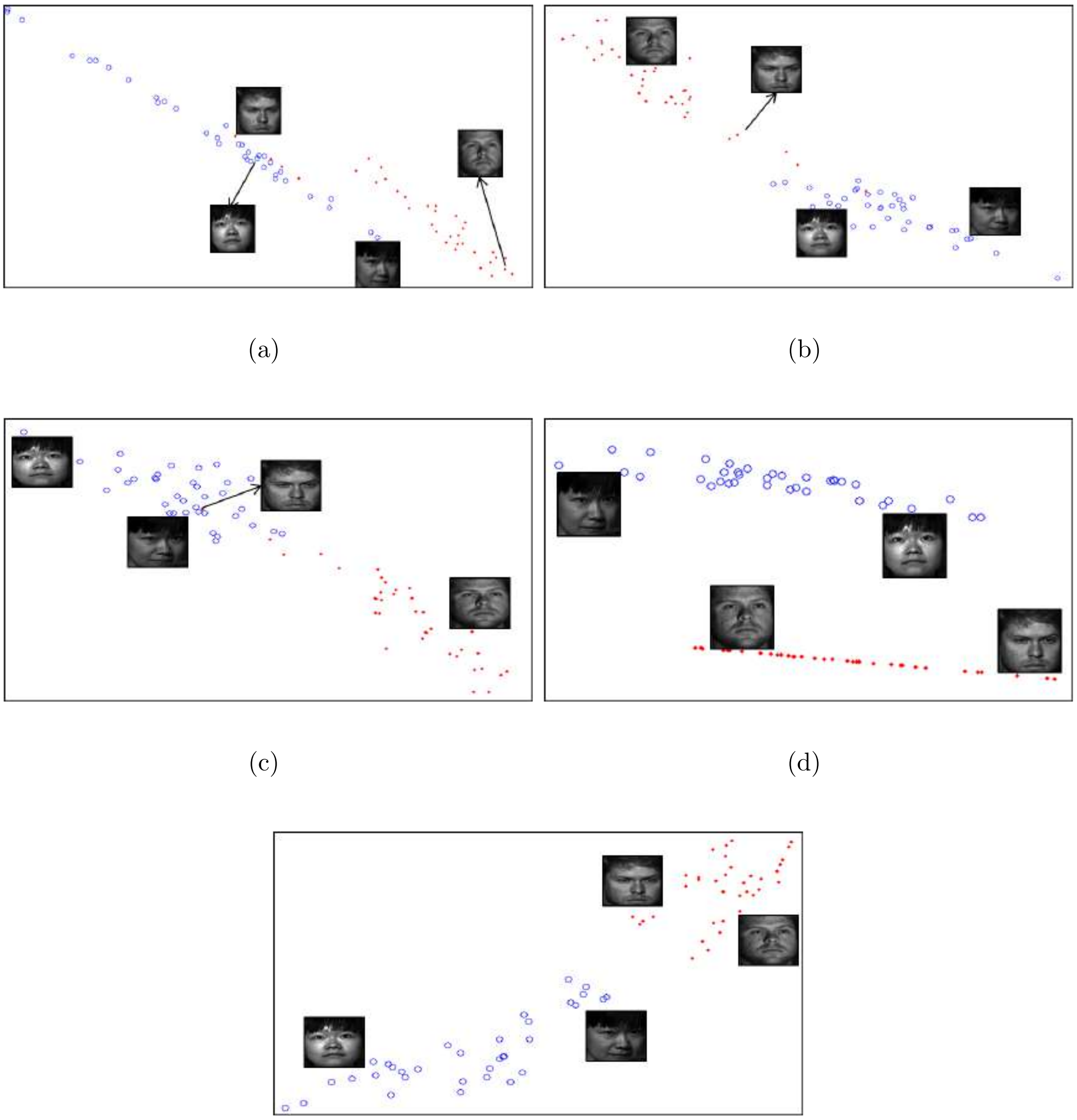


Figure 3.6: Error Rate (%) vs. Number of dimensions for the ORL database (Comparison with kernelized variants of LPP)

represents the next strongest component. Clear discrimination between faces of two persons can be observed using just strongest two dimensions using both ESLPP-MD and LPDP.

Face recognition experiments with varied number of dimensions on the YALE face database B are reported in Figure 3.8. For lower dimensions, the proposed approach performs better, but as the dimensions increase, the performance of LPDP is comparable to ESLPP-MD. Comparison of K-LPDP is also performed with other kernel based variants results of which are shown in Figure 3.9. The current proposal outperforms all the other variants showing better discrimination capability, achieving more than 96% recognition accuracy.

Face recognition experiments on an expression database i.e. JAFFE database are reported in Figure 3.10. All the compared approaches report less than 10% error rates with LPDP giving the least i.e. 1% error rate, whereas SLPP and ESLPP have only 3% and 2% errors. As the error rates for face recognition are very less in case of the JAFFE database, with LPDP giving almost 99% accuracy, experiments using the kernel based variants have not been performed.



(a)

(b)

(c)

(d)

(e)

Figure 3.7: Strongest two dimensional projection of face images having some similar properties from the YALE face database. (a)LPP (b)ELPP (c)SLPP (d)ESLPP-MD (e) LPDP Horizontal axis represents the direction of the strongest component of basis $\mathbf{V}$ while vertical axis represents the $2^{nd}$ strongest component.

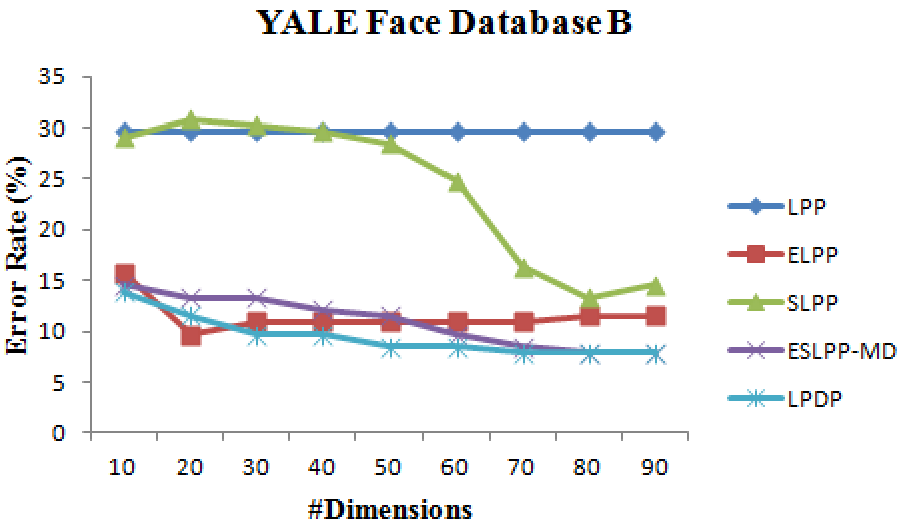


Figure 3.8: Error Rate (%) vs. Number of dimensions for the YALE database (Comparison with variants of LPP)

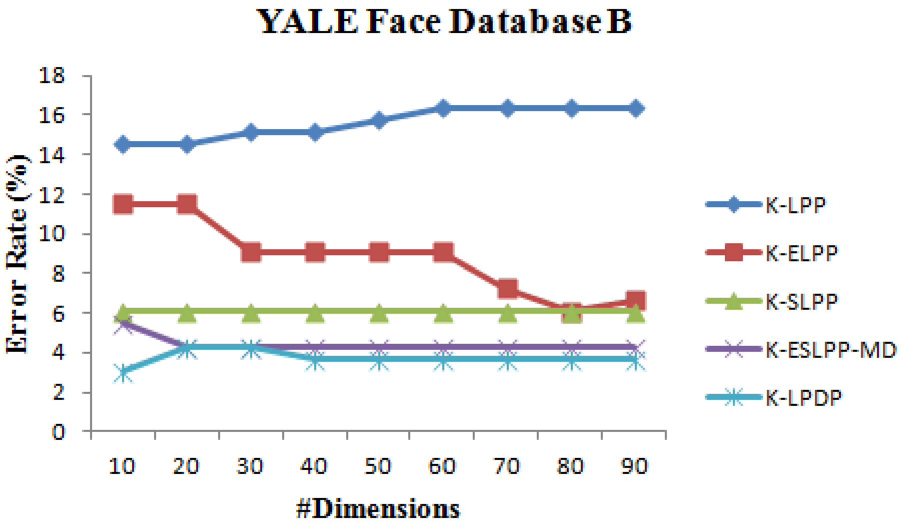


Figure 3.9: Error Rate (%) vs. Number of dimensions for the YALE database (Comparison with kernelized variants of LPP)

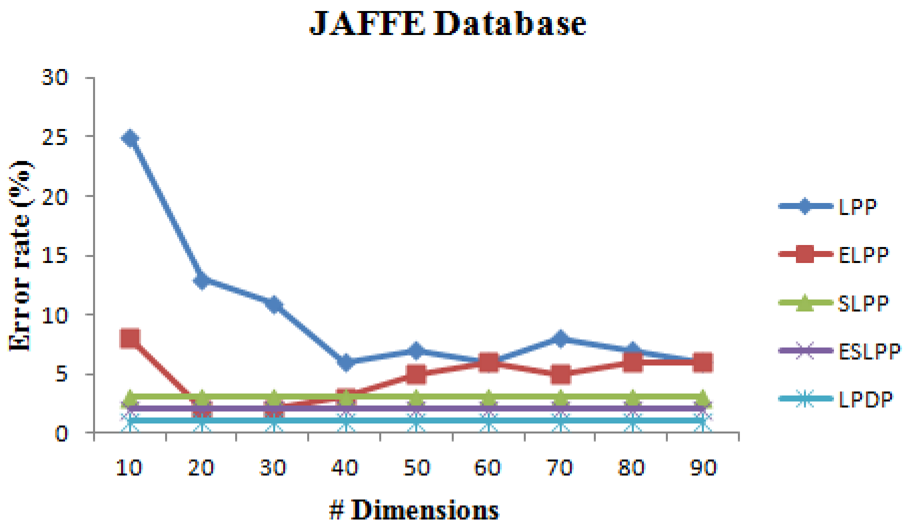


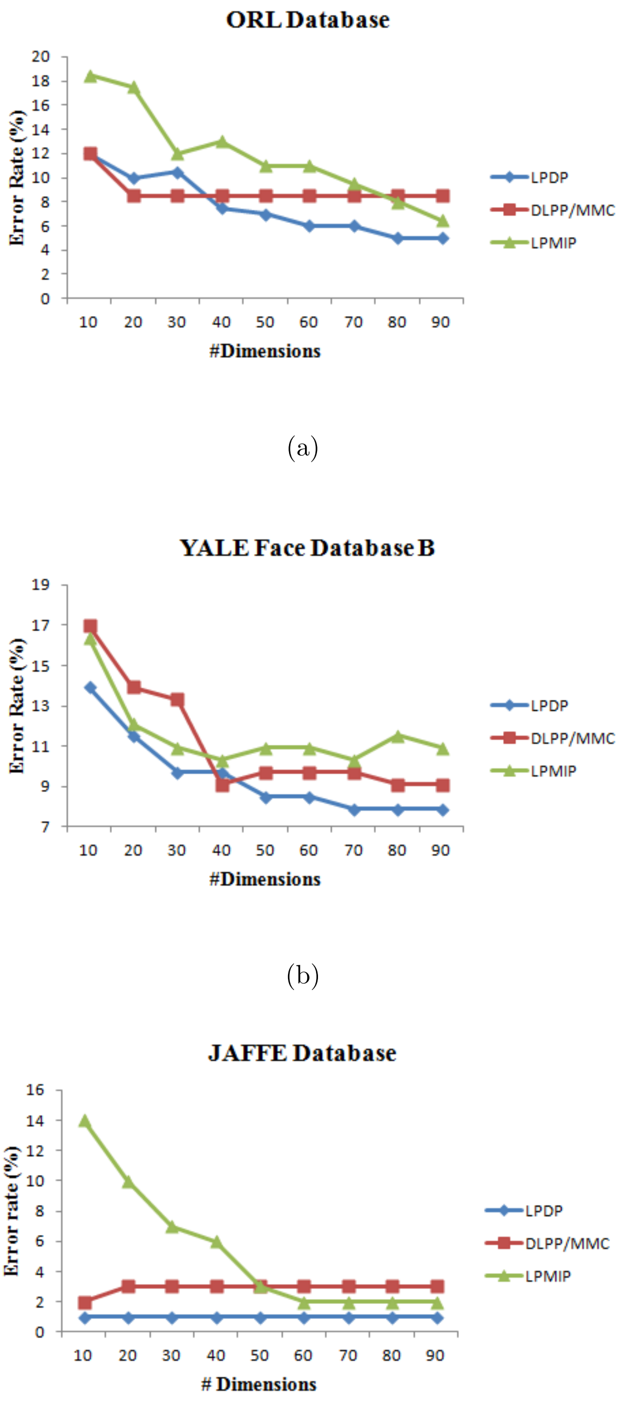
Figure 3.10: Error Rate (%) vs. Number of dimensions for the JAFFE database (Comparison with variants of LPP)

### 3.4.4 Performance analysis of LPDP and discriminant techniques

Proposed LPDP is a dimensionality reduction approach that takes into consideration both the local structure of the data and at the same time emphasizes on separating different classes. To have a complete performance analysis of LPDP, it is important to report comparison with other techniques working on similar idea such as DLPP [1], DLPP/MMC [2, 3, 4], LPMIP [5] etc. DLPP/MMC and LPMIP are reported to be performing better that DLPP and hence face recognition results using these two techniques are compared with LPDP. Also, it is to be noted that the kernel based variants of DLPP/MMC and LPMIP are not available and hence their comparison with K-LPDP is not reported. It can be observed that LPDP consistently outperforms both DLPP/MMC and LPMIP on all three face databases used for experimentation with only $70 - 80$ strongest dimensions producing best results.

## 3.5 Conclusion

A dimensionality reduction technique, locality preserving discriminant projection, that not only preserves the local geometry of the data but also discriminates data points from different classes in the newly found subspace is proposed. Face and fa-



(a)

(b)

(c)

Figure 3.11: Error Rate (%) vs. Number of dimensions for (a) ORL database (b) YALE face database B (c) JAFFE database (Comparison with other discriminant techniques)

cial expression recognition using the face images thus projected in this subspace are performed. It has been observed that the proposed approach significantly improves the performance of face recognition on the used benchmark face databases. In addition to discriminating face images of different persons, the LPDP discriminates various expressions of the same person as well. LPDP tries to maximize the separability between classes while preserving the neighborhood information of the data points in the projection space. Kernelization of LPDP is also proposed with the aim of capturing complex non-linear changes present in the face images and thereby enhance the face recognition performance. Notable improvement is observed in the recognition accuracy using K-LPDP over the conventional non-kernelized LPDP. The proposed approach is also compared with the kernelized variants of other local structure preserving approaches. Higher recognition performances are achieved on both the databases using proposed K-LPDP.

# Chapter 4

# Orthogonalization of Extended LPP

Locality Preserving Projection (LPP), as discussed earlier in detail, is a linear transformation approach that preserves the local structure (neighborhood information of the data) in the transformed domain as well [26]. The transformation matrix i.e. basis learned by LPP are not orthogonal in nature. Non-orthogonality of LPP makes it difficult to reconstruct the data. Also, having orthonormal basis is advantageous in many applications especially in image denoising as there is a need to revert back from the transform domain and project the data in the original space (*i.e.* spatial domain). In case of orthogonal basis $\mathbf{V}$, $\mathbf{V}^{-1} = \mathbf{V}^T$, hence computation of the inverse of the basis matrix is no longer required. Orthogonal variant of LPP and its application to image denoising task is discussed in this chapter.

## 4.1 Orthogonal Locality Preserving Projection

Orthogonal Locality Preserving Projection (OLPP) [32] was introduced to orthogonalize the basis of LPP. The objective function ensuring the basis to be orthogonal thus becomes,

$$\underset{\mathbf{V}}{arg\,min} \mathbf{V}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{V} \tag{4.1}$$

subject to the constraints

$$\mathbf{V}^T\mathbf{X}\mathbf{M}\mathbf{X}^T\mathbf{V} = \mathbf{I}; \mathbf{v_m}^T\mathbf{v_k} = 0, \forall m = \{1, 2, ......, k-1\} \tag{4.2}$$

here, weight matrix $\mathbf{S}$, Laplacian matrix $\mathbf{L}$ and $\mathbf{M}$ are found in the same manner as that in LPP [26]. Weights in $\mathbf{S}$ are assigned using the heat kernel function as

$S_{ij} = e^{-\frac{\left\|\mathbf{x_i}-\mathbf{x_j}\right\|^2}{t}}$ if $\mathbf{x_i}$ belongs to $k$ nearest neighbors ($k$-NN) of $\mathbf{x_j}$.

The basis vectors $\mathbf{v_k}, \forall k$ are found in the iterative manner ensuring that the vector learnt is orthogonal to all the previously found basis vectors. $\mathbf{v_1}$ is the eigenvector associated with the smallest non-zero eigenvalue of $\mathbf{Z}^{(1)} = \left(\mathbf{XMX}^T\right)^{-1}\mathbf{XLX}^T$. Rest of the basis vectors, $\mathbf{v_k}, \forall k$ are computed as follows:

$$\mathbf{Z}^{(k)} = \left\{\mathbf{I} - \left(\mathbf{XMX}^T\right)^{-1}\mathbf{V}^{(k-1)}\left[\mathbf{U}^{(k-1)}\right]^{-1}\left[\mathbf{V}^{(k-1)}\right]^T\right\}\mathbf{Z}^{(1)} \tag{4.3}$$

$\mathbf{v_k}$ is the eigenvector associated with the smallest eigenvalue of $\mathbf{Z}^{(k)}$.

$$\mathbf{V}^{(k-1)} = [\mathbf{v_1}, \ldots, \mathbf{v_{k-1}}]\,;\ \mathbf{U}^{(k-1)} = \left[\mathbf{V}^{(k-1)}\right]^T\left(\mathbf{XMX}^T\right)^{-1}\mathbf{V}^{(k-1)} \tag{4.4}$$

The basis matrix $\mathbf{V}$ thus found is orthonormal.

### 4.1.1 Variants of OLPP

The fundamental requirement for the basis vectors to be orthogonal is $\mathbf{V^TV} = I$. Hence, in order to have orthogonal basis vectors, the constraint $\mathbf{V^TV} = I$ should be satisfied. A new orthogonalization of LPP (NOLPP) is proposed, where we introduce the orthogonalization constraint while minimizing the objective function of conventional LPP as opposed to the normalization constraint $\mathbf{V^TXMX^TV} = 1$. The minimization problem now reduces to the eigenvalue solution of $\mathbf{XLX^TV} = \lambda\mathbf{V}$. The solution is no more the generalized eigenvalue problem and produces orthonormal basis vectors directly in the first step, hence is computationally efficient.

Selection of $t$ in the weight matrix $\mathbf{S}$ according to the heat kernel function plays a major role while obtaining the basis $\mathbf{V}$ which makes it a very crucial and difficult task as it highly depends on the data [47, 46]. Various extensions have been proposed in the literature that try to find optimal dictionary vectors [47, 46, 48, 45]. Weighing as suggested in Section 2.2.1 using the z-shaped weighing function can be performed.

$$S_{ij} = \left\{\begin{array}{ll} 1; & \text{if } d_{ij} \leq a \\ 1 - 2\left(\frac{d_{ij}-a}{b-a}\right)^2; & \text{if } a \leq d_{ij} \leq \frac{a+b}{2} \\ 2\left(\frac{d_{ij}-b}{b-a}\right)^2; & \text{if } \frac{a+b}{2} \leq d_{ij} \leq b \\ 0; \text{otherwise} & \end{array}\right\} \tag{4.5}$$

here, $a$ and $b$ specify range along which the function changes its values and can be controlled. $d_{ij}$ represents the distance between two data points. OLPP basis learned using the z-shaped weighing function are used to perform all the experiments reported in this thesis.

## 4.2 Image Denoising : A New Application

LPP and its variants have been mostly used for object recognition and dimensionality reduction related applications [26, 32, 9] so far. A new application of LPP has been explored for image denoising. The overview of image denoising approaches is discussed first before going into the details of the proposed OLPP based image denoising.

### 4.2.1 Overview of Image Denoising

Noise in images gets produced during the acquisition process by the sensors, the circuitry of a scanner or a digital camera and is unavoidable. It is not part of the object or scene being imaged. To get the original clean image, denoising has to be performed. Hence, denoising becomes an integral part of image generation process if clean image is desired. Most algorithms that convert image sensor data to a digital image, need to incorporate noise removal/reduction processes. Thus, it can be regarded as an important task in the computational imaging community. Gaussian noise is one of the most common form of noise which arises during image acquisition because of poor illumination or high temperature. Other noises such as salt-and-pepper noise, photographic grain noise, shot noise, quantization noise, anisotropic noise etc. are caused during the formation of digital image [85].

Redundancy present in natural images has been explored in many image processing applications, be it various image restoration techniques or dimensionality reduction techniques. High correlation among the neighboring pixels of the image is one of the first properties that was used for image denoising tasks with the spatial domain filters. These techniques used pixel based local correlations. Concept of “non-local self similarity” showed structural similarities between fixed sized patches from different spatial locations of the image [41, 42]. To model the redundancies of the image, linear transformations such as Fourier Transform, Discrete Cosine Transform, Wavelet transform, Block DCT [35], having well-known universal basis are used.

In recent times, there has been great advancement in the performance of image restoration techniques - especially denoising. The state of the art techniques rely on two major hypotheses that hold true for natural images: (1) image patches from distant locations are often structurally similar (“non-local self-similarity”) [42, 41], and (2) image patches can be accurately expressed as a sparse linear combinations of using the linear combinations of basis vectors. Collection of these basis vectors of same or different linear transforms is known as ’dictionary’. This “dictionary” could be a well-known universal basis such as Fourier, DCT, or wavelet (or over-complete extensions of the same) [35, 39], or it could be trained offline on a set of representative images [36]. However, some of the best methods in today's literature actually infer the dictionaries *in situ*, directly from the patches of the noisy or blurry image. Some techniques such as K-Singular Value Decomposition (KSVD) ('K' denotes number of clusters) [44] and its variants [86] learn a single over-complete dictionary for the entire image. Techniques such as clustering based sparse representations (CSR) [37] perform clustering in the space of patches followed by the inference of a single dictionary per cluster. On the other hand, some techniques use principal components analysis (PCA) [87, 34] to learn spatially varying orthonormal set of basis - one per patch in the noisy image, constructed from group of patches that are similar to it. Independent Component Analysis (ICA) [43] is also used to adaptively learn the dictionaries from the input noisy images.

The sparse codes (the coefficients of the linear combination of dictionary vectors) are inferred in an alternating fashion with the dictionary, in methods such as [44]. In case of orthonormal bases (either the universal bases or those learned *in situ*), the sparse codes of the noisy patches are manipulated using hard or soft thresholding [35] or Wiener filtering [35, 34, 87] to obtain the restored image. In recent literature, there is much interest in imposing dependencies between the sparse codes corresponding to different patches, either based on spatial proximity or some notion of structural similarity. For instance, the well-known BM3D technique creates a 3D stack of similar patches, projects it onto a 3D basis (tensor product of 2D-DCT and 1D-Haar). Collaborative filtering is performed using hard thresholding on the transformed coefficients followed by basis inversion, thereby allowing for a coupled update of the coefficients [35].

Local Pixel Grouping based denoising using PCA (LPG-PCA) [34], [87] learn a set of orthonormal PCA basis for each patch of the image from group of patches similar to it. CSR [37] performs clustering operation on the noisy patches, learns separate PCA bases for each cluster, and then minimizes a criterion for the proximity of the coefficients of patches belonging to any given cluster. The Bayesian dictionary learning framework developed in [88] imposes the prior belief that (spatially) nearby patches should be expressible as sparse linear combinations of *similar* subsets of columns from a (possibly over-complete) dictionary. In [36, 89], structurally similar patches are grouped together into a matrix, and restoration is performed by penalizing a matrix norm (such as the trace-norm) subject to a hard constraint derived from the likelihood. K-SVD [44] and its variants learn an over-complete dictionary for the entire noisy image and sparse linear combinations of the dictionary vectors are selected for each patch. Global image denoising [90], learns the dictionary from the informative parts of the image in such a way that each pixel is estimated from all the pixels from the image. Noise elimination is performed by means of shrinkage of the filter eigenvectors. Expected Patch Log Likelihood (EPLL) [91] is maximized to find a reconstructed image in which every patch is likely under the prior while keeping the image still close to corrupted image. EPLL uses GMM for learning the dictionary. GMM is used for learning the dictionary and noisy coefficients are denoised by structured sparse estimation [92].

Another set of image denoising methods, that have evolved recently are based on deep learning and deep neural networks. The noisy image is given as input and the network is trained to predict the clean or original output. Denoising auto encoders and Convolutional Neural Networks (CNNs) are being widely used to address the image denoising task. Multiple (deep) layers of Denoising Auto-encoder (DA) are employed to build a deep neural network for image denoising and inpainting tasks [93]. A new training scheme that trains the DA to reconstruct the clean image from corresponding noisy observation is proposed. For every subsequent layer, hidden layer activations from the previous layer of both the noisy and clean input are calculated to be applied as the training. Xu et al. [94] perform the task of image restoration using deep CNNs by establishing the connection between traditional optimization-based schemes and neural network architecture. The proposed image de-convolution using CNNs is based on separable kernels. Enhanced restoration performances are obtained by initializing the network with separable kernel inversion and outlier rejection. Deep learning based image denoising scheme proposed in [95], uses the de-convolutional layers to obtain the clean images from the noisy ones in addition to the convolutional layers. The corresponding convolutional and de-convolutional layers are connected directly, skipping the intermediate layers, which help in retaining the image details. The convolutional layers act as the feature extractor, which capture the abstraction of image contents while eliminating noise. De-convolutional layers are then used to recover the image details. Though, the deep learning based approaches are becoming popular recently, the focus of this work is to explore the applicability of domain transformation techniques for image denoising, and hence, the proposed denoising framework has not been compared with these techniques.

In the domain transformation based image denoising approaches discussed so far, a dictionary is obtained for each patch or a cluster of similar patches after grouping together structurally similar patches. Also, in most of the non-local self similarity based approaches, a smaller search neighborhood is defined for grouping the similar patches together or maximum number of patches to be grouped together are fixed. An OLPP based denoising scheme is proposed in this chapter that uses a *global dictionary* from all overlapping patches extracted from the entire image. The proposed denoising scheme using OLPP and its suitability for image denoising are discussed later in this section.

### 4.2.2 State of the art denoising approaches

Recent state of the art techniques rely on two major hypotheses that hold true for natural images: (1) image patches from distant locations are often structurally similar ("non-local self-similarity") [42, 41], and (2) image patches can be accurately expressed as a sparse linear combinations of using the linear combinations of basis vectors. Some image denoising techniques working on this principle are discussed in this section.

**Local Pixel grouping - Principle Component Analysis (LPG-PCA) [34, 87]**

Local Pixel Grouping (LPG) refers to finding similar patches to the reference patch. For each reference patch from the noisy input image, PCA basis are learnt from group of patches similar to it. As PCA works on vector data, all the patches are first converted to vector format and then processed. Linear minimum mean square estimation is used to update the patch coefficients in the PCA domain and inverse transformation is used to map the patch back in spatial domain. LPG-PCA is a two stage procedure, denoising image after the first stage is again denoised following the same procedure.

**Block Matching 3D (BM3D) [35]**

Like LPG-PCA, BM3D is also a two stage procedure. In the first stage, reference patch and its similar patches are grouped together and arranged in a 3D stack. The stack is projected onto 3D transform basis typically 3D DCT, or tensor product of 2D DCT and 1D Haar wavelets. The 3D transform coefficients are manipulated usually by hard thresholding. All the patches in the entire stack are reconstructed using an inverse 3D transform. This is repeated for every patch in the image and multiple answers appearing at any pixel are averaged. In the second stage, the output image of the first step is used to compute patch similarities which is more robust than computing the similarities in the noisy image. Patches from the first-stage denoised image are then appropriately assembled into a stack. Corresponding patches from the noisy image are assembled into a second stack. 3D transform coefficients of the second stack are denoised using Wiener filter update rule and inverse 3D transform is used to get back in the original domain. Patch aggregation is performed to obtain the final output image.

**Clustering based Sparse Representation (CSR) [37]**

Instead of grouping together similar patches for each patch from the noisy image, CSR performs clustering operation on all the patches extracted from the noisy image. A separate PCA basis is learned from each cluster, all the patches of the cluster are projected in the PCA domain and a criterion for the proximity of the coefficients of patches belonging to any given cluster is minimized to eliminate noise. Observation of these coefficients suggests that they are not randomly distributed and their location uncertainty is related to non-local self similarity of the image, which implies the possibility of achieving higher sparsity by exploiting such location related constraint. Inverse transform followed by aggregation is performed to obtained the denoised image.

**Expected Patch Log Likelihood (EPLL) [91]**

Expected Patch Log Likelihood (EPLL) is maximized to find a reconstructed image in which every patch is likely under the prior while keeping the image still close to corrupted image. A finite gaussian mixture model is used for learning the dictionary and noisy coefficients are denoised by structured sparse estimation. Dictionary is learned from a large image data base as opposed to many of the patch based methods where dictionary is learnt from the corrupted image itself. Patches of the noisy input image are assumed to be sampled from mixture of finite gaussians. Given a noisy patch, mixing weights for each of the component are calculated and the component having highest mixing weight is selected. Thus, notion of sparsity automatically comes as only one of many components of gaussian mixture is active for a patch. Optimization based on MAP rule is carried out to eliminate noise.

A summery of these image denoising state of the art approaches, respective domain transformation methods and coefficient update rules used in the transformed domain to eliminate noise is reported in Table 4.1.

### 4.2.3 OLPP for Image Denoising

In all the methods that work on the principle of non-local similarity for image restoration tasks [42, 87, 34, 35, 37, 36, 89], all possible patches of fixed size are extracted from the input image from which similar patches of a given reference patch are to be grouped together. Thereafter, an orthonormal basis is inferred for each patch and shrinkage is performed on the coefficients obtained when the patch is projected on that basis. One of the major motivating factors for using OLPP is its property of inherently preserving neighborhood information. Here, by neighborhood, we explic-

Table 4.1: State of the art approaches for image denoising, respective domain transformation methods and coefficient update rules used in the transformed domain to eliminate noise.

| **Image denoising approach** | **Domain transformation technique** | **Coefficient update rule** |
|---|---|---|
| **LPG-PCA** | PCA | Linear minimum mean square error estimation |
| **BM3D** | 2D-DCT & 1D Haar | Hard thresholding & Wiener filter update |
| **EPLL** | GMM | Optimization based on MAP rule |
| **CSR** | PCA | Proximity of coefficients of the patches belonging to same cluster is minimized |

itly refer to patch similarity. Due to this locality preserving property, similar patches are automatically taken into consideration while formation of basis (in the form of weights in matrix $\mathbf{S}$ as explained in Section chap4:variantsOLPP). Hence, a global or semi-global basis is learned for a whole image or a large window at once and the same basis is used for all the overlapping patches present. This allows us to employ one single orthonormal basis for an entire sub-image as opposed to spatially varying orthonormal bases [87], [34]. Most importantly, since an entire set of patches from a large sub-image is represented by a single orthonormal basis, the OLPP method is truly non-local in nature. Coefficients of all the patches in the OLPP domain are obtained by projecting the patches on the OLPP basis. Noise elimination is performed by manipulating the coefficients of the patches in the OLPP domain. In the transform domain, higher magnitude coefficients are the ones that carry information about the true clean image. Hence, by suppressing the lower magnitude coefficients, noise elimination can be performed and the original image can be obtained from the noisy one.

As the basis obtained using OLPP are orthogonal, standard shrinkage rules for orthogonal transforms can be applied. The basic noisy image formation model is $\mathbf{I}_\eta = \mathbf{I} + \eta$ where $\mathbf{I}$ is the original image, $\eta$ is i.i.d. Gaussian noise with standard deviation $\sigma$ and $\mathbf{I}_\eta$ is the noisy image. The model in transformed domain using basis $\mathbf{V}$ can be expressed as, $\mathbf{I}_{\eta\mathbf{V}} = \mathbf{I}_\mathbf{V} + \eta_\mathbf{V}$. Let $\{\mathbf{x}_i\}_{i=1}^N$ be a collection of $N$ patches of size $l \times l$ from $\mathbf{I}_\eta$. The coefficients of the $i^{th}$ patch in the transformed domain are represented as $x_{iv_k} = \mathbf{v_k^T x_i}$. The shrinkage rules on the coefficients are defined as follows:

a. Hard Thresholding: $\tilde{x}_{iv_k} = x_{iv_k} \circ (|x_{iv_k}| \geq \sigma\sqrt{2\log l^2})$

b. Soft Thresholding : $\tilde{x}_{iv_k} = \text{sign}(x_{iv_k})\text{max}(0, |x_{iv_k}| - \sigma\sqrt{2\log l^2})$

c. Wiener Update (squared L2 norm): $\tilde{x}_{iv_k} = \frac{\sigma_{v_k}^2}{\sigma^2+\sigma_{v_k}^2} x_{iv_k}$ where $\sigma_{v_k}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_{iv_k})^2$

The filtered coefficients are transformed back to the spatial domain and the patches are aggregated to get back the clean image. Comparison of the existing approaches with the proposed approach for denoising is shown in Figure 4.1 to give clear insight of the advantage of the proposal.

### 4.2.4 Suitability of OLPP for Image denoising

The proposed image denoising procedure along with the initial set of results geared towards gaining a better understanding of the performance of OLPP and its suitability for image denoising are presented in this section. In all experiments using OLPP, we divide the noisy image into overlapping patches of size 6 × 6. Also, Wiener filter update rule is used to filter the coefficients in the OLPP domain as it produces best results.

**OLPP: Granularity**

Given any noisy reference patch $\mathbf{x}_r$, we can collect together patches $\{\mathbf{x}_i\}_{i=1}^{P}$ that are structurally similar to it, where the similarity is quantified based on knowledge of the standard deviation of noise $\sigma$. Basically, any patch $\mathbf{x}_i$ for which $\|\mathbf{x}_r - \mathbf{x}_i\|^2 \leq 2.7\sigma l^2$ (here, patches are of size $l \times l$), is considered similar to $\mathbf{x}_r$ [96]. These similar patches are assigned weight according to the function in Equation 4.5 while constructing the OLPP basis. The parameter "$b$" of Equation 4.5 for constructing the similarity matrix $\mathbf{S}$ automatically takes care of the similar patches, hence reducing the task of explicitly finding the similar patches for each patch while constructing the global orthonormal basis. This makes the present proposal computationally efficient without compromising the performance of denoising.

Instead of working on each patch and repeating the procedure for all the patches, we divide the image into large-sized windows (*e.g.* an image of size 512 × 512 is divided into windows of size 128 × 128) and learn a single OLPP basis using all the patches within this window. Such a window-based approach saves computational cost considerably, and has been employed in mixture-model learning for image restoration, in prior work such as [92]. After computation of the orthogonal basis, the patch coefficients are computed, and they are modified using the Wiener update rule. The patches are then reconstructed from the modified coefficients. Multiple candidate

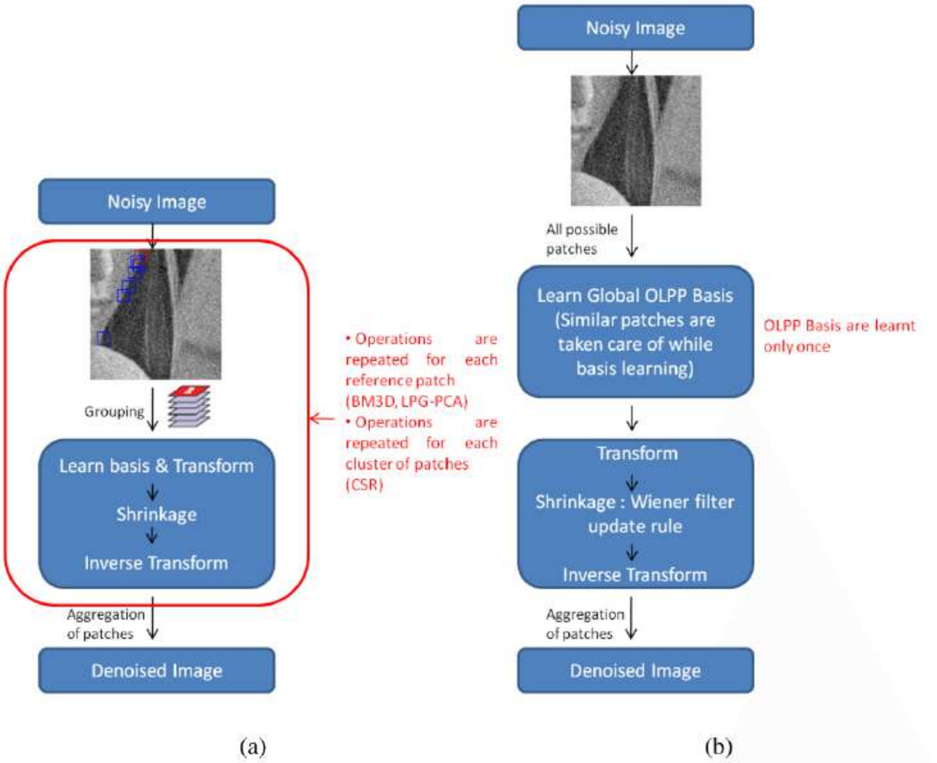

Noisy Image
Grouping
Learn basis & Transform
Shrinkage
Inverse Transform
Aggregation of patches
Denoised Image
• Operations are repeated for each reference patch (BM3D, LPG-PCA)
• Operations are repeated for each cluster of patches (CSR)
(a)
Noisy Image
All possible patches
Learn Global OLPP Basis (Similar patches are taken care of while basis learning)
OLPP Basis are learnt only once
Transform
Shrinkage : Wiener filter update rule
Inverse Transform
Aggregation of patches
Denoised Image
(b)

Figure 4.1: Comparison of existing denoising approaches, such as LPG-PCA, BM3D, CSR vs. proposed orthogonal locality preserving projection (OLPP) based denoising approach. (a) Similar patches of a given reference patch from the image are grouped together, a set of orthonormal basis is inferred for each patch from the grouped patches, shrinkage is performed on the coefficients of the patch in respective projection domain and the modified patches are transformed back to the spatial domain. This process is repeated for every patch (BM3D, LPG-PCA) or a cluster of patches (CSR) and aggregation is performed to obtain the denoised image. (b) A global OLPP basis that takes care of the similarity between the patches is learnt for the entire image or a large portion of the image, shrinkage in the form of Wiener filter update rule is applied on the patches projected in the OLPP domain followed by inverse transform to project the patches back in the spatial domain. Patches are aggregated to acquire denoised image.

values for a single pixel location (since overlapping patches were being used) are averaged to produce a final restored image.

The computation time reduced to ∼5 minutes using window based global approach versus ∼3 hours using the conventional approach of learning basis for each patch of an image of size 512 × 512 on machines of same configuration.

### OLPP, PCA, DCT and Random Orthogonal Basis

We performed an experiment to measure the locality preserving nature of the learnt OLPP bases in comparison to PCA, DCT and a randomly chosen orthogonal basis. The locality preserving property of an orthogonal basis $\mathbf{V} \in \mathbb{O}(l^2, \mathbb{R})$ is quantified by the value of $E(\mathbf{v}_k) = \frac{\mathbf{v}_k^T \mathbf{XLX}^T \mathbf{v}_k}{\mathbf{v}_k^T \mathbf{XMX}^T \mathbf{v}_k}$ (*i.e.* the generalized eigenvalue in $\mathbf{v}_k^T \mathbf{XLX}^T \mathbf{v}_k = \mathbf{v}_k^T \mathbf{XMX}^T \mathbf{v}_k$ of OLPP) for each $k \in \{1, ..., l^2\}$. These values, obtained from denoising experiments on a 128 × 128 texture image with a patch size of 6 × 6, are plotted (in descending order) in Figure 4.2 for all the aforementioned bases. As OLPP seeks to minimize the value of $E(\mathbf{v}_k)$, it can be seen that these values computed from the OLPP basis are smaller than those computed from other bases.

Another experiment is performed to evaluate the patch-wise denoising performance for OLPP and PCA basis. All 6 × 6 patches of 120 × 120 image are considered for comparison. Basis of OLPP and PCA are learnt from the clean (original) as well as noisy image with $\sigma = 20$ for natural and textured gray scale images. Here, the reconstruction error of denoised patch using respective approaches i.e. OLPP and PCA (one basis for the entire image i.e. global) is computed with respect to the clean patch. Table 4.2 contains the number of patches having higher reconstruction error after denoising using PCA based denoising as compared to OLPP. On an average, more than 65% of the patches have higher reconstruction error using global PCA basis, which shows higher reconstruction and hence restoration (here, denoising) accuracy of OLPP.

In the next experiment, PCA and OLPP coefficients of all the overlapping patches of size 5 × 5 from portions of size 128 × 128 of ‘Barbara’ image are arranged in image form in Figure 4.3. Here, each image represents one coefficient of all the patches i.e. first image of Figure 4.3 shows first coefficient value of all the patches and so on. For PCA, the basis were arranged in increasing order of the eigenvalues whereas while computing OLPP basis, no such arrangement of basis was performed which leads to non-uniform arrangement of the coefficients. As we are not truncating any of the coefficients, this arrangement of coefficients does not play any role in denoising experiments and their performance. We are only showing the basis for visual inspection if possible. Also note that the PCA basis found here are local

Table 4.2: Patch-wise denoising performance is evaluated by calculating the reconstruction error with reference to corresponding clean patch. Percentage (%) of patches having higher reconstruction error using PCA based denoising approach than the OLPP based approach for $120 \times 120$ image. Basis are learnt from clean (original) and noisy image using both the approaches.

| Image | # Patches | |
|---|---|---|
| | Clean | Noisy |
| Lena | 78.51 | 74.9 |
| Airplane | 77.05 | 80.43 |
| Brick Wall | 62.07 | 62.32 |

whereas those using OLPP are global.

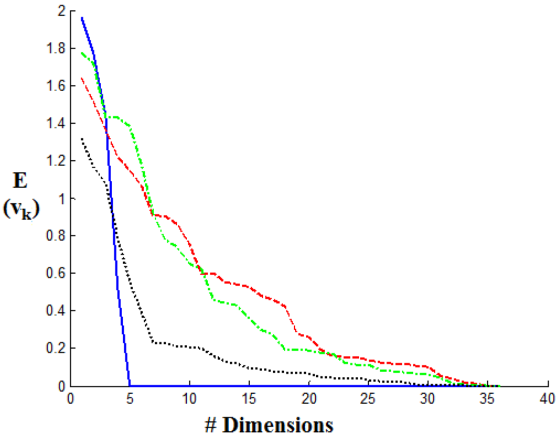


Figure 4.2: Plot of the values of $E(\mathbf{v}_k)$ for OLPP (solid), PCA (dashed), DCT (dash dotted) and random orthogonal (dotted) bases respectively

### Comparison with New OLPP Basis

A new orthogonalization of Locality Preserving Projection (NOLPP) is introduced by relaxing the normalization constraint which makes procedure of obtaining orthogonal LPP basis computationally more efficient. However, we realized that the constraint that we relaxed is very stringent, relaxing which affects the basis computation. Though, in some cases, improvement in the performance over other state-of-the-art approaches is achieved using this approach, conventional OLPP with the normalization constraint performs consistently better specially in case of natural images. Comparison of denoising results using NOLPP and OLPP on two natural images

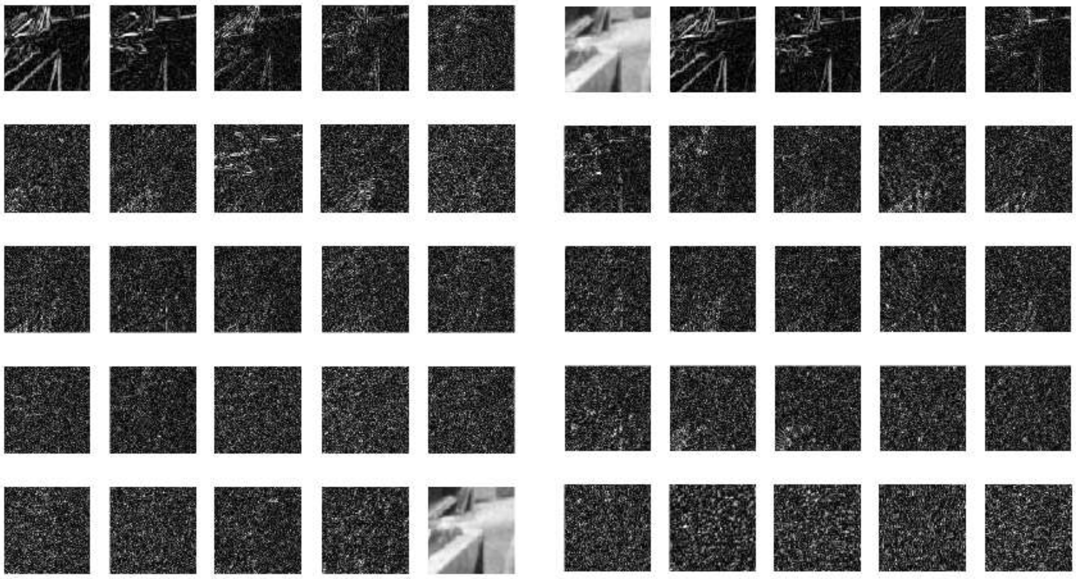


Figure 4.3: Coefficient-wise plot for all the overlapping patches of a $128 \times 128$ portion of barbara image using OLPP (Left) and PCA (Right) basis

is shown in Figure 4.4. It can be observed that the current proposal using OLPP performs better than the NOLPP bases approach visually as well as quantitatively.

### 4.2.5 Time Complexity Analysis

The cost, as far as time complexity is concerned, appears to be much less in case of OLPP as compared to state of the art methods. Assuming total number of overlapping patches in the image is $N$, average number of patches similar to the reference patch is by $P$, average time required to search similar patches for each reference patch is $T_s$, time taken to generate the Laplacian matrix $L$ for $N$ patches is $NT_L$ and size of each patch is $l \times l$. In approaches that require the patch (in matrix format) to be transformed into vector format, size of each representative vector turns out to be $l^2 \times 1$.

In PCA based approaches [34, 87], for each reference patch, similar patches are grouped together and covariance matrix of dimensions $l^2 \times l^2$ is computed, eigenvectors of which form the transformation matrix. Thus, the time complexity becomes $\mathcal{O}(N[T_s + Pl^4 + l^6])$. Note that computation of eigenvectors of an $l \times l$ matrix takes $\mathcal{O}(l^3)$ time. BM3D [35] stacks similar patches of the reference patch and performs 2D and 1D transforms on patches of that stack. The 2D transforms lead to $\mathcal{O}(Pl^3)$ complexity whereas for 1D it is $\mathcal{O}(P^2l^2)$. As the procedure is repeated for all overlapping

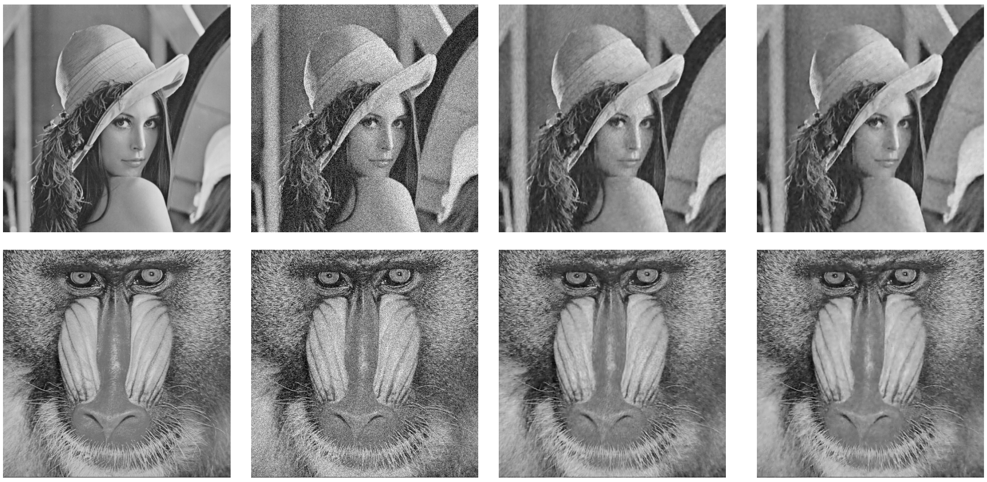


Figure 4.4: From left to right, top to bottom (first row): Original 'Lena' image, noisy image with white Gaussian noise ($\sigma = 30$), results (PSNR, SSIM) of denoising using NOLPP (29.91, 0.78) and OLPP (30.38, 0.81). (second row) Original 'Mandrill' image, noisy image with white Gaussian noise ($\sigma = 30$), results (PSNR, SSIM) of denoising using NOLPP (24.61, 0.71) and OLPP (24.64, 0.72).

patches, overall time complexity turns out to be $\mathcal{O}(N[T_s + Pl^3 + P^2l^2])$.

In case of OLPP where patches are considered in vector format, $\mathcal{O}(NT_L)$ time is required to compute the Laplacian matrix and $\mathcal{O}(l^6)$ for the transformation matrix i.e. basis matrix, as the global basis is computed only once for the whole image. Thus, overall time complexity is $\mathcal{O}(NT_L + l^6)$. Thus, the proposed approach is faster than other state of the art image denoising approaches. Time complexities of the approaches discussed above are summarized in Table 4.3.

Table 4.3: Summary of time complexities of denoising approaches. $N$: Number of patches, $l \times l$: size of a patch, $T_s$: average time required to search similar patches of the reference patch, $P$: average number of similar patches, $NT_L$: time taken to generate the Laplacian matrix $\mathbf{L}$ for $N$ patches.

| Technique | Time Complexity |
|---|---|
| LPG-PCA | $\mathcal{O}(N[T_s + Pl^4 + l^6])$ |
| BM3D | $\mathcal{O}(N[T_s + Pl^3 + P^2l^2])$ |
| OLPP | $\mathcal{O}(NT_L + l^6)$ |

### 4.2.6 Noise level estimation

In this thesis, we are denoising images corrupted with i.i.d. Gaussian noise with zero mean and $\sigma$ standard deviation i.e. $\mathcal{N}(0, \sigma)$. All the approaches that are being compared assume that standard deviation of noise ($\sigma$) i.e. the noise level, is known. In practical scenarios, value of $\sigma$ might not be available, hence methods for estimating $\sigma$ can be used prior to denoising. Donoho and Johnstone [33] derived value of $\sigma$ from the finest scale empirical wavelet coefficient of the given input data. Highest subband of a Daubechies 2 wavelet transform was used in [97] to estimate noise variance. This approach is also used in [98] and [34] to estimate the noise level. Another set of approaches use homogeneous regions of image to estimate noise variance/standard deviation [99]. Methods proposed in [100] used local statistics of image to estimate noise level. Noise level estimation based on principal component analysis has been recently proposed in [101]. Several other approaches have also been proposed in literature to estimate noise level from given noisy image.

### 4.2.7 Evaluation Measures

After denoising the noisy image, apart from visual comparison, in order to compare the results of different approaches as well as to check the quality of denoised image, evaluation measures are required. Various evaluation measures are present in literature that are used to examine the quality of the denoised image. Here are the measure for performance evaluation.

#### PSNR: Peak Signal to Noise Ratio

Peak signal-to-noise ratio (PSNR), is the ratio between the maximum possible power of a signal and the power of corrupting noise represented in decibels (db). It is most commonly used quality measure between the original image and the reconstructed denoised image. The higher the value of PSNR, the better the quality of denoised image.

Let $I_\eta$ be the original image of size $M \times N$ and $J$ be is its noise-free approximation (i.e. the denoised image).

$$MeanSquareError(MSE) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\Big(I_\eta(i,j) - \hat{J}(i,j)\Big)$$

$$PSNR = 10log_{10}\left(\frac{L^2}{MSE}\right) \tag{4.6}$$

where $L$ is maximum intensity level present in the image $I_\eta$ and $MSE$ is same as defined above.

### SSIM: Structural Similarity Index Measure

Structural Similarity Index Measure (SSIM) [102] is another widely used image quality measure given the original clean image. Natural images have high dependencies between the neighboring pixels which carry important information about the structure of the objects in the image in visual sense. This measure compares the changes in the structural information in local patterns of pixel intensities.

$$SSIM\_local(x,y) = \frac{(2\mu_x\mu_y + \epsilon_1)(2\sigma_{xy} + \epsilon_2)}{(\mu_x^2 + \mu_y^2 + \epsilon_1)(\sigma_x^2 + \sigma_y^2 + \epsilon_2)}$$

$$SSIM = \frac{1}{M}\sum_{j=1}^{M} SSIM\_local(x_j, y_j) \tag{4.7}$$

where $\epsilon_1$, $\epsilon_2$ ensure stability when either $(\mu_x^2 + \mu_y^2)$ or $(\sigma_x^2 + \sigma_y^2)$ us close to zero. The SSIM_local is defined over a local window centred at $(x, y)$ and average over such windows gives a single measure SSIM for entire image.

## 4.3 Experiments

Image denoising experiments on natural and textured gray-scale as well as color image databases are performed. Gray scale image denoising is performed on the Lansel database of natural images (airplane, barbara, boats, couple, elaine, fingerprint, goldhill, house, man, mandrill, peppers, stream, zelda) and UIUC texture database [103] (100 texture images with 17 different textures). Color image databases i.e. Kodak database [1] and Brodatz color texture database [2] are used.

We have compared the performance of OLPP against that of popular state of the art approaches for which code was publicly available - BM3D (stage 1 and stage 2) [35], expected patch log-likelihood model (EPLL) [91], LPG-PCA (stage 1 and stage 2) [87], and the clustering based sparse representation (CSR) from [37]. For each competing method, we use author-specified parameter settings for the sake of fairness.

[1] Kodak Image Database: `http://r0k.us/graphics/kodak/`

[2] Color Brodatz Texture database: `http://multibandtexture.recherche.usherbrooke.ca/colored_brodatz_more.html`

### 4.3.1 Texture Preservation Property

We believe that the dependencies imposed by the OLPP method enhance the preservation of finer textural features during denoising, in comparison with approaches like PCA which do not impose such a dependency. To provide evidence for this, we performed extensive experiments on a database of 100 texture images from UIUC. These texture images consisted of 17 different texture categories such as bark, water, carpet, wood, *etc.* The average PSNR and SSIM values for some categories are reported in Figures 4.7 and 4.8 for OLPP and all competing methods, for 3 different noise levels, *i.e.* $\sigma \in \{20, 30, 40\}$. The results are shown in graphical format for better visual comparison. For each method, the resultant value is shown using different symbol. Note that the results are not continuous, just for better visual comparison, results of all the methods are joined by a line for each noise level. As the methods LPG-PCA and BM3D work in two stages, results for both stages are reported. In most of the cases, the OLPP method was able to perform better in terms of PSNR, SSIM as well as subjective visual quality with better preservation of textural details at higher noise levels. Both LPG-PCA and BM3D tend to smooth out the finer textures. In fact, BM3D uses a tensor product of DCT and Haar wavelet bases, the latter of which is akin to performing diffusion filtering which can erase subtle textures [104]. Though EPLL performed very well in most cases, it also produced some undesired spiky artifacts as can be observed clearly. Iterative behavior of both EPLL and CSR methods increase their time complexity in comparison to the proposed approach. Though, the proposed approach is not outperforming CSR, the results are comparable with much less computational and time complexities. Examples of reconstruction for texture databases are shown in Figure 4.5 and 4.6. Sharp edges and minute textural details can be observed by zooming in the images. Average PSNR and SSIM values for the UIUC texture databases are reported in Table 4.4.

Table 4.4: Average denoising results for the entire UIUC Texture Database containing 100 images ($\sigma \in \{20, 30, 40\}$)

| Denoising Approach | $\sigma = 20$ | | $\sigma = 30$ | | $\sigma = 40$ | |
|---|---|---|---|---|---|---|
| | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| LPG-PCA | 28.25 | 0.807 | 26.24 | 0.722 | 24.96 | 0.654 |
| BM3D | 28.58 | 0.819 | 26.74 | 0.747 | 25.49 | 0.686 |
| EPLL | 28.46 | 0.824 | 26.58 | 0.746 | 25.37 | 0.669 |
| CSR | 28.54 | 0.817 | 26.75 | 0.748 | **25.57** | 0.683 |
| OLPP | **28.67** | **0.829** | **26.79** | **0.757** | 25.56 | **0.696** |

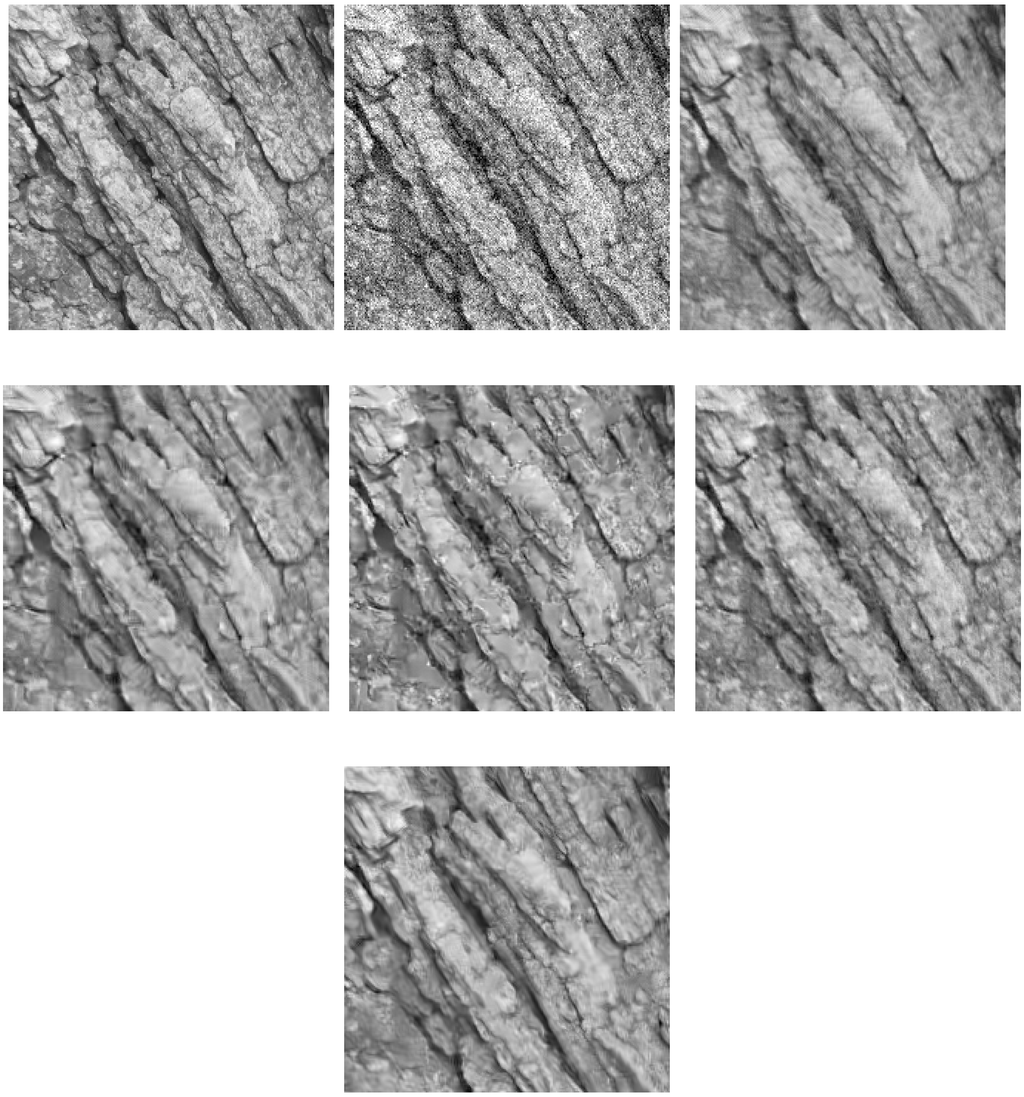


Figure 4.5: From left to right, top to bottom: Original texture image ('Bark'), noisy image with white Gaussian noise ($\sigma = 30$), Results (PSNR, SSIM) of denoising using LPG-PCA (23.48, 0.700), BM3D (23.69, 0.71), EPLL (18.19, 0.32), OLPP (23.98, 0.746), CSR (23.7, 0.71).

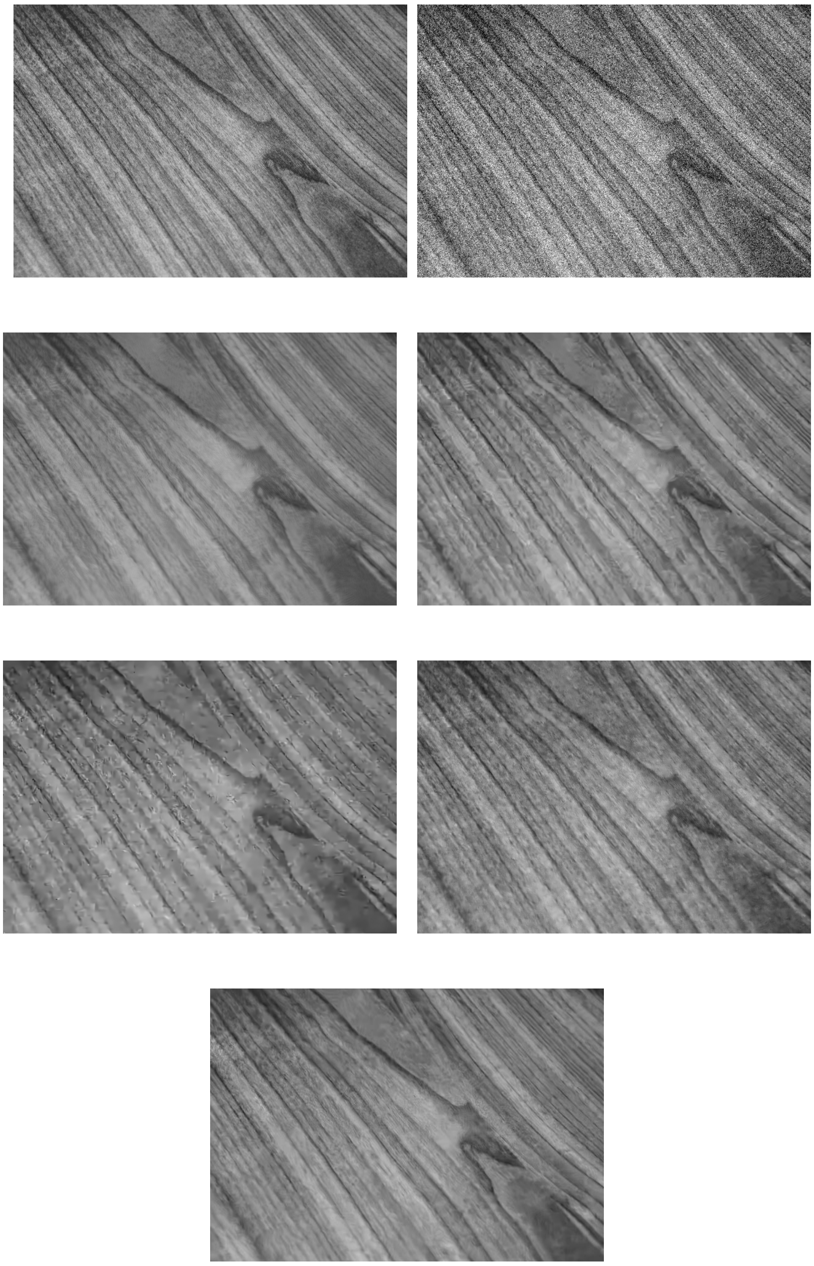
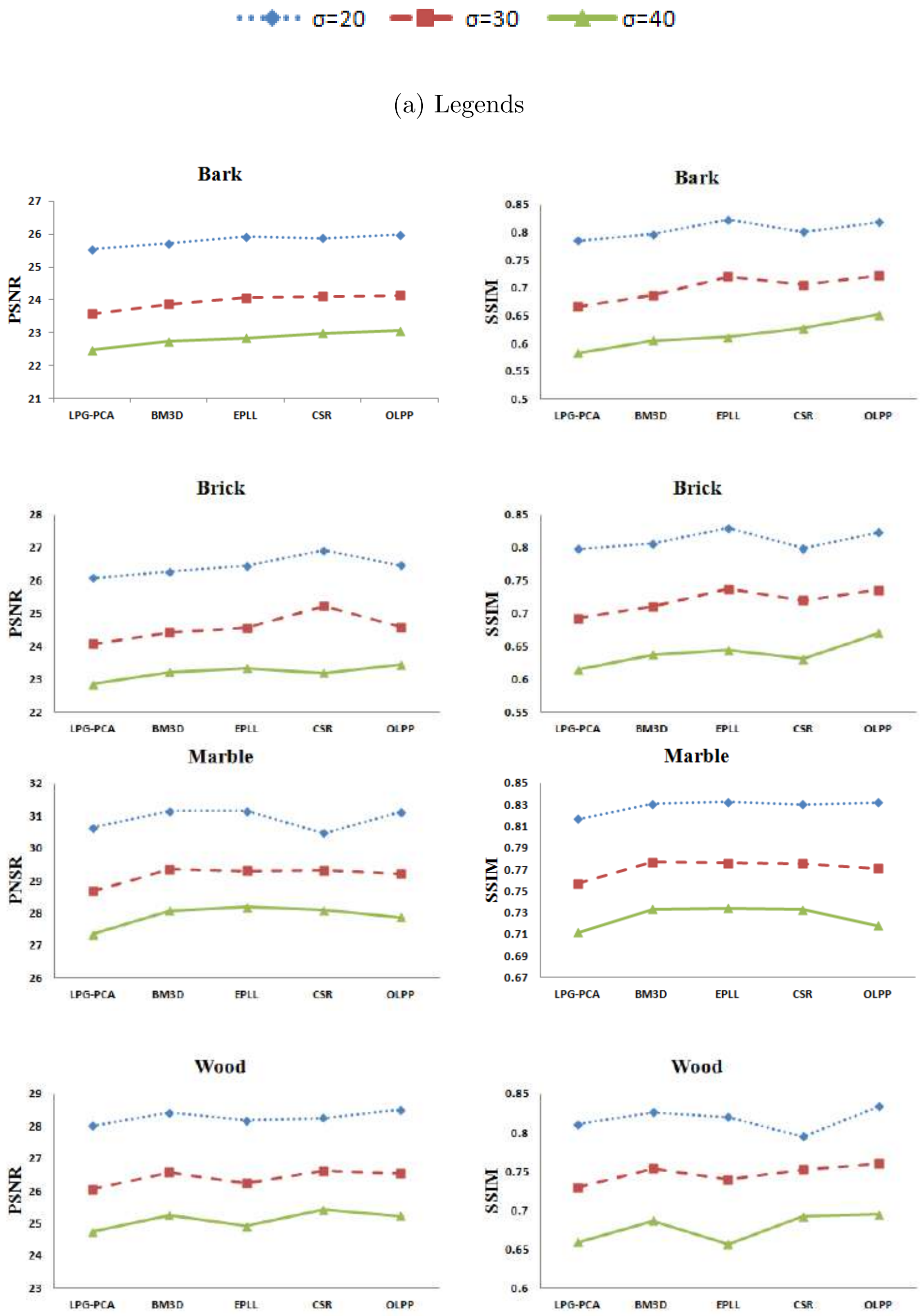
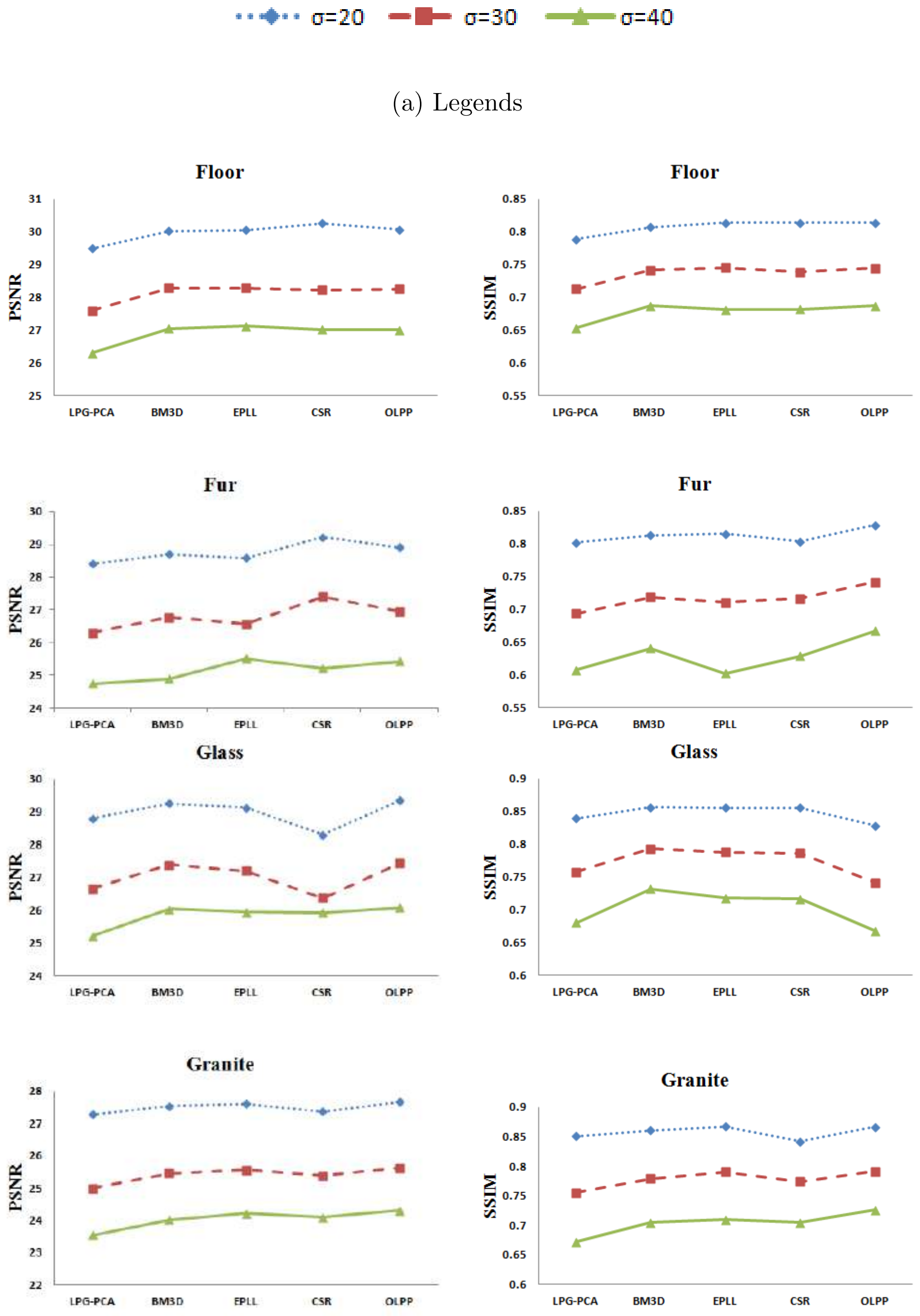


Figure 4.6: Denoising experiment on a texture image ('Wood'). From left to right, top to bottom: Original image, noisy image ($\sigma = 40$), results (PSNR, SSIM) of denoising using LPG-PCA (24.73, 0.659), BM3D (25.26, 0.686), EPLL (24.94, 0.656), OLPP (25.23, 0.695) and CSR (25.41, 0.691) based denoising.



(a) Legends

Figure 4.7: Denoising results for some images from UIUC Texture Database. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated in Figure 4.7(a)



(a) Legends

Figure 4.8: Denoising results for some images from UIUC Texture Database. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated in Figure 4.8(a)

### 4.3.2 Results on Natural Images (Grayscale)

Denoising experiments are also performed on the 'Lansel database' of natural images. Results of denoising a natural image 'Stream' are shown in Figure 4.9 for noise level $\sigma = 40$. The average results for the whole database are reported in Table 4.5. Denoising results on some natural images are reported in graphical format for three noise levels $\sigma = 20, 30, 40$ in Figure 4.10 and 4.11 using LPG-PCA, BM3D, EPLL, CSR and OLPP based denoising methods. The proposed method preserves the finer details of the images and produces PSNR/SSIM values comparable to the state of the art, but it has a tendency to slightly under-smooth the homogeneous regions of the image.

Table 4.5: Average denoising results for all the images from Lansel Database ($\sigma \in \{20, 30, 40\}$)

| Denoising Approach | $\sigma = 20$ | | $\sigma = 30$ | | $\sigma = 40$ | |
|---|---|---|---|---|---|---|
| | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| LPG-PCA | 30.42 | 0.834 | 28.48 | 0.78 | 27.08 | 0.73 |
| BM3D | **30.78** | 0.84 | **29.03** | **0.796** | 27.66 | 0.754 |
| EPLL | 30.46 | 0.839 | 28.6 | 0.788 | 27.48 | 0.747 |
| CSR | 30.69 | **0.841** | 28.9 | 0.793 | **27.76** | **0.755** |
| OLPP | 30.45 | 0.838 | 28.52 | 0.787 | 27.2 | 0.743 |

### 4.3.3 Results on Color Images

Extensive experiments on two color image databases are reported - (1) a portion of the Brodatz color texture database (first 28 images), and (2) all 24 images of the Kodak image database. For the OLPP method, we used patches of size $6 \times 6 \times 3$ represented as vectors of length 108. This allows for easy coupled updates of the RGB values at a pixel. The competing methods were BM3D (which uses a decorrelated color space - YCbCr) and LPG-PCA (which denoises the R, G, B channels independently). We did not perform comparisons with EPLL and CSR for which no color denoising code is available. Besides producing better PSNR and SSIM values, one can again notice the superior texture preservation property of OLPP over BM3D. This can also be seen in a sample result for the Kodak database shown in Figure 4.12. Some of the results (PSNR, SSIM) for the Brodatz database are presented in Figures 4.14 with the average PSNR and SSIM values for all the images in Table 4.6.

Experiments were also performed on all 24 images of the Kodak database for OLPP, BM3D (stage 2) [35] and LPG-PCA (stage 2) [87]. We did not compare with EPLL (by Zoran and Weiss) [91] or CSR (by Dong et al ) [41] because the available

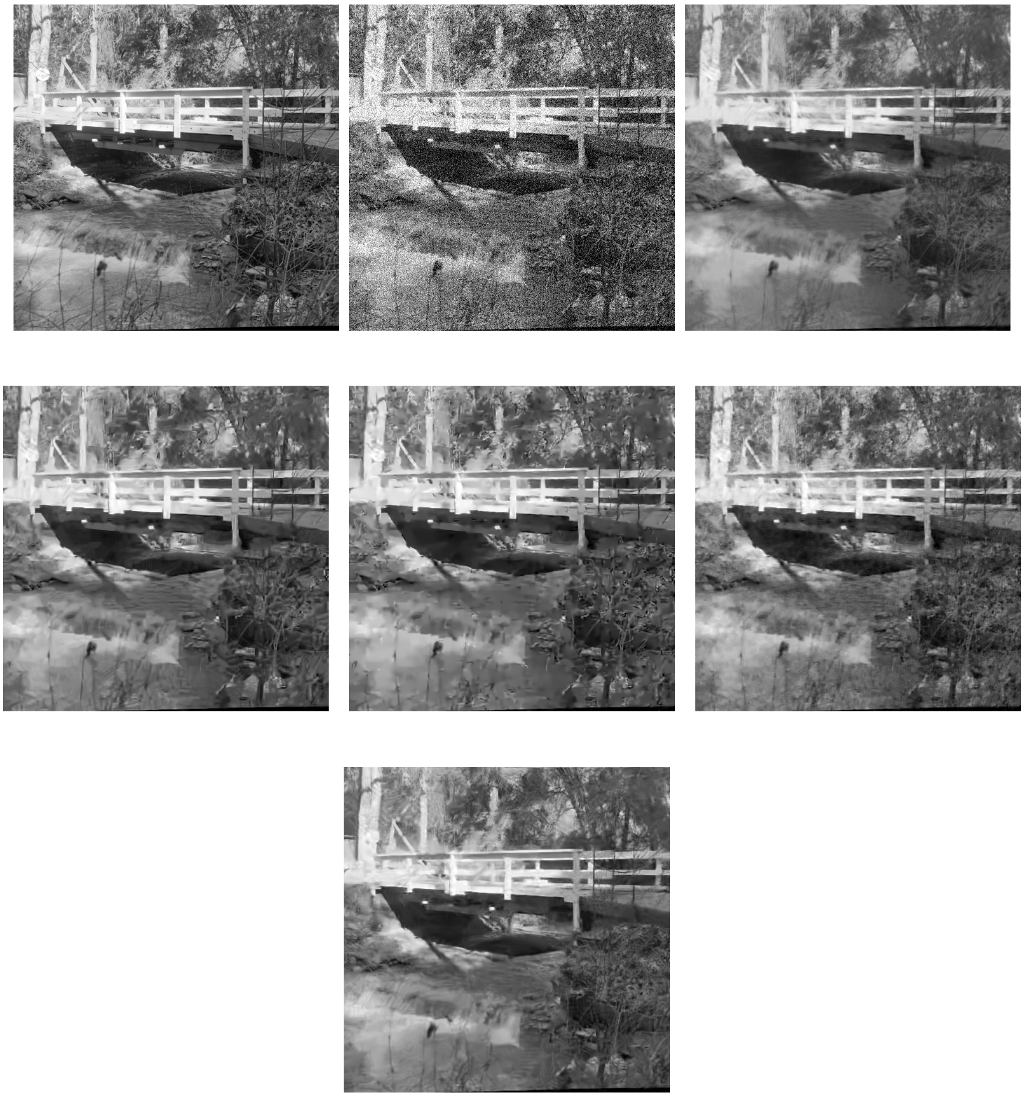
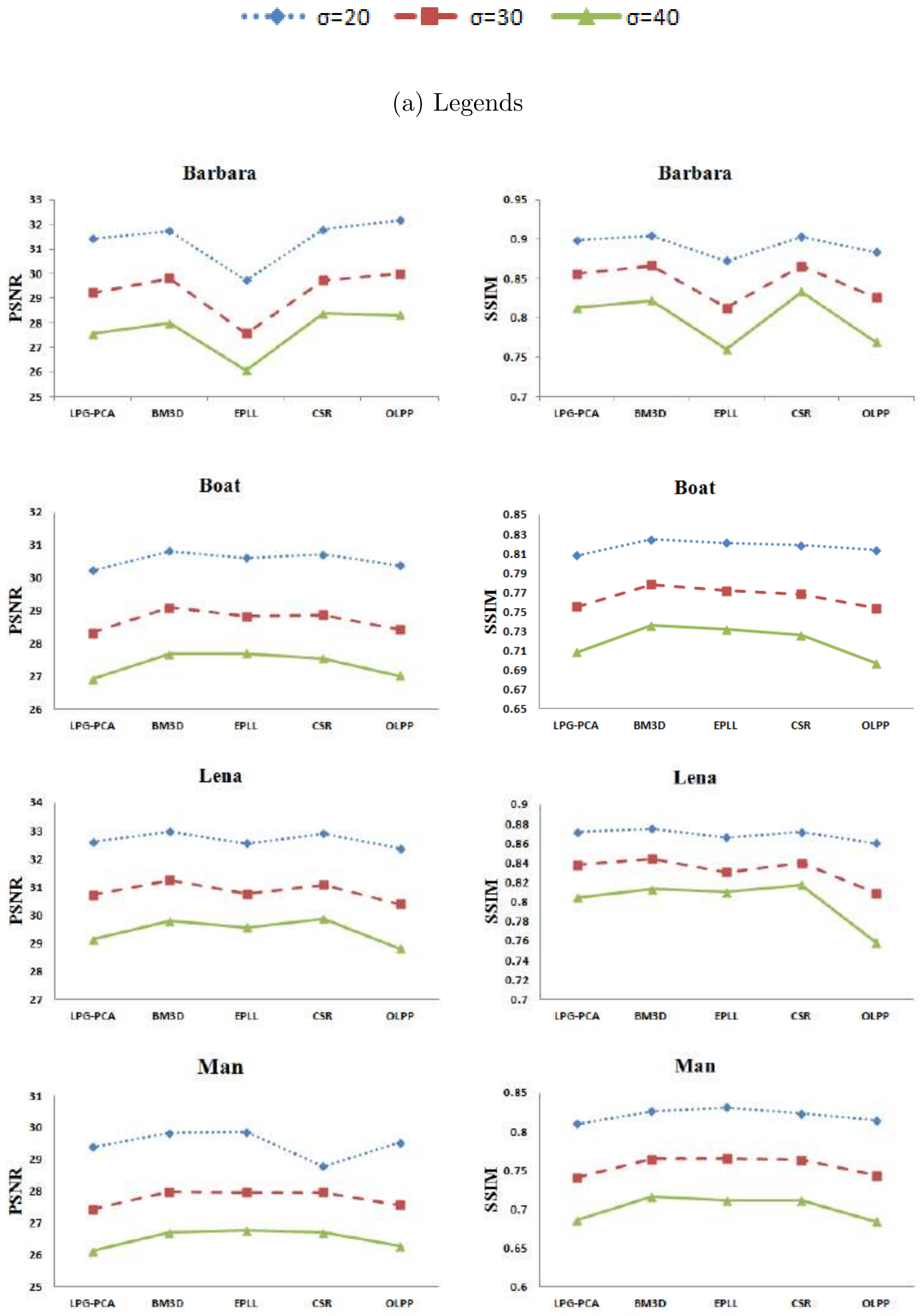
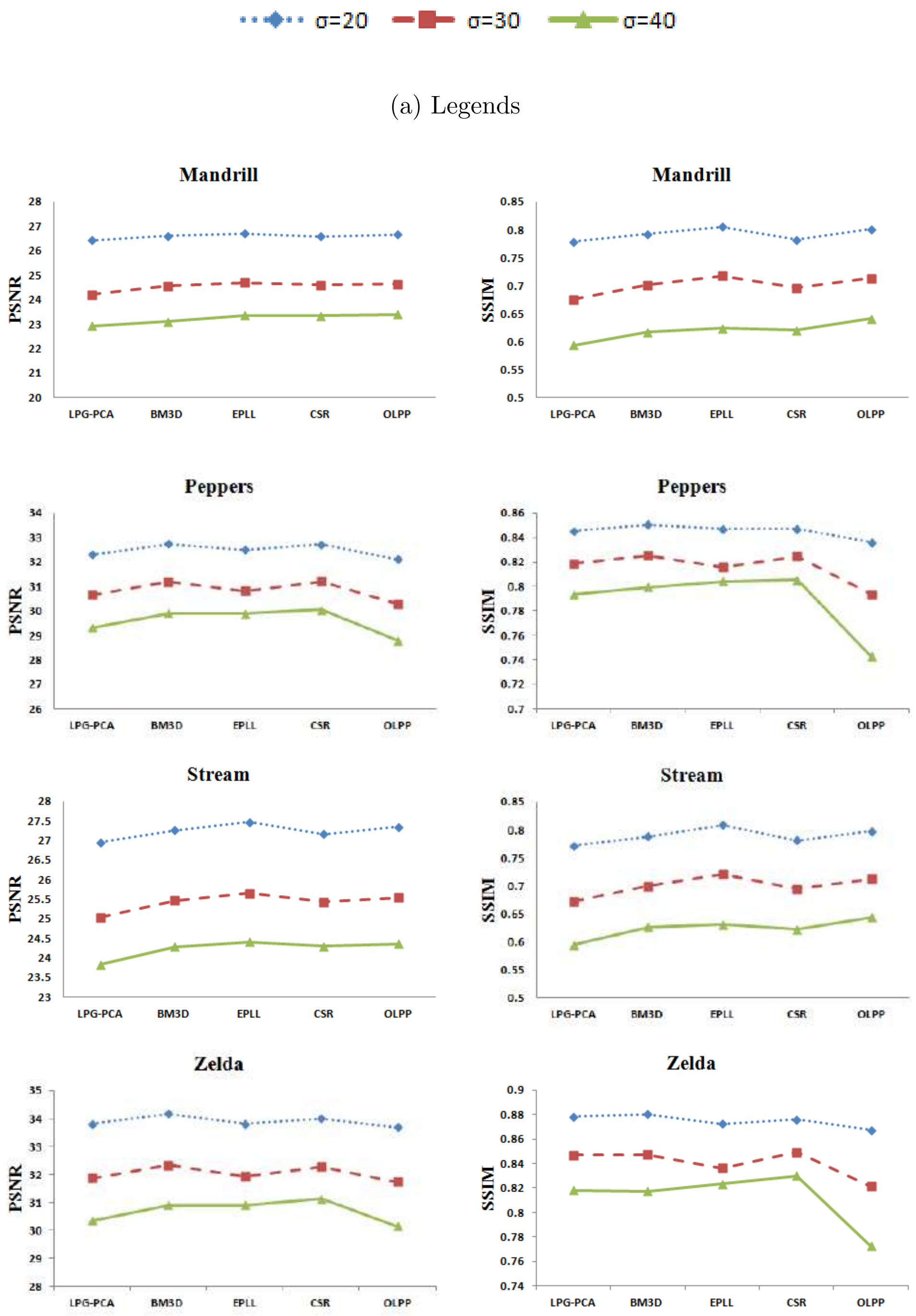


Figure 4.9: Denoising experiment on a natural image (‘Stream’). From left to right, top to bottom: Original image, noisy image ($\sigma = 40$), LPG-PCA (23.83, 0.5944), BM3D (24.29, 0.6260), EPLL (24.41, 0.6309), OLPP (24.36, 0.6438) and CSR (24.30, 0.6218).



(a) Legends

Figure 4.10: Denoising results for some images from Lansel database of gray-scale natural images. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated in Figure 4.10(a)



(a) Legends

Figure 4.11: Denoising results for some images from Lansel database of gray-scale natural images. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated by legends in Figure 4.11(a)

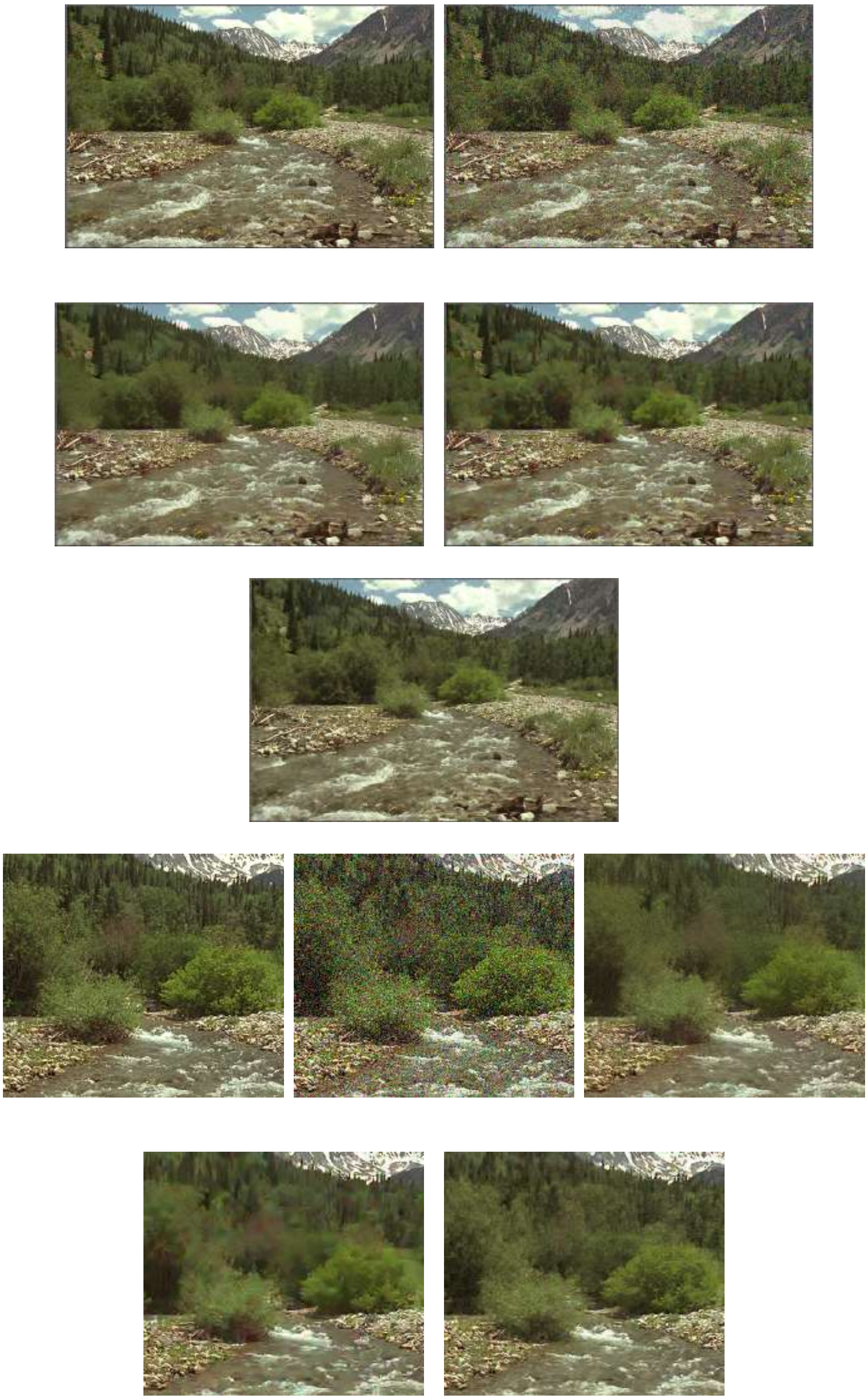


Figure 4.12: Denoising experiment on a color image from Kodak Database. Left to right, top to bottom: Original image, noisy image ($\sigma = 40$), denoising using LPG-PCA (PSNR: 23.97, SSIM: 0.6286), BM3D (24.94, 0.6833), and OLPP (24.90, 0.7159). Zoomed-in portions of the images (in same order).

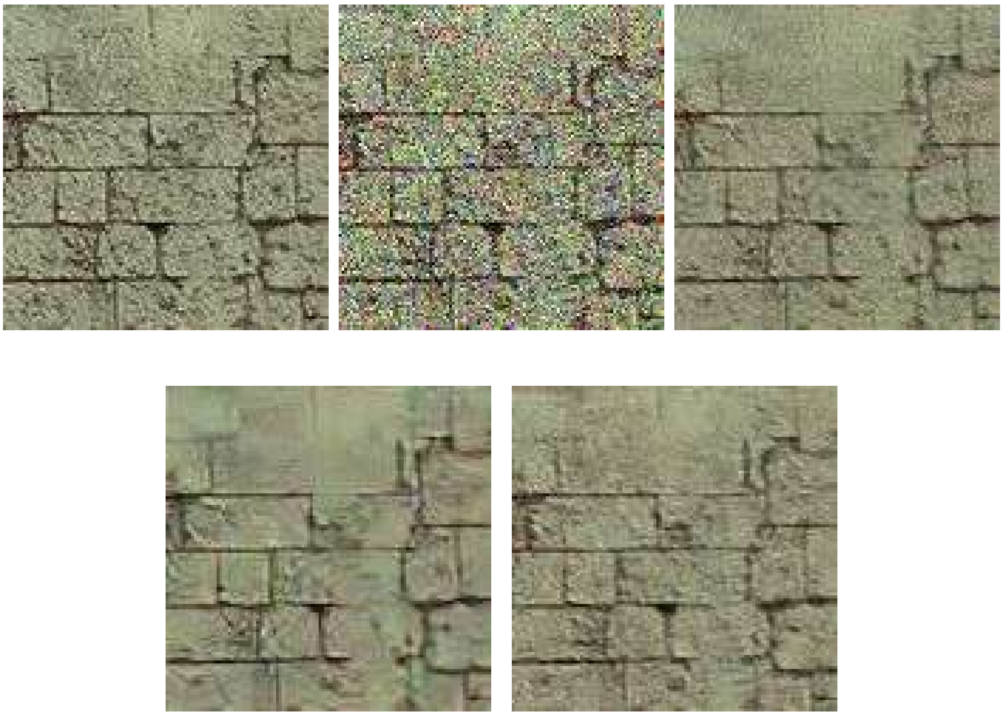
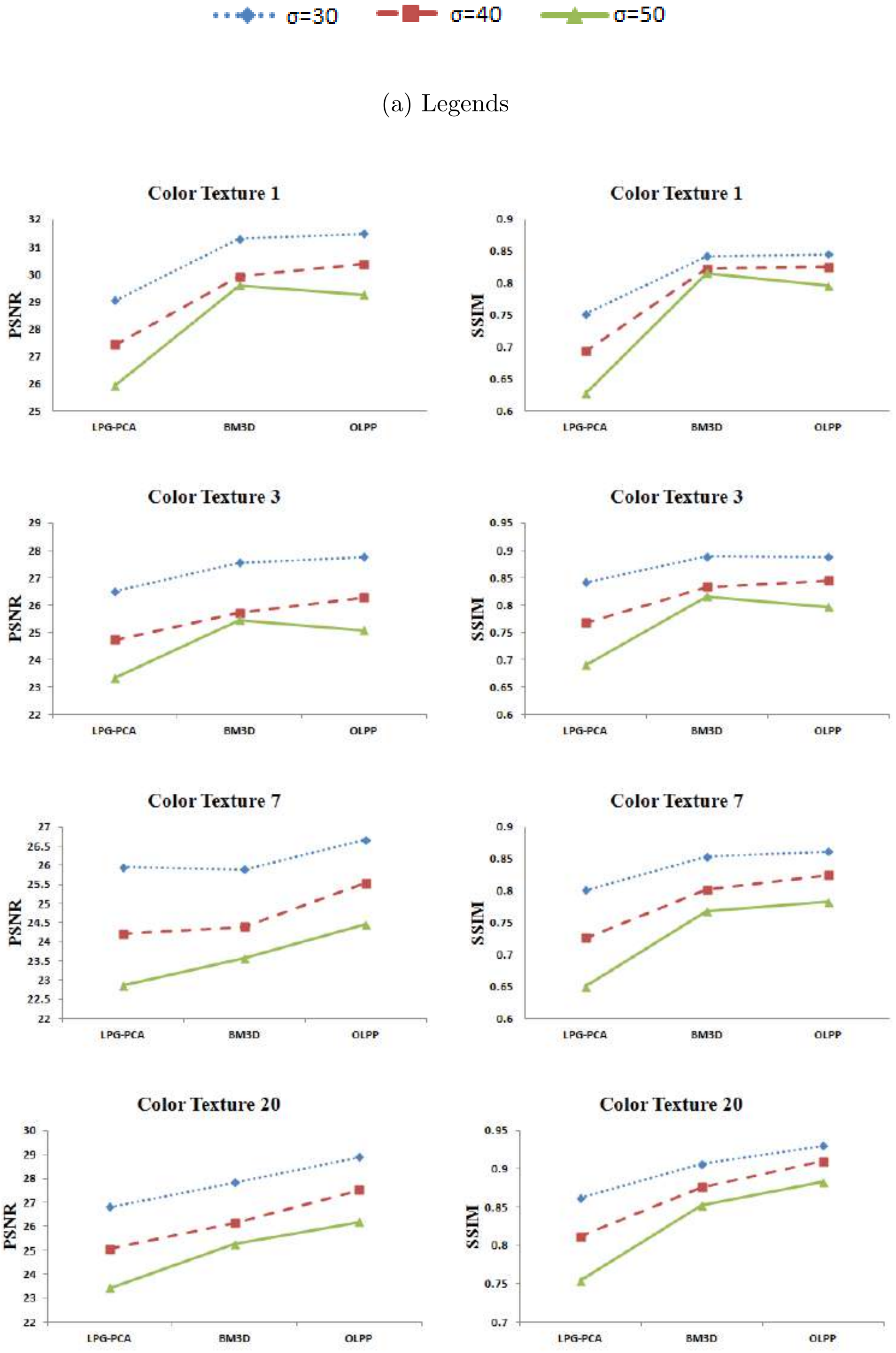


Figure 4.13: Denoising experiments for color texture. Left to right, top to bottom: original texture image, degraded image with white Gaussian noise ($\sigma = 30$), Results (PSNR, SSIM) of denoising using LPG-PCA (24.02, 0.648), BM3D (24.02, 0.627), OLPP (24.34, 0.677).



(a) Legends

Figure 4.14: Denoising results for some images from Bordatz Color Texture Database. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated in 4.14(a)

Table 4.6: Average denoising results on 28 images of Brodatz Color Texture Database ($\sigma \in \{30, 40, 50\}$)

| Noise level | LPG-PCA | | BM3D | | OLPP | |
|---|---|---|---|---|---|---|
| | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| $\sigma = 30$ | 26.62 | 0.816 | 27.28 | 0.866 | **27.82** | **0.876** |
| $\sigma = 40$ | 24.92 | 0.75 | 25.67 | 0.817 | **26.53** | **0.841** |
| $\sigma = 50$ | 23.52 | 0.68 | 25.18 | 0.795 | **25.4** | **0.802** |

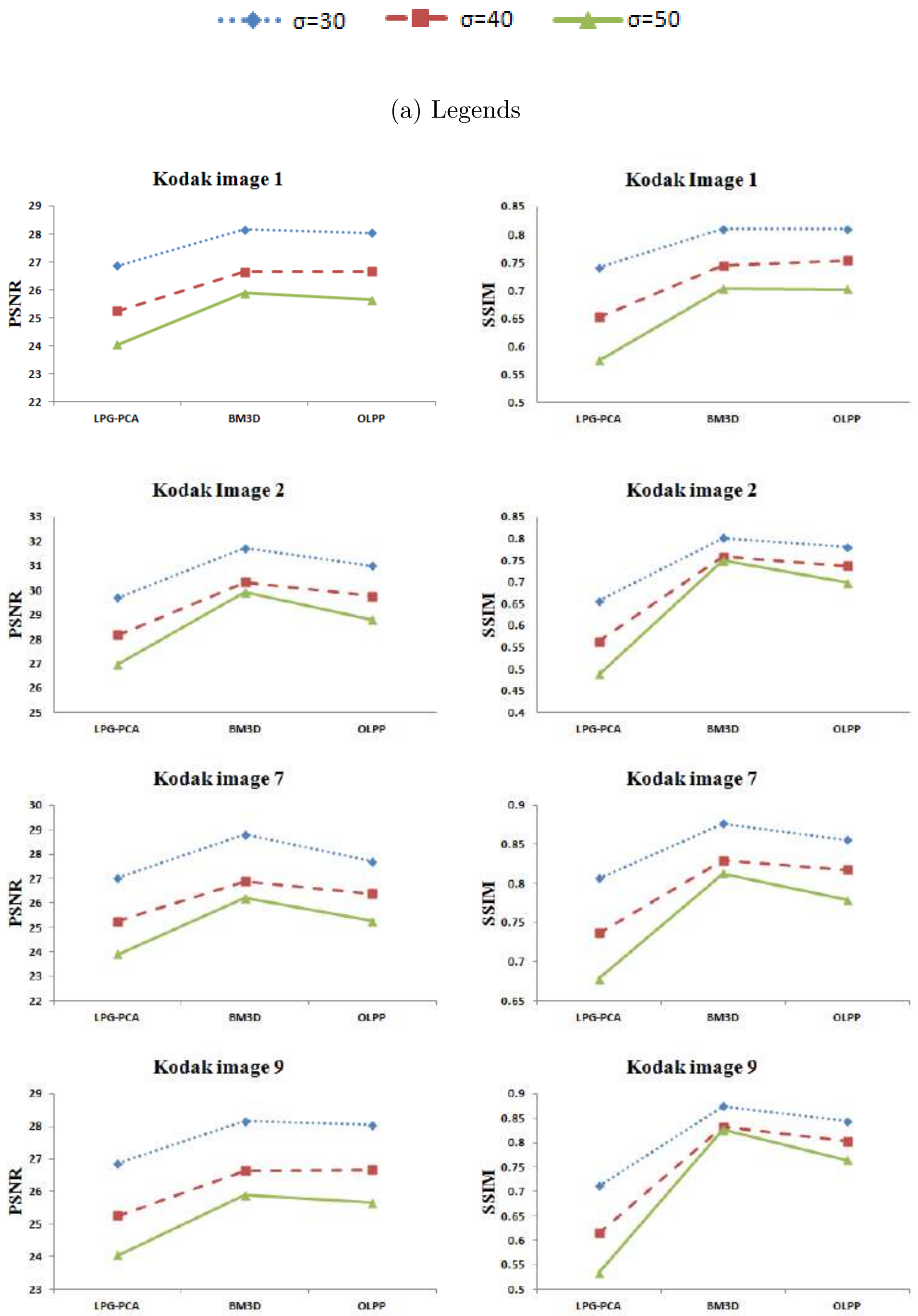
code for these two methods did not handle color images separately. Graphical results for image quality measures PSNR and SSIM [105] for some of the images from the database are shown in Figure 4.15 for three noise levels - $\sigma \in \{30; 40; 50\}$. For some images, the OLPP method did not compare favorably with BM3D in terms of PSNR (SSIM results were much better). In some homogeneous portions of the image, the performance of OLPP needs improvement (in a subjective sense). Average results using PSNR and SSIM measures for all the 24 images of the Kodak database are reported in Table 4.7.

Table 4.7: Average denoising results on all images from Kodak database ($\sigma \in \{30, 40, 50\}$)

| Noise level | LPG-PCA | | BM3D | | OLPP | |
|---|---|---|---|---|---|---|
| | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| $\sigma = 30$ | 28.32 | 0.72 | **30.2** | **0.839** | 29.51 | 0.825 |
| $\sigma = 40$ | 26.63 | 0.628 | **28.51** | **0.783** | 28.16 | 0.776 |
| $\sigma = 50$ | 25.36 | 0.556 | **27.9** | **0.761** | 27.1 | 0.732 |

## 4.4 Conclusion

In this chapter, we presented a new technique for image denoising which imposes the prior belief that similar patches from the noisy image should map onto similar transform coefficients, and explicitly learns an orthonormal basis to optimize for this criterion. The proposed technique is simple to understand and affords easy, efficient implementation. We demonstrate the excellent performance of this technique, especially with regard to texture preservation, on a large set of experiments across varied noise levels. The finer details present in the image are restored very efficiently. Also, in case of color images, superior texture preservation is achieved. This is so, even when we learn a single basis for large-sized windows from the image. This leads to an edge of the present method over other existing techniques where basis are learnt for each single patch.



(a) Legends

Figure 4.15: Denoising results for some images from Kodak database of color natural images. Horizontal axis represent various methods of denoising and vertical axis represents the PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated in Figure 4.15(a).

LPG-PCA groups together similar patches of the patch under consideration and learns PCA basis for all such overlapping patches in the image. It is a two stage procedure and in both the stages, similar computations are performed. Noise removal is carried out in the transformed domain. It is to be noted that the proposed technique outperforms both the stages of LPG-PCA in almost all the results reported even with the global OLPP basis which reduces the computational time to a large extent.

BM3D though does not learn the basis for each stack of similar patches as it uses 2D-DCT and 1D-Haar transforms, it tends to smooth out textural details due to the diffusion process in the Haar wavelet basis. Due to this property, BM3D looses some of the textural information which can be observed visually. Also, higher SSIM values are obtained for most of the texture images, both gray-scale and color, using current proposal.

EPLL is an iterative procedure of denoising, basis of which resemble PCA basis. The approach works well in the structural regions but produces spiky artifacts. Though the PSNR/SSIM values of the restored images are good, visual inspection clearly shows the undesired spikes. The proposed technique does not totally outperform the CSR approach, in most of the cases it is better than or comparable in terms of PSNR/SSIM values. Also, the iterative behavior of CSR makes it computationally more expensive as compared to one stage performance of the present OLPP approach.

The method has been tested on noise levels $\sigma$ ranging from 10 to 60. The strength of the current proposal is to suppress even the higher noise levels and that is why the same is reported. The proposal is working well on lower noise levels also. It is also noticed that there is considerable scope for improving the performance of this technique in the homogeneous regions of an image. The approach can also be adapted for various noise levels for single image as only the filtering process in the transformed domain is dependent on the noise level. So, using the same dictionary (global), the filtering of coefficients can be carried out depending upon the noise level of the particular part/portion i.e. signal dependent noise. This problem is not being addressed in this work, as our aim was to show the strength of OLPP as a denoising tool. A proper noise (signal dependent/independent) estimation technique can be added as a component. Besides working towards this, other directions for future work include combining the sparsity criterion with the basis learning and applications of this framework to other image processing tasks such as deblurring and inpainting.

# Chapter 5

# Two Dimensional Orthogonal Locality Preserving Projection

The dimensionality reduction approaches discussed so far such as PCA, LDA, LLE, ISOMAP, LPP as well as the ones proposed in the thesis i.e. ESLPP-MD, LPDP, OLPP and its extensions are vectorized in nature. Each data point in matrix format such as image, is required to be converted to the vector format before processing. Due to vectorization, spatial neighborhood information gets lost. The vectorization procedure leads to high dimensional data vectors resulting in high dimensional basis matrix and increase in computational time. Also, most of the linear dimensionality reduction approaches such as LDA, LPP, NPP often suffer from small sample size problem i.e. number of data samples is much less than the dimension of the data [50, 51, 9]. In such cases, the matrices become singular, hence cannot be processed [9]. To overcome the issue, PCA is applied on raw data to reduce the dimensions of the data after which the other dimensionality reduction procedure is applied. Two dimensional variants of PCA, LDA, LPP, NPP etc. have been proposed in literature [6, 9, 8] that directly process the data points in matrix format. These approaches are not only computationally more efficient but also enhance the performances of their one dimensional counterparts [6, 9, 8]. The two dimensional variants also overcome the problem of singularity implicitly, thus pre-processing of the data using PCA is no longer required. Brief discussion about some of the popular two dimensional dimensionality reduction approaches is carried out.

## 5.1 Overview of Two Dimensional (2D) Dimensionality Reduction Approaches

### 5.1.1 Two Dimensional PCA (2D-PCA) [6]

Conventional Principle Component Analysis (PCA) requires conversion of image data to the vector format leading to high dimensional image vector space which makes the covariance matrix computation difficult [6]. Alternate approach using SVD has been suggested that obtains the eigenvectors without actually generating the covariance matrix. It is reported in Yang et al. [6] that obtaining the basis vectors in this manner does not guarantee accuracy and may differ from the ones obtained from the covariance matrix. A two dimensional extension of PCA (2D-PCA) directly processes the images in 2D matrix format. The covariance matrix is thus formed from the original image matrices which is much smaller in size than the one formed using conventional PCA. Smaller covariance matrix results in faster computation of the basis matrix. The procedure for obtaining 2D-PCA basis is discussed below.

Let us consider we have $N$ data points $X_1, X_2, \cdots, X_N$, that are images of $m \times n$ dimensions. As per the principle of PCA, a set of basis $\mathbf{V}$ is to be obtained such that the scatter of the resulting projected images gets maximized. The eigen vectors of the image covariance matrix satisfy this criteria. Image covariance matrix $\mathbf{C}$ can be obtained as follows:

$$\mathbf{C} = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{X_i} - \bar{\mathbf{X}}\right)^T \left(\mathbf{X_i} - \bar{\mathbf{X}}\right) \tag{5.1}$$

here, $\mathbf{X_i}$ is the $i^{th}$ training sample and $\bar{\mathbf{X}}$ denotes average of all the training samples. Eigenvectors of $\mathbf{C}$ form the basis matrix $\mathbf{V}$ of 2D-PCA. Mathematical justification for the same can be traced from [6].

### 5.1.2 Two Dimensional LDA (2D-LDA) [7]

Linear Discriminant Analysis (LDA) has been widely used for various applications dealing with the high dimensional data. As discussed earlier, to take care of the small sample size problem, dimensionality reduction using PCA has to be performed before LDA. This results in higher time and space complexities. Two dimensional LDA (2D-LDA) [7] directly processes images in matrix format with the aim of maximizing between class scatter $\mathbf{S_B}$ and minimizing within class scatter $\mathbf{S_W}$. Projection of $i^{th}$ image $\mathbf{X_i}$ on the learned 2D-LDA basis $\mathbf{V}$, denoted by $\mathbf{Y_i}$, can be obtained as

$\mathbf{Y_i} = \mathbf{V}\mathbf{X_i}$. As this is a supervised approach, class labels of all the data points are known. Number of training samples is denoted by $N$, $\bar{\mathbf{X}}$ denotes mean of all the training samples, number of classes is denoted by $L$, number of samples $i^{th}$ class $L_i$ is $N_{L_i}$ and mean of each class is denoted by $\bar{\mathbf{X_i}}$. The 2D-LDA basis are obtained by solving the following generalized eigenvalue problem:

$$\mathbf{S_B}\mathbf{V} = \lambda\mathbf{S_W}\mathbf{V} \tag{5.2}$$

here,

$$\mathbf{S_B} = \sum_{i=1}^{L} N_{L_i} \left[\left(\bar{\mathbf{X_i}} - \mathbf{X}\right)\mathbf{V}\right]\left[\left(\bar{\mathbf{X_i}} - \mathbf{X}\right)\mathbf{V}\right]^T$$

$$\mathbf{S_W} = \sum_{i=1}^{L} \sum_{\mathbf{X_k} \in L_i} \left[\left(\mathbf{X_k} - \bar{\mathbf{X_i}}\right)\mathbf{V}\right]\left[\left(\mathbf{X_k} - \bar{\mathbf{X_i}}\right)\mathbf{V}\right]^T$$

The eigenvectors corresponding to the highest eigenvalues form the optimum basis.

### 5.1.3 Two Dimensional NPP (2D-NPP) [8]

Like PCA and LDA, a two dimensional extension of Neighborhood Preserving Projection (NPP) is also proposed on the lines of LLE [21] that each data points and its $k$ nearest neighbors lie on or close to a locally linear patch on a manifold [8]. Thus, NPP tries to reconstruct each data point in such a way that the reconstruction errors are minimized. $\mathbf{Y_i} = \mathbf{V}^T\mathbf{X_i}$ is the projection of $i^{th}$ image $\mathbf{X_i}$ on the learned 2D-NPP basis. The affinity matrix $\mathbf{W}$, each component $w_{ij}$ of which represents the optimal reconstruction weight between each data point $\mathbf{X_i}$ and its neighbor $\mathbf{X_j}$, is computed in the same manner as that of NPP as discussed in Chapter 1. The basis vectors of 2D-NPP are obtained by solving the following generalized eigenvalue solution:

$$\mathbf{X}\left[\left(\mathbf{I} - \mathbf{W}^T\right)\left(\mathbf{I} - \mathbf{W}\right) \otimes \mathbf{I_n}\right]\mathbf{X}^T\mathbf{V} = \lambda\mathbf{X}\mathbf{X}^T\mathbf{V} \tag{5.3}$$

here, $\otimes$ is the Kronecker product operator. As we are solving the problem of error minimization, the eigenvectors corresponding to the lowest non-zero eigenvalues are strongest.

### 5.1.4 Two Dimensional LPP (2D-LPP) [9]

Two Dimensional Locality Preserving Projection (2D-LPP) [9] works on the idea of preserving local information of the data. As in case of LPP, the data points close to each other in the original domain are intended to be projected close to each other in the projection domain thus keeping the proximity of the data intact. To process the images as it is in matrix format, arrangement of the data points i.e. images in the data matrix $\mathbf{X}$ is changed. All the training images are arranged in column $\mathbf{X} = \left[\mathbf{X_1}^T, \mathbf{X_2}^T, \cdots, \mathbf{X_N}^T\right]$. Weights in the similarity matrix $\mathbf{S}$ are assigned using the Heat-Kernel function and the Laplacian matrix is also obtained in the same manner as that of LPP (refer Chapter 1 for details). The basis matrix $\mathbf{V}$ is obtained by solving the following generalized eigenvalue problem:

$$\mathbf{X}^T \left(\mathbf{L} \otimes \mathbf{I_m}\right) \mathbf{X}\mathbf{V} = \lambda \mathbf{X}^T \left(\mathbf{M} \otimes \mathbf{I_m}\right) \mathbf{X}\mathbf{V} \tag{5.4}$$

It is to be noted that the basis matrix thus obtained in much smaller in size resulting in faster computation. Due to 2D data processing, the spatial neighborhood information of the image remains intact resulting in enhanced performance over LPP. As discussed for 2D-LDA and 2D-NPP, 2D-LPP also overcomes the problem of under-sampled data.

It has been reported that 2D-PCA, 2D-LDA, 2D-NPP and 2D-LPP all outperform their one dimensional counterparts. Apart from the approaches discussed here, some other two dimensional extensions of the dimensionality reduction techniques have also been proposed in literature [106, 50, 51, 107]. In this chapter, we formulate Two Dimensional Orthogonal Locality Preserving Projection (2D-OLPP). The approach directly processes images in two dimensional (2D) format i.e. matrix format, hence the overhead of transforming them in vectors gets reduced and the spatial neighborhood information remains intact. Due to 2D data processing, the basis matrix turns out to be much more compact than the one obtained by OLPP, reducing the time and space complexities of the algorithm considerably. Orthogonal nature makes the approach easier to be used in many applications that require orthogonal basis.

## 5.2 Two Dimensional Orthogonal Locality Preserving Projection

As discussed earlier, it is advantageous to process two dimensional data directly. Two dimensional extensions of PCA, LPP and NPP proposed in [6, 9, 8] respectively, directly process the data matrix and have shown improvement in the performance over their one dimensional (1D) counterparts. 2D-LPP/MSD [107], an extension of LPP motivated by the idea of maximum scatter difference, maximizes the difference of between-class scatter and within-class scatter. The approach is supervised in nature and the basis vectors obtained are also non-orthogonal, hence it is not suitable for image denoising. In case of 2D-LPP and 2D-NPP also, the basis found are not orthogonal. Orthogonal basis in both the cases can be directly obtained by ignoring the normalization constraint of the data [8]. However, the results show drastic decrease in the performance of these algorithms, which demonstrates a strong need for the normalization constraint ($\mathbf{V^T X^v M X^{v^T} V = 1}$), especially in case of two dimensional data processing [8]. Hence, it is important to retain this constraint while obtaining the orthogonal basis. With this aim in mind, two dimensional orthogonal locality preserving projection (2D-OLPP) is formulated and derived mathematically.

Let us consider we have $N$ data points $X_1, X_2, \cdots, X_N$, that are images of $m \times n$ dimensions sampled from an $mn$-dimensional space. In case of OLPP, two-dimensional data points are first converted to vector format to form a data matrix. The data matrix $X$ is formed by arranging the data points in columns i.e. $X^v = [X_1^v, X_2^v, \cdots, X_N^v]$ where $X_i^v$s represent vector forms of the corresponding 2-dimensional images $X_i$. The data matrix thus turns out to be of $mn \times N$ dimensions. Also, by converting the 2-dimensional images in vector form, the spatial neighborhood information is lost.

In this thesis, we are suggesting 2-dimensional OLPP which directly works on 2-dimensional images. The data matrix $X$ of dimensions ($mN \times n$) is formed by arranging all the image matrices in row format $X = [X_1; X_2; \cdots; X_N]$. As per the formulation of LPP, the aim is to preserve the local structure of the data. Hence, the objective function remains same as that of LPP and OLPP:

$$min \sum_{ij} (\mathbf{Y_i} - \mathbf{Y_j})^2 S_{ij} \tag{5.5}$$

Here, $\mathbf{Y_i}$ is the projection of $\mathbf{X_i}$ in the projection space using the transformation matrix $\mathbf{V} = [\mathbf{v_1}, \mathbf{v_2}, \cdots, \mathbf{v_k}]$ where $\mathbf{v_i}$s are orthogonal basis vectors. Given $\mathbf{V}$ and $\mathbf{X_i}$, $\mathbf{Y_i}$ is found as follows:

$$\mathbf{Y_i} = \mathbf{X_i V} \tag{5.6}$$

Here, $\mathbf{S}$ is a symmetric weight matrix that represents the weights assigned to the edges of the adjacency graph $G$. The two step procedure of constructing the weight matrix is as follows:

### Constructing the adjacency graph:

In the adjacency graph $G$ of $N$ nodes, all the images are represented by nodes i.e. $i_{th}$ node corresponds to the image $X_i$. Two nodes $X_i$ and $X_j$ are joined by an edge if they are neighbors. In general, the neighborhood is decided using $k$-nearest neighbor approach or the $\epsilon$ neighborhood. In both the cases, it is not clear how to select the values of $k$ or $\epsilon$. Also, a little or no attention in paid in the overlapping regions of the two or more classes. Many a times it happens that nearest neighbor of a data point is a data point belonging to the other class. In such cases, though the points belong to different classes, they could be connected because of their closeness. Hence, in order to have exploit natural grouping of the data, a semi-supervised approach is used to establish the neighborhood/adjacency. $k$-means clustering is performed on the data. Now, two data points $X_i$ and $X_j$ are considered neighbors if they belong to the same class according to the clustering performed earlier and are joined by an edge.

**Selection of weights:**

Here, the weight matrix $\mathbf{S}$, Laplacian matrix $\mathbf{L}$ and $\mathbf{M}$ are found in the same manner as that in ELPP [48]. Weights in the matrix $S$ are assigned in a monotonically decreasing z-shaped fashion. Based on the range of values given as input, weights are assigned to the distances over the complete scale as per Equation 5.7. As the distance between the data points increases, weight at that point decreases.

$$\mathbf{S}_{ij} = \left\{ \begin{array}{ll} 1; & \text{if } x \leq a \\ 1 - 2\left(\frac{x-a}{b-a}\right)^2; & \text{if } a \leq x \leq \frac{a+b}{2} \\ 2\left(\frac{x-b}{b-a}\right)^2; & \text{if } \frac{a+b}{2} \leq x \leq b \\ 0; \text{otherwise} & \end{array} \right\} \tag{5.7}$$

Here, $a$ and $b$ specify the range of values along which the function changes its values and can be controlled. Up to $a$, the value of function is kept constant (1) while after $b$, the value is set to 0. Between $a$ and $b$ two functions are used making the final output function the Z-shaped one. The slope of the function is dependent on the parameters $a$ and $b$. Here, $x$ represents the Euclidean distance between two data points i.e. $x = \|X_i - X_j\|$. As opposed to the conventional LPP where only few neighbors are considered, here the data points that are at a moderate distance from the point of interest are also taken into consideration and weighed accordingly.

After simplifying, the objective function turns out to be:

$$\begin{aligned} arg\,min \sum_{ij} (\mathbf{Y_i} - \mathbf{Y_j})^2 S_{ij} &= arg\,min \sum_{ij} (\mathbf{X_i V} - \mathbf{X_i V})^2 S_{ij} \\ &= arg\,min\, \mathbf{V^T X^T} \left(\mathbf{L} \otimes \mathbf{I_m}\right) \mathbf{XV} \end{aligned} \tag{5.8}$$

subject to the constraint,

$$\mathbf{V^T X^T} \left(\mathbf{M} \otimes \mathbf{I_m}\right) \mathbf{XV} = 1 \tag{5.9}$$

where, $\mathbf{L} = \mathbf{M} - \mathbf{S}$ is the Laplacian matrix and $M_{ii} = \sum_j S_{ij}$. $\mathbf{I_m}$ is the identity matrix of size $m \times m$ and $\otimes$ is the Kronecker product operator. As, we need to obtain orthogonal basis matrix, all the basis vectors should be orthogonal to each other. Hence, one more constraint is to be added.

$$\mathbf{v_i^T v_k} = 0, \forall i = \{1, 2, \cdots, k-1\} \tag{5.10}$$

For the first eigenvector $\mathbf{v_1}$, the problems reduces to minimize Equation 5.8 subject to Equation 5.9 and hence $v_1$ is the eigenvector corresponding to the smallest non-zero eigenvalue of the following generalized eigenvalue solution:

$$\mathbf{X^T}\left(\mathbf{L} \otimes \mathbf{I_m}\right)\mathbf{XV} = \lambda \mathbf{X^T}\left(\mathbf{M} \otimes \mathbf{I_m}\right)\mathbf{XV} \tag{5.11}$$

For rest of the basis vectors $\mathbf{v_i}$s, the problem now formulates to minimizing Equation 5.8 subject to constraints in Equation 5.9 and Equation 5.10. Using Lagrange's multiplier approach to solve the minimization problem,

$$\begin{aligned} C^{(K)} = \mathbf{v_k^T X^T}\left(\mathbf{L} \otimes \mathbf{I_m}\right)\mathbf{Xv_k} - \lambda\left[\mathbf{v_k^T X^T}\left(\mathbf{M} \otimes \mathbf{I_m}\right)\mathbf{Xv_k} - 1\right] \\ - \mu_1 \mathbf{v_k^T v_1} - \mu_2 \mathbf{v_k^T v_2} - \cdots - \mu_{k-1} \mathbf{v_k^T v_{k-1}} \end{aligned}$$

For optimizing the minimization problem, by equating partial derivatives of $C^{(K)}$ with respect to $\lambda$, $\mu_i$ and $\mathbf{v_k}$ to zero.

$$\frac{\partial C^{(K)}}{\partial \lambda} = 0 \Rightarrow \mathbf{V^T X^T}\left(\mathbf{M} \otimes \mathbf{I_m}\right)\mathbf{XV} = 1$$

$$\frac{\partial C^{(K)}}{\partial \mu_i} = 0 \Rightarrow \mathbf{v_k^T v_i} = 0$$

$$\begin{aligned} \frac{\partial C^{(K)}}{\partial \mathbf{v_k}} = 0 \Rightarrow 2\mathbf{X^T}\left(\mathbf{L} \otimes \mathbf{I_m}\right)\mathbf{Xv_k} - 2\lambda \mathbf{X^T}\left(\mathbf{M} \otimes \mathbf{I_m}\right)\mathbf{Xv_k} \\ - \mu_1 \mathbf{v_1} - \mu_2 \mathbf{v_2} - \cdots - \mu_{k-1} \mathbf{v_{k-1}} = 0 \end{aligned} \tag{5.12}$$

Multiplying Equation 5.12 by $\mathbf{v_k^T}$, we get

$$2\mathbf{v_k^T X^T}\left(\mathbf{L} \otimes \mathbf{I_m}\right)\mathbf{Xv_k} - 2\lambda \mathbf{v_k^T X^T}\left(\mathbf{M} \otimes \mathbf{I_m}\right)\mathbf{Xv_k} = 0$$

$$\Rightarrow \lambda = \frac{\mathbf{v_k^T X^T}\left(\mathbf{L} \otimes \mathbf{I_m}\right)\mathbf{Xv_k}}{\mathbf{v_k^T X^T}\left(\mathbf{M} \otimes \mathbf{I_m}\right)\mathbf{Xv_k}} \tag{5.13}$$

From the above value of $\lambda$ and Equation 5.11, it can be observed that $\lambda$ is the equation to be minimized. Since, $\left[\mathbf{X^T}\left(\mathbf{M}\otimes\mathbf{I_m}\right)\mathbf{X}\right]$ is positive definite and non-singular [26], [108], its inverse exists. Now, multiplying Equation 5.12 successively by $\mathbf{v_i^T}\left[\mathbf{X^T}\left(\mathbf{M}\otimes\mathbf{I_m}\right)\mathbf{X}\right]^{-1}$; $\forall i=1,2,\cdots,k-1$, we get a set of $(k-1)$ equations:

$$\begin{aligned}&\mu_1\mathbf{v_i^T}\left[\mathbf{X^T}\left(\mathbf{M}\otimes\mathbf{I_m}\right)\mathbf{X}\right]^{-1}\mathbf{v_1}+\\&\cdots+\mu_{k-1}\mathbf{v_i^T}\left[\mathbf{X^T}\left(\mathbf{M}\otimes\mathbf{I_m}\right)\mathbf{X}\right]^{-1}\mathbf{v_{k-1}}\\&=2\mathbf{v_i^T}\left[\mathbf{X^T}\left(\mathbf{M}\otimes\mathbf{I_m}\right)\mathbf{X}\right]^{-1}\left[\mathbf{X^T}\left(\mathbf{L}\otimes\mathbf{I_m}\right)\mathbf{X}\right]\mathbf{v_k}\end{aligned}\tag{5.14}$$

Converting the set (k-1) of equations stated in Equation 5.14 in matrix format, we get,

$$\mathbf{U^{(k-1)}}\mu^{\mathbf{(k-1)}}=2\left[\mathbf{V^{(k-1)}}\right]^T\left[\mathbf{X^T}\left(\mathbf{M}\otimes\mathbf{I_m}\right)\mathbf{X}\right]^{-1}\left[\mathbf{X^T}\left(\mathbf{L}\otimes\mathbf{I_m}\right)\mathbf{X}\right]\mathbf{v_k}\tag{5.15}$$

where,

$$\begin{aligned}\mathbf{V^{(k-1)}}&=\left[\mathbf{v_1},\mathbf{v_2},\ldots,\mathbf{v_{(k-1)}}\right]\\\mu^{\mathbf{(k-1)}}&=\left[\mu_1,\mu_2,\ldots,\mu_{(k-1)}\right]\\\mathbf{U^{(k-1)}}&=\left[\mathbf{V^{(k-1)}}\right]^T\left[\mathbf{X^T}\left(\mathbf{M}\otimes\mathbf{I_m}\right)\mathbf{X}\right]^{-1}\mathbf{V^{(k-1)}}\end{aligned}$$

From Equation 5.15, $\mu^{\mathbf{(k-1)}}$ can be obtained by multiplying the equation with $\left[\mathbf{U^{(k-1)}}\right]^{-1}$

$$\mu^{\mathbf{(k-1)}}=2\left[\mathbf{U^{(k-1)}}\right]^{-1}\left[\mathbf{V^{(k-1)}}\right]^T\left[\mathbf{X^T}\left(\mathbf{M}\otimes\mathbf{I_m}\right)\mathbf{X}\right]^{-1}\left[\mathbf{X^T}\left(\mathbf{L}\otimes\mathbf{I_m}\right)\mathbf{X}\right]\mathbf{v_k}\tag{5.16}$$

Multiplying Equation 5.12 by $\left[\mathbf{X^T}\left(\mathbf{M}\otimes\mathbf{I_m}\right)\mathbf{X}\right]^{-1}$:

$$\begin{aligned}&2\left[\mathbf{X^T}\left(\mathbf{M}\otimes\mathbf{I_m}\right)\mathbf{X}\right]^{-1}\left[\mathbf{X^T}\left(\mathbf{L}\otimes\mathbf{I_m}\right)\mathbf{X}\right]\mathbf{v_k}\\&\qquad-2\lambda\mathbf{v_k}-\left[\mathbf{X^T}\left(\mathbf{M}\otimes\mathbf{I_m}\right)\mathbf{X}\right]^{-1}\mathbf{V^{(k-1)}}\mu^{\mathbf{(k-1)}}=0\end{aligned}$$

Replacing $\mu^{\mathbf{(k-1)}}$ with the value found in Equation 5.16 we get,

$$\begin{aligned}&2\left[\mathbf{X^T}\left(\mathbf{M}\otimes\mathbf{I_m}\right)\mathbf{X}\right]^{-1}\left[\mathbf{X^T}\left(\mathbf{L}\otimes\mathbf{I_m}\right)\mathbf{X}\right]\mathbf{v_k}\\&\quad-2\lambda\mathbf{v_k}-2\left[\mathbf{X^T}\left(\mathbf{M}\otimes\mathbf{I_m}\right)\mathbf{X}\right]^{-1}\mathbf{V^{(k-1)}}\left[\mathbf{U^{(k-1)}}\right]^{-1}\left[\mathbf{V^{(k-1)}}\right]^T\\&\qquad\left[\mathbf{X^T}\left(\mathbf{M}\otimes\mathbf{I_m}\right)\mathbf{X}\right]^{-1}\left[\mathbf{X^T}\left(\mathbf{L}\otimes\mathbf{I_m}\right)\mathbf{X}\right]\mathbf{v_k}=0\end{aligned}\tag{5.17}$$

$$\mathbf{A}=\left[\mathbf{X^T}\left(\mathbf{M}\otimes\mathbf{I_m}\right)\mathbf{X}\right]^{-1}\mathbf{V^{(k-1)}}\left[\mathbf{U^{(k-1)}}\right]^{-1}\left[\mathbf{V^{(k-1)}}\right]^T$$

$$\Rightarrow \{\mathbf{I} - \mathbf{A}\} \left[\mathbf{X^T} \left(\mathbf{M} \otimes \mathbf{I_m}\right) \mathbf{X}\right]^{-1} \left[\mathbf{X^T} \left(\mathbf{L} \otimes \mathbf{I_m}\right) \mathbf{X}\right] \mathbf{v_k} = \lambda \mathbf{v_k}$$

As stated before, $\lambda$ is the criterion to be minimized and hence, $\mathbf{v_k}$ turn out to be eigen vectors of the following:

$$\mathbf{Z^{(k)}} = \{\mathbf{I} - A\} \left[\mathbf{X^T} \left(\mathbf{M} \otimes \mathbf{I_m}\right) \mathbf{X}\right]^{-1} \left[\mathbf{X^T} \left(\mathbf{L} \otimes \mathbf{I_m}\right) \mathbf{X}\right] \tag{5.18}$$

This is an iterative procedure, and in each iteration $\mathbf{v_k}$ is the eigenvector associated with the smallest non-zero eigenvalue of $Z^{(k)}$.

The main advantage of performing 2D-OLPP instead of the conventional 1D-OLPP is computational efficiency. In case of OLPP (1D), the basis matrix turns out to be of $mn \times mn$ i.e. it takes $mn$ iterations to build the complete basis matrix $\mathbf{V}$, whereas for 2D-OLPP, dimensions of $\mathbf{V}$ are $n \times n$. Hence it takes only $n$ iterations to compute $\mathbf{V}$ which makes 2D-OLPP computationally more efficient. Also, due to the compact data arrangement, the space complexity reduces. The time complexity analysis is discussed in detail in the next section.

## 5.3 2D-OLPP for Image Denoising

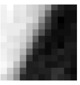
As discussed in Section 4.2.1, recent state of the art techniques for image denoising rely on two statistics of natural images: (1) There exists self-similarity between the patches from different locations of the same image [41], (2) Image patches can be sparsely represented by linear combinations of the basis vectors. The OLPP based image denoising approach uses the principle of similarity preservation and thus maintains structural similarity while learning the basis, but it requires the patches to be converted in vector format and works on a large window of the image at a time, instead of the whole image. The proposed 2D-OLPP approach is also applied for image denoising. In 2D-OLPP, patches from the entire input image are considered as it is (in matrix format) and weighed according to their structural similarity during the basis learning process. Hence, a *global dictionary* is learnt from the noisy input image, which remarkably reduces the computational complexity of the approach, and at the same time implements the underlying idea of non-local self similarity in dictionary learning process.

Suitability of 2D-OLPP for image denoising, parameter selection and noise removal process in the transformed domain are discussed in this section followed by experiments on some of the benchmark databases for image denoising in the next Section.

### 5.3.1 Incorporating non-local self similarity

The basis learning procedure of 2D-OLPP automatically takes care of non-local self similarity i.e. the patches that share more structural similarity are assigned higher weights using the z-shaped weighing function while constructing weight matrix **S**. Similarity between patches is measured by Euclidean distance. The reference patch, some of its structurally similar patches from the entire image, their euclidean distances from the reference patch and respective weights are shown in Table 5.1. In Figure 5.1, the reference patch is shown using blue boundary in the original image. Three of its similar patches i.e. patch #1780, patch #4059 and patch #692 are shown with red boundary. Though patch #692 is local neighbor of the reference patch, patches #1780 and #4059 are structurally more similar to it and hence are assigned higher weights.

Table 5.1: Reference patch, some of its structurally similar patches from the entire image, their euclidean distances from the reference patch and respective weights.

| Patch # | Reference Patch (150) | 1780 | 3191 | 4059 | 1021 | 692 | 8291 |
|---|---|---|---|---|---|---|---|
| Patch Image | | | |  | | | |
| Euclidean Distance | 0 | 109.749 | 128.768 | 139.309 | 188.155 | 164.213 | 173.465 |
| Weight | 0 | 0.931 | 0.905 | 0.889 | 0.798 | 0.846 | 0.828 |

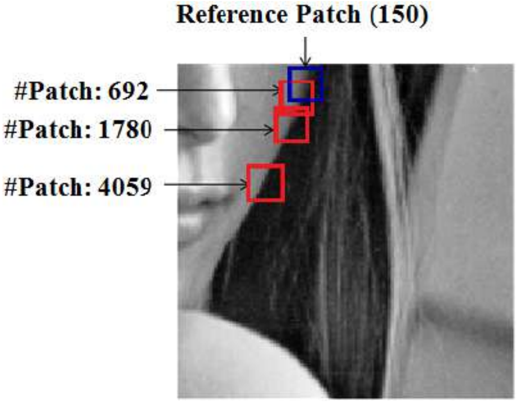


Figure 5.1: Reference patch from ‘Lena’ image and some of its similar patches from the whole image.

### 5.3.2 Sparsity

Although, the number of basis vectors in case of 2D-OLPP for an $m \times n$ image reduces from $mn$ to $n$, the coefficients in the 2D-OLPP domain are still sparse. To demonstrate the sparseness, coefficient plots using each of the $n$ basis vectors are shown in Figure 5.2 for a $120 \times 120$ portion of 'Lena' image. All $12 \times 12$ overlapping patches are used for learning the basis. Sparsity in the coefficients can still be observed in case of 2D-OLPP with only 12 basis vectors as compared to 144 for OLPP or PCA (all the approaches that work in vector format). As it is not possible to show all 144 OLPP basis vectors for $12 \times 12$ patches, coefficient plots considering all $6 \times 6$ patches from the same image are presented in Figure 5.3.

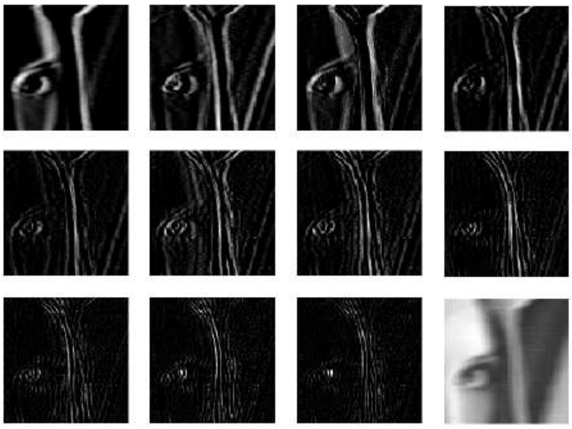


Figure 5.2: Coefficient-wise plot for all the $12 \times 12$ overlapping patches of a $120 \times 120$ portion of 'Lena' image using 2D-OLPP

### 5.3.3 Time complexity analysis

The cost, as far as time complexity is concerned, appears to be much less in case of 2D-OLPP as compared to state of the art methods. Time complexities of some of the state of the art approaches has been discussed in detail in Section 4.2.5. The time complexity analysis of the proposed 2D-OLPP based denoising approach is derived here. Total number of overlapping patches in the image is represented by $N$, average number of patches similar to the reference patch is $P$, average time required to search similar patches for each reference patch is $T_s$, time taken to generate the Laplacian matrix $L$ for $N$ patches is $NT_L$ and size of each patch is $l \times l$. In approaches that require the patch (in matrix format) to be transformed into vector format, size of each representative vector turns out to be $l^2 \times 1$.

In case of OLPP where patches are considered in vector format, $\mathcal{O}(NT_L)$ time is required to compute the Laplacian matrix and $\mathcal{O}(l^6)$ for the transformation matrix

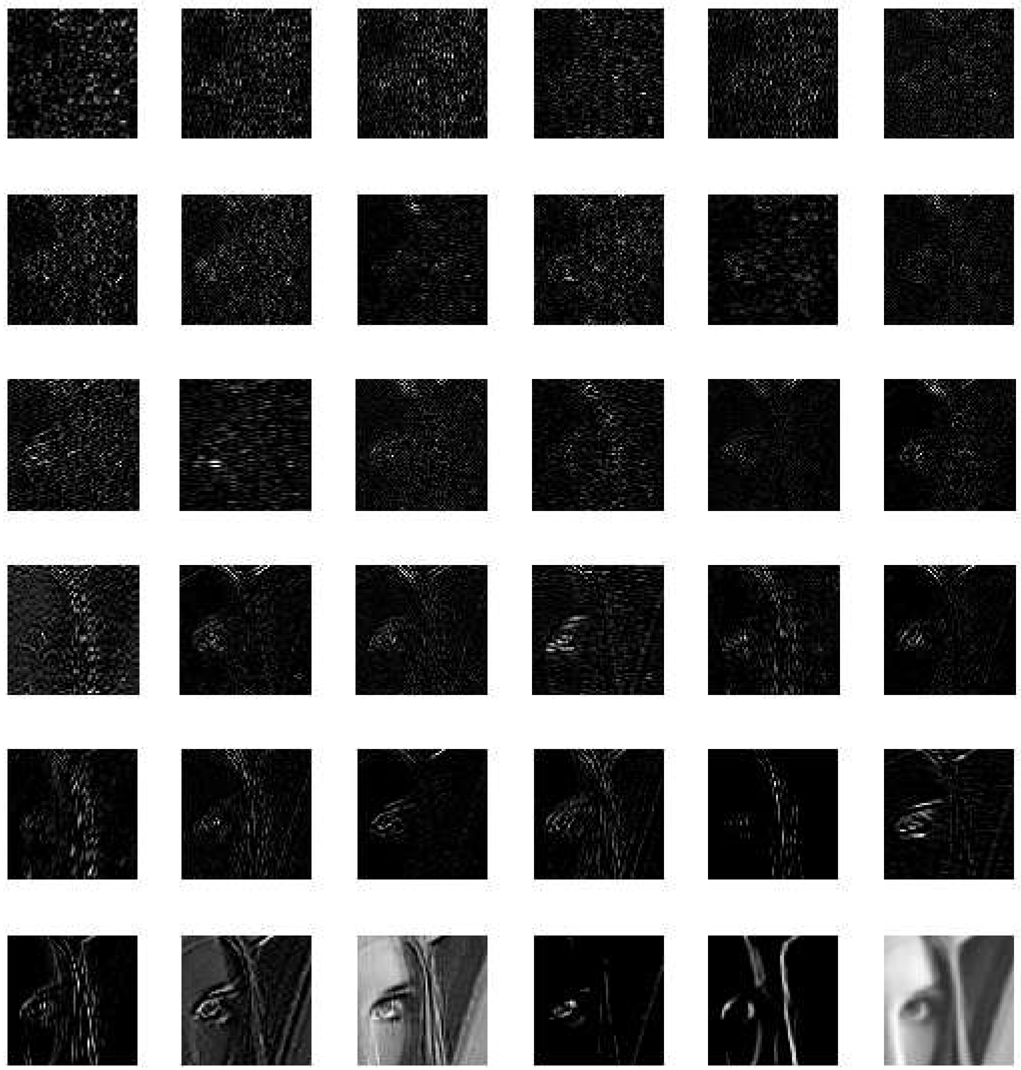


Figure 5.3: Coefficient-wise plot for all the 6 × 6 overlapping patches of a 128 × 128 portion of 'Lena' image using OLPP

i.e. basis matrix, as the global basis is computed only once for the whole image. Thus, overall time complexity is $\mathcal{O}(NT_L + l^6)$. Because of 2D patch processing in the proposed approach, the eigen decomposition problem reduces to $l \times l$ matrix instead of $l^2 \times l^2$ and hence requires $\mathcal{O}(l^3)$ time which leads to $\mathcal{O}(NT_L + l^3)$ complexity. Thus, the proposed approach is faster than other state of the art image denoising approaches. Time complexities of the state of the art approaches discussed in Section 4.2.5 along with that of the current proposal are summarized in Table 5.2.

Table 5.2: Summary of time complexities of denoising approaches. $N$: Number of patches, $l \times l$: size of a patch, $T_s$: average time required to search similar patches of the reference patch, $P$: average number of similar patches, $NT_L$: time taken to generate the Laplacian matrix $\mathbf{L}$ for $N$ patches.

| Technique | Time Complexity |
|---|---|
| LPG-PCA | $\mathcal{O}(N[T_s + Pl^4 + l^6])$ |
| BM3D | $\mathcal{O}(N[T_s + Pl^3 + P^2l^2])$ |
| OLPP | $\mathcal{O}(NT_L + l^6)$ |
| 2D-OLPP | $\mathcal{O}(NT_L + l^3)$ |

### 5.3.4 Selection of patch size

An important parameter in the denoising procedure is patch size. All the reported experiments use fixed patch size i.e. 12 × 12. Denoising experiments with different patch sizes ranging from 6 × 6 to 22 × 22 were carried out on the Lansel database of natural images. Average denoising results in terms of PSNR and SSIM values for various patch sizes are reported in Table 5.3. Though the performances of some of the patch sizes are not significantly different, patches with 12 × 12 size produce best PSNR and SSIM results. This performance has remained consistent across other databases, hence we have used patch-size 12 × 12 for the experiments reported in this chapter.

Table 5.3: Average denoising results for all the images from Lansel Database ($\sigma = 20$) using the proposed approach with various patch sizes

| Patch size | 6 | 8 | 10 | 12 | 15 | 20 | 22 |
|---|---|---|---|---|---|---|---|
| PSNR | 31.19 | 31.68 | 31.80 | **31.93** | 31.77 | 31.68 | 31.58 |
| SSIM | 0.822 | 0.859 | 0.867 | **0.873** | 0.872 | 0.871 | 0.868 |

### 5.3.5 Noise Removal

Experiments carried out so far suggest that the proposed 2D-OLPP can be effectively applied for image denoising tasks. Noisy image formation model in spatial domain is represented as $\mathbf{I}_\eta = \mathbf{I} + \eta$ and in transformed domain, using basis matrix $\mathbf{V}$, the model turns out to be $\mathbf{I}_{\eta\mathbf{V}} = \mathbf{I}_\mathbf{V} + \eta_V$. Here, $\eta \in \mathcal{N}(0, \sigma)$ i.e. i.i.d. Gaussian noise with zero mean and $\sigma$ standard deviation. A data matrix $\mathbf{X}$ consisting of all the patches $\mathbf{X_i}$ from the noisy input image $\mathbf{I}_\eta$, is constructed as discussed in Section 5.2. The basis matrix $\mathbf{V}$ is learnt following the iterative procedure and patch $\mathbf{X_i}$ in spatial domain is transformed to $\mathbf{X_{iv}}$ in the 2D-OLPP domain. Coefficients of patches in transformed domain are altered to eliminate the noise.

Given that $\mathbf{V}$ is an orthonormal basis, some of the standard forms of eliminating noise i.e. shrinkage rules can be directly used. Some of the most popular shrinkage rules are hard thresholding, soft thresholding [33], [35] and Wiener filter update rule (1D) [35, 34, 87] have already been discussed in Chapter 4. The rule to update the coefficients in the transformed domain using Wiener filter is given as $\tilde{X}_{iv_k} = \frac{\sigma^2_{v_k}}{\sigma^2+\sigma^2_{v_k}} X_{iv_k}$ where $X_{iv_k}$ represents $k^{th}$ coefficient of the $\mathbf{X_{iv}}$. Here, $\sigma$ is the standard deviation of noise. For a particular coefficient position $k$, $\sigma^2_{v_k}$ is computed by considering the $k^{th}$ coefficients from all the similar patches of the reference patch.

The Wiener filter update rule is modified for two dimensional (2D) patches. Instead of using only one coefficient across patches for estimating variance at a particular coefficient position, coefficients falling in 3×3 window around the coefficient of interest are used. Median of the variance of the particular window is used to compute the update rule *i.e.* $\sigma^2_{v_{k_n}ew} = median \left[\sigma^2_{v_k} \in N_k\right]$ where $N_k$ defines the neighboring window of the $k^{th}$ coefficient position. Note that, similar concept of median is the simplest form of filter for data denoising. Experiments have been performed to compare the proposed modified Wiener filter (2D-Wie) update rule with the other shrinkage rules such as hard thresholding (HardT) and conventional Wiener filter update (1D-Wie) in the 2D-OLPP denoising framework. Average PSNR and SSIM results for Lansel database reported in Table 5.4 show improvement over the other shrinkage rules.

Table 5.4: Average denoising results for Lansel database for different noise removal (filtering) techniques

| Filter | $\sigma = 20$ | | $\sigma = 30$ | | $\sigma = 40$ | |
|---|---|---|---|---|---|---|
| Type | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| HardT | 29.47 | 0.801 | 28.02 | 0.737 | 26.66 | 0.673 |
| 1D-Wie | 31.67 | 0.855 | 29.89 | 0.795 | 28.63 | 0.732 |
| 2D-Wie | **31.69** | **0.872** | **30.65** | **0.821** | **29.43** | **0.762** |

Denoising experiments on a large set of natural and texture databases on both gray-scale and color images are included in the next section.

## 5.4 Experiments

Algorithmic comparison of the proposed 2D-OLPP denoising with general framework of non-local self similarity based transformed domain image denoising approaches is shown in Figure 5.4. Basis vectors obtained using PCA are shown in the general framework in Figure 5.4(a). As it is 1D processing, patches are converted to vector format because of which size of the basis turns out to be $l^2 \times l^2$ where $l \times l$ is the patch size. First step involves grouping together similar patches and learning the local dictionary. Generally grouping is carried out based on the Euclidean distance between patches and defining a threshold either based on distance or number of patches. For LPG-PCA, this process is repeated for all the overlapping patches whereas for CSR, new set of PCA basis are generated for each cluster of patches. BM3D uses fixed 2D-DCT and 1D-Haar basis on the grouped patches. As it uses 2D transform, patch to vector conversion is not required. Coefficients of the patches in the respective transform domain are modified by applying one of the shrinkage rules discussed in 5.3.5 to eliminate noise. Denoised patch is transformed back to the spatial domain and is placed back at its original position after aggregation.

OLPP based denoising approach, being a global one, learns the basis only once for a large portion of image. Due to its one dimensional processing, the patches are required to be converted into vector format making the size of the dictionary $l^2 \times l^2$. The proposed approach directly uses image patches in matrix format for dictionary learning. For patches of size $l \times l$, basis i.e. 2D-OLPP dictionary turns out to be of size $l \times l$ only as shown in Figure 5.4(b). Also, the basis learning process itself takes care of similarity between patches as per the objective of 2D-OLPP, *i.e.* patches that are similar in spatial domain should remain close and have similar coefficients in the transformed domain as well. Hence, there is no need to explicitly group similar patches and a global dictionary is sufficient. Coefficients of patches in the 2D-OLPP domain are altered using modified Wiener filter update rule for 2D patch processing and inverse transform is performed to map denoised patches back to the spatial domain. Final restored image is obtained by averaging multiple candidate values for a particular pixel position.

A global 2D-OLPP basis for the entire image is learned using all the overlapping patches of size 12 × 12. For a given reference patch $\mathbf{Y}_r$, any patch $\mathbf{Y}_i$ for which $\|\mathbf{Y}_r - \mathbf{Y}_i\| \leq 1.7\sigma l$ is considered structurally similar and hence neighbor [96], where $l$ is the patch size. Weight, according to Equation 5.7, is assigned to patches that are similar to the reference patch while constructing the 2D-OLPP basis. Parameter '$b$' of Equation 5.7 is assigned value $1.7\sigma l$ for constructing weight matrix $\mathbf{W}$. This automatically takes care of similar patches, hence explicit search for similar patches of each patch is not required.

The proposed approach is compared with some of the state of the art approaches. Publicly available codes with the author specified parameter values are used for all the competing approaches to ensure unbiased comparison. As discussed in Chapter 4, all the approaches that are being compared assume that standard deviation of noise ($\sigma$) i.e. the noise level, is known, which may not always be the case. Some already available noise estimation approaches have been discussed in Section 4.2.6. All the experiments are performed in MATLAB R2012b environment on Lenovo Z510 with Intel core i7 windows 8 laptop. Restored image quality evaluation measures Peak Signal to Noise Ratio (PSNR) in terms of decibel (db) scale and Structural Similarity Index Measure (SSIM) [102] are used.

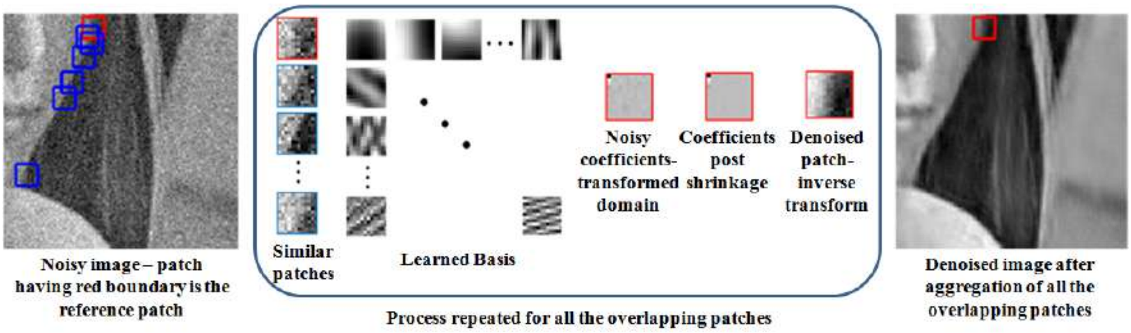
Apart from the existing approaches, to have fair comparison between the dictionary learning schemes and to establish the claim that locality preserving property plays an important role for the global basis to be suffice for the entire image, experiments using PCA in the global dictionary learning framework have also been performed. Instead of grouping together similar patches (as in some of the PCA based state of the art approaches), only one global PCA basis is learnt from all possible overlapping patches of the image to be denoised. Denoising procedure as explained for the proposed approach is followed.

Average denoising results for various databases using the Global PCA based denoising approach along with the other state of the art approaches are reported. Image denoising experiments on some of the benchmark gray-scale and color image databases are reported along with results of noise to noise test and method noise comparison for various approaches.

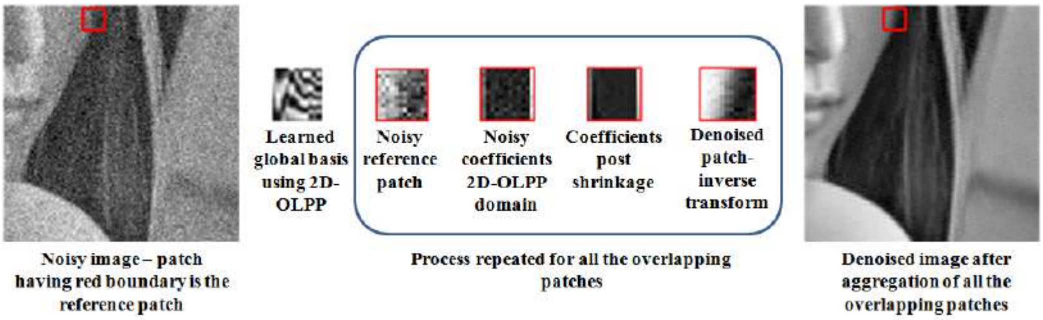
### 5.4.1 Gray-scale Image Denoising

Experiments on both natural and textured gray-scale images are performed. The proposed approach is compared with some of the state of the art denoising approaches i.e. principal component analysis with local pixel grouping (LPG-PCA) [87], block matching 3D (BM3D) [35], expected patch log-likelihood (EPLL) [91], clustering based sparse representation (CSR) [37] and OLPP based denoising discussed in Chapter 4.

For natural images, the Lansel database containing 13 widely used images for denoising (airplane, barbara, boats, couple, elaine, fingerprint, goldhill, house, man, mandrill, peppers, stream, zelda) is used. Noisy images with additive white Gaussian noise i.e. $\mathcal{N}(0, \sigma)$ having three noise levels $\sigma \in \{20, 30, 40\}$ are considered for experimentation. A portion from 'Barabara' image containing face part of the image, noisy image with $\sigma = 30$ and denoising results using various approaches are shown


Noisy image – patch having red boundary is the reference patch
Similar patches
Learned Basis
Noisy coefficients-transformed domain
Coefficients post shrinkage
Denoised patch-inverse transform
Process repeated for all the overlapping patches
Denoised image after aggregation of all the overlapping patches

(a) Framework for non-local self similarity based transformed domain image denoising.


Noisy image – patch having red boundary is the reference patch
Learned global basis using 2D-OLPP
Noisy reference patch
Noisy coefficients 2D-OLPP domain
Coefficients post shrinkage
Denoised patch-inverse transform
Process repeated for all the overlapping patches
Denoised image after aggregation of all the overlapping patches

(b) Framework for the proposed 2D-OLPP based image denoising.

Figure 5.4: Algorithmic comparison of the denoising approaches. All the patches are assumed to be of fixed size $l \times l$. (a) First step is to group together similar patches for a given reference patch. In case of 1D processing, the patches are converted to vector format before learning the dictionary which increases the size of dictionary i.e. number of basis vectors to be learned. In case of 1D data processing, $l \times l$ 2D patch is required to be converted to $l^2 \times 1$ sized vector, thus $l^2$ basis vectors are to be learned. The noisy reference patch is then projected on these basis vectors, coefficients in the transform are filtered to eliminate noise and inverse transformation is performed to get back to the spatial domain. This process is repeated for all overlapping patches as separate dictionary for all/cluster of patches is learned. (b) In case of the proposed denoising approach, only one global dictionary is sufficient for the entire image. Due to 2D data processing, the basis learnt are much compact i.e. $l$ as explained in Section 5.2. All the noisy patches are then projected on the same global basis, coefficients are modified using the modified Wiener filter update rule (refer Section 5.3.5), patches are transformed back to the spatial domain and aggregated.

in Figure 5.5. It can be observed that features of the face are nicely preserved using the proposed approach. Another denoising experiment on a portion of 'Lena' image with similar set of results is shown in Figure 5.6. LPG-PCA performs well in homogeneous regions of the image but oversmooths finer textural details. BM3D tends to over sharpen some of the edges (lips and shoulder portions of Lena image). BM3D also smooths out finer textures as it uses a tensor product of DCT and Haar wavelet basis, the latter of which is akin to performing diffusion filtering which can erase subtle textures [104]. EPLL restores textural details but sometimes produces spiky artifacts in the denoised image which can be clearly observed by zooming in. CSR based denoising approach produces small blocky artifacts specially in smooth (homogeneous) regions which are not present in outputs produced by the proposed approach. It is observed that the proposed approach outperforms all other approaches visually as well as quantitatively. It is to be noted that the proposed approach is able to preserve the finer textural details even at higher noise levels.

2D-OLPP is a two-dimensional extension of OLPP. It is not only computationally more efficient as compared to OLPP (as discussed in Section 5.3.3) but also processing two dimensional patches directly improves the denoising performance. In the process of matrix to vector conversion, neighborhood information of a pixel is not retained. This may not play any role in maintaining the similarity criterion, but while learning the basis, neighbors of a pixel turn out to be informative. It seems that finer textural details are restored well by the OLPP based approach but it undersmooths the homogeneous regions, hence performance of OLPP degrades for natural images. Artifacts are visible specially on the skin portions of both the 'Barbara' and 'Lena' images (denoised) using OLPP based method. 2D-OLPP overcomes these limitations and produces superior results both quantitatively and visually.

The PSNR and SSIM values for some of the images are reported in Figure 5.7 for 3 noise levels $\sigma \in \{20, 30, 40\}$ using various denoising approaches. The results are shown in graphical format for better visual comprehension. For each noise level, the resultant value is shown using a different symbol. Note that the results are not continuous, results of all the methods are joined by a line for each noise level for ease of comparison.

In order to show fine structure preservation property, experiments are performed on the UIUC Texture database (`http://www-cvr.ai.uiuc.edu/ponce_grp/data/index.html#texture`) containing 100 images of 17 different textures such as bark, brick, wood, carpet, wall etc. Average PSNR and SSIM values for some of the textures (by averaging all the images having same texture) are reported in Figure 5.9 and 5.10 for 3 different noise levels $\sigma \in \{20, 30, 40\}$. It can be observed that for both the gray-

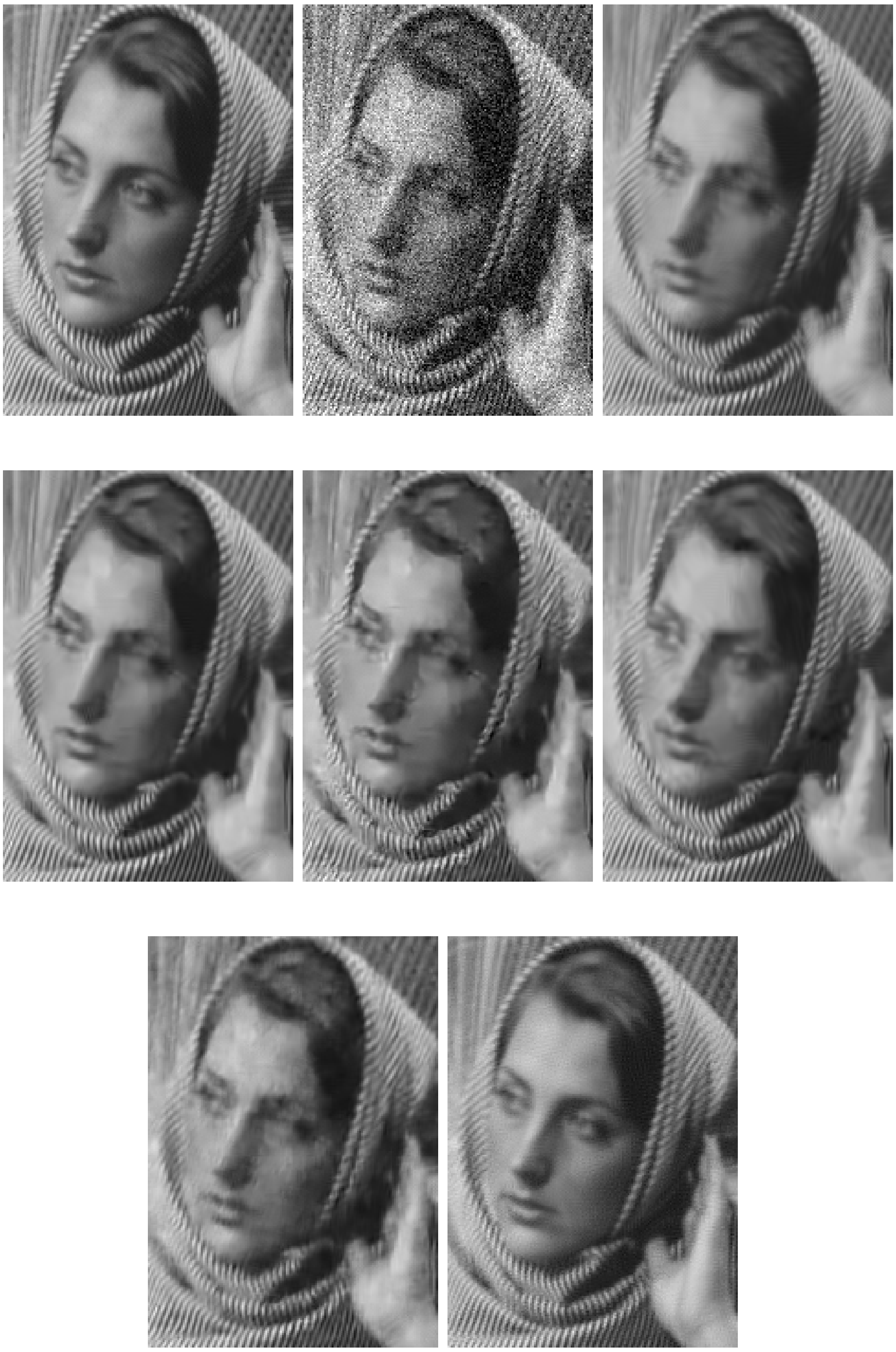


Figure 5.5: Denoising experiment on a part of natural image ('Barbara'). Left to right, top to bottom: Clean Image, Noisy image ($\sigma = 30$), LPG-PCA(28.39, 0.843), BM3D(28.93, 0.857), EPLL(27.44, 0.816), CSR(28.78, 0.850), OLPP(27.45, 0.821), 2D-OLPP(30.32, 0.875).

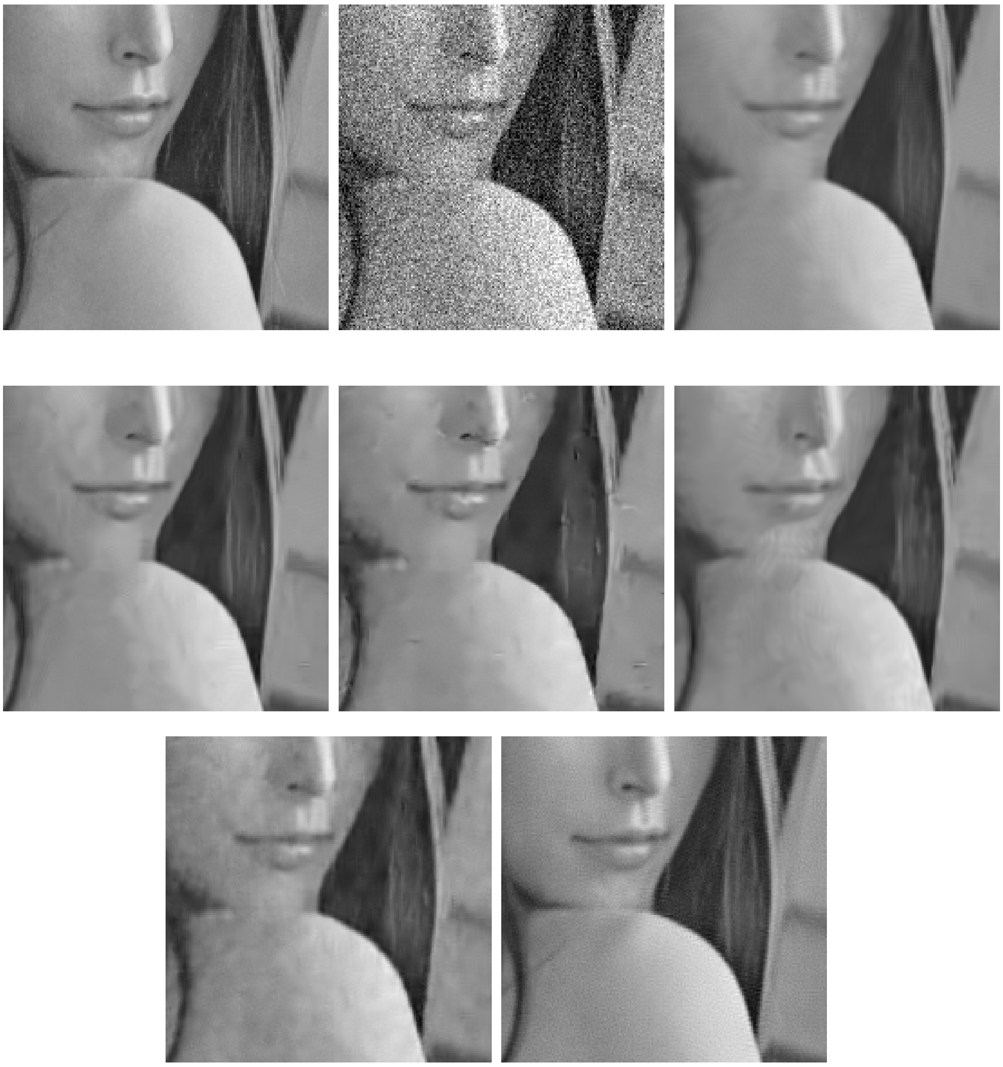
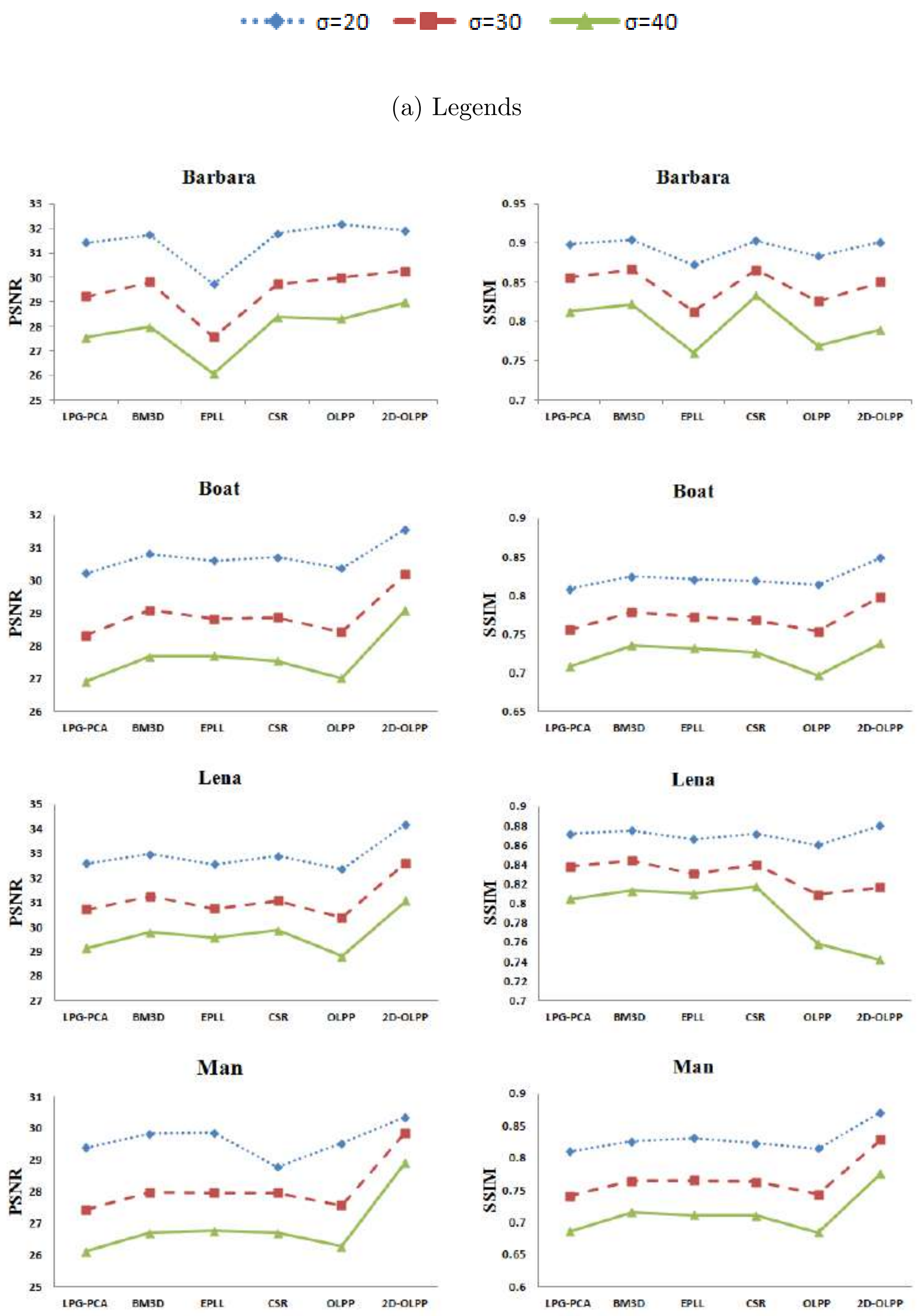
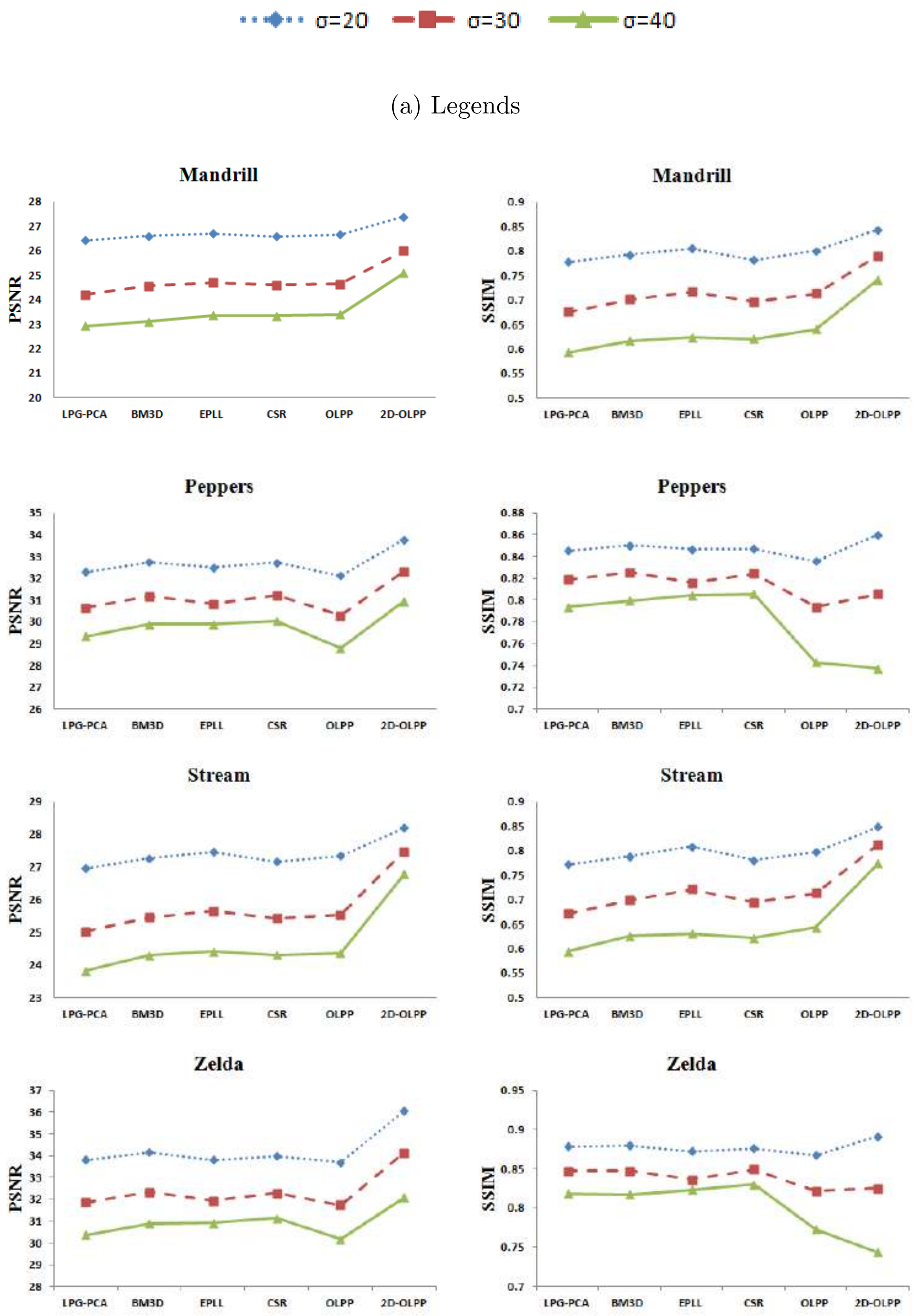


Figure 5.6: Denoising experiment on a portion of 'Lena' image. Left to right, top to bottom: Clean image, noisy image ($\sigma = 30$), LPG-PCA(31.73, 0.847), BM3D(32.38, 0.851), EPLL(31.78, 0.831), CSR(32.36, 0.850), OLPP(31.42, 0.820), 2D-OLPP(33.49, 0.842).



(a) Legends

Figure 5.7: Denoising results for some images from Lansel database of gray-scale natural images. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated in Figure 5.7(a)



(a) Legends

Figure 5.8: Denoising results for some images from Lansel database of gray-scale natural images. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM. Results are shown for three noise levels as indicated in Figure 5.8(a)

Table 5.5: Average denoising results for all the images from gray-scale databases: the Lansel Database and the UIUC Texture database ($\sigma \in \{20, 30, 40\}$)

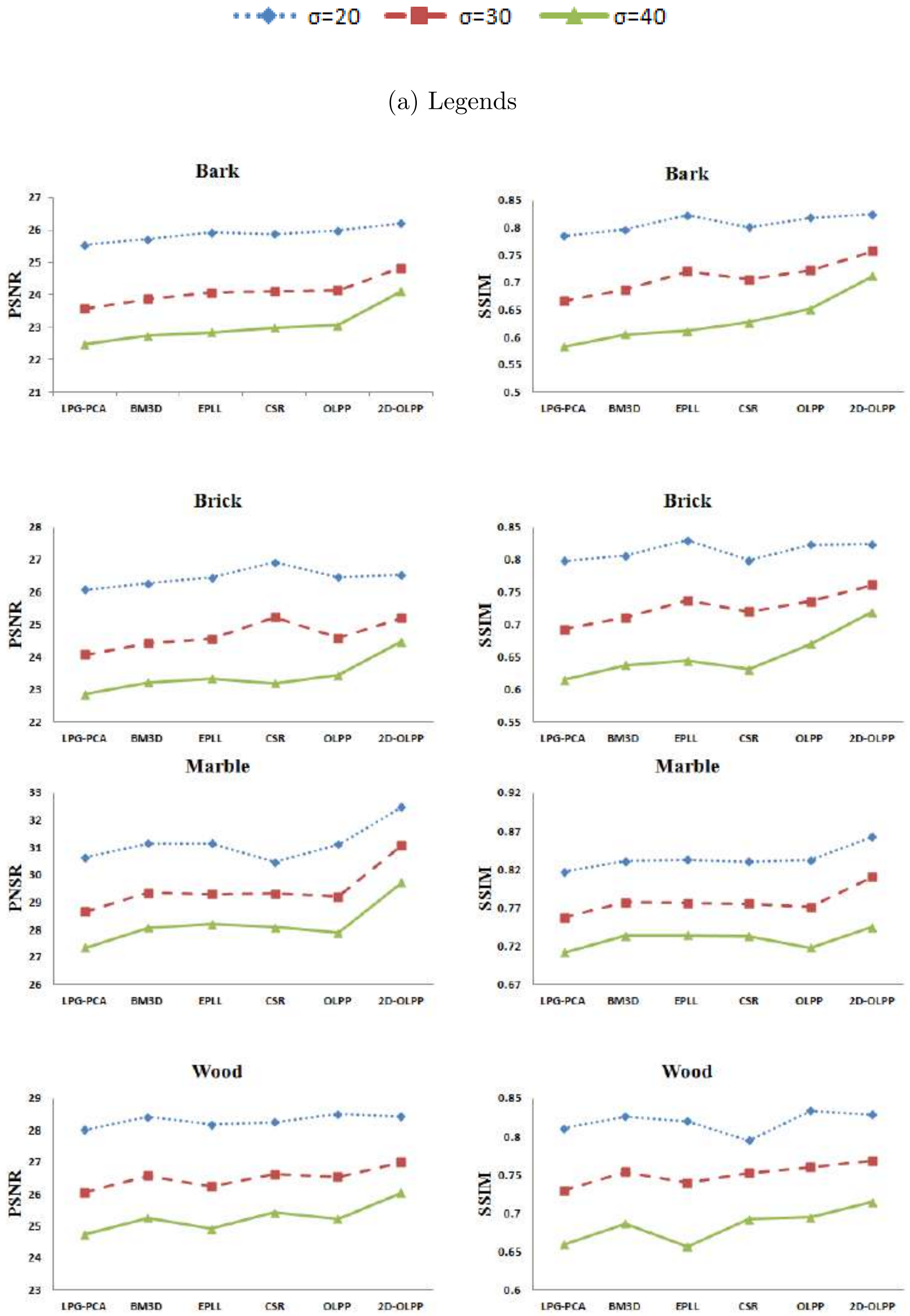
| Database | Denoising Approach | $\sigma = 20$ | | $\sigma = 30$ | | $\sigma = 40$ | |
|---|---|---|---|---|---|---|---|
| | | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| Lansel | Global PCA | 30.31 | 0.829 | 28.32 | 0.767 | 26.92 | 0.710 |
| | LPG-PCA | 30.42 | 0.834 | 28.48 | 0.78 | 27.08 | 0.73 |
| | BM3D | 30.78 | 0.84 | 29.03 | 0.796 | 27.66 | 0.754 |
| | EPLL | 30.46 | 0.839 | 28.60 | 0.788 | 27.48 | 0.747 |
| | CSR | 30.69 | 0.841 | 28.90 | 0.793 | 27.76 | 0.755 |
| | OLPP | 30.45 | 0.838 | 28.52 | 0.787 | 27.20 | 0.743 |
| | 2D-OLPP | **31.68** | **0.872** | **30.65** | **0.821** | **29.43** | **0.762** |
| UIUC Texture | Global PCA | 28.44 | 0.829 | 26.59 | 0.756 | 25.27 | 0.693 |
| | LPG-PCA | 28.25 | 0.807 | 26.24 | 0.722 | 24.96 | 0.654 |
| | BM3D | 28.58 | 0.819 | 26.74 | 0.747 | 25.49 | 0.686 |
| | EPLL | 28.46 | 0.824 | 26.58 | 0.746 | 25.37 | 0.669 |
| | CSR | 28.54 | 0.817 | 26.75 | 0.748 | 25.57 | 0.683 |
| | OLPP | 28.67 | 0.829 | 26.79 | 0.757 | 25.56 | 0.696 |
| | 2D-OLPP | **29.20** | **0.843** | **27.91** | **0.789** | **26.92** | **0.739** |

scale image databases, the proposed approach outperforms all the approaches both in terms of PSNR and SSIM values. Average results for all 100 images from UIUC texture database are reported in Table 5.5. For both natural and texture databases, it can be observed that both the similarity preserving transformed domain approaches, i.e. OLPP and 2D-OLPP outperform the denoising results obtained using global PCA in the same framework, which shows the impact of locality preserving property while learning the basis.

### 5.4.2 Color Image Denoising

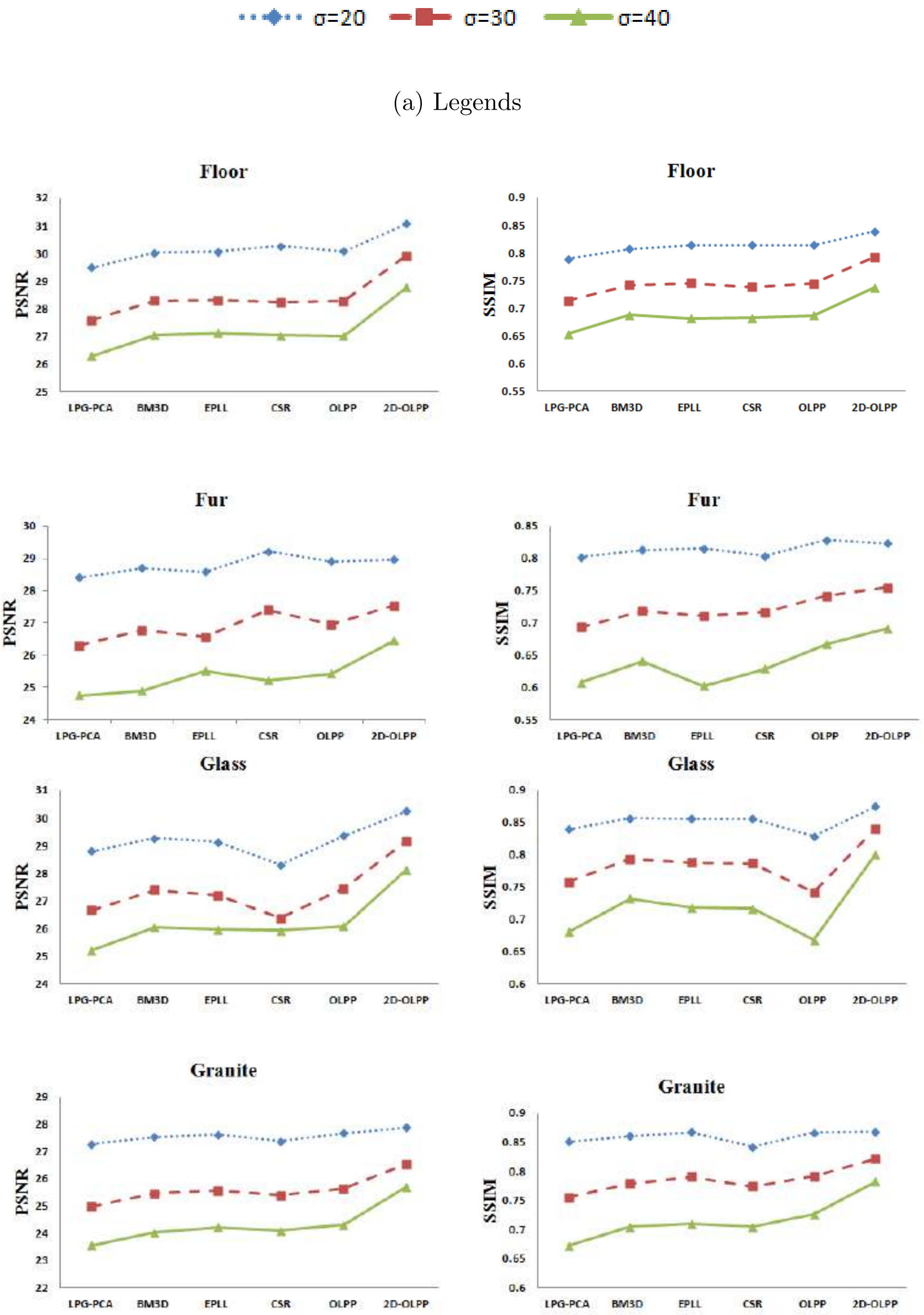
The proposed denoising approach is also applied for color image denoising. As the codes for EPLL and CSR approaches were not available for color images, comparison of the proposed approach with global PCA, LPG-PCA, BM3D and OLPP is reported. BM3D uses decorrelated color space YCbCr for the denoising purpose. Noise removal is performed only in the Y-channel of the image while Cb and Cr channels are kept as it is and after denoising the Y-channel, the image is transformed back to the original RGB color space. Thus, the information in Cb and Cr channels is ignored completely. On the other hand LPG-PCA denoises R, G and B channels independently, treating each of them as a separate gray-scale image. In this case, the dependency or correlation between the R, G, B channels is lost. In case of global PCA and OLPP, denoising is performed by converting the 3D patch in vector format.

In order to preserve the dependencies between R, G, B channels as well as to



(a) Legends

Figure 5.9: Denoising results for some images from UIUC Texture Database. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated in Figure 5.9(a)



(a) Legends

Figure 5.10: Denoising results for some images from UIUC Texture Database. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated in Figure 5.10(a)

process the 3D block/patch altogether, matricization, also known as unfolding of the 3-dimensional block, formally called 3D tensor [109], is carried out. As we have R, G and B- 3 channels, the 3-dimensional array is reordered into 3-mode matrices using the unfolding process. Each mode is denoised using the proposed technique and folded back to form the 3D block. Due to the 3-mode unfolding, resultant 3 folds are averaged to produce the final denoised block. The process of unfolding is described below.

A tensor is a multidimensional array and order/mode of a tensor is the number of dimensions [109], [110]. For an n-order tensor $\mathbf{X} \in \mathbf{R}^{\mathbf{I_1} \times \mathbf{I_2} \times \cdots \times \mathbf{I_N}}$, mode-n unfolding is denoted by $\mathbf{X_{(n)}}$. The tensor element $(i_1, i_2, \ldots, i_N)$ gets mapped to the matrix element $(i_n, j)$. Value of $j$ is computed as follows (refer [109] for details):

$$j = 1 + \sum_{k=1,k\neq n}^{N} (i_k - 1)J_k; J_k = \prod_{m=1,m\neq n}^{k-1} I_m \tag{5.19}$$

For better understanding of the unfolding procedure, an example showing unfolding $\mathbf{X_{(1)}}, \mathbf{X_{(2)}}, \mathbf{X_{(3)}}$ of a third-order tensor $\mathbf{X}$ are shown in Figure 5.12.

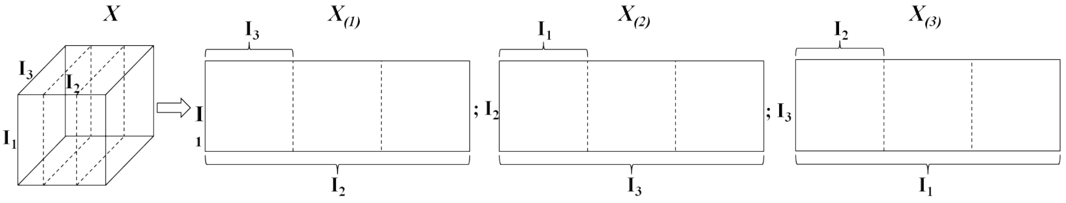


Figure 5.11: Mode-1, 2 and 3 unfolding of a third-order tensor

$$X_1 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, X_2 = \begin{bmatrix} 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \end{bmatrix}$$

$$X_{(1)} = \begin{bmatrix} 1 & 10 & 2 & 11 & 3 & 12 \\ 4 & 13 & 5 & 14 & 6 & 15 \\ 7 & 16 & 8 & 17 & 9 & 18 \end{bmatrix} X_{(2)} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 \\ 2 & 5 & 8 & 11 & 14 & 17 \\ 3 & 6 & 9 & 12 & 15 & 18 \end{bmatrix}$$

$$X_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 \end{bmatrix}$$

Figure 5.12: An example showing Mode-1, 2 and 3 unfolding of a third-order tensor

Two color image data sets namely the Kodak image database (`http://r0k.us/graphics/kodak/`) containing 24 natural images and first 28 images of the Brodatz color texture database (`http://multibandtexture.recherche.usherbrooke.`

`ca/colored_brodatz_more.html`) are used for experimentation. 12 × 12 × 3 image patches are used for denoising process. To establish the unfolding based approach for color image denoising, experiments have been conducted using both YCbCr and separate RGB channels based denoising in the 2D-OLPP based framework. Average results using YCbCR, separate RGB and proposed unfolding for some of the images from Kodak database with noise level ($\sigma = 20$) in terms of (PSNR, SSIM) are (29.59, 0.420), (29.39, 0.753) and (34.84, 0.828) respectively. It can be observed that, in the proposed 2D-OLPP denoising framework, unfolding based processing of color images is surpassing the other schemes, and hence all the experiments have been performed using the unfolding based procedure.

Denoising experiment on a color image from the Kodak database is shown in Figure 5.13. The resultant PSNR and SSIM values clearly indicate superiority of the proposed approach over LPG-PCA, BM3D and OLPP. Smoothness achieved in the denoised image using BM3D in homogeneous regions is better but it also smooths edges in those regions. Some undesired textures get generated while restoring the image in case of both LPG-PCA and BM3D. As mentioned earlier, restored image using OLPP is under-smoothed and contains some blob-like artifacts.

PSNR and SSIM values for some of the images from the Kodak database and Brodatz color texture database are reported in graphical format in Figure 5.14 and Figure 5.15 for 3 different noise levels $\sigma \in \{30, 40, 50\}$, along with the average results for all the images in Table 5.6. Proposed denoising scheme beats the other approaches with a margin of almost 1$db$ (PSNR) for all the three noise levels in case of the Kodak database.

Table 5.6: Average denoising results for color image databases: the Kodak database and the Brodatz texture database ($\sigma \in \{30, 40, 50\}$)

| Database | Denoising Approach | $\sigma = 30$ | | $\sigma = 40$ | | $\sigma = 50$ | |
|---|---|---|---|---|---|---|---|
| | | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| Kodak | Global PCA | 28.56 | 0.757 | 26.89 | 0.700 | 25.62 | 0.649 |
| | LPG-PCA | 28.32 | 0.72 | 26.63 | 0.628 | 25.36 | 0.556 |
| | BM3D | 30.20 | 0.839 | 28.51 | 0.783 | 27.90 | 0.761 |
| | OLPP | 29.51 | 0.825 | 28.16 | 0.776 | 27.10 | 0.732 |
| | 2D-OLPP | **31.34** | **0.842** | **29.84** | **0.806** | **28.85** | **0.790** |
| Brodatz color texture | Global PCA | 24.37 | 0.766 | 22.82 | 0.720 | 21.80 | 0.690 |
| | LPG-PCA | 26.62 | 0.816 | 24.92 | 0.750 | 23.52 | 0.680 |
| | BM3D | 27.28 | 0.866 | 25.67 | 0.817 | 25.18 | 0.795 |
| | OLPP | **27.82** | **0.876** | **26.53** | **0.841** | 25.40 | 0.803 |
| | 2D-OLPP | 27.55 | 0.868 | 26.37 | 0.834 | **25.61** | **0.808** |

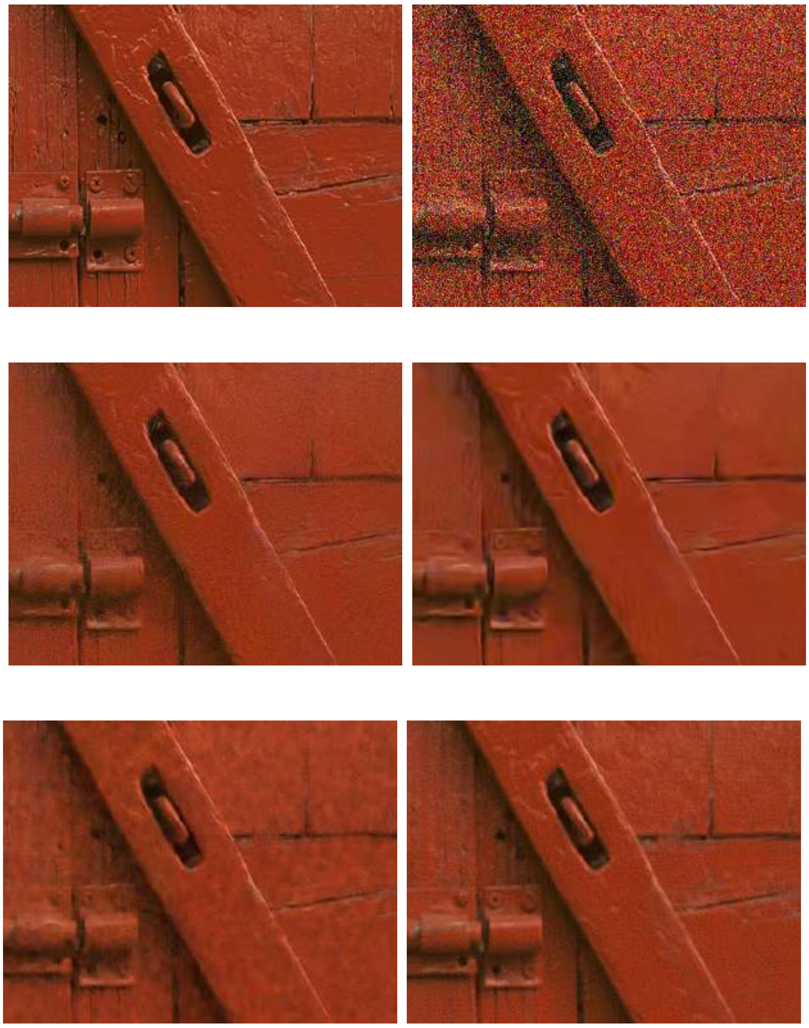
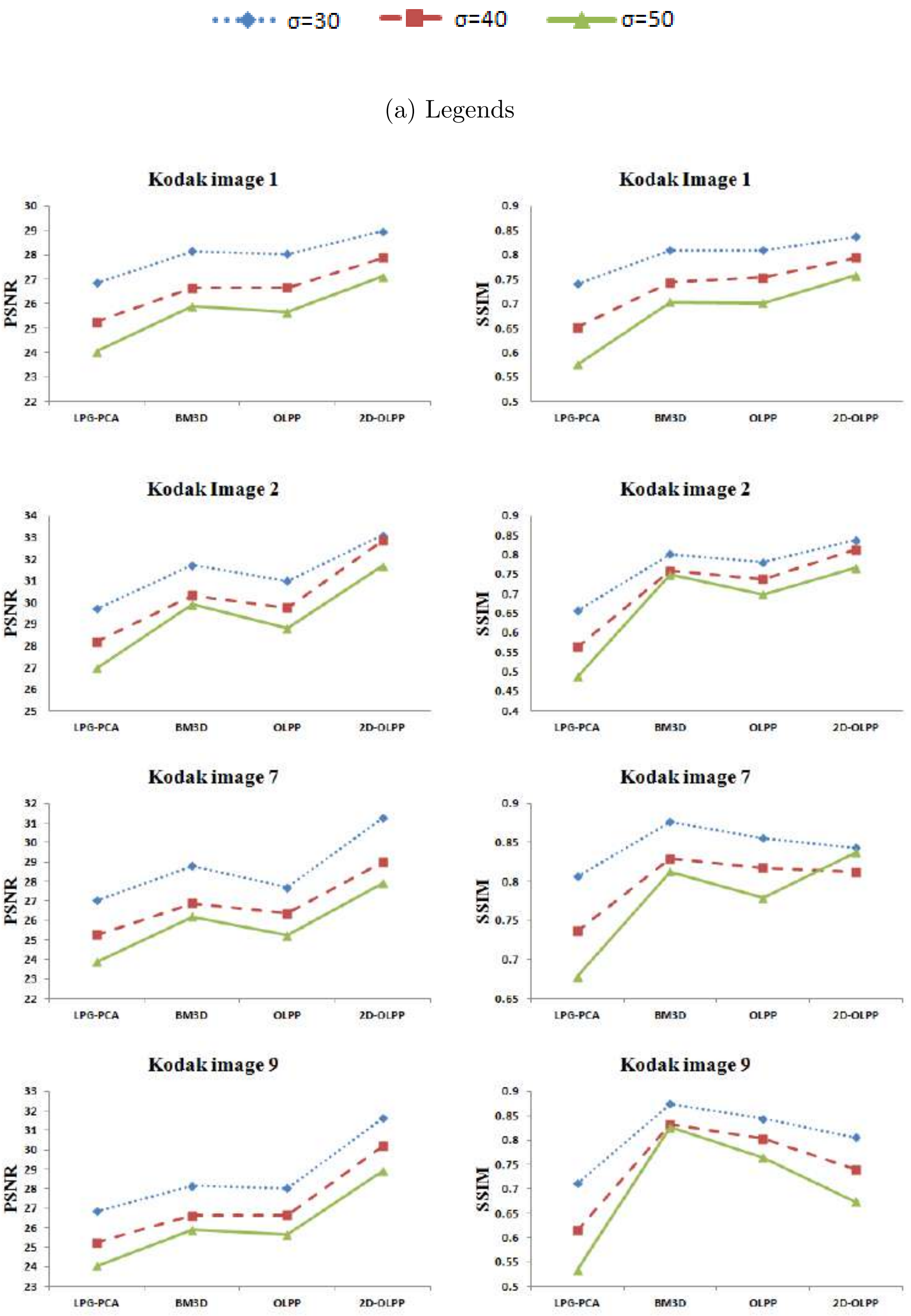
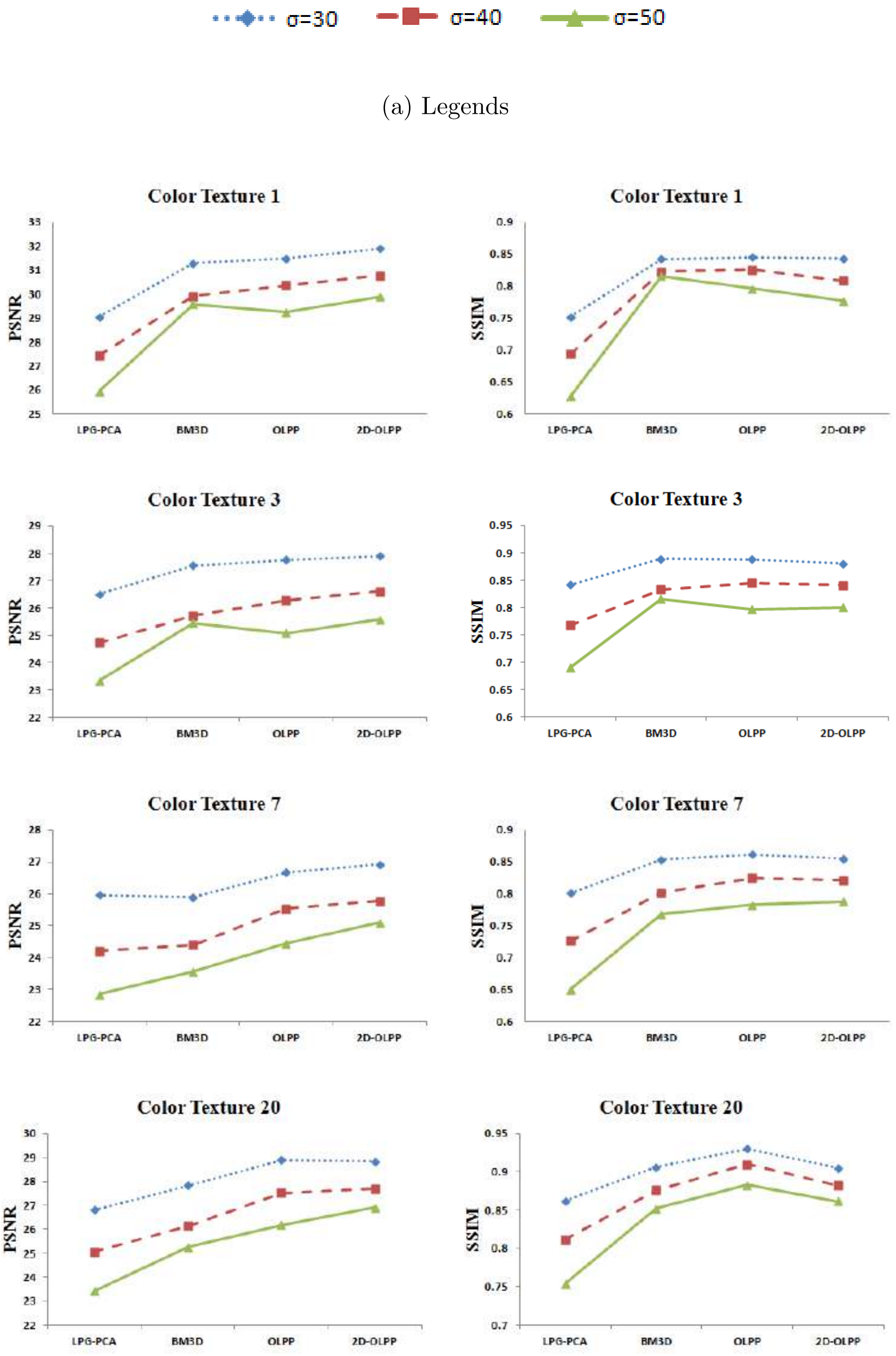


Figure 5.13: Denoising experiment on a color image from the Kodak database. Left to right, top to bottom: Clean image, noisy image ($\sigma = 50$), LPG-PCA (26.96, 0.487), BM3D (29.91, 0.749), OLPP (28.80, 0.699), 2D-OLPP (31.67, 0.765).



(a) Legends

Figure 5.14: Denoising results for images from Kodak database of color natural images. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated in Figure 5.14(a).



(a) Legends

Figure 5.15: Denoising results for some images from Bordatz Color Texture database. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated in 5.15(a)

### 5.4.3 The noise test

As suggested in [111], one of the way to characterize artifact free approaches is to check how the approach behaves on white noise image. Given a white noise image as input of the denoising approach, it should get transformed into white noise as only white noise is free from structure [112]. This is called the noise-to-noise criterion. Hence, if the output still remains white noise with same or lower variance, it will not have any artifacts in the denoised image. To test the noise-to-noise criterion, on the proposed 2D-OLPP based approach, a white noise image ($\mathcal{N}(o, \sigma), \sigma = 40$) was given as input. The resultant image along with the input image is shown in Figure 5.16. The proposed approach transforms white noise into white noise and hence satisfies the noise-to-noise criterion.

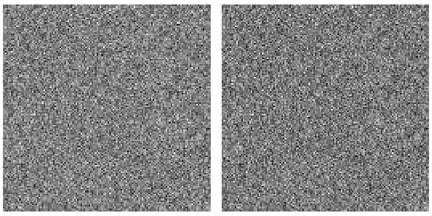


Figure 5.16: Noise-to-noise criterion: white noise image ($\mathbf{N}(0, \sigma), \sigma = 40$) (left), output of the proposed 2D-OLPP approach (right)

Method noise, also known as the residual image, is defined as the difference between a noisy image and its denoised version [41], [111]. It is one of the most powerful visual evaluation measures for denoising approaches. Ideally it should look like pure noise (white noise). Many denoising approaches remove image information (mainly the edges and fine textures) while eliminating noise. Method noise shows what part of noisy image has been removed by denoising process. Hence, the more method noise resembles white noise, lesser information content has been removed while denoising and the better is the denoising approach. Method noise for 'Lena' image using different denoising approaches is shown in Figure 5.17. All the prominent edges/features are clearly visible in method noise produced by the LPG-PCA approach. BM3D and EPLL both have tendency to smooth out sharp edges, which is evident from the residual images as most prominent edges of the respective images are present. Still, method noise is reduced considerably in both the approaches. In comparison with LPG-PCA, method noise is lesser in case of CSR, but structure of the original image is present in the residual image. Residual image produced from the denoised image using OLPP is close to noise, but the structural details can be observed by zooming in. We request the readers to zoom in the pdf to have better view. For 2D-OLPP, method noise for 'Lena' image completely resembles white noise and no structural details are present.

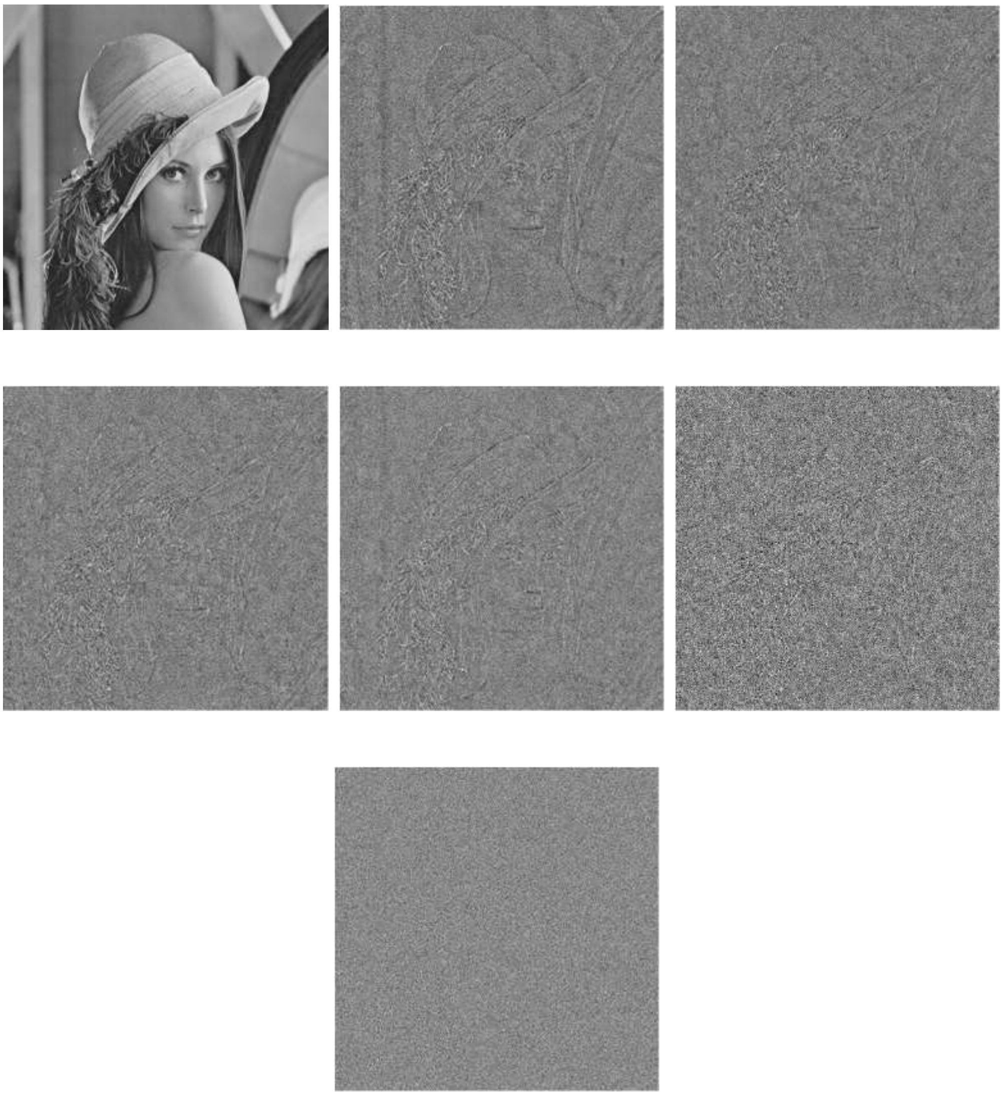


Figure 5.17: Method noise comparison on a natural image ('Lena'). Left to right, top to bottom: clean image, LPG-PCA, BM3D, EPLL, CSR, OLPP, 2D-OLPP.

## 5.5 Conclusions

Formulation of Two Dimensional Orthogonal Locality Preserving Projection (2D-OLPP) along with its application to image denoising is presented in this chapter. For denoising, 2D-OLPP processes two-dimensional image patches directly that preserves the spatial information. Computational complexity of 2D-OLPP is derived and it has been shown that it is significantly lesser than the other denoising algorithms. In contrast to the state of the art algorithms for denoising where basis are computed for each image patch, a global basis is sufficient for the entire image in 2D-OLPP. The approach is tested extensively on several benchmark data sets. The results obtained are very encouraging and appeared to be comparable with the forerunner approaches of image denoising. Finer textural details are well preserved even at higher noise levels. The proposed approach can further be extended for image deblurring and inpainting tasks.

# Chapter 6

# Two Dimensional Orthogonal Locality Preserving Discriminant Projection

## 6.1 Two Dimensional Orthogonal Locality Preserving Discriminant Projection

For better clustering of data, it is important to explore the information that discriminates dissimilar data points. So, along with preserving similarity information, separating the dissimilar data or or data belonging to different classes (in case of supervised approach), is always advantageous. Locality Preserving Discriminant Projection (LPDP), discussed in Chapter 3 is evolved on this idea.

LPP and LPDP are vectorized in nature, hence two dimensional data points (represented in matrix format) can not be processed directly and have to be converted to the vector format beforehand. This process not only adds computational complexity and overhead of matrix to vector conversion, but the neighborhood information of the pixels is also lost. As discussed in Section 4.2.1, neighboring pixels are highly correlated, disruption of which may cause information loss. One can take care of the neighbors while rearranging the matrices in vectors, which will again increase the computational complexity. Hence, it is always advantageous to process the matrices directly. Also, as both LPP and LPDP solve generalized eigenvalue problem, the basis obtained are non-orthogonal in nature. In many applications in image processing, attaining the data back in the original format is a requirement. In such cases orthogonal basis are beneficial as for orthogonal matrices $\mathbf{V}^T = \mathbf{V^{-1}}$ due to which the inverse transformation becomes much simpler. Orthogonal basis for LPP and

LPDP can be obtained directly if the normalization constraint is ignored as discussed in Section 4.1.1 and [8]. It has been observed that the arbitrary scaling in the data needs to be normalized, ignoring which may adversely affect the performance [8].

Two Dimensional Orthogonal Locality Preserving Projection (2D-OLPDP) that processes the data points in matrix format and learns orthogonal basis is proposed in this chapter. The basis are tuned in such a way that structurally similar data points get mapped close to each other, while the ones that are not similar get mapped far apart from each other in the projection space. Thus, better clustering of the data points is obtained.

### 6.1.1 Mathematical Formulation of 2D-OLPDP

Two Dimensional Orthogonal Locality Preserving Projection (2D-OLPDP) is derived here which directly processes the data in matrix form and obtains orthogonal basis. The objective of proposed approach is same as LPDP i.e. obtain maximum separability between dissimilar data keeping the local structure intact. Hence, the mapping of data points directly depends on the structural similarity between them. The objective function of 2D-OLPDP ensures that, data points, when projected in the 2D-OLPDP subspace, get clustered in such a way that structurally similar data points are mapped close by and dissimilar data points are mapped apart from each other. The objective function for 2D-OLPDP, modified for the matrix operation is as follows:

$$min \sum_{ij} (\mathbf{Y_i} - \mathbf{Y_j})^2 S_{ij} \ \& \ max \sum_{ij} (\mathbf{Y_i} - \mathbf{Y_j})^2 D_{ij} \tag{6.1}$$

here, $\mathbf{Y_i} = \mathbf{X_i}\mathbf{V}$ represents projection of the $\mathbf{X_i}^{th}$ data point in matrix format in the 2D-OLPDP domain using basis $\mathbf{V}$. The similarity and dissimilarity matrices $\mathbf{S}$, $\mathbf{D}$ are computed the same way as explained in LPDP (refer Chapter 3). $N$ training data points $\mathbf{X_i}s$ are arranged by stacking in row form as $\mathbf{X} = [\mathbf{X_1}; \mathbf{X_2}; \cdots; \mathbf{X_N}]$, thus forming the data matrix of size $mN \times n$. The objective function in the simplified matrix format after combining the both the minimization and maximization problems can be represented as:

$$argmin \, \mathbf{V^T X^T} \left\{ (\mathbf{L_s} - \mathbf{L_D}) \otimes \mathbf{I_m} \right\} \mathbf{XV} \tag{6.2}$$

here, $\mathbf{L_S} = \mathbf{M_S} - \mathbf{S}$, $\mathbf{L_D} = \mathbf{M_D} - \mathbf{D}$ and $\mathbf{M_S} = \sum_i S_{ij}$, $\mathbf{M_D} = \sum_{ij} D_{ij}$ are diagonal matrices. Details can be traced back from Chapter 3.

To take care of the arbitrary scaling, normalization constraint on the projected data is imposed:

$$\mathbf{Y^TY = I \Rightarrow V^TX^TXV = I} \tag{6.3}$$

In addition to the normalization constraint, to obtain the basis vectors that are orthogonal, orthogonalization constraint on the learned basis vectors is also added.

$$\mathbf{v^Tv = I}; \mathbf{v_i^T v_k} = 0; \forall i \in \{1, 2, \ldots, k-1\} \tag{6.4}$$

As it is not possible to obtain all the basis vectors that minimize the objective function satisfying both the constraints in one step, iterative procedure is followed. In this approach, each iteration produces one basis vector that is orthonormal to all the previously obtained vectors.

The very first basis vector $\mathbf{v_1}$ is obtained by solving the optimization problem stated in Equation 6.2 with just the normalization constraint i.e. Equation 6.3. Hence, $\mathbf{v_1}$ turns out to be the eigenvector corresponding to the smallest non-zero eigenvalue of the generalized eigenvalue solution represented in Equation 6.5:

$$\mathbf{X^T} \left\{ (\mathbf{L_S} - \mathbf{L_D}) \otimes \mathbf{I_m} \right\} \mathbf{X^TV} = \lambda \mathbf{X^TXV} \tag{6.5}$$

After obtaining the first basis vector, rest of them are computed by taking into consideration both the normalization and orthogonalization constraints. The problem now formulates to minimizing Equation 6.2 subject to constraints in Equations 6.3 and 6.4. Using Lagrange's multiplier approach to solve the minimization problem,

$$\begin{aligned} C^{(K)} = {} & \mathbf{v_k^T X^T} \left\{ (\mathbf{L_S} - \mathbf{L_D}) \otimes \mathbf{I_m} \right\} \mathbf{X v_k} \\ & - \lambda \left[ \mathbf{v_k^T X^T X v_k} - 1 \right] - \mu_1 \mathbf{v_k^T v_1} - \mu_2 \mathbf{v_k^T v_2} - \cdots - \mu_{k-1} \mathbf{v_k^T v_{k-1}} \end{aligned}$$

For optimizing the minimization problem, by equating partial derivatives of $C^{(K)}$ with respect to $\lambda$, $\mu_i$ and $\mathbf{v_k}$ to zero:

$$\frac{\partial C^{(K)}}{\partial \lambda} = 0 \Rightarrow \mathbf{V^TX^TXV = I}$$

$$\frac{\partial C^{(K)}}{\partial \mu_i} = 0 \Rightarrow \mathbf{v_k^T v_i} = 0$$

$$\begin{aligned} \frac{\partial C^{(K)}}{\partial \mathbf{v_k}} = 0 \Rightarrow {} & 2\mathbf{X^T} \left\{ (\mathbf{L_S} - \mathbf{L_D}) \otimes \mathbf{I_m} \right\} \mathbf{X v_k} \\ & - 2\lambda \mathbf{X^T X v_k} \\ & - \mu_1 \mathbf{v_1} - \mu_2 \mathbf{v_2} - \cdots - \mu_{k-1} \mathbf{v_{k-1}} = 0 \end{aligned} \tag{6.6}$$

Multiplying Equation 6.6 by $\mathbf{v_k^T}$, we can obtain $\lambda$:

$$\Rightarrow \lambda = \frac{\mathbf{v_k^T X^T} \left\{ (\mathbf{L_S} - \mathbf{L_D}) \otimes \mathbf{I_m} \right\} \mathbf{X v_k}}{\mathbf{v_k^T X^T X v_k}} \tag{6.7}$$

From the above value of $\lambda$ and Equation 6.5, it can be observed that $\lambda$ is to be minimized. Since, $\left[\mathbf{X^T X}\right]$ is positive definite and non-singular [26], its inverse exists. Now, multiplying Equation 6.6 successively by $\mathbf{v_i^T} \left[\mathbf{X^T X}\right]^{-1}$; $\forall i = 1, 2, \cdots, k-1$, we get a set of $(k-1)$ equations:

$$\begin{aligned} &\mu_1 \mathbf{v_i^T} \left[\mathbf{X^T X}\right]^{-1} \mathbf{v_1} + \\ &\cdots + \mu_{k-1} \mathbf{v_i^T} \left[\mathbf{X^T X}\right]^{-1} \mathbf{v_{k-1}} \\ &= 2\mathbf{v_i^T} \left[\mathbf{X^T X}\right]^{-1} \left[\mathbf{X^T} \left\{ (\mathbf{L_S} - \mathbf{L_D}) \otimes \mathbf{I_m} \right\} \mathbf{X}\right] \mathbf{v_k} \end{aligned} \tag{6.8}$$

Converting the set of (k-1) equations stated in Equation 6.8 in matrix format, we get,

$$\mathbf{U^{(k-1)}} \mu^{(\mathbf{k-1})} = 2 \left[\mathbf{V^{(k-1)}}\right]^T \left[\mathbf{X^T X}\right]^{-1} \left[\mathbf{X^T} \left\{ (\mathbf{L_S} - \mathbf{L_D}) \otimes \mathbf{I_m} \right\} \mathbf{X}\right] \mathbf{v_k} \tag{6.9}$$

where,

$$\begin{aligned} \mathbf{V^{(k-1)}} &= \left[\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_{(k-1)}}\right] \\ \mu^{(\mathbf{k-1})} &= \left[\mu_1, \mu_2, \ldots, \mu_{(k-1)}\right] \\ \mathbf{U^{(k-1)}} &= \left[\mathbf{V^{(k-1)}}\right]^T \left[\mathbf{X^T X}\right]^{-1} \mathbf{V^{(k-1)}} \end{aligned}$$

From Equation 6.9, $\mu^{(\mathbf{k-1})}$ can be obtained by multiplying the equation with $\left[\mathbf{U^{(k-1)}}\right]^{-1}$:

$$\mu^{(\mathbf{k-1})} = 2 \left[\mathbf{U^{(k-1)}}\right]^{-1} \left[\mathbf{V^{(k-1)}}\right]^T \left[\mathbf{X^T X}\right]^{-1} \left[\mathbf{X^T} \left\{ (\mathbf{L_S} - \mathbf{L_D}) \otimes \mathbf{I_m} \right\} \mathbf{X}\right] \mathbf{v_k} \tag{6.10}$$

Multiplying Equation 6.6 by $\left[\mathbf{X^T X}\right]^{-1}$ and replacing $\mu^{(\mathbf{k-1})}$ with the value found in Equation 6.10 we get,

$$\begin{aligned} 2 \left[\mathbf{X^T X}\right]^{-1} \left[\mathbf{X^T} \left\{ (\mathbf{L_S} - \mathbf{L_D}) \otimes \mathbf{I_m} \right\} \mathbf{X}\right] \mathbf{v_k} - 2\lambda \mathbf{v_k} \\ - 2 \left[\mathbf{X^T X}\right]^{-1} \mathbf{V^{(k-1)}} \left[\mathbf{U^{(k-1)}}\right]^{-1} \left[\mathbf{V^{(k-1)}}\right]^T \\ \left[\mathbf{X^T X}\right]^{-1} \left[\mathbf{X^T} \left\{ (\mathbf{L_S} - \mathbf{L_D}) \otimes \mathbf{I_m} \right\} \mathbf{X}\right] \mathbf{v_k} = 0 \end{aligned} \tag{6.11}$$

$$\mathbf{A} = \left[\mathbf{X^T X}\right]^{-1} \mathbf{V^{(k-1)}} \left[\mathbf{U^{(k-1)}}\right]^{-1} \left[\mathbf{V^{(k-1)}}\right]^T$$

$$\Rightarrow \left\{ \mathbf{I} - \mathbf{A} \right\} \left[\mathbf{X^T X}\right]^{-1} \left[\mathbf{X^T} \left\{ (\mathbf{L_S} - \mathbf{L_D}) \otimes \mathbf{I_m} \right\} \mathbf{X}\right] \mathbf{v_k} = \lambda \mathbf{v_k}$$

As stated before, $\lambda$ is the criterion to be minimized and hence, $\mathbf{v_k}$ turn out to be eigen vectors of the following:

$$\mathbf{Z^{(k)}} = \left\{ \mathbf{I} - A \right\} \left[\mathbf{X^T X}\right]^{-1} \left[\mathbf{X^T} \left\{ (\mathbf{L_S L_D}) \otimes \mathbf{I_m} \right\} \mathbf{X}\right] \tag{6.12}$$

This is an iterative procedure, each iteration of which computes $\mathbf{v_k}$, the eigenvector associated with the smallest non-zero eigenvalue of $Z^{(k)}$. The basis vectors thus found are orthogonal to each other, preserve similarity information in the transformed domain and also try to discriminate dissimilar data points. The current proposal, 2D-OLPDP has been used for image denoising in the same manner as OLPP and 2D-OLPP.

The proposed formulation is expected to achieve better patch grouping, as it will also incorporate the dissimilarity information while learning. Structurally different group of patches get projected far apart from each other in the projection space (transformed domain), hence prominent clusters of patches are attained. Experiments to support the claim have been reported in the next section.

## 6.2 Image Denoising using 2D-OLPDP

The proposed domain transformation technique is applied for image denoising task. Images go through various processing steps which may introduce undesired distortions called noise. It is always desired to have as clean an image as possible and hence denoising is performed to obtain a noise free image.

As discussed in Section 4.2.1, current state of the art image denoising approaches are based on patch based transform domain techniques. Image denoising in the transform domain works on the assumption that original signal can be represented as a linear combination of only a few basis vectors i.e. not all the basis vectors are required to represent the signal and hence the signal can be sparsely represented. Also, in the transform domain, the higher magnitude coefficients are the ones that carry information about the true clean image. Hence, by surpassing the lower magnitude coefficients, noise elimination can be performed and the original image can be obtained from the noisy one.

Here, noise elimination process is carried out in the proposed 2D-OLPDP domain. The proposed denoising algorithm is outlined as follows:

### 6.2.1 Algorithm : Image Denoising

**Algorithm 1.** **Input**: Noisy Image ($I_n$), patch size ($l \times l$), noise standard deviation ($\sigma$)

**Output**: Denoised Image ($\hat{I_n}$)

a.Extract all the overlapping patches of size $p \times p$ from the noisy image $I_n$.

```
    b.Learn a global 2D-OLPDP basis $\mathbf{V}$ considering all the extracted patches.
    c.Transform the patches using the obtained basis to the 2D-OLPDP domain i.e. $\mathbf{Y_i} = \mathbf{X_i V}$
    d.Eliminate noise by applying the modified Wiener filter update rule on the coefficients of the patches in the 2D-OLPDP domain.
    e.Inverse transform the denoised patches $\tilde{\mathbf{X_i}} = \tilde{\mathbf{Y_i}}\mathbf{V^T}$
    f.Aggregate the patches in the spatial domain to regenerate the image.
end
```

As outlined in the algorithm, the input arguments are noisy image $I_n$, size of the patches to be extracted from the noisy image $l \times l$ and level of noise $\sigma$ in the input image. Patch size selection plays very important role in the denoising process as very small patch size may result in a very large pool of data points to learn the basis and the basis matrix also turns out to be too small. The basis matrix in case of 2D-OLPDP is of size $l \times l$, which is the same as the patch size. Too large patch size may reduce the effect of self similarity within the image as the size of the patch increases, number of similar patches to the reference patch within the image decreases. We performed experiments using various patch sizes and based on the best denoising results using the patches of size $12 \times 12$, all the experiments reported here are performed using patches of size $12 \times 12$.

The other input variable is the noise level of the image to be denoised. As discussed in previous chapters, we assume the standard deviation of the noise to be known beforehand. In practical scenarios, this information may not always be available and in such cases, noise estimation approaches discussed in in Section 4.2.6 can be used.

In the process of learning 2D-OLPDP basis, structurally similar patches are assigned higher weights in the similarity criteria in order to keep them closer in the transform domain. Similarity of patches with other patches from the noisy image is decided based on a threshold proposed in [96]. If the Euclidean distance of the reference patch from any other patch is less than $1.7\sigma p^2$, where the patch is of size $p \times p$, they are considered to be similar and weighed accordingly by the similarity function. To achieve better discrimination between dissimilar patches and eventually better clustering of patches in the transform domain, the patches that are not similar are projected far apart. In order to do so, higher weights are assigned to such patches in the dissimilarity matrix. For dissimilarity measure again, Euclidean distance between the patches is considered. The farthest patch is assigned highest weight in the dissimilarity matrix.

Noise elimination is performed by manipulating the coefficients of the patches in the 2D-OLPDP domain. In order to attenuate the noise, standard forms for orthogonal transforms can be used such as soft thresholding [33], hard thresholding [35] and Wiener filter update rule (1D) [35, 34, 87]. In this work, noise in the transformed domain is suppressed using the modified Wiener filter update rule proposed in Section 5.3.5. According to this rule, instead of obtaining the update coefficient weight based only on one coefficient location, a small $3 \times 3$ window of nearby pixel locations is considered, median of which is selected as the new update weight.

The patches, thus denoised are transformed back to the spatial domain. As all the overlapping patches are considered while constructing the image back in the spatial domain, aggregation is performed for all the pixel positions.

### 6.2.2 Automatic clustering of patches

In Section 6.1 and 6.2.1, it has been mentioned that the proposed two dimensional basis learning procedure i.e. 2D-OLPDP naturally clusters the data points in the transformed domain. Locality preserving property tries to keep structurally similar patches close by in the transform domain whereas discrimination property maps the dissimilar patches far apart, resulting in better clustering. A small experiment is reported to justify the argument. Strongest three dimensional projection of some of the patches extracted from portions of 'Lena' and 'Barbara' image is shown in Figure 6.1. Each color and symbol denotes a different set of similar patches. It can be observed that in both the cases, similar patches are projected close to each other. Some of the sample patches are shown near the corresponding cluster. In the 2D-OLPDP domain, distinct clusters of structurally similar patches show the property of grouping, and hence, explicit grouping of the patches is no longer required.

### 6.2.3 Complexity Analysis

Besides analyzing the effectiveness of the proposed image denoising scheme quantitatively in terms of image quality evaluation measures, it is also important to study the computational efficiency of the same. As discussed in Section 6.1, similarity and dissimilarity matrices $\mathbf{S}$ and $\mathbf{D}$ are to be built in order to learn the basis that achieve better separation between clusters of structurally unlike patches. Time taken to compute $\mathbf{S}$ and $\mathbf{D}$ for a set of $M$ such patches is denoted by $NT_{L_s}$ and $NT_{L_d}$ respectively. Processing the patches in the matrix format results in considerably smaller basis matrix of size $l \times l$ thus reducing the operations to $\mathcal{O}(l^3)$. The overall computational complexity of the 2D-OLPDP based denoising approach turns out to be

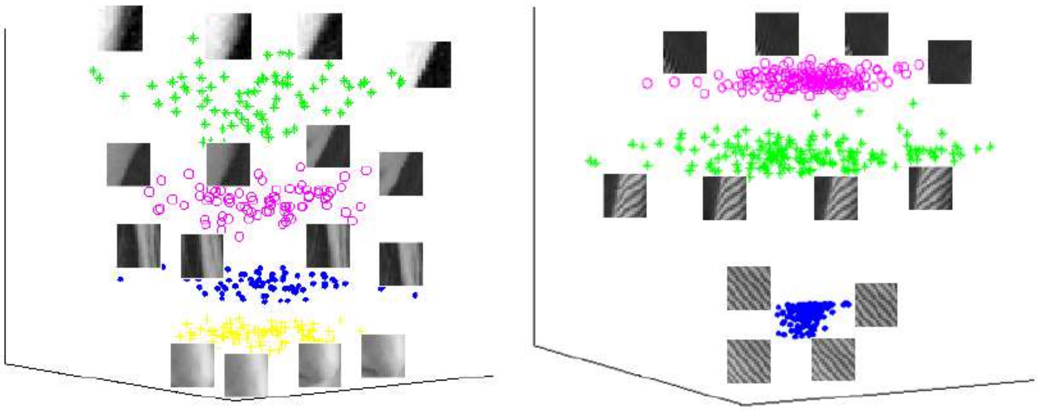


Figure 6.1: Strongest three dimensional projection of patches extracted from 'Lena' and 'Barbara' images. The patches automatically get clustered when projected on 2D-OLPDP basis.

$\mathcal{O}(NT_{L_s} + NT_{L_d} + l^3)$. 2D-OLPP takes care of preserving structurally similar patches but does not pay attention to the discriminant information while learning, hence its computational complexity is $\mathcal{O}(NT_{L_s} + l^3)$. It can be observed that incorporation of discriminant information while learning adds some extra computations to the current proposal as compared to 2D-OLPP, but at the same time, it provides better clustering of patches in the transformed domain resulting in improved image denoising.

Detailed discussion about 2D-OLPP and OLPP can be found in Section 5.3.3 and Section 4.2.5 along with the complexity analysis of some other state of the art approaches. Computational complexities are listed in Table 6.1 for reference. It is to be noted that the proposed 2D-OLPDP based approach is more efficient than the other state of the art image denoising methods. The symbols used to denote the complexities are described as: $N$: Number of patches, $l \times l$: size of a patch, $T_s$: average time required to search similar patches of the reference patch, $P$: average number of similar patches, $NT_{L_s}$: time taken to generate the similarity matrix $\mathbf{S}$ for $N$ patches, $NT_{L_d}$: time taken to generate the dissimilarity matrix $\mathbf{D}$ for $N$ patches.

## 6.3 Experiments

Image denoising experiments on both gray scale and color image databases are performed as reported in Chapters 4 and 5. Results are compared with several transformed domain image denoising techniques namely, block matching 3D (BM3D) [35], local pixel grouping based principle component analysis (LPGPCA) [87], clustering

Table 6.1: Summary of computational complexities of state of the art denoising approaches. $N$: Number of patches, $l \times l$: size of a patch, $T_s$: average time required to search similar patches of the reference patch, $P$: average number of similar patches, $NT_{L_s}$: time taken to generate the similarity matrix $\mathbf{S}$ for $N$ patches, $NT_{L_d}$: time taken to generate the dissimilarity matrix $\mathbf{D}$ for $N$ patches.

| Technique | Time Complexity |
|---|---|
| Technique | Time Complexity |
| LPG-PCA | $\mathcal{O}(N[T_s + Pl^4 + l^6])$ |
| BM3D | $\mathcal{O}(N[T_s + Pl^3 + P^2l^2])$ |
| OLPP | $\mathcal{O}(NT_L + l^6)$ |
| 2D-OLPP | $\mathcal{O}(NT_L + l^3)$ |
| 2D-OLPSP | $\mathcal{O}(NT_L + T_D + l^3)$ |

based sparse representation (CSR) [37], expected patch log-likelihood (EPLL) [91], OLPP based denoising discussed in Chapter 4 and image denoising using 2D-OLPP proposed in 5. Codes provided by the authors are used to report the results of the competing image denoising approaches. Quality of the denoised image is evaluated using Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [102].

### 6.3.1 Gray scale Image Denoising

Same databases for image denoising, as used in previous chapters, the Lansel database of gray scale natural iamges and the UIUC texture database [103] are used for performing the experiments. Denoised images using recent state of the art image denoising approaches are reported in Figures 6.2, 6.3 and 6.4. A cropped portion from the 'Barbara' image having both smooth as well as the textured regions is shown in Figure 6.2. The proposed approach surpasses or is comparable to all other denoising approaches. Face part of the 'Elaine' image is reported in Figure 6.3. Superior restoration of the hair texture using the proposed approach can be observed. All the competing approaches except 2D-OLPP tend to smooth out the hair. Though denoised images using 2D-OLPP and 2D-OLPDP are visually very similar, both the quality evaluation measures show improvement in the performance.

Figure 6.4 shows restoration of noisy 'Airplane' image (noise level $\sigma = 40$) along with zoomed portion of a small portion from the image containing some digits marked by a box in the original image. In the restored image using LPG-PCA, the digits are not clearly visible and blurring around the boundaries of the digits can observed. The smooth i.e. homogeneous portions of the zoomed part are restored very well by both

BM3D and EPLL but the digits are mixed up and not readable. CSR produces best result so far, as most of the digits in the denoised image are readable. The results obtained using OLPP based approach are not that encouraging whereas processing two dimensional patches directly turns out to be beneficial in case of 2D-OLPP as the resultant denoised image is comparable to the one produced using CSR. The restored digits are blurred yet visible. Output using the proposed 2D-OLPDP based denoising scheme produces sharper edges and the blurring effects are less as compared to all the approaches discussed so far. Still some blocky effects can be observed in the homogeneous region which show scope of improvement in such cases. BM3D clearly out performs the other compared methods as far as the homogeneous regions are considered. However, the overall denoising performance of the proposed approach is surpassing other schemes both visually as well as qualitatively.

PSNR and SSIM values for some of the images from Lansel database are reported in Figure 6.5 and 6.6 for 3 noise levels $\sigma \in \{20, 30, 40\}$ using various denoising approaches. The results are shown in graphical format for better visual comprehension. For each noise level, the resultant value is shown using a different symbol. Note that the results are not continuous, results of all the methods are joined by a line for each noise level for ease of comparison.

Average denoising results for both the gray scale databases comparing state of the art denoising approaches along with OLPP, 2D-OLPP and 2D-OLPDP based denoising approaches proposed in this work are reported in Table 6.2. Improvement in the performance of the current proposal over the other competitive approaches can be noticed.

### 6.3.2 Color Image Denoising

Along with gray scale images, denoising experiments on color image databases have also been performed. Color image denoising experiments are performed on Kodak [113] and Brodatz texture database [114]. Color images are processed in the same manner as discussed in 5. On 3D blocks comprised of R, G and B channels, the matricization procedure is carried out to preserve the dependency between all three channels, each unfolded block is denoised separately and the results are averaged to obtain the final output image. Denoising results on a texture image from Brodatz database are shown in Figure 6.9. Comparison of the proposed approach with LPG-PCA, BM3D, OLPP and 2D-OLPP based denoising methods on some images from both Kodak database and Brodatz color texture database are reported in Figure 6.10 and Figure 6.11 respectively. Average denoising results for Kodak database and a part of Brodatz texture database in terms of PSNR (in db) and SSIM values is included

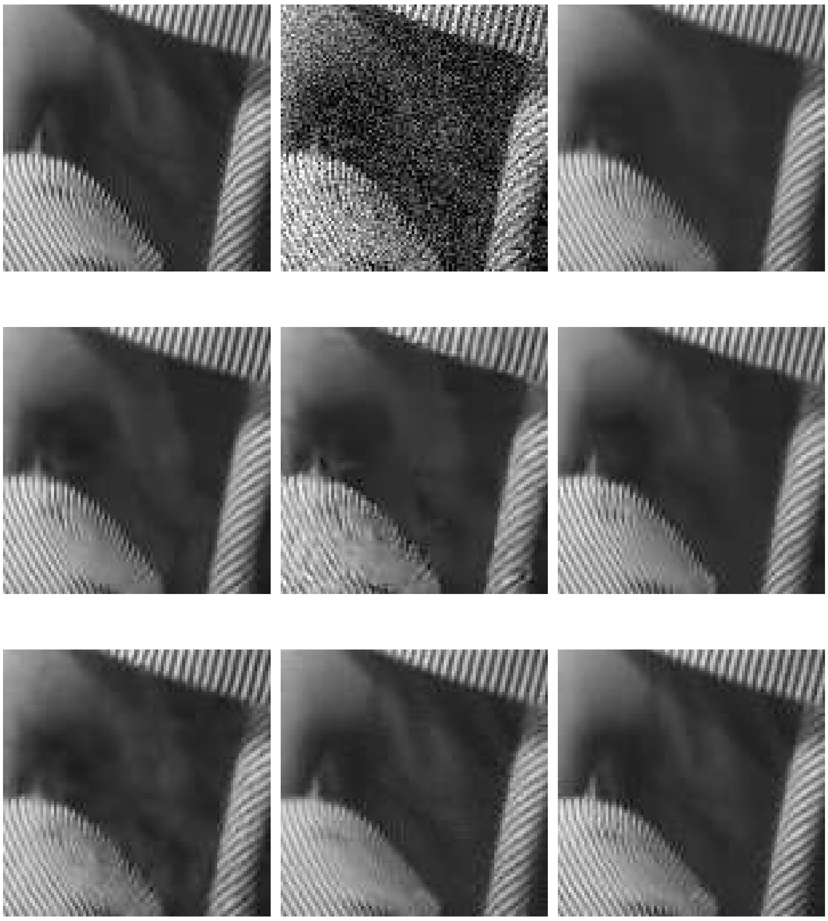


Figure 6.2: Denoising experiment on a part of natural image (‘Barbara’). Left to right, top to bottom: Clean image, noisy image ($\sigma = 30$), LPG-PCA(29.17, 0.879), BM3D(29.56, 0.880), EPLL(27.07, 0.844), CSR(29.86, 0.889), OLPP(28.02, 0.841), 2D-OLPP(28.28, 0.820), 2D-OLPDP(30.00, 0.879).

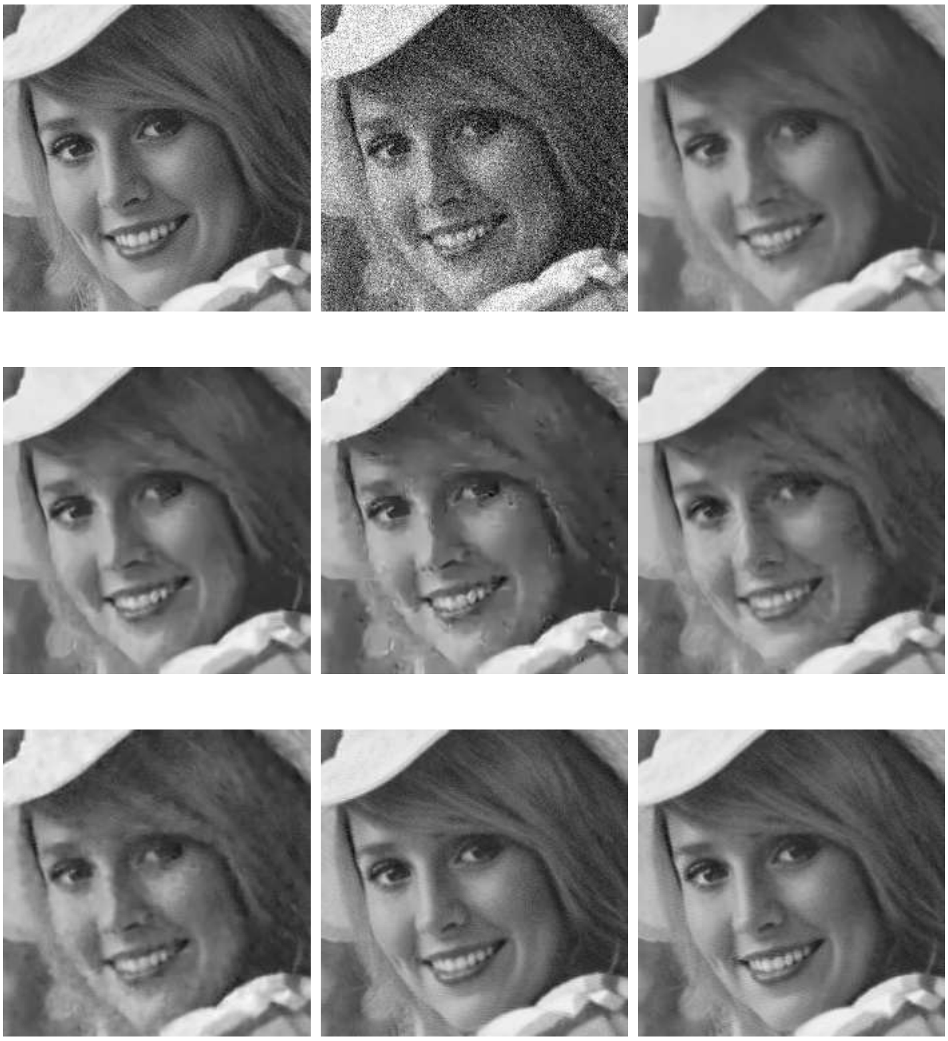


Figure 6.3: Denoising experiment on a part of natural image ('Elaine'). Left to right, top to bottom: Clean image, noisy image ($\sigma = 30$), LPG-PCA(28.81, 0.678), BM3D(29.17, 0.688), EPLL(28.77, 0.678), CSR(29.79, 0.682), OLPP(28.86, 0.692), 2D-OLPP(31.22, 0.774), 2D-OLPDP(31.50, 0.784).

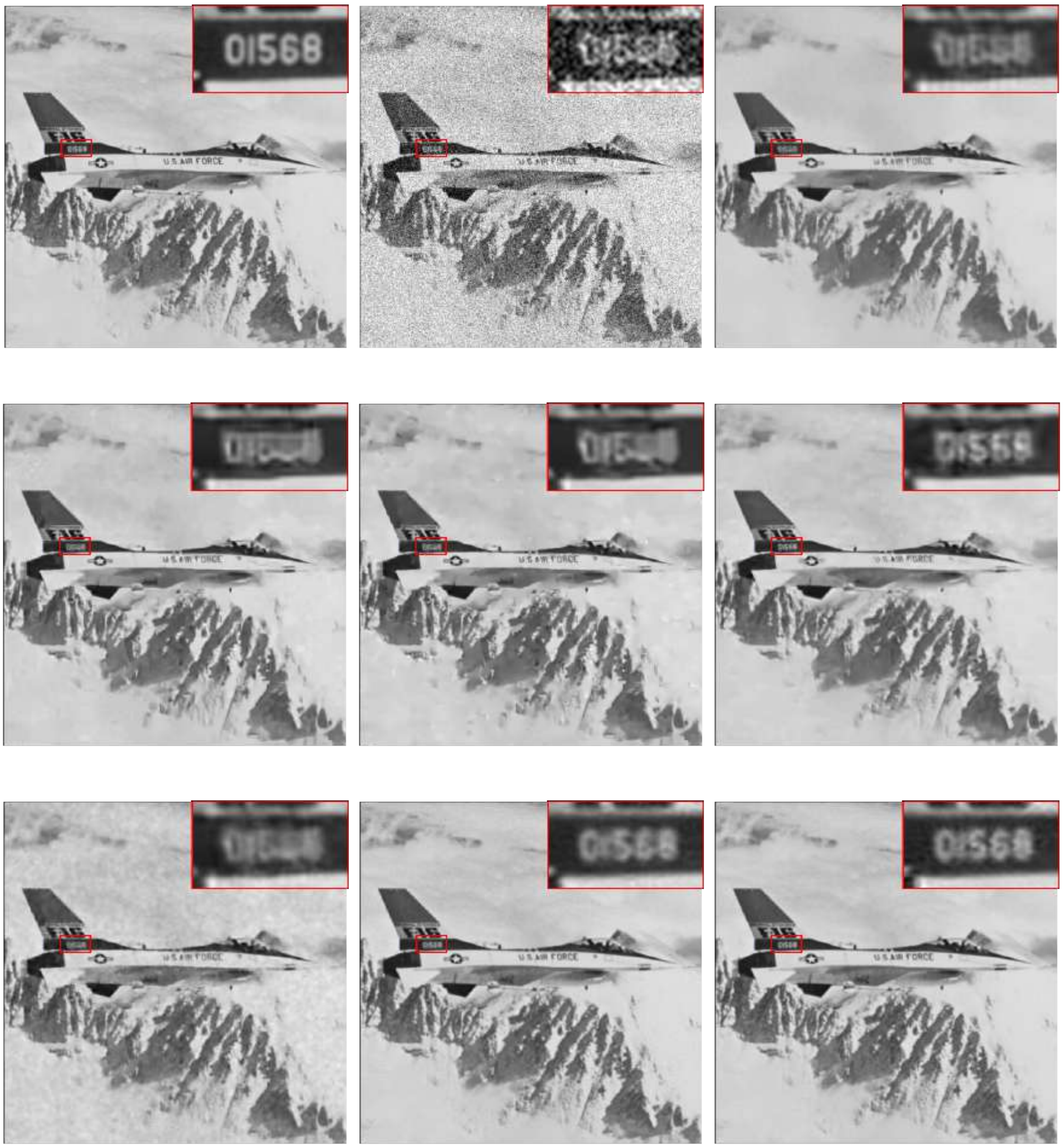
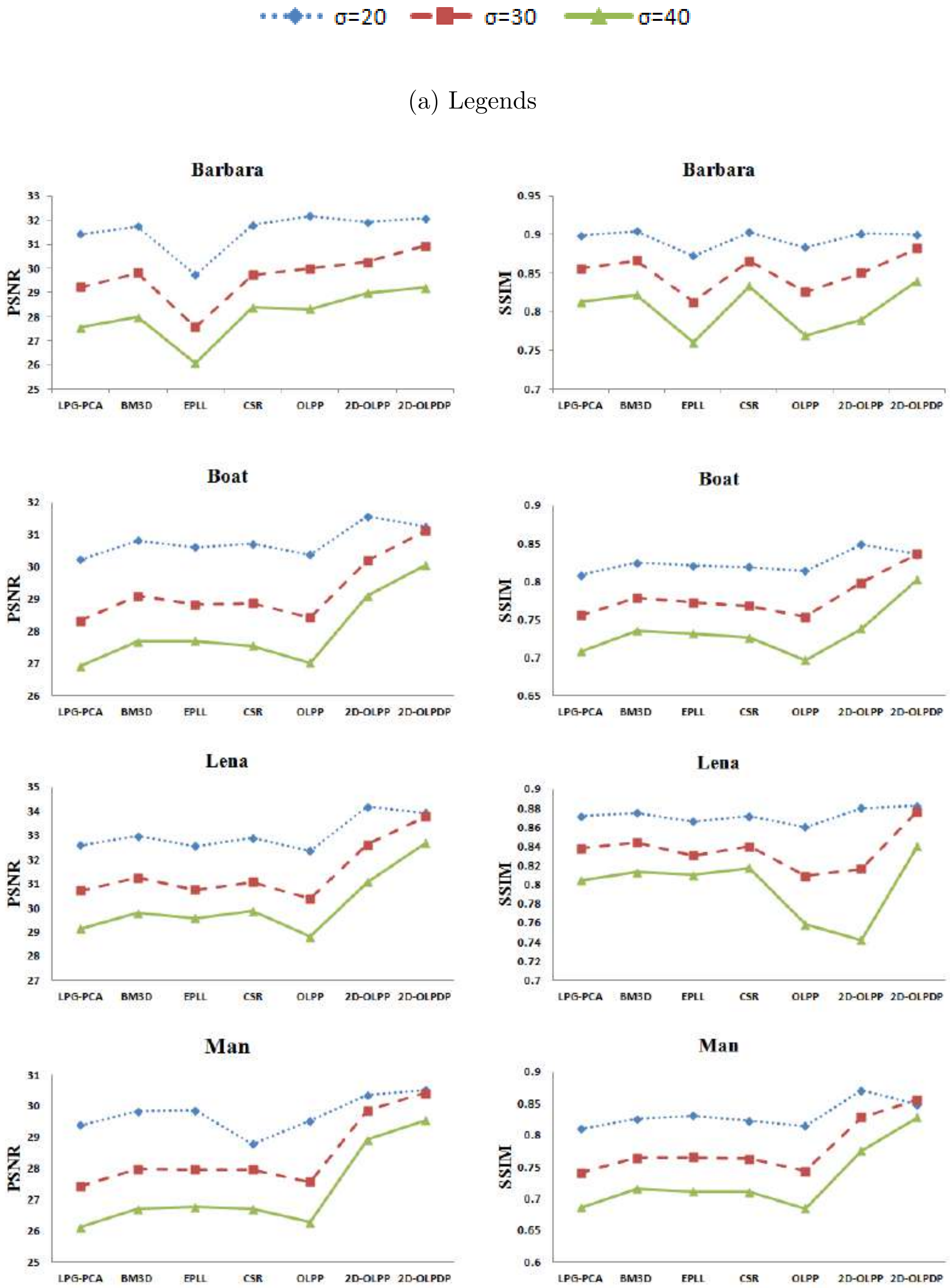
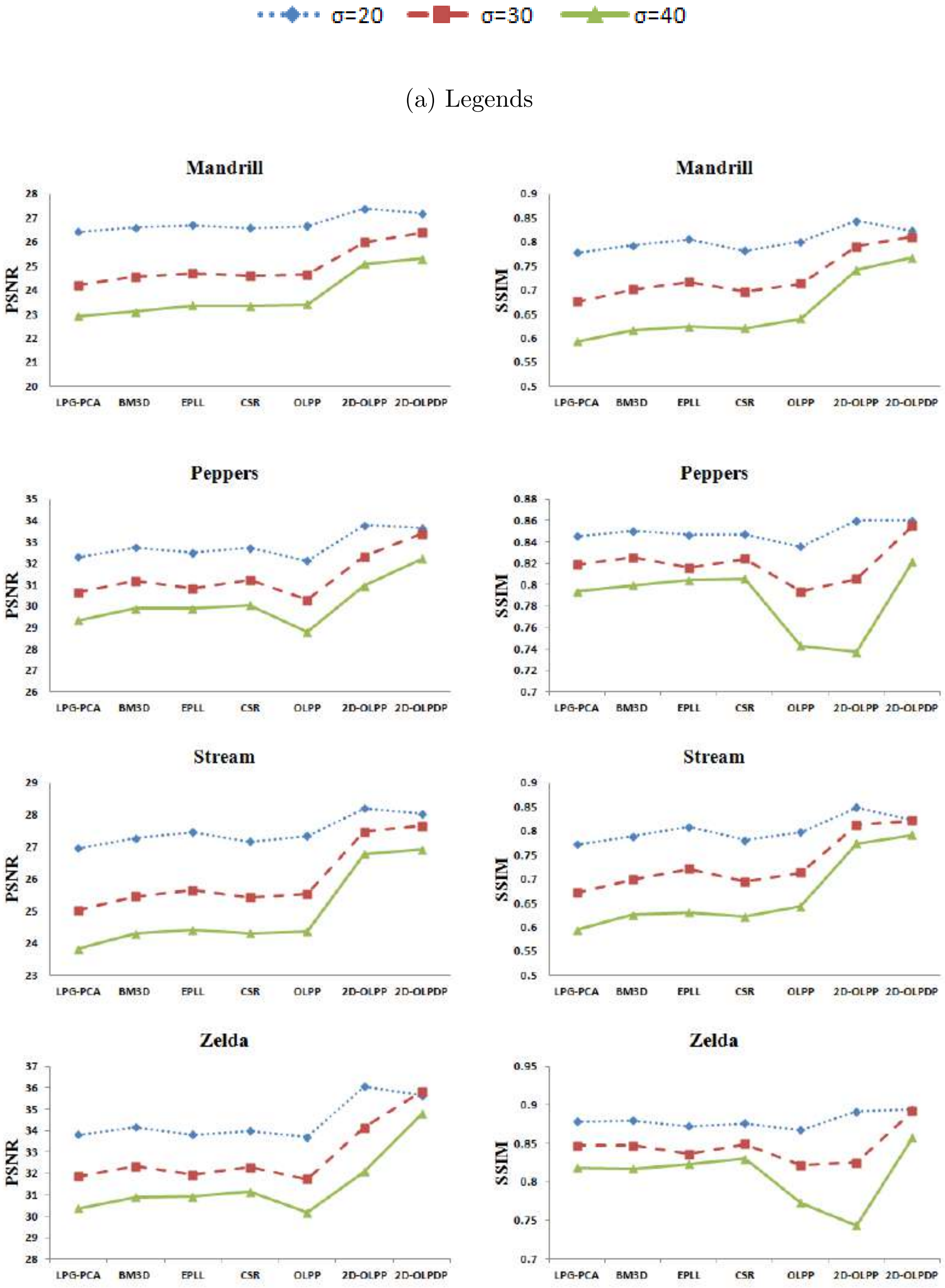


Figure 6.4: Denoising experiment a natural image ('Airplane'), a small part from the image is also zoomed to show the restoration of finer texture. Left to right, top to bottom: Clean image, noisy image ($\sigma = 40$), LPG-PCA(32.23, 0.894), BM3D(32.55, 0.898), EPLL(32.45, 0.897), CSR(32.68, 0.899), OLPP(32.18, 0.882), 2D-OLPP(32.42, 0.88), 2D-OLPDP(33.46, 0.895).



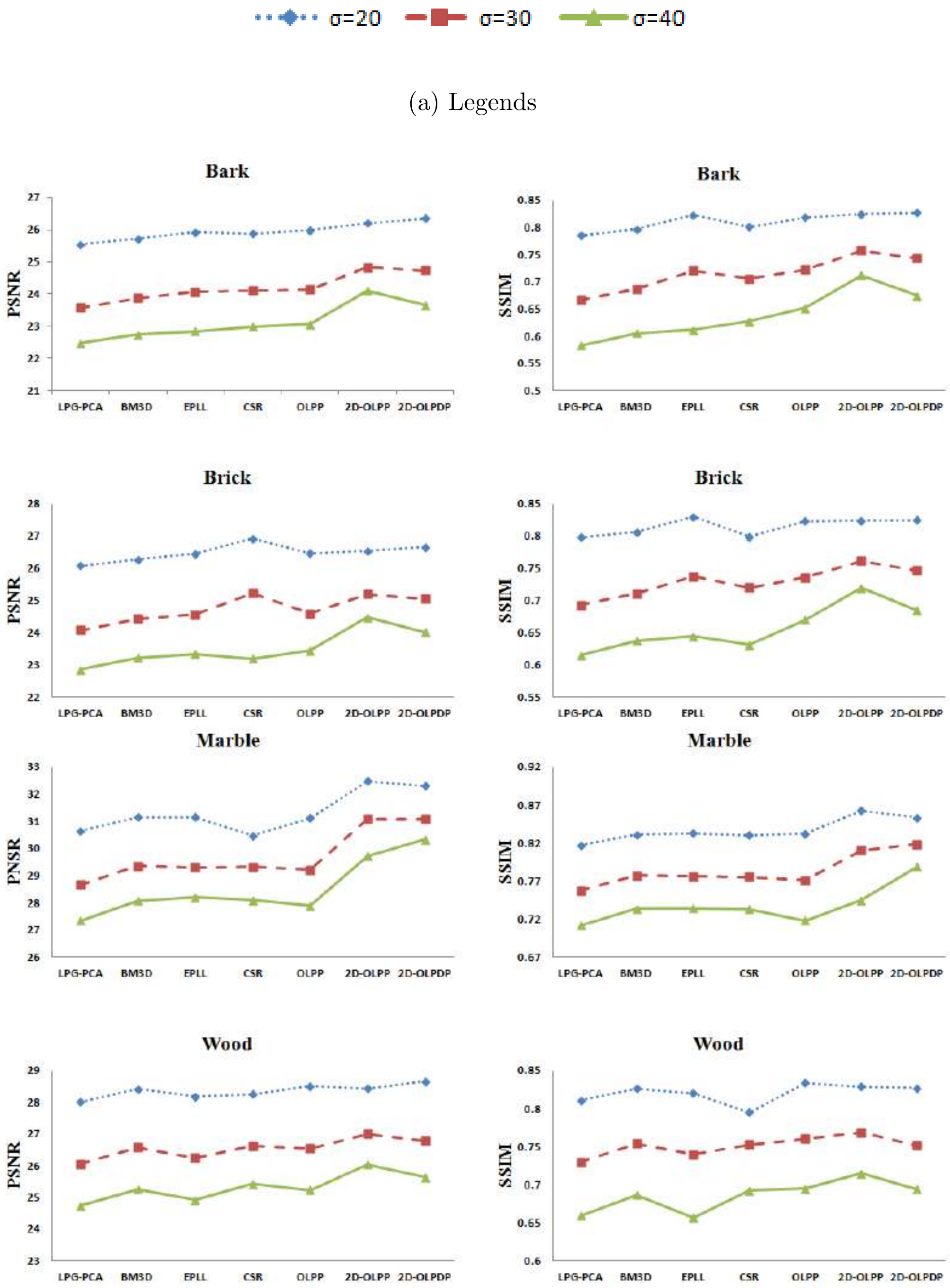
(a) Legends

Figure 6.5: Denoising results for some images from Lansel database of gray-scale natural images. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated in Figure 6.5(a)



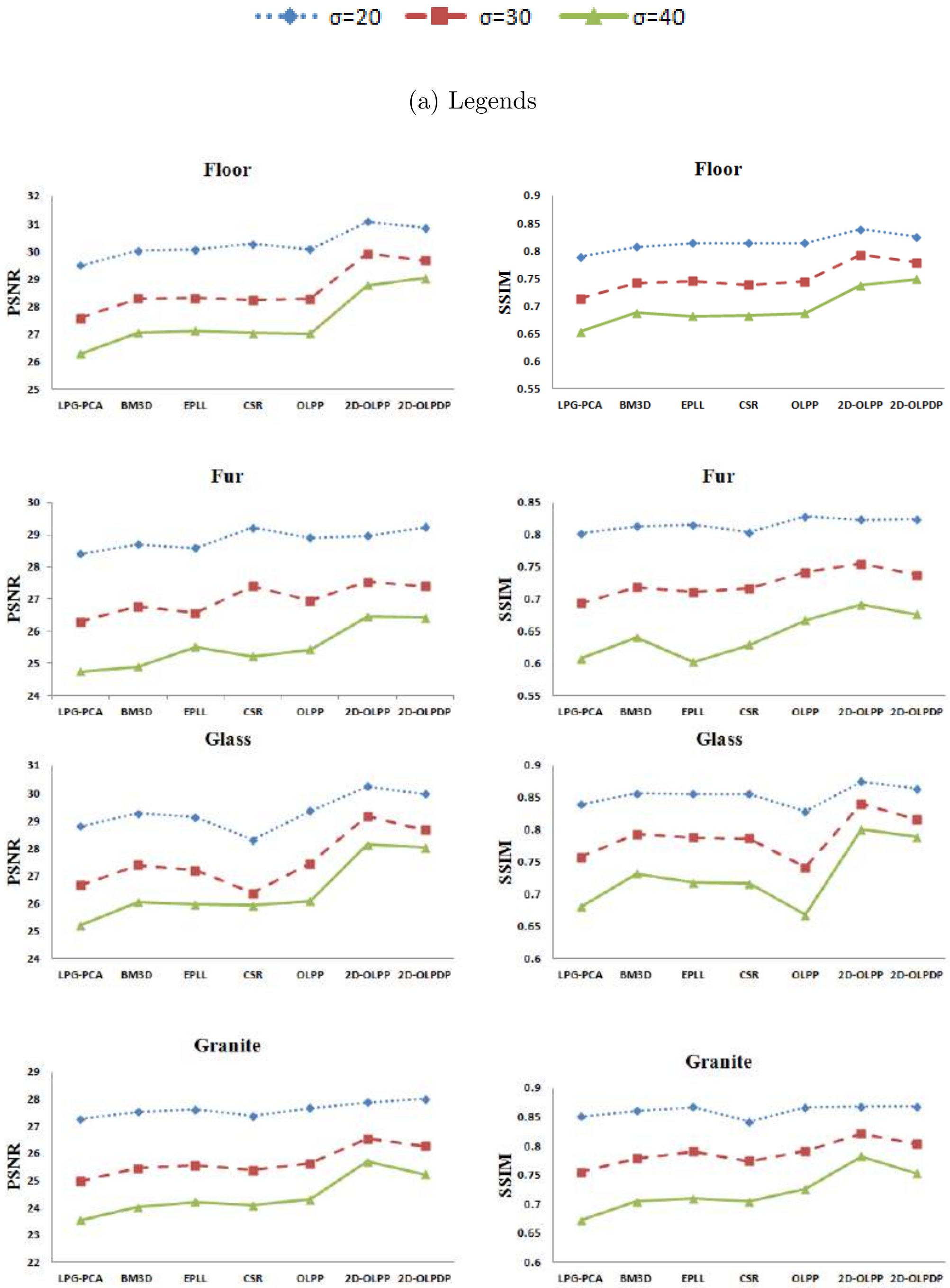
(a) Legends

Figure 6.6: Denoising results for some images from Lansel database of gray-scale natural images. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values for all the images. Results are shown for three noise levels as indicated in Figure 6.6(a)



(a) Legends

Figure 6.7: Denoising results for some images from UIUC Texture Database. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated in Figure 6.7(a)



(a) Legends

Figure 6.8: Denoising results for some image from UIUC Texture Database. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated in Figure 6.8(a)

Table 6.2: Average denoising results for all the images from gray-scale databases: the Lansel Database and the UIUC Texture database ($\sigma \in \{20, 30, 40\}$)

| Database | Denoising Approach | $\sigma = 20$ | | $\sigma = 30$ | | $\sigma = 40$ | |
|---|---|---|---|---|---|---|---|
| | | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| Lansel | LPG-PCA | 30.42 | 0.834 | 28.48 | 0.78 | 27.08 | 0.73 |
| | BM3D | 30.78 | 0.84 | 29.03 | 0.796 | 27.66 | 0.754 |
| | EPLL | 30.46 | 0.839 | 28.60 | 0.788 | 27.48 | 0.747 |
| | CSR | 30.69 | 0.841 | 28.90 | 0.793 | 27.76 | 0.755 |
| | OLPP | 30.45 | 0.838 | 28.52 | 0.787 | 27.20 | 0.743 |
| | 2D-OLPP | **31.68** | **0.872** | 30.65 | 0.821 | 29.43 | 0.762 |
| | 2D-OLPDP | 31.61 | 0.860 | **31.37** | **0.821** | **30.33** | **0.824** |
| UIUC Texture | LPG-PCA | 28.25 | 0.807 | 26.24 | 0.722 | 24.96 | 0.654 |
| | BM3D | 28.58 | 0.819 | 26.74 | 0.747 | 25.49 | 0.686 |
| | EPLL | 28.46 | 0.824 | 26.58 | 0.746 | 25.37 | 0.669 |
| | CSR | 28.54 | 0.817 | 26.75 | 0.748 | 25.57 | 0.683 |
| | OLPP | 28.67 | 0.829 | 26.79 | 0.757 | 25.56 | 0.696 |
| | 2D-OLPP | 29.20 | 0.843 | **27.91** | **0.789** | 26.92 | 0.739 |
| | 2D-OLPDP | **29.39** | **0.845** | 27.88 | 0.786 | **26.96** | **0.743** |

in Table 6.3. It can be observed that for Brodatz texture database, denoising results are comparable to the other approaches. For Kodak database, the PSNR values are comparable but the SSIM values seem to drop. Decline in SSIM values might be because of the roughness present in the denoised color images.

Table 6.3: Average denoising results for color image databases: the Kodak database and the Brodatz texture database ($\sigma \in \{30, 40, 50\}$)

| Database | Denoising Approach | $\sigma = 30$ | | $\sigma = 40$ | | $\sigma = 50$ | |
|---|---|---|---|---|---|---|---|
| | | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| Kodak | LPG-PCA | 28.32 | 0.72 | 26.63 | 0.628 | 25.36 | 0.556 |
| | BM3D | 30.20 | 0.839 | 28.51 | 0.783 | 27.90 | 0.761 |
| | OLPP | 29.51 | 0.825 | 28.16 | 0.776 | 27.10 | 0.732 |
| | 2D-OLPP | **31.34** | **0.842** | **29.84** | **0.806** | **28.85** | **0.790** |
| | 2D-OLPDP | 30.74 | 0.810 | 29.44 | 0.748 | 28.35 | 0.685 |
| Brodatz color texture | LPG-PCA | 26.62 | 0.816 | 24.92 | 0.750 | 23.52 | 0.680 |
| | BM3D | 27.28 | 0.866 | 25.67 | 0.817 | 25.18 | 0.795 |
| | OLPP | **27.82** | **0.876** | 26.53 | **0.841** | 25.40 | 0.803 |
| | 2D-OLPP | 27.55 | 0.868 | 26.37 | 0.834 | 25.61 | **0.808** |
| | 2D-OLPDP | 27.78 | 0.852 | **26.65** | 0.812 | **25.74** | 0.795 |

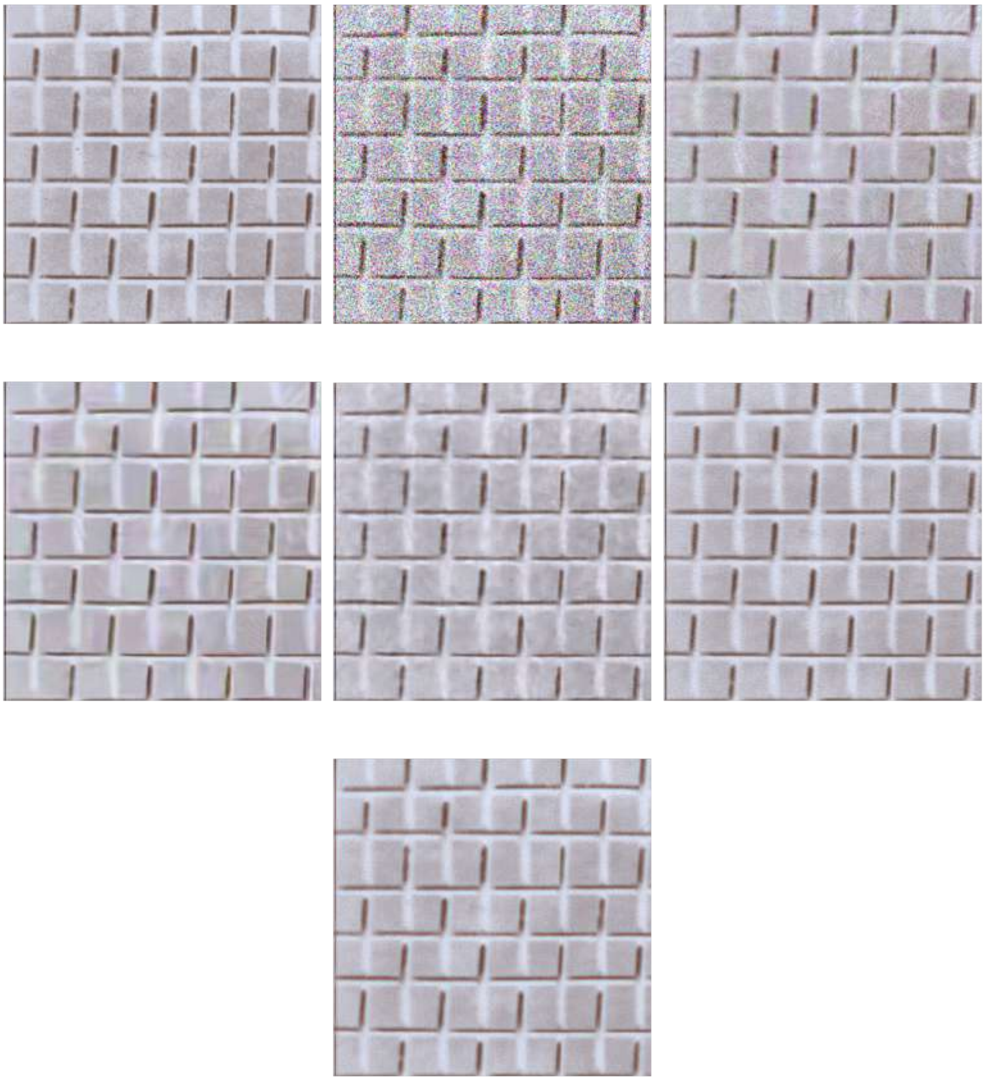
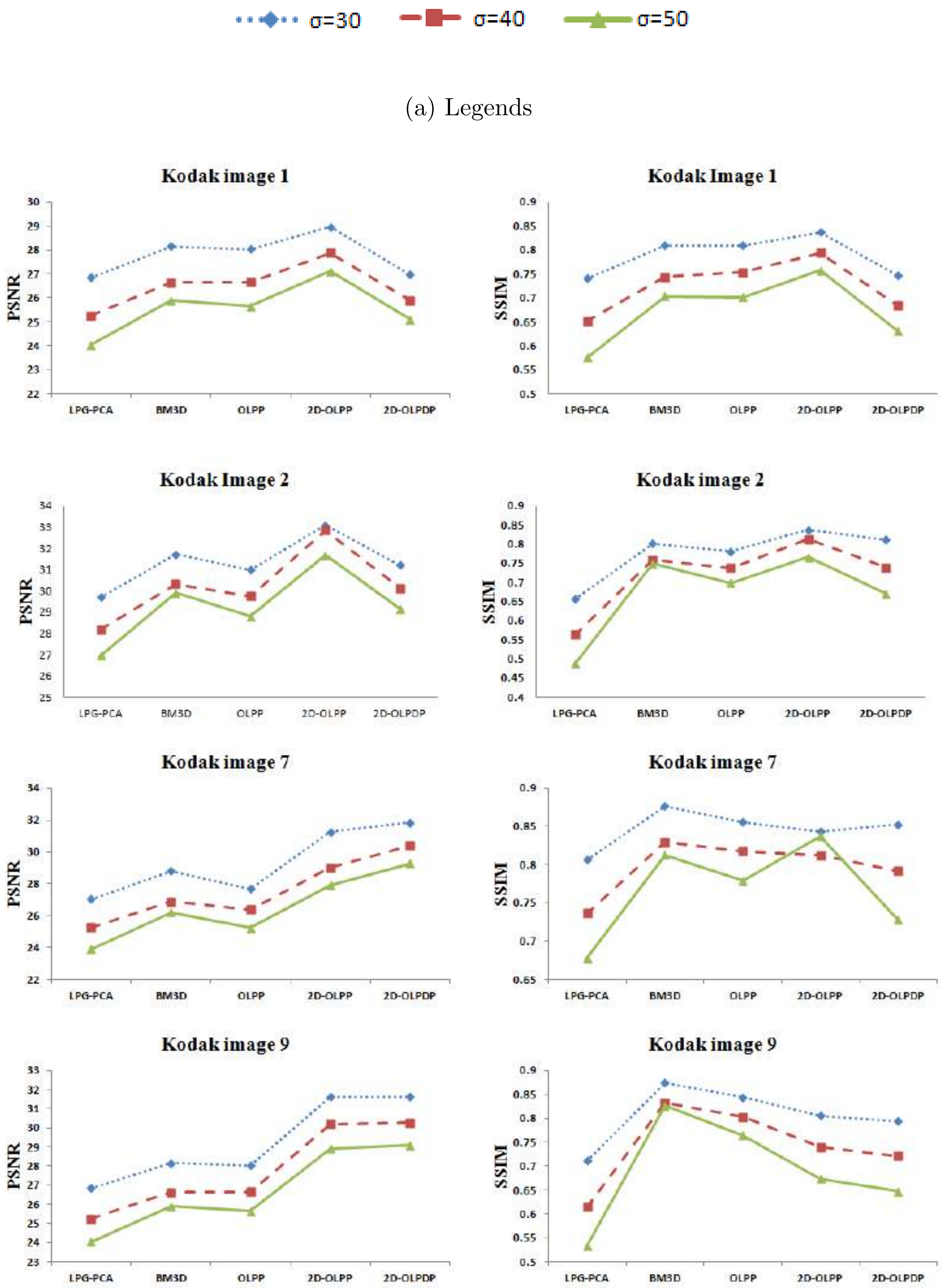
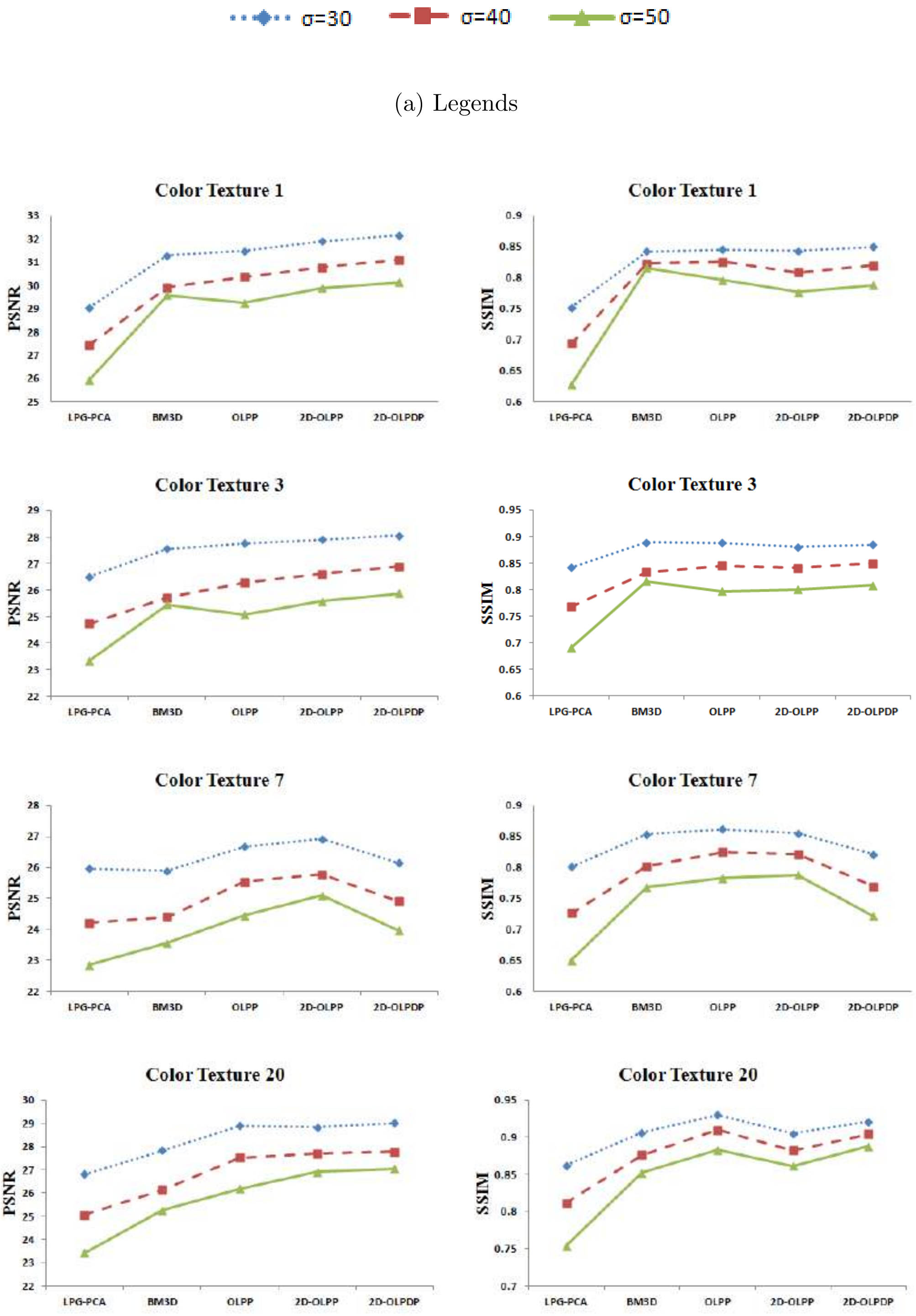


Figure 6.9: Denoising experiment on a color texture image. Left to right, top to bottom: Clean image, noisy image ($\sigma = 50$), LPG-PCA(25.92, 0.627), BM3D(29.58, 0.815), OLPP(29.25, 0.796), 2D-OLPP(29.89, 0.776), 2D-OLPDP(30.13, 0.787).



(a) Legends

Figure 6.10: Denoising results for some images from Kodak database of color natural images. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated in Figure 6.10(a).



(a) Legends

Figure 6.11: Denoising results for some images from Bordatz Color Texture database. Horizontal axis represent various methods of denoising and vertical axis represents PSNR (in db) and SSIM values. Results are shown for three noise levels as indicated in 6.11(a)

### 6.3.3 Method noise comparison

As discussed in Section 5.4.3, method noise or residual image is the most powerful evaluation measure for image denoising approaches. It shows the difference between a noisy image and its denoised version [41], [111] thus indicating the information from the image that has been eliminated along with the noise. Hence, method noise should resemble white noise. In addition to method noise for Lena image produced by all the competing denoising schemes reported in Figure 5.17 in Section 5.4.3, method noise using the current proposal 2D-OLPDP is also reported in Figure 6.12. It can be observed that 2D-OLPP and 2D-OLPDP both do not eliminate structural information while denoising and the residual image looks like white noise.

Method noise for a portion of image from 'Goldhill' from the Lansel database using various image denoising approaches is also reported in Figure 6.13. As in case of 'Lena', most of the prominent edges of the 'Goldhill' image have been removed in the method noise produced using LPG-PCA. BM3D, EPLL and CSR perform better in terms of preserving the edges of the original image. The residual images produced from the denoised images using OLPP and 2D-OLPP have very little structure specially window portions from the input image visible. All other parts of the residual image visible look similar to white noise. Method noise produced using 2D-OLPDP resembles noise and no structural information can be observed.

### 6.3.4 Comparison with denoising bound

As discussed at various places during the course of this thesis, image denoising is an age old problem and research has been carried out for decades to eliminate noise and obtain noise free images. With the extent of research carried out in this domain, a pertinent question arises that whether there is a theoretical limit to performance of image denoising approaches [115]? A lower bound on the Mean Square Error (MSE) of the denoised result according to the noise level of the image to be denoised has been suggested in [115]. Given the noise free original image, the proposed approach computes how well a noisy image can be denoised in terms of MSE. The image is first clustered into geometrically heterogeneous patches and analysis of denoising performance is carried out independently for each such cluster of similar patches. The MSE bounds for each of these clusters are aggregated to obtain MSE lower bound for the entire image. It has been shown that despite the phenomenal recent progress in the quality of denoising algorithms, some room for improvement still remains for a wide class of general images, and at certain signal-to-noise levels. Here, we have compared MSE obtained using various state of the art denoising approaches with the

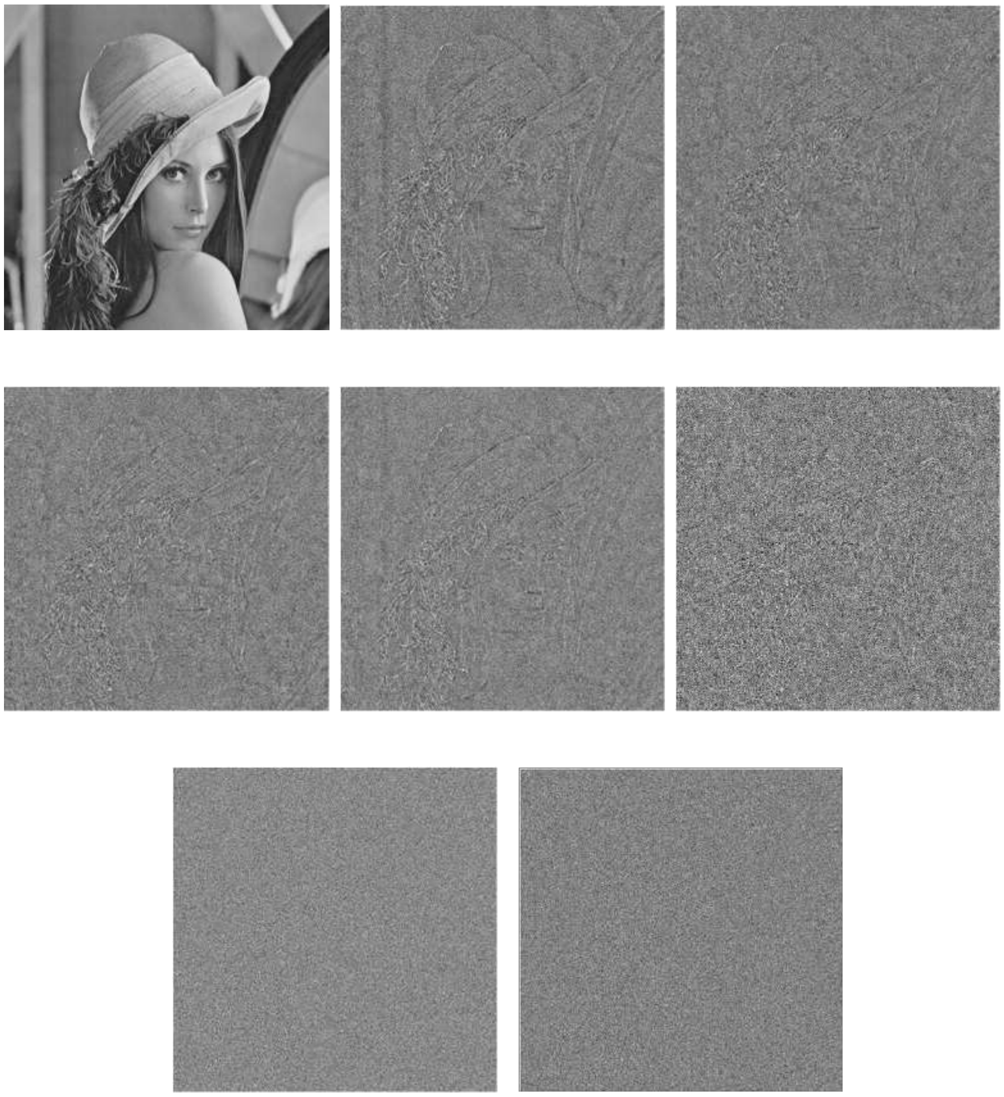


Figure 6.12: Method noise comparison on a natural image ('Lena'). Left to right, top to botom: Clean image, LPG-PCA, BM3D, EPLL, CSR, OLPP, 2D-OLPP and 2D-OLPDP.

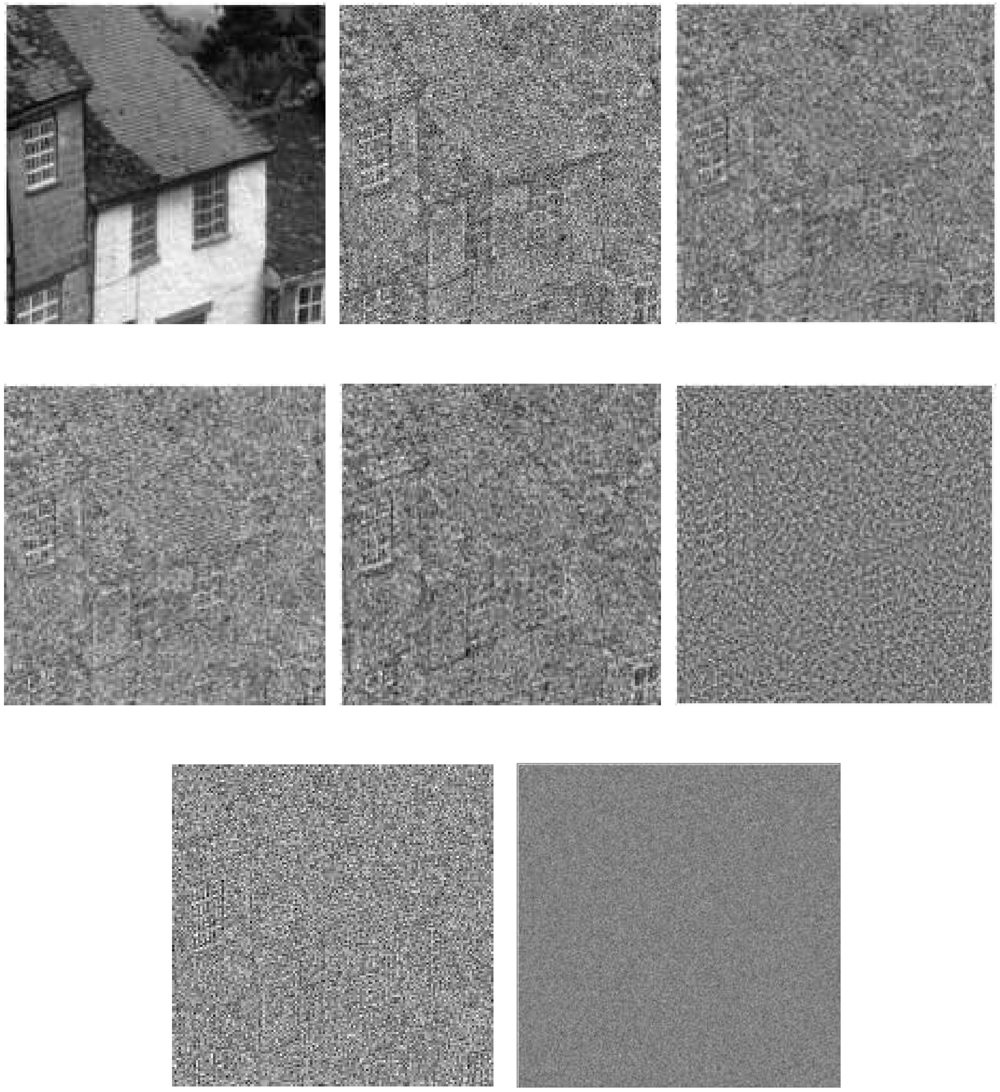


Figure 6.13: Method noise comparison on a natural image (‘Goldhill’). Left to right, top to bottom: Clean image, LPG-PCA, BM3D, EPLL, CSR, OLPP, 2D-OLPP and 2D-OLPDP. $N$: Number of patches, $l \times l$: size of a patch, $T_s$: average time required to search similar patches of the reference patch, $P$: average number of similar patches, $NT_{L_s}$: time taken to generate the similarity matrix $\mathbf{S}$ for $N$ patches, $NT_{L_d}$: time taken to generate the dissimilarity matrix $\mathbf{D}$ for $N$ patches.

MSE lower bound on 'Lena' and 'Barbara' images. The MSE values using various denoising approaches along with the bound calculated using the available online code with author specified parameters are reported in Table 6.4. It can be observed that the mean square error obtained using the 2D-OLPDP based denoising approach is very close to the MSE bounds for all the noise levels and are considerably less than that of other state of the art image denoising methods.

Table 6.4: The lower bounds for Mean Square Error (MSE) of the denoised image on three noise levels ($\sigma \in \{20, 30, 40\}$) on 'Lena' and 'Barbara' image along with the MSE values produced using different image denoising approaches.

| Image | Denoising Approach | $\sigma = 20$ | $\sigma = 30$ | $\sigma = 40$ |
|---|---|---|---|---|
| Lena | MSE Bound | **14.89** | **24.60** | **34.25** |
| | LPG-PCA | 35.70 | 55.26 | 79.40 |
| | BM3D | 32.84 | 48.81 | 68.30 |
| | EPLL | 36.18 | 54.74 | 71.91 |
| | CSR | 33.35 | 50.90 | 67.19 |
| | OLPP | 37.78 | 59.67 | 85.78 |
| | 2D-OLPP | 24.74 | 35.61 | 50.83 |
| | 2D-OLPDP | 19.71 | 27.34 | 35.22 |
| Barbara | MSE Bound | **37.20** | **63.68** | **89.93** |
| | LPG-PCA | 46.92 | 77.94 | 114.43 |
| | BM3D | 43.68 | 68.08 | 103.66 |
| | EPLL | 69.30 | 113.91 | 160.60 |
| | CSR | 42.87 | 69.17 | 94.45 |
| | OLPP | 54.54 | 89.74 | 130.24 |
| | 2D-OLPP | 41.80 | 67.20 | 93.66 |
| | 2D-OLPDP | 42.00 | 66.50 | 92.12 |

## 6.4 Conclusion

The proposed domain transformation technique considers both similarity and dissimilarity between the patches extracted from the image while learning the basis. The aim is to learn the basis such that in the projection space, structurally similar patches get projected close by whereas dissimilar patches far apart from each other thus achieving better clustering of the patches. Due to automatic clustering of patches, only one basis for the image to be denoised are adequate. Two dimensional data processing also assures faster basis learning. Image denoising experiments are carried out on some of the benchmark databases and compared with state of the art techniques as well as OLPP and 2D-OLPP based methods which work on the similar principle.

Projection of patches in the 2D-OLPDP domain shows distinct clusters of patches from the image. Image denoising performance of the proposed approach is superior to that of the state of the art approaches. The results are on par with the 2D-OLPP based denoising, however, smaller structural details get restored very well using the current proposal.

# Chapter 7

# Conclusions and Future Work

Advances in imaging technologies have resulted in high resolution image acquisition requiring more pixels to represent the image. Handling such high dimensions of images for various image processing, pattern recognition related applications is very challenging, hence dimensionality reduction is applied before processing the images. This thesis emphasizes on the dimensionality reduction techniques based on local structure preservation, which ensures mapping of similar data points close by in the lower dimensional subspace.

This work is an attempt to develop more robust and efficient dimensionality reduction techniques for image data. Discriminant information is combined with similarity preservation property to achieve better separation between distinct clusters of data points. Orthogonalization and matricization procedures are formulated to allow faster reconstruction and reduced computational complexity. Applications such as face recognition and image denoising are addressed using the proposed dimensionality reduction variants.

The work accomplished in the thesis is summarized in this chapter followed by some discussion on possibilities of extension in future.

## 7.1 Contributions

Image data typically consists of very high dimensions and processing such large data is a challenge. Dimensionality reduction is the technique of representing the data with much less dimensions without losing the information carried by it. Many a times, local neighborhood structure of the data reveals the underlying manifold of the data in a much better way that the global euclidean structure. In this work, we have explored similarity preserving dimensionality reduction methods and applied them to various applications.

Preserving the similarity present in data in the reduced dimensions helps in capturing the data non-linearity. Data classification and recognition related applications also require discriminant property of the data to be retained. Dimensionality reduction approaches that take care of both, local information preservation and discriminant property of the data while learning the basis are suggested. Extended Locality Preserving Projection with Modified Distance (ESLPP-MD), a supervised approach utilizes the class labels of the data points to achieve better class discrimination by shrinking or diverging the pairwise distances depending upon whether they belong to same class or not. ESLPP-MD achieves class discrimination by manipulating the distances, which clusters the data very well, but may suffer ambiguities for unknown data points. A more robust technique is proposed in which the objective function itself ensures maximum separation between classes by projecting similar data points close by whereas dissimilar data points far apart in the learnt Locality Preserving Discriminant Projection (LPDP) space.

Projection and recognition experiments performed on face data are encouraging and show enhanced class discrimination ability with only a few strongest dimensions. Improvement in face recognition performance on some challenging face databases has been observed. Kernelization of these linear approaches aims at further exploring non-linear structure of data, which is supported by improved recognition accuracies over the non-kernelized counterparts.

Non-orthogonality of dimensionality reduction approaches restricts their use for applications that require reconstruction of data back to the original space. Orthogonal ELPP imposes a prior belief that similar data points should map onto similar transform coefficients, and explicitly learns an orthonormal basis to optimize for this criterion. This orthogonal formulation for basis learning makes it suitable for image denoising. Excellent performance of this technique, at par with or surpassing the state of the art approaches, especially with regard to texture preservation is achieved. This is so, even when we learn a single basis for large-sized windows from the image. This leads to an edge of the present method over state of the art techniques where bases are learnt for each patch. However, there is considerable scope for improving the performance of this technique in the homogeneous regions of an image.

All higher dimensional data points used for learning are required to be converted into vector format due to vectorized nature of LPP, OLPP and other approaches, which not only increases the computational complexity, but also causes loss of spatial neighborhood information. Two Dimensional Orthogonal Locality Preserving Projection (2D-OLPP) processes data in matrix format making it faster and efficient. A global basis is sufficient for entire image to be denoised. The approach appears to be comparable with the forerunner approaches of image denoising. Finer structural details as well as homogeneous regions of the denoised image are restored well, both visually as well as quantitatively. Discriminant information is always advantageous for data clustering and classification. Two Dimensional Orthogonal extension of LPDP is formulated and applied for image denoising. Similar patch clustering within the image has improved resulting in enhanced denoising performance.

The summery of contributions of the thesis is listed below:

- Local information preservation and class separation properties are combined in the proposed dimensionality reduction approaches. Enhanced class separation results in superior projection and face recognition results.

- Face images having vast expression, illumination and pose changes may not necessarily lie on a linear manifold. Kernel functions are used to map the data in feature space to explore the non-linear manifold of the data before pursuing dimensionality reduction leading to improved recognition accuracies over the non-kernelized counterparts.

- Orthogonalization of Extended LPP basis is suggested assuring faster and efficient data reconstruction. Basis thus learnt are orthogonal in nature and project structurally similar data points close to each other, thereby makes it suitable for image denoising. Patches extracted from the noisy image are treated as data points and a global OLPP basis is learnt for the entire image. Image denoising performance of the proposed approach is on par with or surpasses the state of the art methods, especially with regard to texture preservation.

- Two Dimensional OLPP processes data in matrix format directly, which not only preserves spatial neighborhood information, but also improves the computational efficiency considerably. The results obtained are very encouraging and are comparable with the state of the art approaches of image denoising. Finer structural details as well as homogeneous regions of the denoised image are restored well, both visually as well as quantitatively.

- Two Dimensional Orthogonal LPDP (2D-OLPDP) takes into account the dissimilarity between patches along with the similarity while learning the basis directly. Discriminant information improves patch clustering within image resulting in enhanced denoising performance.

## 7.2 Overall conclusion

An attempt is made in this thesis to build dimensionality reduction techniques based on similarity preservation. Neighborhood information, revealing the local structure of the data is utilized while learning the basis for domain transformation. These techniques are suitable for applications where local information preservation is desirable. Similarity preserving dimensionality reduction has been applied to face recognition and image denoising in this work.

In addition to local structure, discriminant information, if considered, can play pivotal role in achieving superior class separation. Extended Supervised Locality Preserving Projection with Modified Distance (ESLPP-MD) changes distance between data points based on the class label information to achieve better class separation. Enhanced face recognition performance as compared to the other locality preserving approaches is observed. Manipulating distances in this manner assures better class discrimination ability of training data points but unknown data points may face difficulty in classification. To overcome this issue, Locality Preserving Discriminant Projection (LPDP) is proposed with an objective function which itself takes care of projecting similar data points close by and dissimilar data points from different classes far apart. This method seems to be performing on par with or surpassing ESLPP-MD and other locality preserving as well as discriminant approaches. Kernel based variants of these approaches improve face recognition accuracies even further.

Dimensionality reduction techniques do not always guarantee enhanced performance with increase in number of dimensions used to represent the data. This may be due to redundant information getting added with more number of dimensions. However, in this thesis, we have not studied the redundancy pattern of the dimensions and experiments have also not been carried out for the same. Also, the emphasis of this thesis is on dimensionality reduction approaches and their usability for various challenges. For face recognition, proposed dimensionality reduction techniques are directly applied on the raw face images from different databases and nearest neighbor (rank one) classifier is used to classify the test image in the reduced dimensions. Most of the benchmark face recognition techniques are designed with various pre-processing stages along with more robust classifiers to deal with the face images captured in unconstrained environment. As main scope of the thesis is not to resolve face recognition issues, but to devise more robust dimensionality reduction approaches, study of benchmark face recognition systems has not been carried out.

The basis obtained using dimensionality reduction approaches based on LPP are non-orthogonal. Orthogonalization of Extended LPP basis (OLPP) is suggested in Chapter 4. Similarity preserving property of OLPP makes it suitable for image denoising which seeks to explore non-local self similarity between the noisy image patches. Due to locality preserving property, similar patches get clustered when projected on OLPP basis, thus a global basis is sufficient for the entire image. Noise elimination is performed in the transformed domain. OLPP based image denoising technique demonstrates excellent performance, especially with regard to texture preservation, on a large set of experiments across varied noise levels. The finer details present in the image are restored very efficiently. However, we have noticed that there is considerable scope of improving performance of this technique as it undersmooths the homogeneous regions of the image.

Two Dimensional OLPP (2D-OLPP) directly processes the images in matrix format, thus reducing the computational complexity of the approach considerably. It also retains the spatial neighborhood information which gets lost due to vectorization of data points in all the previously discussed approaches. 2D-OLPP based image denoising shows significant improvement in the performance of denoising natural images as homogeneous regions are restored well along with the structural details. The results are encouraging and comparable to the state of the art methods of image denoising. In addition to preserving the neighborhood information, Two dimensional Orthogonal LPDP (2D-OLPDP) also takes into account the dissimilarity between the noisy image patches to attain superior clustering of the patches in the projection domain. The results of 2D-OLPDP based denoising are on par with that of 2D-OLPP, however, smaller structural details get restored very well using the current proposal.

## 7.3 Future Work

The scope of this thesis lies in proposing a robust and efficient similarity preserving dimensionality reduction technique. Face recognition is taken up as an application to showcase the efficiency of the proposal. In this work, dimensionality reduction techniques are applied on raw face images as it is and the simplest classifier, i.e. nearest neighbor classifier is used to recognize the class of unknown data points (Chapter 2 and 3). A complete face recognition system consists of a combination of various pre-processing stages to take care of illumination, pose and other external environmental changes as well as more sophisticated classifiers which boost the performance resulting in higher accuracy. Research can be further extended to build a complete face recognition system.

Another application that is explored in this thesis is Image Denoising. The locality preserving property of the learnt basis takes care of projecting similar patches close to each other in the projection space thus clustering structurally similar patches automatically (Chapter 4, 5, 6). To ensure such grouping, weights in the similarity matrix are assigned based on the distance between the patches. Patch similarity measure proposed by Rajwade et al. [96] is used throughout this thesis which is based on the Euclidean distance between patches. Patch similarity measure considering correlations and structural patterns between the patches can be developed. This could result in enhanced clustering of patches further improving the image denoising performance.

In Chapter 5, dimensionality reduction technique which processes the images in matrix format is proposed. In this 2D-OLPP approach, the data is no longer required to be converted to vector format before processing. This technique can be further extended for tensorized data i.e. to deal with the data in 3 or more dimensional format.

Though the dimensionality reduction techniques proposed during the course of this thesis are applied for face recognition and image denoising, their usage is not limited to them. Applicability of the proposals can be extended for numerous applications. Other image restoration techniques such as image deblurring, inpainting, super resolution etc. can be solved using the locality preserving dimensionality reduction approaches. It is to be noted that patch based transformed domain approaches have already been applied for such applications [92], [91]. Hence, similarity preserving dimensionality reduction approaches are expected to perform well for these applications.

An application that has been explored widely in last few decades is Content Based Image Retrieval (CBIR). As the proposed approaches have property of projecting similar data points close to each other in the projection space, it could well be used for image retrieval task. The images having similar structures are expected to get projected nearer thus making image retrieval possible. Instead of directly using the images for retrieval, patch based similarity within different images can be investigated to obtain more robustness. The modern CBIR design requires similarity measures invariant of rotation, translation, shift and illumination. Such variants of some of the dimensionality reduction techniques i.e. PCA and LDA have already been proposed in literature. More robust versions of the current proposals of LPP, to handle such variations, can be explored in future.

Dimensionality reduction approaches, that have been discussed so far, have an objective function to be optimized to obtain the basis/transformation matrix that is used to project the data in the reduced dimensional space, mainly preserving desired properties of the raw high dimensional data. Depending on the objective function, the basis are obtained from the data in a pre-determined manner, thus, training and learning from the data is not performed as in case of neural network based approaches.

Deep learning has emerged as a very successful area of research, in which the deep network gets trained from the data itself to obtain the desired output. Many different problems/applications are being tackled using the deep networks, dimensionality reduction being one of them. Auto-encoders are used as dimensionality reduction tools, wherein the loss function at the output layer is made similar to that of the objective functions of the well known dimensionality reduction approaches, letting neural network learn the basis. Hence, depending upon the loss function, the basis are learned using deep network architecture, which can give freedom of learning the non-linearity of the data as well. In particular, the objective function of LPP, which assures projection of similar data points close by in the LPP domain, can be used as the loss function for training the deep network.

Another direction of work could be combining the dimensionality reduction and deep learning frameworks to boost the performance. One way could be, feeding the reduced dimensional data using the proposed dimensionality reduction techniques as input to the deep learning framework, instead of the high dimensional raw data. The basis learned using dimensionality reduction approaches are adapted according to the training data, thus another way of combining dimensionality reduction with deep learning could be using the trained basis as the filters to the initial input layers of the network. Hence, instead of learning the filters to be used for CNN, the basis obtained using the proposed approach can directly be used as filters.

# REFERENCES

[1] W. Yu, X. Teng, and C. Liu. Face recognition using discriminant locality preserving projections. *Image and Vision Computing*, 24(3):239 – 248, 2006.

[2] GF. Lu, Z. Lin, and Z. Jin. Face recognition using discriminant locality preserving projections based on maximum margin criterion. *Pattern Recognition*, 43(10):3572–3579, 2010.

[3] X. Ma, Y. Tan, Y. Zhao, and H. Tian. Face recognition based on maximizing margin and discriminant locality preserving projection. *Journal of Information and Computational Science*, 7(7):1551–1559, 2010.

[4] J. Gui, W. Jia, L. Zhu, SL. Wang, and DS. Huang. Locality preserving discriminant projections for face and palmprint recognition. *Neurocomputing*, 73(13):2696–2707, 2010.

[5] H. Wang, S. Chen, Z. Hu, and W. Zheng. Locality-preserved maximum information projection. *Neural Networks, IEEE Transactions on*, 19(4):571–585, 2008.

[6] J. Yang, D. Zhang, A. Frangi, and J.Yang. Two-dimensional pca: A new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 26(1):131–137, 2004.

[7] M. Li and B. Yuan. 2d-lda: A statistical linear discriminant analysis for image matrix. *Pattern Recognition Letters*, 26(5):527–532, 2005.

[8] H. Zhang, Q. J. Wu, T. Chow, and M. Zhao. A two-dimensional neighborhood preserving projection for appearance-based face recognition. *Pattern Recognition*, 45(5):1866–1876, 2012.

[9] S. Chen, H. Zhao, M. Kong, and B. Luo. 2d-lpp: A two-dimensional extension of locality preserving projections. *Elsevier Neurocomputing*, 70:912–921, 2007.

[10] J. Chen and Z. Ma. Locally linear embedding : A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(7):985–1008, 2011.

[11] I. K. Fodor. A survey of dimension reduction techniques. Technical report, Lawrence Livermore National Lab., CA (US), 2002.

[12] D. L. Donoho. High-dimensional data analysis: the curses and blessings of dimensionality. In *American Mathematical Society Conf. Math Challenges of the 21st Century*, 2000.

[13] K. Fukunaga. *Introduction to statistical pattern recognition.* Academic press, 2013.

[14] L. O. Jimenez and D. Landgrebe. Supervised classification in high-dimensional space: geometrical, statistical, and asymptotical properties of multivariate data. *IEEE Transactions on Systems, Man, and Cybernetics*, 28(1):39–54, 1998.

[15] Laurens JP van der Maaten, Eric O Postma, and H Jaap van den Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10(1-41):66–71, 2009.

[16] I.T. Jolliffe. *Principal Component Analysis.* Springer, Berlin, New York, 2002.

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

[18] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriengman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 19(13):711–720, 1997.

[19] A. Hyvrinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[20] J.B. Tenenbaum, V.D. Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[21] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[22] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computing*, 15:1373–1396, 2003.

[23] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, volume 14, pages 585–591, 2001.

[24] X. He, D. Cai, S. Yan, and H.J Zhang. Neighborhood preserving embedding. In *IEEE International Conference on Computer Vision*, volume 2, pages 1208–1213, 2005.

[25] E. Kokiopoulou and Y. Saad. Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2143–2156, 2007.

[26] X. He and P. Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems*, 2003.

[27] L. Qiao, S. Chen, and X. Tan. Sparsity preserving projections with applications to face recognition. *Pattern Recognition*, 43(1):331 – 341, 2010.

[28] D. Arpit, I. Nwogu, and V. Govindaraju. Dimensionality reduction with subspace structure preservation. In *Advances in Neural Information Processing Systems*, pages 712–720, 2014.

[29] Xiaogang Wang and Xiaoou Tang. Unified subspace analysis for face recognition. In *IEEE International Conference on Computer Vision*, pages 679–686, 2003.

[30] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using lapacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, March 2005.

[31] F. R. K. Chung. *Spectral Graph Theory.* American Mathematical Society, 1997.

[32] D. Cai, X. He, J. Han, and H. Zhang. Orthogonal laplacianfaces for face recognition. *IEEE Transactions on Image Processing*, pages 3608–3614, 2006.

[33] D.L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41:613–627, 1995.

[34] D. Darian Muresan and Thomas W. Parks. Adaptive principal components and image denoising. In *ICIP*, pages 101–104, 2003.

[35] K. Dabov, A. Foi, V. Katkovnik, and K. O. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[36] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *International Conference on Computer Vision*, 2009.

[37] W. Dong, X. Li, L. Zhang, and G. Shi. Sparsity-based image denoising via dictionary learning and structural clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 457–464, 2011.

[38] X. Jiang. Linear subspace learning-based dimensionality reduction. *Signal Processing Magazine, IEEE*, 28(2):16–26, 2011.

[39] R. R. Coifman and D. L. Donoho. Translation-invariant de-noising. *Wavelets and Statistics*, pages 125–150, 1995.

[40] D. L. Donoho and J. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[41] A. Buades, B. Coll, and J. Morel. A review of denoising algorithms, with a new one. *Multiscale Modeling and Simulation*, 4(2):490–530, 2005.

[42] A. Buades, B. Coll, and J. Morel. Nonlocal image and movie denoising. *International Journal of Computer Vision*, 76(2):123–139, 2008.

[43] P. O. Hoyer. *Independent Component Analysis in Image Denoising.* PhD thesis, Helsinki University of Technology, 1999.

[44] M. Elad and M. Aharon. Image denoising via learned dictionaries and sparse representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6, 2006.

[45] Y. Xu, A. Zhong, J. Yang, and D. Zhang. Lpp solution schemes for use with face recognition. *Pattern Recognition*, 43(12):4165–4176, 2010.

[46] J. Wei G. Yu, H. Peng and Q. Ma. Enhanced locality preserving projections using robust path based similarity. *Neurocomputing*, 71:598–605, 2011.

[47] F. Dornaika and A. Assoum. Enhanced and parameterless locality preserving projections for face recognition. *Neurocomputing*, 99:448–457, 2013.

[48] G. Shikkenawis and S. K. Mitra. Improving locality preserving projection for dimensionality reduction. In *Emerging Applications of Information Technology*, pages 161–164, 2012.

[49] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[50] Z. Liang, Y. Li, and P. Shi. A note on two-dimensional linear discriminant analysis. *Pattern Recognition Letters*, 29(16):2122 – 2128, 2008.

[51] Jieping Ye, Ravi Janardan, and Qi Li. Two-dimensional linear discriminant analysis. In *Advances in Neural Information Processing Systems*, pages 1569–1576, 2004.

[52] Z. Zhenga, F. Yanga, W. Tana, J. Jiaa, and J. Yangb. Gabor feature-based face recognition using supervised locality preserving projection. *Signal Processing*, 87:2473–2483, 2007.

[53] J. Lu and Y. P. Tan. Regularized locality preserving projections and its extensions for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (TSMC-B)*, 40(3):958–963, 2010.

[54] X. Liu C. Lu and W. Liu. Face recognition based on two dimensional locality preserving projections in frequency domain. *Neurocomputing*, 98(3):135–142, 2012.

[55] Mnist database of hand written digits. `http://yann.lecun.com/exdb/mnist/`.

[56] Interview video dataset. `http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/data/talingface/`.

[57] C. Zhang, F. Nie, and S. Xiang. A general kernelization framework for learning algorithms based on kernel pca. *Neurocomputing*, 73(4-6):959–967, 2010.

[58] B. Scholkopf, A. Smola, and K. R. Muller. Kernel principal component analysis. In *Artificial Neural Networks ICANN'97*, Lecture Notes in Computer Science, pages 583–588, 1997.

[59] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12:2385–2404, 2000.

[60] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, KR. Müller, Y-H Hu, J. Larsen, E. Wilson, and S. Douglas. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing*, pages 41–48, 1999.

[61] H. Lu J. Cheng, Q. Liu and Y. W. Chen. Supervised kernel locality preserving projections for face recognition. *Neurocomputing*, 67:443–449, 2005.

[62] M. A. Aizerman, E. A. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. In *Automation and Remote Control,*, number 25, pages 821–837, 1964.

[63] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[64] B. Schölkopf, C. Burges, V. Vapnik, F. Uthurusamy, et al. Extracting support data for a given task. In *First International Conference on Knowledge Discovery Data Mining (KDD-95)*, pages 252–257. AAAI Press, 1995.

[65] G. Dai and D. Y. Yeung. Nonlinear dimensionality reduction for classification using kernel weighted subspace method. In *IEEE International Conference on Image Processing*, volume 2, pages 838–841, 2005.

[66] K. W. Bowyer, K. Chang, and P. Flynn. A survey of approaches and challenges in 3d and multi-modal 3d+ 2d face recognition. *Computer vision and image understanding*, 101(1):1–15, 2006.

[67] L Torres. Is there any hope for face recognition? In *International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pages 21–23, 2004.

[68] I. Craw, D. Tock, and A. Bennet. Finding face features. In *European Conference on Computer Vision*, pages 92–96, 2004.

[69] S. Li and A. Jain. *Hand book of Face Recognition*. Springer, 2011.

[70] A. Yuille, D. Cohen, and P. Hallinan. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1989.

[71] C. Huang and C. Chen. Human facial feature extraction for face interpretation and recognition. *Pattern Recognition*, 25(12):1435–1444, 1992.

[72] K. Lam and H. Yan. Locating and extracting the eye in human face images. *Pattern Recognition*, 29(5):771–779, 1996.

[73] G. Chow and X. Li. Towards a system for automatic facial feature detection. *Pattern Recognition*, 26(12):1739–1755, 1993.

[74] V. Govindaraju, S. N. Srihari, and D. B. Sher. A computational model for face location. In *International Conference on Computer Vision*, pages 718–721, 1990.

[75] H. P. Graf, T. Chen, E. Petajan, and E. Cosatto. Locating faces and facial parts. In *Int. Workshop on Automatic Face and Gesture Recognition*, pages 41–46, 1995.

[76] M. C. Burl, T. K. Leung, and P. Perona. Face localization via shape statistics. In *Int. Workshop on Automatic Face and Gesture Recognition*, pages 154–159, 1995.

[77] S. Jeng, H. M. Liao, Y. Liu, and M. Chern. Facial feature detection using geometrical face model: An efficient approach. *Pattern Recognition*, 31(3):273–282, 1996.

[78] K. Delac, M. Grgic, and S. Grgic. Statistics in face recognition: analyzing probability distributions of pca, ica and lda performance results. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 289–294. IEEE, 2005.

[79] Ferdinando S Samaria and Andy C Harter. Parameterisation of a stochastic model for human face identification. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pages 138–142, 1994.

[80] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 23(6):643–660, 2001.

[81] L. Ding and A. Martinez. Features versus context: An approach for precise and detailed detection and delineation of faces and facial features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):2022–2038, March 2010.

[82] X. Yu and X. Wang. Uncorrelated discriminant locality preserving projections. *Signal Processing Letters, IEEE*, 15:361–364, 2008.

[83] Jun Yan, Benyu Zhang, Shuicheng Yan, Qiang Yang, Hua Li, Zheng Chen, Wensi Xi, Weiguo Fan, Wei-Ying Ma, and Qiansheng Cheng. Immc: incremental maximum margin criterion. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 725–730. ACM, 2004.

[84] M. J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba. Coding facial expressions with gabor wavelets. In *Automatic Face and Gesture Recognition, IEEE Computer Society*, pages 200–205, 1998.

[85] A. C. Bovik. *Handbook of image and video processing.* Academic Press, 2010.

[86] J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for image and video restoration. *SIAM Multiscale Modeling and Simulation*, 17(1):214–241, 2008.

[87] L. Zhang, W. Dong, D. Zhang, and G. Shi. Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recognition*, 43(4):1531–1549, April 2010.

[88] M. Zhou, H. Yang, G. Sapiro, D. Dunson, and L. Carin. Dependent hierarchical beta process for image interpolation and denoising. *Journal of Machine Learning Research - Proceedings Track*, 15:883–891, 2011.

[89] H. Ji, C. Liu, Z. Shen, and Y. Xu. Robust video denoising using low rank matrix completion. In *CVPR*, pages 1791–1798, 2010.

[90] H. Talebi and P. Milanfar. Global image denoising. *Image Processing, IEEE Transactions on*, 23(2):755–768, 2014.

[91] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, pages 479–486, 2011.

[92] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity. *IEEE Transactions on Image Processing*, 21(5):2481–2499, 2012.

[93] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems*, pages 341–349, 2012.

[94] L. Xu, J. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *Advances in Neural Information Processing Systems*, pages 1790–1798, 2014.

[95] X. J. Mao, C. Shen, and Y. B. Yang. Image denoising using very deep fully convolutional encoder-decoder networks with symmetric skip connections. *arXiv preprint arXiv:1603.09056*, 2016.

[96] A. Rajwade, A. Rangarajan, and A. Banerjee. Image denoising using the higher order singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):849–862, 2013.

[97] I.M. Johnstone and B.W. Silverman. Wavelet threshold estimators for data with correlated noise. *Journal of Royal Statist. Soc.*, 59:319–351, 1997.

[98] G. Chang, B. Yu, and M. Vetterli. Spatially adaptive wavelet thresholding with context modeling for image denoising. *IEEE Trans. Image Processing*, 9(9):1522–1531, 2000.

[99] S. Abramov, V. Lukin, B. Vozel, K. Chehdi, and J. Astola. Segmentation-based method for blind evaluation of noise variance in images. *IEEE Trans. Consumer Electronics*, 2(1):23533, 2008.

[100] M. Ghazal and A. Amer. Homogeneity localization using particle filters with application to noise estimation. *IEEE Trans. Image Processing*, 20(7):1788–1796, 2011.

[101] S. Pyatykh, J. Hesser, and L. Zheng. Image noise level estimation by principal component analysis. *IEEE Trans. Image Processing*, 22(2):687–699, 2013.

[102] Z. Wang, H. R. Sheikh A.C. Bovik, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[103] S. Lazebnik, C. Schmid, and J.Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, 2005.

[104] M. Welk, G. Steidl, and J. Weickert. Locally analytic schemes: A link between diffusion filtering and wavelet shrinkage. *App. and Comp. Harm. Anal.*, 24(2):195 – 224, 2008.

[105] A. Mittal, A. Moorthy, and A. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012.

[106] D. Zhang and Z. Zhou. Two-directional two-dimensional {PCA} for efficient face representation and recognition. *Neurocomputing*, 69(13):224 – 231, 2005.

[107] S. Wang, N. Zhang, X. Peng, and C. Zhou. Two-dimensional locality preserving projection based on maximum scatter difference. *Journal of Information and Computational Science*, 8(3):484–494, 2011.

[108] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using lapacianfaces. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 27(3):328–340, 2005.

[109] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[110] R. Costantini, L. Sbaiz, and S. Ssstrunk. Higher order svd analysis for dynamic texture synthesis. *IEEE Trans. Image Processing*, 17(1):42–52, 2008.

[111] A. Buades, B. Coll, and J. Morel. Image denoising methods. a new nonlocal principle. *SIAM Review*, 52(1):113–147, 2010.

[112] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61(3):183–193, 1954.

[113] Kodak image database. `http://r0k.us/graphics/kodak/`.

[114] Color brodatz texture database. `http://multibandtexture.recherche.usherbrooke.ca/colored_brodatz_more.html`.

[115] P. Chatterjee and P. Milanfar. Is denoising dead? *Image Processing, IEEE Transactions on*, 19(4):895–911, 2010.

# Publications

The list of publications out of the work is as under:

- Journal

  (1) G. Shikkenawis, S. K. Mitra, Two Dimensional Orthogonal Locality Preserving Discriminant Projection and its Application to Image Denoising (Manuscript communicated to IEEE Transactions on Image Processing (TIP), 2016).

  (2) G. Shikkenawis, S. K. Mitra, Two Dimensional Orthogonal Locality Preserving Projections for Image Denoising, IEEE Transactions on Image Processing (TIP), 25(1), pp. 262-273, 2016.

  (3) G. Shikkenawis, S. K. Mitra, On some variants of Locality Preserving Projection, Special Issue on Visual Big Data, Neurocomputing, Elsevier, 173 part 2, pp. 196-211, 2016.

  (4) G. Shikkenawis, S. K. Mitra, A. Rajwade, Image Denoising using Orthogonal Locality Preserving Projections, Journal of Electronic Imaging, 24 (4), 2015.

- Conference

  (1) G. Shikkenawis and S. K. Mitra, Kernelization of Locality Preserving Discriminant Projection, National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2015.

  (2) G. Shikkenawis, S. K. Mitra, Locality Preserving Discriminant Projection, IEEE International Conference on Identity Security and Behavior Analysis (ISBA), 2015.

  (3) G. Shikkenawis, S. K. Mitra, A. Rajwade, A New Orthogonalization of Locality Preserving Projection and Applications, International Conference on Pattern Recognition and Machine Intelligence (PReMI), LNCS 8251, pp. 277-283, 2013.

  (4) D. Jain, G. Shikkenawis, S. K. Mitra, S. Parulkar, Face and facial expression recognition using Extended Locality preserving projection, National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013.