DOCTOR OF PHILOSOPHY

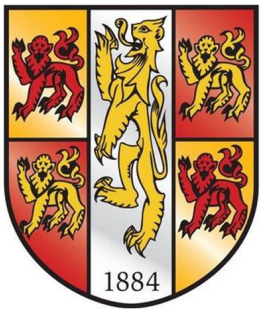
Faithful visualization and dimensionality reduction on graphics processing unit

Najim, Safa

Award date:
2014

Awarding institution:
Bangor University

Faithful Visualization and Dimensionality Reduction on Graphics Processing Unit



Safa Amir Najim

School of Computer Science

University of Bangor

A thesis submitted for the degree of

Doctor of Philosophy

March 2014

I would like to dedicate this thesis to my loving father and the loving memory of my mother for their support and unconditional love.

To my wife “Widad”, who has accompanied me with her love, unlimited patience, understanding, help and encouragement. Without her support, I would never have been able to accomplish this work.

I would like to dedicate this thesis to my brother Dr. “Alaa”, whose love and support has sustained me throughout my life.

To my kids “Manar”, “Mustafa” and “Mariam”, who make life fun.

To all my beloved brothers and sisters families, I dedicate this thesis.

Acknowledgements

First and foremost I thank Almighty "Allah", the compassionate, the almighty, the merciful, who kindly helped me to complete my thesis.

My sincere thanks to my country (Iraq). I recognize that this thesis would not have been possible without their financial support.

The success of this thesis is attributed to the extensive support and assistance from my supervisor, Dr. Ik Soo Lim. I would like to express my gratitude and sincere appreciation to him for his guidance, valuable advice, supervision and encouragement to me throughout this study.

A very special thanks goes out to the head of research, Prof. Nigel W. John for his kindness and support. I thank him very much for reading my thesis and for his important comments.

I would like to give my sincere and warm thanks to Dr. William J. Teahan for taking the time to read my thesis and for his advice. I express my deeply felt thanks to him and Prof. Lusy Kuncheva for their important comments.

I am indebted to Dr. Sa'ad Mansoor, Head of the School of Computer Science, Bangor University, for his kind support and help in my work. My sincere thanks also go to all members of School for their helpful attitude and constant support.

Last, but not least, I sincerely thank my family, brothers, sisters, relatives and their families, who stood by me during this challenging time.

Abstract

Information visualization is a process of transforming data, information and knowledge to the geometric representation in order to see unseen information. Dimensionality reduction (DR) is one of the strategies used to visualize high-dimensional data sets by projecting them onto low-dimensional space where they can be visualized directly. The problem of DR is that the straightforward relationship between the original high-dimensional data sets and low-dimensional space is lost, which causes the colours of visualization to have no meaning.

A new nonlinear DR method which is called faithful stochastic proximity embedding (*FSPE*) is proposed in this thesis to visualize more complex data sets. The proposed method depends on the low-dimensional space rather than the high-dimensional data sets to overcome the main shortcomings of the DR by overcoming the false neighbour points, and preserving the neighbourhood relation to the true neighbours. The visualization by our proposed method displays the faithful, useful and meaningful colours, where the objects of the image can be easily distinguished. The experiments that were conducted indicated that the *FSPE* is higher in accuracy than many dimension reduction methods because it prevents as much as possible the false neighbourhood errors to occur in the results.

In addition, in the results of other methods, we have demonstrated that the *FSPE* has an important role in enhancing the low-dimensional space which are carried by other DR methods. Choosing the worst efficient points to update the rest of the points has helped in improving the visualization information. The results showed the proposed method has an impacting role in increasing the trustworthiness of the visualization by retrieving most of the local neighbourhood points, which they missed during the projection process.

The sequential dimensionality reduction (*SDR*) method is the second proposed method in this thesis. It redefines the problem of DR as a sequence of multiple DR problems, each of which reduces the dimensionality by a small amount. It maintains and preserves the relations among neighbour points in low-dimensional space. The results showed the accuracy of the proposed *SDR*, which leads to a better visualization with minimum false colours compared to the direct projection of the DR method, where those results are confirmed by comparing our method with 21 other methods.

Although there are many measurement metrics, our proposed point-wise correlation metric is the better. In this metric, we evaluate the efficiency of each point in the visualization to generate a grey-scale efficiency image. This type of image gives more details instead of representing the evaluation in one single value. The user can recognize the location of both the false and the true points.

We compared the results of our proposed methods (*FSPE* and *SDR*) and many other dimension reduction methods when applied to four scenarios: (1) the unfolding curved cylinder data sets; (2) projecting a human face data sets into two dimensions; (3) classifing connected networks and (4) visualizing a remote sensing imagery data sets. The results showed that our methods are able to produce good visualization by preserving the corresponding colour distances between the visualization and the original data sets.

The proposed methods are implemented on the graphic processing unit (GPU) to visualize different data sets. The benefit of a parallel implementation is to obtain the results in as short a time as possible. The results showed that compute unified device architecture (CUDA) implementation of *FSPE* and *SDR* are faster than their sequential codes on the central processing unit (CPU) in calculating floating-point operations, especially for a large data sets. The GPU is also more suited to the implementation of the metric measurement methods because they do a large computation. We illustrated that this massive speed-up requires a parallel structure to be suitable for running on a GPU.

Contents

Contents v

List of Figures viii

List of Tables xiii

Nomenclature xiv

1 Introduction 1

1.1 Problems of Dimensionality Reduction (DR) 2
1.2 Hypothesis and Research Goals 2
1.3 Contributions 3
1.4 Thesis Organization 5

2 Dimensionality Reduction 6

2.1 Introduction 6
2.2 Types of Dimensionality Reduction 8
2.2.1 Linear Dimension Reduction Methods 10
2.2.2 Nonlinear Dimension Reduction Methods 13
2.2.3 Local Nonlinear Methods 18
2.2.4 Trustworthiness Methods 20
2.3 Information Visualization 22
2.4 Quality of Visualization 26
2.5 Conclusion 28

3 Visualization of Remote Sensing Imagery Data Sets Using Faithful Stochastic Proximity Embedding **29**
3.1 Introduction . . . 30
3.2 Visualization of Remote Sensing Imagery Data Sets by Dimensionality Reduction . . . 32
3.3 Faithful Stochastic Proximity Embedding . . . 32
3.3.1 Stochastic Proximity Embedding (SPE) . . . 32
3.3.2 *FSPE* . . . 33
3.4 Point-wise Quality Metric . . . 36
3.5 Experimental Results . . . 36
3.5.1 Quantitative Comparison . . . 38
3.5.2 Faithful Visualization . . . 44
3.6 Discussion . . . 52
3.7 Conclusion . . . 56

4 Visualization of High-dimensional Data Sets by Sequential Dimensionality Reduction **57**
4.1 Introduction . . . 58
4.2 Methodology: Sequential Dimensionality Reduction . . . 58
4.3 Experimental Results . . . 62
4.4 Conclusions . . . 66

5 Faithful Visualization of Different Data Sets **75**
5.1 Introduction . . . 75
5.2 Data Sets . . . 77
5.3 Unfolding The Curved Cylinder Data Sets . . . 77
5.4 Projecting The Frey Face Data Sets into Two dimensional Space . . . 81
5.5 Network Classification . . . 86
5.6 Conclusion . . . 94

6 Speed-up The Processing by Using Graphics Processing Unit **95**
6.1 Introduction . . . 96
6.2 Our Contributions in GPU Application . . . 101
6.2.1 Parallel *FSPE* . . . 102

6.2.2 Parallel Correlation Metric 103
6.2.3 Parallel Residual Variance Metric 104
6.3 Experimental Results 105
6.4 Conclusion 107

7 Enhancing The Visualization by Using Faithful Stochastic Proximity Embedding 109
7.1 Introduction 109
7.2 Enhancement Method 110
7.3 Experimental Results and Discussion 111
7.4 Conclusion 115

8 Final General Conclusion 117
8.1 Conclusion 117
8.2 Future Work 119

Appendix A 121

References 130

List of Figures

2.1 *Three dimensional spiral data sets are unfolded to a one dimensional straight string line* . . . 7

2.2 *Geodesic distance idea. The geodesic distance between the two red points is the length of the geodesic path* . . . 9

2.3 *1000 points of three dimension Swiss Roll data sets. The data sets generated by* $\{x1 = \theta cos(\theta),\ x2 = \theta sin(\theta),\ x3\}$ . . . 12

2.4 *Linear DR methods (PCA and MDS) cannot unfold the Swiss Roll data sets, in Figure 2.3, in two dimension representation* . . . 13

2.5 *Efficiency of nonlinear projection is clear in unfolding Swiss Roll data sets in Figure 2.3* . . . 14

2.6 *LLE and Laplacian methods can unfold the Swiss Roll data sets, in Figure 2.3, in a satisfactory low-dimensional manifold.* . . . 19

2.7 *Trustworthiness methods, CCA and CDA, can unfold the Swiss Roll data sets. CDA is better than CCA* . . . 21

2.8 *DR might cause the points which are outside local neighborhood in high dimension space to be inside local neighborhood* . . . 25

3.1 *Distortions cannot be overcome when the amount of reduction is equal to 3 or less.* . . . 31

3.2 *The main idea of FSPE. At step* t *of iteration process.* p_1 *preserves its distance with true neighbours* . . . 35

3.3 *The visualizations of 12 regions of VIRIS Moffet Field data sets from the southern end of San Francisco Bay, California, done in 1997* . . . 37

3.4 *The DR methods have different efficiency when using different values of neighbourhood points for Region_0.* . . . 39

3.5 *FSPE's projected space of Region_0 is projected into CIE Lab colour space. The largest standard deviation among three channels* 42
3.6 *Gray-scale images of point-wise correlation values. The top two rows show the colour display of a remote sensing imagery* 43
3.7 *p_1 and p_2 are neighbours in the original space and they are far from p_3 (see Table 3.6. SPE do not preserve the distance* 46
3.8 *PCA versus FSPE in visualizing the Region_0. The distance between p_1 and p_2 was not preserved in the PCA's visualization* 47
3.9 *Colour distance between p_1 and p_2 is not preserved in the CDA's visualization, , although they are neighbours in the original space* 48
3.10 *Colour distance between p_1 and p_2 is not preserved in the Isomap's visualization, although they are neighbours in the original space* 49
3.11 *LLE lets non-neighbour regions (as in p_2 and p_3) in the original space to be neighbours in the visualization* 50
3.12 *Qualitative CIE Lab visualization of data sets with 1800x600x224 dimension. FSPE's visualization reveals different vivid colour variations* 53
3.13 *Visualizing the large data sets (1800x600x224) by FSPE is better when compared with CCA* 54
3.14 *Linear projection for the large data sets (1800x600x224) by PCA caused to lose more information, and nonlinear projection by FSPE* 55

4.1 *a) Visualizations of regions_0 by using DR methods (SPE, CCA and FSPE); b) Visualizations by SDR methods (SSPE, SCCA and SFSPE)* 64
4.2 *Visualization of Region_4 in remote sensing imagery data sets by SDR methods are better than their corresponding DR methods* 67
4.3 *SDR methods generate higher quality visualizations of Region_5, and SFSPE is better than SSPE and SCCA* 68
4.4 *Visualization of Region_0 by using 20 DR methods and 3 SDR methods. According to Table 4.5, the worst visualization* 70
4.5 *The average of comparisons, in the Table 4.5, of 3 SDR methods (SSPE, SCCA and SFSPE) and 20 DR methods.* 71
4.6 *The execution times of SDR methods in CPU (SCCA_c, SSPE_c and SFSPE_c) are very high, and the role of using GPU* 72

4.7 *DR and SDR use the same number of iterations. In DR, there is no significant impact on the change in DR's efficiency through iterations* 73
4.8 *The correlation measurement value of SSPE is very high (which is equal to 0.991 for region_9) when S (amount of DR)* 74

5.1 *a) Three dimension of curved cylinder is described by* $x = (7+2\,cos(u))\,cos(v)$, $y = (7+2\,cos(u))\,sin(v)$, $z = 2\,sin(u)$ 76
5.2 *Unfolded space of a three-dimensional curved cylinder by PCA, CCA, CDA, Isomap, LLE, SPE and tSNE and FSPE.* 78
5.3 *The grey-scale efficiency images by the point-wise correlation metric of the unfolded space of curved cylinder in Figure* 79
5.4 *Projecting the Frey face data sets into 2-dimensional space by using FSPE.* . 83
5.5 *Projecting the Frey face data sets into 2-dimensional space by using SFSPE.* 84
5.6 *Grey-scale efficiency image by the point-wise correlation metrics of results in Figure* 85
5.7 *Low-dimensional representation of the American College Football Network by PCA, Prob_PCA, SNE, tSNE, FSPE and SFSPE.* 87
5.8 *The grey-scale efficiency images by the point-wise correlation metric of the low-dimensional representations of the American College Football Network* 88
5.9 *The results of the comparisons, in Table 5.3, of our methods (FSPE and SFSPE) and the 20 DR methods show that our methods (FSPE and SFSPE)* . . 90
5.10 *Low-dimensional representation of the second network by PCA, CCA, CDA, Prob_PCA, FSPE and SFSPE.* 92
5.11 *The grey-scale efficiency images by the point-wise correlation metric of the low-dimensional representations of the second network* 93

6.1 *The performance of the GPU has increased exponentially in recent years, which makes it deliver superior performance in parallel computation* 96
6.2 *The general structure of the CPU and the GPU. a) The CPU contains very few cores that can work in parallel. b) The GPU contains hundreds of small cores* 97
6.3 *GPU consist of many threads which are grouped into blocks, which are then grouped into grids of blocks* 98

6.4 *CPU and GPU are cooperative together in executing CUDA program. Data should be copied from CPU memory to GPU memory* 100
6.5 *Evaluation of one point requires to move N points from CPU to GPU. These points are processing in parallel to evaluate their relation with point in the hand.* 101
6.6 *FSPE is implemented on the GPU by using CUDA C++ codes. This kernel attempts to update the coordinates of all points in the data sets* 102
6.7 *First step to compute correlation metric on the GPU (CUDA C++). This kernel attempts to compute the* $\overline{A}$, $\overline{B}$ *and* $A^T\ B/|A|$ *for the point i* 103
6.8 *Second step to compute correlation metric on the GPU (CUDA C++). This kernel attempt to compute the* σ_A *and* σ_B *for the point pi* 104
6.9 *CUDA C++ codes of residual variance. This kernel computes the stress error (*Stress_i*) for the point "i".* 105
6.10 *The computation time of the FSPE algorithm for different data sets size. The GPU implementation is able to speed up the execution of FSPE* 107
6.11 *Time of moving data from the CPU to the GPU is greater than pure execution in the GPU. Moving the results back to the CPU is also greater than* 108

7.1 *a) The gray-scale image shows the upper right area of the visualization of remote sensing imagery, which was carried by CDA* 112
7.2 *Visual comparison between traditional and enhanced visualizations. a) The gray efficiency image shows there are many points in the CDA* 113
7.3 *a) The grey efficiency image shows that many colours have low efficiency in the SPE's visualization. CIE Yxy colour space of this visualization* 115
7.4 *a) The visualization by Isomap has several false colours. b) The enhancement visualization when FSPE is applied on the selected worst efficiency colours.* 116

1 *Unfolded space of three dimension curved cylinder by many methods. The colour of each point in the unfolded space is the same colour* 122
2 *Unfolded space of three dimension curved cylinder by many methods. The colour of each point in the unfolded space is the same colour* 123
3 *The gray-scale efficiency images by point-wise correlation metric of the unfolded shapes of curved cylinder in the Figure 1* 124
4 *The gray-scale efficiency images by point-wise correlation metric of the unfolded shapes of curved cylinder in the Figure 2* 125

5 *Projecting the Frey face dataset into 2-dimensional space by many dimension reduction methods.* 126
6 *Projecting the Frey face dataset into 2-dimensional space by many dimension reduction methods.* 127
7 *Gray-scale effiency image by point-wise correlation metric of results in the Figure 5.* 128
8 *Gray-scale effiency image by point-wise correlation metrics of results in the Figure 6.* 129

List of Tables

3.1 *By using Euclidean distance, correlation measurement values of the comparison between FSPE and PCA, SPE, Isomap, CDA and LLE* 40

3.2 *By using geodesic distance, correlation measurement values of the comparison between FSPE and PCA, SPE, Isomap, CDA and LLE* 40

3.3 *Performance of DR projections measured according to the LC. The higher, the better.* 41

3.4 *LC metric. FSPE has a few number of false neighbourhood points, which have zero measurements,* 44

3.5 *For all Regions: by using LC metric, comparison among methods by using the number of false neighbourhood points* 44

3.6 *The distances among the three points p_1, p_2 and p_3 in the original space and their corresponding distance in the projected spaces of SPE and FSPE.* . . . 46

3.7 *The distances among the three points p_1, p_2 and p_3 in the original space and their corresponding distance in the projected spaces of PCA and FSPE.* . . . 47

3.8 *The distances among the three points p_1, p_2 and p_3 in the original space and their corresponding distance in the projected spaces of CDA and FSPE.* . . . 48

3.9 *The distances among the three points p_1, p_2 and p_3 in the original space and their corresponding distance in the projected spaces of Isomap and FSPE.* . . 49

3.10 *The distances among the three points p_1, p_2 and p_3 in the original space and their corresponding distance in the projected spaces of LLE and FSPE.* . . . 50

4.1 *Methods employed in comparison. The first column contains the name of methods, second column shows the type of method* 62

4.2 *Results of comparison the SDR method (represented by SSPE), when $S = 1$, with the DR method (represented by SPE)* 63

4.3 *Results of comparison the SDR method (represented by SCCA), when $S = 1$, with the DR method (represented by CCA)* 65
4.4 *Results of comparison the SDR method (represented by SFSPE), when $S = 1$, with the DR method (represented by FSPE)* 65
4.5 *Correlation, LC and stress values of comparisons among 23 methods for three regions (Region_0, Region_4 and Region_5)* 69
4.6 *The computation time, in seconds, of the SDR methods in the CPU and the GPU. the GPU is the best way to speed up the SDR methods* 69

5.1 *Unfolded spaces, in Figure 5.2, of the curved cylinder data sets by using 20 methods are measured by correlation, LC and stress metrics.* 80
5.2 *The results of reducing the dimensionality of the Frey face data sets into two dimensions, which are carried by 21 dimension reduction methods* 82
5.3 *Measuring the low-dimensional representation of the American College Football Network, which was carried out by 21 methods,* 89
5.4 *Measuring the low-dimensional representation of the second network, which was carried out by 21 methods,* 91

6.1 *FSPE speed performance comparisons between the CPU codes and the CUDA C++ codes with different size of remote sensing imagery data sets* 106
6.2 *Performance comparisons of the CPU codes and CUDA C++ codes in small data sets (300x300x224). Correlation or Stress computation* 106

7.1 *The enhanced visualization got the highest correlation and least stress values. (Correlation: the highest is better, Stress: the lowest is better).* 114

Chapter 1

Introduction

Information visualization can be considered a process of transforming similarity relationships between data points to a geometric representation in order to see unseen information. Results are analysed for the following questions: Does this visualization help to make new discoveries when analysing large quantities of data? What is the relationship between original data points and transformed data points? How can the original information be preserved in projected space?

A high-dimensional data set is one of the main problems of information visualization. Dimensionality reduction (DR) is therefore a useful strategy to project high-dimensional data sets onto low-dimensional space, which can be visualized directly. The application of this technique has two benefits:

1. DR can minimize the amount of storage needed by reducing the size of the data sets.

2. DR helps to understand the data sets by discarding any irrelevant features, and to focus on the main important features.

DR can enable the discovery of rich information, which assists the task of data analysis. Visualization of a high-dimensional data sets is widely used in many fields, such as remote sensing imagery, biology, computer vision, and computer graphics.

The colour of each pixel in the visualization should somehow be a compendium of information contained in the original data at the corresponding data point at which their most salient features are captured. The visualization is a simple way to understand

the high-dimensional data sets because the relationship between original data points is incomprehensible.

Therefore, DR might face the following questions: Has DR the ability to preserve all pairwise relationships between data points? Can people observe the proximity relationships amongst data points in a projected space? This has led to the introduction and development of a large number of DR methods that attempt to minimize the loss of original information. However, preserving original information is becoming more and more difficult, as we will see in the next section.

1.1 Problems of Dimensionality Reduction (DR)

Using DR in the visualization of data sets can cause two kinds of errors to occur; namely, *continuity* errors and *false neighbourhood* errors. In *continuity* errors, the nearby neighbourhood data points in the original space can be projected further away in the projected space. On the other hand, *false neighbourhood* errors cause the farther away data points in the original space to be projected nearby in the projected space.

Although many methods have been introduced to overcome the above problems, a fully optimal method does not exist. Thus, the colours of visualization typically have no meaning because the straightforward relationship with the original high-dimensional data sets is lost.

1.2 Hypothesis and Research Goals

This section explains the main purpose of this research by proposing some hypotheses.

Useful information exists in low-dimensional space despite transforming the data from a higher dimensional space. Thus, the DR problems can be overcome. This statement is very important, and it represents the main point for this research. The methodologies in this dissertation have been dedicated to improving and developing methods for deriving conclusions and have satisfied this hypothesis. Quantitative and qualitative measurements are the good ways to prove the efficiency of our method. Using many well-known measurement methods are necessary to give a good concept about what the visualization will be. Massive dimensionality of data sets is handled in

order to make almost certain that our methods have minimized as much as possible the error of losing information.

If we suppose that such information does exist in low-dimensional manifold, the missing neighbourhood points can be retrieved. While this hypothesis might be simple to address, it is yet to be proven. Indeed, several methods including ours attempt to prove this statement.

If the projected space has been transferred to the appropriate competent colour space, the derived information will be accurate and richly represented. Several DR methods mapped their results to the *CIE Lab* colour space to get high quality colours. The important thing in this dissertation is to define a simple, efficient and direct way to achieve what was imposed.

The way of applying DR can be redefined to increase the efficiency of the projected space. Due to the difference between the topological structures of data points in high-dimensional data sets with the topological structure of projected space, it is difficult to preserve the neighbourhood relations between data points. One of our proposed methodologies is trying to redefine the DR as sequential DR, which can lead to preserving the original information as much as possible.

The users role is important in evaluating the efficiency of the visualization. If there is a mechanism to give him a chance to see the site of strengths and weaknesses, that can help in the analysis of those data sets. Most traditional measurement methods compute numerical value to reflect the efficiency of the projected space. This way is important, but the user wants the easier way. In this project, this hypothesis has been studied with interest, and we were able to achieve convergence between the results and the user.

1.3 Contributions

Below is a list of contributions used in our project:

1. The faithful dimension reduction method, called *Faithful Stochastic Proximity*

Embedding (FSPE), is proposed to visualize data sets. *FSPE* can overcome the false neighbourhood points as much as possible to derive higher quality colours in its visualization.

2. A new technique called *Sequential Dimensionality Reduction (SDR)* is also introduced in this thesis. Unlike DR, our proposed new technique redefines the problem of DR as a sequence of multiple DR problems, each of which reduces the dimensionality by a small amount. The efficiency of the visualization is increased when applying it.

3. We have visualized very complex data sets, such as remote sensing imagery and human faces. In addition, we have classified connected networks and unfolded curved cylinder.

4. We propose the point-wise metric of "good" visualization. Unlike the existing metrics, which only provide a global measure of the whole visualization with a single number, the proposed metric provides a quantitative measure for each pixel of the data sets, and tells us how reliable the colour is per pixel. This measure is a more stringent criterion to estimate the quality of the projection.

5. *FSPE* is used after a dimension reduction method as a supplementary stage for enhancing the final results by retrieving the missing neighbourhood points. This system has succeeded in enhancing the visualization and overcoming the false neighbourhood errors that might happen through projection.

6. The computational speed of our methods are improved by using the graphics processing unit (GPU) as the computation engine rather than the central processing unit (CPU).

List of Publications

1. Najim S., Lim I. and Saeed M. (2013). Trustworthy Enhancing the Visualization of Remote Sensing Imagery Dataset on GPU. In Proceedings of 6th International Conference on Developments in eSystems Engineering (DESE 2013), Abu Dhabi, UAE, 16-18 December.

2. Najim S. and Lim I. (2014). Visualization of Remote Sensing Imagery by Sequential Dimensionality Reduction on Graphics Processing Unit. In Proceedings of the 5th International Conference on Information Visualization Theory and Applications (IVAPP 2014), , pages 71-79, Lisbon, Portugal, 5-8 January.

3. S.A. Najim, I.S. Lim, Trustworthy dimension reduction for visualization different data sets, Inform. Sci. (2014), http://dx.doi.org/10.1016/j.ins.2014.03.048.

4. Najim S., Lim I, Wittek P. and Jones M. FSPE: Visualisation of Hyperspectral Imagery Using Faithful Stochastic Proximity Embedding. Accepted for publication in IEEE Geoscience and Remote Science Letters 25/3/2014.

1.4 Thesis Organization

The organization of the dissertation is as follows. Chapter 2 gives an overview of DR, where different categories are explained together with their advantages and limitations. The methodology of the first proposed method, *FSPE*, is explained in Chapter 3. In this chapter, we use *FSPE* in the visualization of a remote sensing imagery data sets, and compare its results with well-known methods to demonstrate the ability of our method. Chapter 4 explains the second new method, *SDR*, and this is compared with other methods. In Chapter 5, we compare our proposed methods (*FSPE* and *SDR*) and 19 DR methods by using three data sets: unfolding curved cylinder data sets, projecting human face data sets into 2-dimensional space and classifying two connected networks. The primary aim of this chapter is to evaluate the efficiency of our methods in neighbourhood preservation. Chapter 6 illustrates the parallel processing in the GPU, where a parallel method to visualize data sets and measure its efficiency on the GPU are presented. In Chapter 7, we use *FSPE* as a supplementary stage after a dimension reduction method for enhancing the final results. Finally, Chapter 8 concludes the dissertation.

Chapter 2

Dimensionality Reduction

In this chapter we will address the following topics:

1. Formal definition of dimensionality reduction.
2. Overviews of dimensionality reduction methods with advantages and disadvantage of each one.
3. Measurement metrics to evaluate the result of dimensionality reduction

2.1 Introduction

Visualisation of high-dimensional data sets is widely used to analyse data in many fields of study, including remote sensing imagery, biology, computer vision, and computer graphics. Its purpose is to provide rich information to assist with data analysis [Zhang, 2008]. Dimensionality reduction (DR) is an important step for data preprocessing in visualisation and knowledge discovery, and it is used for different purposes, such as information visualisation, noise reduction, and imaging applications [Borg and Groenen, 2005] [France and Carroll, 2011] [Nishisato, 2006]. Formally, for a set of n input points $X \subset \mathcal{R}^D$, $\phi(X)$ is used to project the D dimensional data points $x_i \in X$ to d dimensional data points $y_i \in Y$, where $d \ll D$.

$$\phi : \mathcal{R}^D \rightarrow \mathcal{R}^d \tag{2.1}$$

$$x_i \mapsto y_i \;\; \forall \; 1 \leq i \leq n \tag{2.2}$$

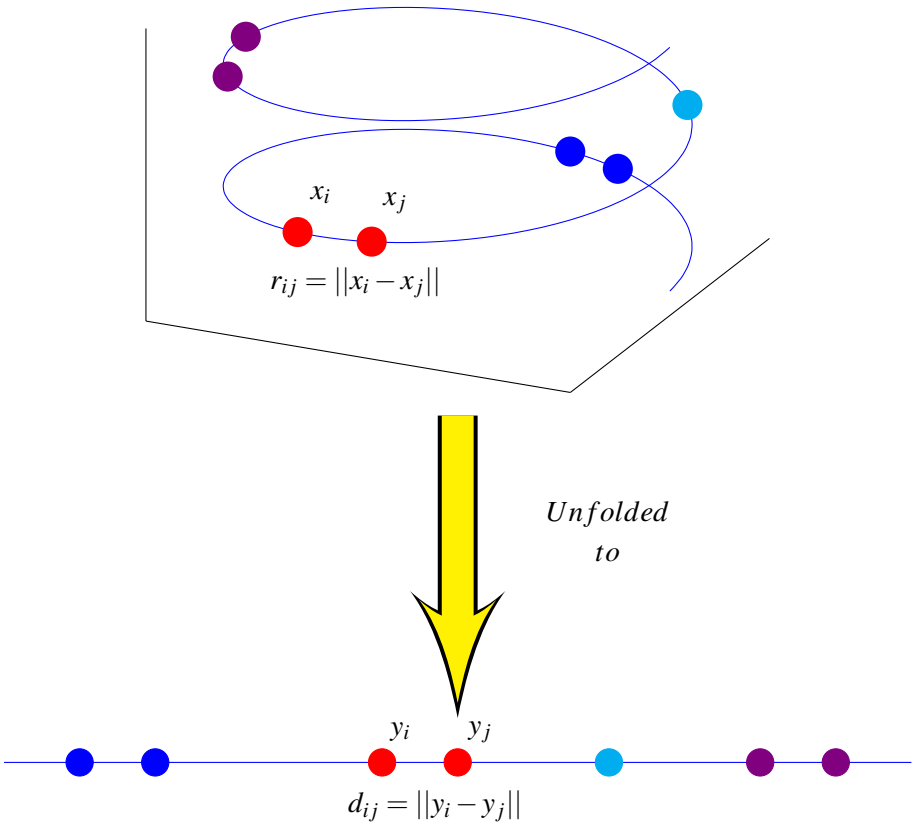


Figure 2.1: *Three dimensional spiral data sets are unfolded to a one dimensional straight string line. The neighbourhood relations between points in the unfolding space are preserved with their correspoinding relations in the high-dimensional data sets.*

In Equation 2.1, ϕ attempts to approximate the output pairwise distance $d(x_i, x_j)$ with their corresponding in input space $r(y_i, y_j)$, i.e, $r(x_i, x_j) \approx d(y_i, y_j) \ \forall \ 1 \leq i \leq n$ to project X's data point correctly in Y space.

The high-dimensional data sets have several features; however, some might not be relevant to specific data analysis. DR is used to discover the main and important features by which to make analysis and visualisation possible. The fundamental information in the original data sets is reflected in the distances between pairs of data points, and this information should be preserved by using a gradient step and fitting the input distances r_{ij} to output distances d_{ij}. Thus, the goal of preserving the distance

is to represent the original data sets in a projectec space [Meng et al., 2011] [Lee and Verleysen, 2007]. A simple example of applying DR is shown in Figure 2.1, where the spiral data sets are unfolded to the straight string. Each point in the straight string preserves its local neighbourhood relations, and preserving neighbourhood relations between points indicates the efficiency of the low-dimensional space.

In reality, reducing dimensionality of large data sets to a low-dimensional space without losing information might be impossible. In general, DR attempts to minimize as in the following equation:

$$\phi = \sqrt{\sum_{i,j=1}^{n} (r_{ij} - d_{ij})^2} \tag{2.3}$$

where $r_{ij} = ||x_i - x_j||$ and $d_{ij} = ||y_i - y_j||$ for the four points x_i, x_j, y_i and y_j. The cost function in Equation 2.3 measures the difference between the distances in the input space and the corresponding distances in the projected space, and the final values should be minimized according to the data in a projected space.

Unfolding a complex high-dimensional data sets into low-dimensional representation should focus on preserving the nearby neighbourhood relationship between points rather than on creating additional points. There are two ways by which to define neighbouring points for a point. The first supposes that all points are neighbours for a point, but the nearest k points are strong neighbours [Yang, 2011]. Each point has a fixed number of neighbours and this number will not change through the projection process. The second method uses a fixed circular radius r_c, where the neighbouring points are inside this domain for a point [Agrafiotis, 2003]. Thus, the number of neighbours is not the same for all points in the space.

2.2 Types of Dimensionality Reduction

A variety of strategies has resulted in the development of many different DR methods, which can be classified as being either linear or nonlinear. Most of the linear methods are non-iterative, which assumes that the data are distributed close to a hyperplane of original space to reflect their uniqueness [Borg and Groenen, 2005] [Agarwal et al.,

2007]; therefore, they are constrained in many applications. To overcome this problem, most nonlinear methods adopt iterative optimisation to get more flexibility in searching for a representation of data points in a projected space. Iterative methods start with a random configuration of low-dimensional space, and iteratively refine it until the error measure of low-dimensional space is minimised to the defined threshold or the specified number of iterations has been sapped. Local and trustworthiness methods are two other types of DR. The codes of 34 DR methods have been programmed in a Matlab toolbox and can be found in [Maaten, 2013].

Preservation of high-dimensional data sets distances depends on the distance similarity measure. Distance, in nature, is used to measure the dissimilarity between two data points in a space. Euclidean and geodesic distances are usually used in this matter. The notion of Euclidean distance was originally defined as the distance of a straight line between two points, but the efficiency of Euclidean distance depends on the type of area. For example, the distance between two nearby points is computed exactly by Euclidean distance; however, in nonlinear space, Euclidean distance cannot measure the correct distance. For this, the concept of geodesic distance is suitable for computing the shortest path between two points that lie on a surface, as in Figure 2.2. Although both ideas are used for distance preserving DR, many methods are enhanced when geodesic distance is used because it provides a better measurement of distance between pairs of data points than Euclidean distance.

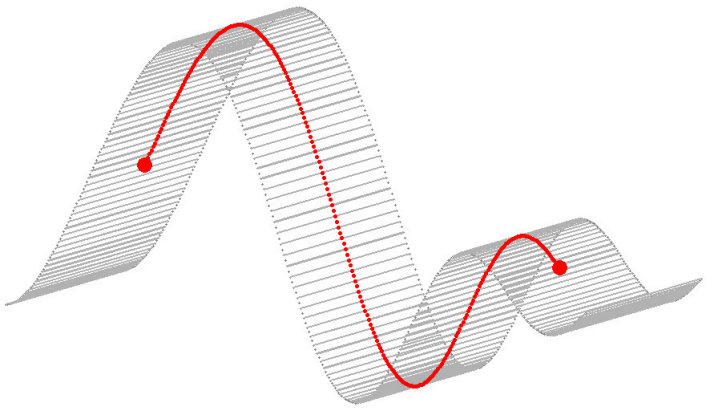


Figure 2.2: *Geodesic distance idea. The geodesic distance between the two red points is the length of the geodesic path, which is the shortest path between the points that lie on the surface.*

2.2.1 Linear Dimension Reduction Methods

The well-known methods of linear DR are principle component analysis (PCA), multidimensional scaling (MDS), and linear discriminant analysis (LDA). The benefits of these methods are that their results are simple and can be easily interpreted.

PCA is the oldest of the three methods, and has been used in different applications because of its simplicity. PCA supposes the linear projection between original data and projected space [Jolliffe, 2002], but this principle does not work well with nonlinear data sets. The neighbourhood relationships between points are lost in projected space and do not preserve their nonlinear relationships of original data. Figure 2.4(a) shows PCA cannot unfold Swiss Roll data sets. PCA uses orthogonal linear combination to find linear transformation space of data sets $X \subset \mathcal{R}^D$. The steps of applying it are:

1. Compute the centroid of original data sets:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2.4}$$

   and subtract it from x_i to generate new origin

$$z_i = x_i - \mu \tag{2.5}$$

2. Compute the covariance matrix

$$cov = \frac{1}{n}\sum_{i=1}^{n} z_i \, z_i^T \tag{2.6}$$

   where z_i^T is the transpose of the z_i

3. According to the target dimension of low dimension space, PCA computes eigenvectors $u_1, u_2, ..., u_d$ and eigenvalues $\lambda_1, \lambda_2, ..., \lambda_d$ of covariance matrix cov.

4. Compute principal component transformation :

$$y = P^T z \tag{2.7}$$

   where $P = [u_1, u_2, ..., u_d]$ and its transpose is P^T.

MDS is a method used for fitting a set of data points in a space corresponding to a given set of objects. The distances between data points of high-dimensional data sets are known, and the coordinates of data points in projected space are generated according to these distances [Borg and Groenen, 2005] [Forero and Giannakis, 2012] [Lee and Verleysen, 2007]. There are two types of MDS methods: iterative and classical. Iterative MDS can be classified into two types, which are metric and non-metric scaling. Metric MDS uses the actual distance values between data points, and non-metric uses only their rank. The input can be coordinates rather than a distance matrix, and, in this case, MDS generates a distance matrix among points by computing pairwise distance d_{ij} between points y_i and y_j. The distance matrix should be symmetrical where $d_{ij} = d_{ji}$ and $d_{ij} = 0$. The iterative MDS attempts to minimize the Kruskal stress function in 2.8 which indicates to the goodness of fit value by minimizing the difference between pairwise Euclidean data points $d_{ij} = ||y_i - y_j||$ distances in projected space with their corresponding Euclidean distances in original space $r_{ij} = ||x_i - x_j||$. $\sum_{i,j=1}^{n}(d_{ij})^2$ in Equation 2.8 is used to normalize the results to fall in the range [0, 1]:

$$S = \sqrt{\frac{\sum_{i,j=1}^{n}(r_{ij} - d_{ij})^2}{\sum_{i,j=1}^{n}(d_{ij})^2}}. \tag{2.8}$$

Classical MDS is a non-iterative method and uses linear algebra to solve a problem. It calculates the eigenvalues and eigenvectors of a covariance matrix. Classical MDS is identical to PCA, except MDS input should be pairwise similarity. The only difference between iterative and classical MDS is the way they are applied. As with PCA, MDS is not efficient enough to find appropriate projected space of a nonlinear data sets where there are big differences between d_{ij} and r_{ij} Euclidean distance measurements cause incorrect adjustments of the locations of data points in projected space. Although iterative MDS has more flexibility than classical MDS, it suffers from heavy computations, and it is a purely global method that cannot work with local structure. Figure 2.4(b) shows the limitation of MDS in unfolding Swiss Roll data sets.

The objective of linear discriminant analysis (LDA) is to perform DR while preserving discriminatory information [Martinez and Zhu, 2005] [Kim et al., 2011]. As with PCA, it seeks to find direction along with the classes that are good separated, but unlike PCA, LDA does classify data. LDA is a supervised method whereby class labels

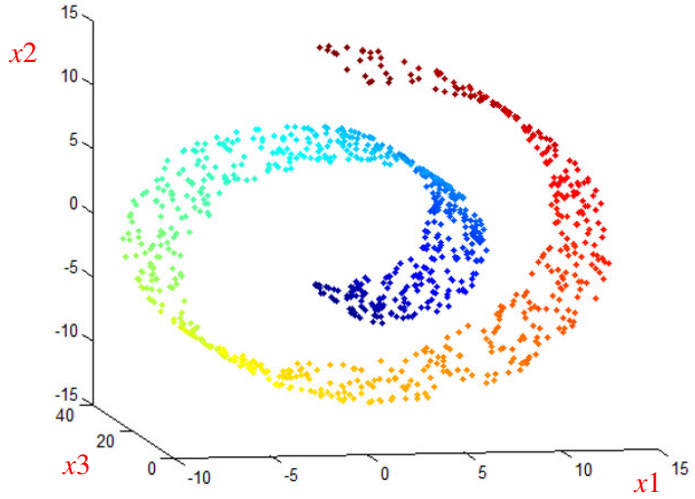


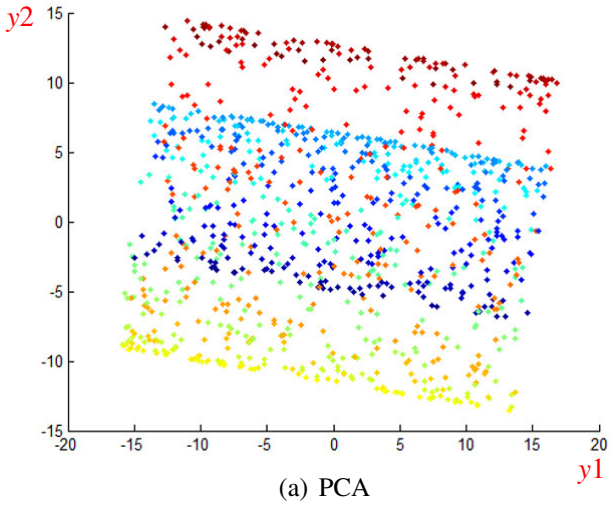
Figure 2.3: *1000 points of three dimension Swiss Roll data sets. The data sets generated by* $\{x1 = \theta cos(\theta),\ x2 = \theta sin(\theta),\ x3\}$*, where* θ *and* $x3$ *are random numbers in the ranges [5, 14] and [0, 30], respectively. The colours have been added to describe the relationship between points.*

should be available, which leads it to be more effective than unsupervised PCA [Ji and Ye, 2008]. It preserves the location of the original data sets when projected to different space dimensions with increasing class linear separability among them. LDA classifies separability into within-class scatter (S_w) and between-class scatter (S_b), where LDA transformation maximises S_b but minimises S_w, where their matrix are defined as

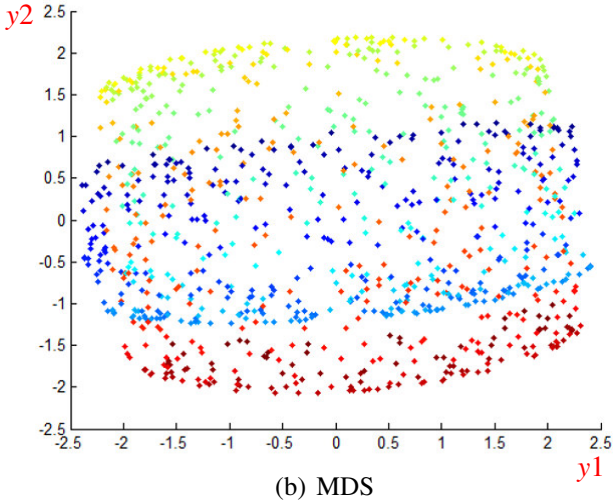
$$S_w = \sum_{i=1}^{C} \sum_{j=1}^{M_i} (y_j - \mu_i)(y_j - \mu_i)^T \tag{2.9}$$

$$S_b = \sum_{i=1}^{C} (\mu_i - \mu)(\mu_i - \mu)^T \tag{2.10}$$

where C is the number of classes, M_i is the number of samples within class i, μ_i is mean vector of class i, and μ is mean of entire data sets defines as $\mu = \frac{1}{C}\sum_{i=1}^{C}(\mu_i)$. The class separability is done while reducing the variation; however, when dimensionality of data is larger than the sample size, LDA cannot be applied because all scatter matrices will be singular [Ji and Ye, 2008]. LDA can be applied when the data sets have many classes, therefore, it is not suitable to unfold the Swiss Roll data sets, in Figure 2.3 because it consist from one class.



(a) PCA



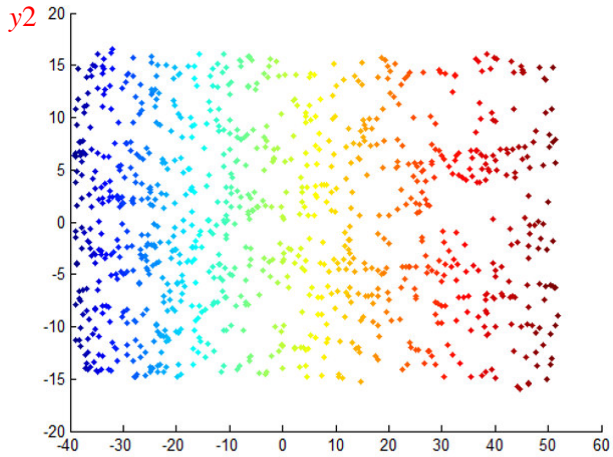
(b) MDS

Figure 2.4: *Linear DR methods (PCA and MDS) cannot unfold the Swiss Roll data sets, in Figure 2.3, in two dimension representation, where overlap in neighbourhood relation is the predominate status of these unfolded representations.*

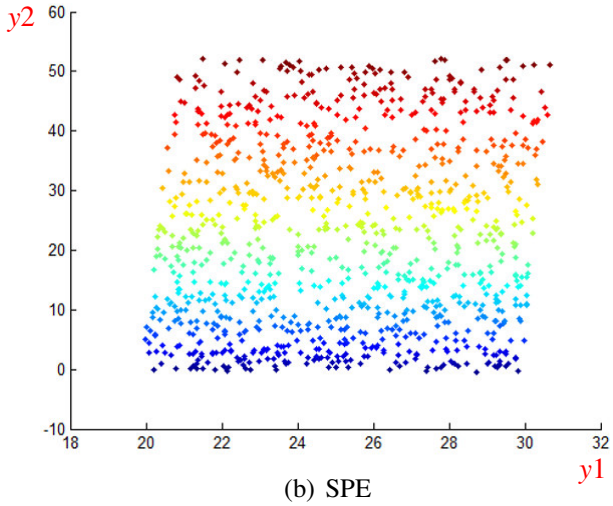
2.2.2 Nonlinear Dimension Reduction Methods

Nonlinear DR uses nonlinear mapping to preserve the nonlinear properties of the original data. Isometric feature mapping (Isomap), stochastic proximity embedding (SPE), and stochastic neighbour embedding (SNE) are the more general methods that have been used to reduce dimensionality of nonlinear data.

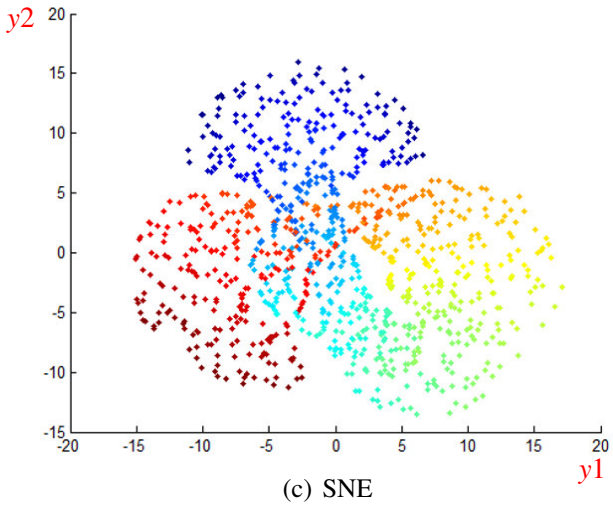
Isomap is a well-known global nonlinear DR method; it was produced to overcome



(a) Isomap



(b) SPE



(c) SNE

Figure 2.5: *Efficiency of nonlinear projection is clear in unfolding Swiss Roll data sets in Figure 2.3. Two dimension representations by Isomap and SPE preserve as well as the neighbourhood relation of original space data points. SNE preserves most of the points, and some of them are overlapped with the others.*

the limitations of MDS. Isomap attempts to preserve and hold the global structure of the original high-dimensional data sets [Tenenbaum et al., 2000] by computing the geodesic distance $r_{ij} = ||x_i - x_j||$ between two high-dimensional pairs of data points x_i and x_j. Geodesic distance is better than Euclidean distance in recognizing distances between neighbours near and far. Isomap is a continuity method that attempts to preserve the geodesic distance r_{ij} in the original space with its corresponding linear distance d_{ij} in the projected space. The algorithm of Isomap can be summarized in the following steps:

1. If the data sets are very large, select n random data points. Otherwise, all data points are selected.

2. Constructing the neighbourhood graph by connecting each data points to its k nearest neighnours.

3. The shortest paths among data points are computed by using Dijkstra's algorithm [Clark and Holton, 2005] to construct a geodesic distance matrix.

4. Applying MDS to find low-dimensional space by minimizing the following stress function:
$$\phi(Y) = \sqrt{\sum_{i<j}(r_{ij} - d_{ij})^2} \tag{2.11}$$

5. If step 1 is applied, the interpolation is applied on the remaining data points to generate final projected space.

Isomap has been used in different applications by reducing the high-dimensional data sets into two or three dimensions [Samko et al., 2006] [Hamarneh et al., 2011] [Verma et al., 2007] [Geng et al., 2005]. It has the ability to deal with nonlinear data sets and can discover the information and details that were hidden on MDS. For example, Figure 2.5(a) shows an Isomap unfolded Swiss Roll data sets in a two-dimensional space, where the neighbourhood distances between original data points are preserved with their corresponding points in projected space; however, an extensive time commitment and storage requirements make Isomap inappropriate for use on large data sets. In addition, Isomap fails to generate a projected space when some points are not connected to a graph, where the number of points in the projected and original spaces

are not equal [Akkucuk and Carroll, 2010]. Another method, PARAMAP [Akkucuk and Carroll, 2006], attempts to overcome the Isomap problems and gives better results than Isomap on circular shapes, although its efficiency decreases with large data sets. To accelerate the projection process, it is better to use Isomap first and use PARAMAP to complete the process [Akkucuk and Carroll, 2010].

Stochastic proximity embedding (SPE) is a nonlinear method that proceeds by calculating Euclidean distance for global neighbourhood points within a fixed radius [Agrafiotis and Xu, 2002] [Agrafiotis, 2003]. SPE is an enormous step in computational efficiency over MDS, and is faster than Isomap. SPE is used in different applications and has succeeded in getting satisfactory results [Agrafiotis et al., 2010] [Ameer and Jacob, 2012]. The objective of SPE is to find representation that has points distances that are identical to their corresponding distances in high-dimensional data sets. The method starts by selecting a random point from original data, in time t, to be projected in the low-dimensional space.

Projected space starts with initial coordinates, and is updated iteratively by placing y_i and y_j onto the projected space in such a way that their Euclidean distance ($d_{ij} = ||y_i - y_j||$) is close to the corresponding distance ($r_{ij} = ||x_i - x_j||$) in original high-dimensional data sets. Thus, SPE minimizes the following Equation:

$$Stress = \sqrt{\frac{\sum_{i<j}(d_{ij} - r_{ij})^2 / r_{ij}}{\sum_{i<j} r_{ij}}}. \tag{2.12}$$

The points in projected space are updated according to the following constraint:

$$\begin{aligned} &if\ (r_{ij} \leq r_c)\ or\ ((r_{ij} > r_c)\ and\ (d_{ij} < r_{ij})) \\ &y_i \leftarrow y_i + \lambda(t)\frac{r_{ij} - d_{ij}}{d_{ij} + \varepsilon}\ (y_i - y_j) \\ &y_j \leftarrow y_j + \lambda(t)\frac{r_{ij} - d_{ij}}{d_{ij} + \varepsilon}\ (y_j - y_i) \end{aligned} \tag{2.13}$$

where $\lambda(t)$ is learning rate at t time, and r_c is a fixed circular radius of neighbourhood points.

In order to increase the efficiency of SPE, its modified version [Rassokhin and

Agrafiotis, 2003] selects one random point y_i to update all the remaining points, which are satisfy the following:

$$\begin{aligned} &if\ (r_{ij} \leq r_c)\ or\ ((r_{ij} > r_c)\ and\ (d_{ij} < r_{ij})) \\ &\qquad y_j \leftarrow y_j + \lambda(t)\frac{r_{ij} - d_{ij}}{d_{ij} + \varepsilon}\ (y_j - y_i) \end{aligned} \tag{2.14}$$

Figure 2.5(b) showed how the SPE unfolds the Swiss Roll data sets, where the distances among projected space points are preserved with their corresponding distances in the original data sets. However, the difficulty with using SPE is in determining the value of r_c. The results will be torn and very bad if r_c is very small, and SPE will be equivalent to MDS when r_c is very large.

The stochastic neighbour embedding (SNE) method is an iterative nonlinear method that attempts to preserve structural properties between low-dimensional space pairwise data points with their corresponding distance in high-dimensional data sets. It computes asymmetric probability p_{ij} between two neighbour points x_i and x_j in high-dimensional data sets.

$$p_{ij} = \frac{e^{-||x_i - x_j||^2/2\sigma_i^2}}{\sum_{k \neq i} e^{-||x_i - x_k||^2/2\sigma_i^2}} \tag{2.15}$$

where $p_{ii} = 0$, and σ_i is variance of Gaussian centered around x_i set by user.

Asymmetric probability q_{ij} of the corresponding points in low-dimensional space y_i and y_j of original high-dimensional points x_i and x_j are computed by:

$$q_{ij} = \frac{e^{-||x_i - x_j||^2}}{\sum_{k \neq i} e^{-||x_i - x_k||^2}} \tag{2.16}$$

where $q_{ii} = 0$.

SNE attempts to find low-dimensional space by matching p_{ij} and q_{ij} as much as possible by minimizing stress function in Equation 2.17 instead of using squared differences between p_{ij} and q_{ij}:

$$S = \sum_i \sum_j p_{ij}\ log\ \frac{p_{ij}}{q_{ij}} \tag{2.17}$$

and S can be minimizing by using a gradient descent method. Figure 2.5(c) showed

the ability of SNE in unfolding Swiss Roll data sets in a satisfactory projected space. However, some of the points are overlapped with others that causes to loss some information.

2.2.3 Local Nonlinear Methods

Locally linear embedding (LLE) is a nonlinear dimension reduction method that attempts to project the nearby neighbourhood points to a locally linear projected space. LLE uses three steps to do this task; first, it finds k nearest neighbourhood points for each point in high-dimensional data sets. Second, it computes a weight matrix among neighbours W by minimizing Equation 2.18. Weights among neighbours represent a value of strength relation among them.

$$\varepsilon(W) = \sum_{i}^{n} \|x_i - \sum_{j \neq i} w_{ij} x_j\|^2 \tag{2.18}$$

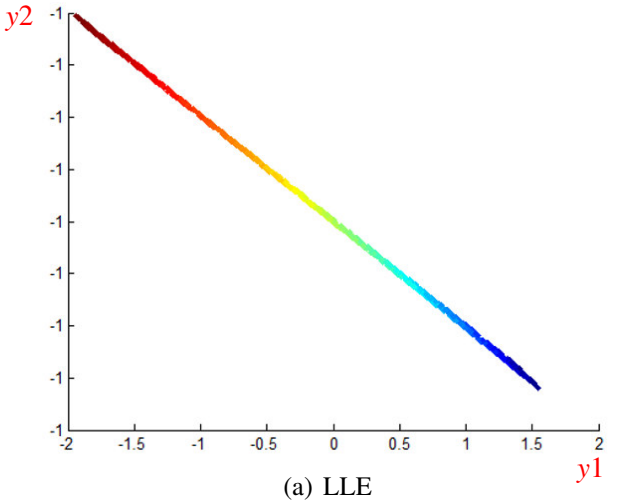
where $w_{ij} = 0$, if $x_j \neq x_i$. $\forall i, \sum_j w_{ij} = 1$. In the final step, LLE lies the D dimensions of original data into d dimensions space, where $d \ll D$. It uses weight matrix W to find the coordinates representation of low-dimensional space in order to preserve the topology properties of the original data. Based on locally linear reconstruction, LLE minimizes the following equation to find the projected space

$$\Phi(Y) = \sum_{i}^{n} \|y_i - \sum_{j \neq i} w_{ij} y_j\|^2 \tag{2.19}$$

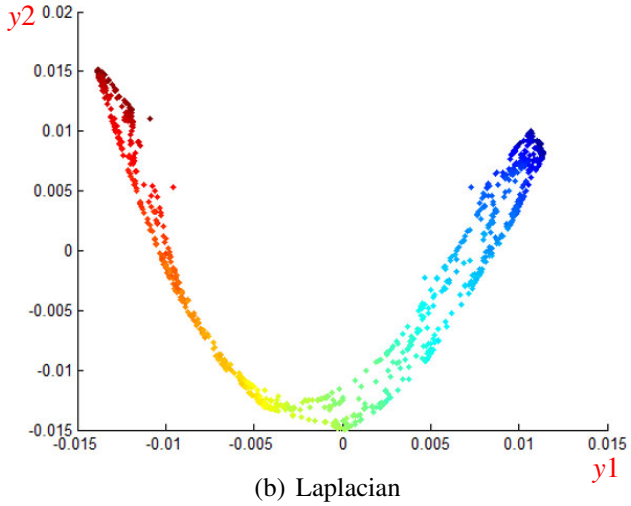
subject to $\frac{1}{N}\sum_i y_i = 0$, and $\frac{1}{N} Y^T Y = I$.

As an example, Figure 2.6(a) shows that LLE can unfold the three dimension of the Swiss Roll data sets. However, in many applications, LLE has poor generalization because it is not dynamic [Zeng and Luo, 2008]; it forces projected space to have a representation according to the weight matrix of high-dimensional data sets. Thus, LLE is poor in dealing with more complex data sets.

The Laplacian Eigenmaps (LE) method is similar to LLE and finds low-dimensional space by preserving the local properties of the original data [Belkin and Niyogi, 2003] [Tu et al., 2012]. Low-dimensional space is generated by preserving the distance



(a) LLE



(b) Laplacian

Figure 2.6: *LLE and Laplacian methods can unfold the Swiss Roll data sets, in Figure 2.3, in a satisfactory low-dimensional manifold.*

among k nearest neighbouring data points. Each data point gives weight to their neighbours, where the weight of first nearest neighbour is higher than second neighbour weight. That means large weight w_{ij} indicates the distance between y_i and y_j is very small. The steps of applying the Laplacian Eigenmaps method are:

1. It constructs a neighborhood graph G by connecting each data point y_j is connected to k nearest neighbors.

2. The weights between neighbors are computed by using Gaussian kernel func-

tion:

$$w_{ij} = e^{-\frac{||y_i - y_j||^2}{2*\sigma^2}} \tag{2.20}$$

where σ is the variance of the Gaussian.

3. The low dimension space is constructed. First computes diagonal weight matrix $D_{ii} = \sum_j w_{ij}$, then laplacian matrix is computed $L = D - W$. Thus, low dimension space is computed by minimizing the following cost function:

$$\sum_{i,j} (y_i - y_j)^2 \, w_{ij} \tag{2.21}$$

In order to ensure the projected data points are close to their corresponding data points in the original space, the following equation should be satisfied [Belkin and Niyogi, 2003]:

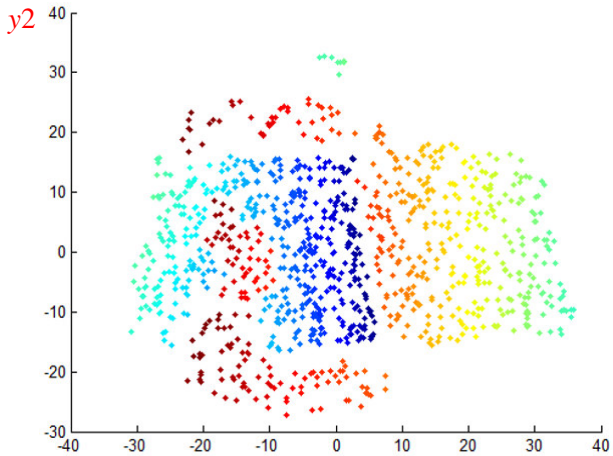
$$\sum_{i,j} (y_i - y_j)^2 \, w_{ij} = 2y^T Ly \tag{2.22}$$

2.2.4 Trustworthiness Methods

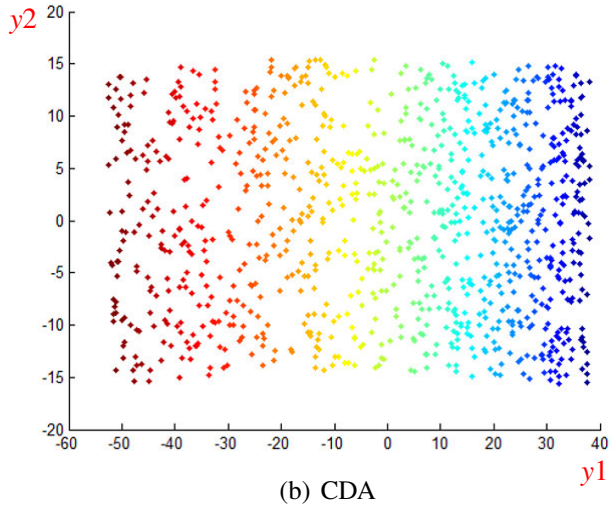
The curvilinear component analysis (CCA) method preserves the pairwise distances in the low-dimensional space with their corresponding pairwise distance in the original high-dimensional data sets by minimizing the following cost function:

$$\phi(Y) = \sum_{i<j} (r_{ij} - d_{ij})^2 F(d_{ij}, \lambda_t) \tag{2.23}$$

where $r_{ij} = ||x_i - x_j||$ and $d_{ij} = ||y_i - y_j||$ are the Euclidean distances between data points i and j in original high-dimensional and low-dimensional spaces, respectively. F is a bounded decreasing function, and allows CCA to preserve the distances on different scales depending on the time dependent value of λ_t which is started with large value to cover all data points, and then gradually decreased throughout processing. It is defined as:



(a) CCA



(b) CDA

Figure 2.7: *Trustworthiness methods, CCA and CDA, can unfold the Swiss Roll data sets. CDA is better than CCA because the last one generates tearing low dimension manifold.*

$$F(d_{ij},\lambda_t) = \begin{cases} 1, & \text{if } d_{ij} \leq \lambda_t \\ 0, & Otherwise \end{cases} \tag{2.24}$$

CCA can find the projected space of some nonlinear data sets, but it fails with well-known nonlinear data sets, as in the unfolding Swiss Roll data sets in Figure 2.7(a). Thus, preserving the neighbours distance is not guaranteed by CCA.

The curvilinear distance analysis (CDA) method is a version of CCA, and attempts

to preserve the pairwise distances in the low-dimensional space with their corresponding pairwise distance in the original high-dimensional data sets [Lee et al., 2004]. Like Isomap, CDA uses geodesic distance to keep the preservation between two distances. CDA uses the same CCA Equations in 2.23 and 2.24, but r_{ij} is the geodesic distance between data points i and j in the original high-dimensional data sets, and d_{ij} is the Euclidean distance between data points in the low-dimensional space. The general steps of applying CDA are:

1. If the data sets are very large, select n random data points. Otherwise, all data points are selected.
2. Constructing the neighbourhood graph by connecting each data-point to its k nearest neighnours.
3. The shortest paths among data points are computed by using Dijkstra's algorithm [Clark and Holton, 2005] to construct geodesic distance matrix.
4. Applying CCA algorithm using processed geodesic distances.
5. If step 1 is applied, the interpolation is applied on the remaining data points to generate final projected space.

CDA can find unfolded space of traditional nonlinear data sets, such as Swiss Roll in Figure 2.7(b). The preserving original information by CDA is better than other nonlinear dimension reduction methods, especially with complex data sets. It prevents points in the projected space from overlapping [Li, 2004]. Thus, geodesic distance with the efficiency of CCA gives the CDA more performance. However, the low-dimensional manifold might be torn when the neighbourhood relation of the original high-dimensional data sets are lost.

2.3 Information Visualization

Information visualization is a way to present the information in a way that allows people to see things that were unseen before. It makes the information easy to share among the people by asking each other, "Do you see this thing?" In addition, it leads to more enhancement, such as, "If we made an enhancement to that area, what would

happen?" Recently, information visualization has become the art of science, where there are many methods that can be used to help the people understand the meaning of the data by observing the proximity relationship among data points in a projected space. The colour of each pixel in the visualization is a compendium of information in the original data in which their most salient features are captured [Kaski and Peltonen, 2011] [Peltonen, 2009] [Lespinats et al., 2007].

Modern visualization tools have improved our ability to study many things directly from data. While there are different types of data, the way in which we visualize them are also different [Telea, 2008]. Visualizing scalar data is a popular way that is used in different sciences, where colours are mapped to the points. The visualization uses the same geometry as the original scale data, which can be 1D, 2D or 3D. The main advantage of this method is its simplicity and its ease of understanding. The visualization of vector field data sets has become more interesting in recent years. Vector data is scalar data with description. For example, to visualize fluid data sets, arrows, streamlines or animation are used to represent the direction. The visualization of vector field data sets is based on the dimensionality of the data sets. Three-dimensional vector data sets are a much more challenging problem, and the challenge increases when adding more descriptions (dimensions). The reconstruction of a three-dimensional volume model from a sequence of two-dimensional image slices of the human body is an example of volume rendering, which is helps in planning treatment or surgery.

Information visualization faces many challenge. For example, the visualization of flow data is difficult. Its visualization requires us to use different things to imitate the fluidity of the flow. In addition, the visualization of a connected network requires us to represent the relationships between nodes. However, finding a good representation containing all the information is a challenge. Many visualizations of data sets are pretty, but they do not show the important information. More powerful tools are required to meet the users needs. For example, the parallel coordinates method is used to represent high-dimensional data points into 2-dimensional space by using parallel axes. Although this method allows us to view the useful information, it is limited when it comes to representing large data sets. Information visualization can be used for different purposes:

1. Explanation: The data sets are visualized to explain something. For example, it is used to convince the viewer that one solution is better than another.

2. Exploration: The visualization of scientific data sets helps the researcher to explore the relationship between data sets.

3. Expression: Visualization is a way to represent data sets in a pretty way without giving more detail, in that we are focusing on the aesthetics.

The DR method is a strategy used to visualize a high-dimensional data sets by projecting a high-dimensional data set onto a low-dimensional space where it can be visualized directly [Schreck et al., 2010]. The problem of DR methods arises in discovering the low-dimensional space in the high-dimensional complex data sets. In general, there are two types of DR methods: continuity and trustworthiness. Continuity methods, such as Isomap and SPE, attempt to inherit the relations among global neighbourhood points to local corresponding points in projected space. Although this procedure has advantages in strengthening the coherence of local corresponding points, continuity errors may occur in low-dimensional manifold. Continuity errors mean the nearby neighbourhood data points in the original space can be projected further away in the projected space; they cause projected space to tear. On the other hand, trustworthiness methods, such as CCA and CDA, depend on the point relations in low-dimensional manifold, rather than those relations in original space, to find low-dimensional manifold. The points coordinates can be updated through a projection process in low-dimensional manifold in a flexible way without constraints to preserve the structure of original space. CCA and CDA might face false neighbourhoods in their low-dimensional manifold [Lespinats and Aupetit, 2009] [Lespinats and Aupetit, 2011].

In false neighbourhoods, the farther away data points in original space can be projected nearby in the projected space; they cause the projected space to be overlapping, as in Figure 2.8. Some techniques, as in [Kaski et al., 2003] and [Venna and Kaski, 2005], are used to overcome the false neighbourhood problems by focusing on the projected space distance to preserve the original distance. The researchers in [Kaski et al., 2003] fixed the false neighbourhoods by sending some of them away to improve the trustworthiness of projected space. Although some regions of their projected space are improved by this method, the final visualization might be tearing because the discarded data points left holes in their location when sending them away. In visualization, the false neighbourhoods are more dangerous than the continuity errors because of the

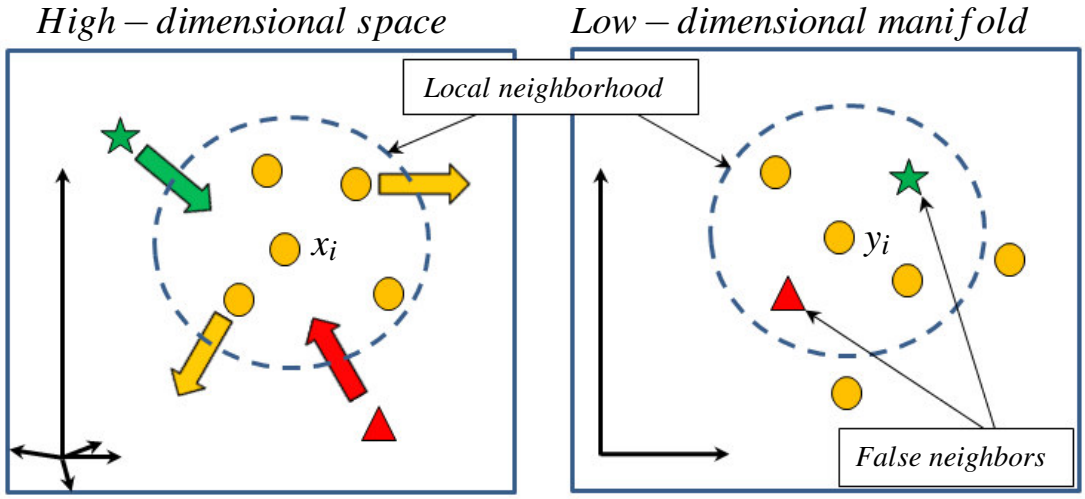


Figure 2.8: *DR might cause the points which are outside local neighborhood in high dimension space to be inside local neighborhood in low dimension manifold. These points are called false neighborhood.*

points that take the incorrect colours are unrelated to the original information. Thus, the colours of visualization have no meaning because the straightforward relationship with original high-dimensional data sets is lost [Bachmann et al., 2005].

Remote sensing imagery is a well-known technique to observe the earth and urban scenes by producing a large number of spectral bands [Smith, 2012]. However, the challenge is how to display the abundant information contained in these images in a way that is more interactive and easy to analyse, in a 3-D image cube, for example, by a user [Tyo et al., 2003]. Due to the difficulty of using these bands which are greater than 100, several DR algorithms are produced to overcome this problem by finding the better relationships among colour values in three colour channels after applying complex formulas to shrink the dimensionality of the original space. The precision of the results depends on the type of algorithm used, which should preserve and gather the relation among neighbours in the original space [Venna and Kaski, 2001], [Onclinx et al., 2009]. DR provides a good way to visualize remote sensing imagery by generating its colour image [Du et al., 2008] [Cui et al., 2009] [Bachmann et al., 2006] [Kotwal and Chaudhuri, 2010] [Mahmood and Scheunders, 2011] [Chen and Zhang, 2011] [Mohan et al., 2007].

The visualization of connected networks is a sub-field of information visualization

that is used to visualize a set of nodes with their edges. A connected network is a simple model that directly explains the features without more interpretation. It has been used in various fields such as computer science, biology, art, and psychology, because of its ability to represent a system in a simple structure with a set of nodes and edges. Recently, the size of data has grown exponentially, and representing this large data in a network is essential in order to present the idea to the viewer in an easy way. However, the study and visualization of large connected networks is difficult, and the analysis of large connected networks might not be feasible where the time complexity might be greater than $O(n^3)$. The understanding of a large connected network is essential; therefore, embedding it into a simple representation is necessary. The challenge involving an embedded connected network is: what is an optimum projected space that can represent this network in different tasks in such a way as to effectively deliver information to the viewers? Many DR methods have been introduced for connected network visualization in order to discover meaningful representations that will aid the reader to see and easily explore the relationship among items [Hu, 2006][Dwyer et al., 2007][Harel and Koren, 2002].

2.4 Quality of Visualization

When measuring the quality of visualization for a given data sets, it is important to know which DR method is suitable for the task at hand. Furthermore, the user cannot compare the quality of a given visualization with the original data by visual inspection due to its high-dimensionality. Thus, the formal measurements should evaluate the amount of the preserving neighbourhood colour distances in the visualization with their corresponding distance in original data. Residual variance (*Stress*), correlation (γ) and local continuity (LC) are the well-known metrics used in this matter. If we suppose X is a vector of all points of the data sets in original space and Y is a vector of all the corresponding points in projected space. A and B are the vectors of all pairwise distance of X and Y, respectively, then:

Residual Variance (*Stress*) is a metric used to compute the standard error of difference between visualization and original space [Tenenbaum et al., 2000]. It calculates the sum of squares of differences between original data point distances and projected

colour distances, as in the following equation:

$$Stress = \sqrt{\frac{\sum_{i=1}^{N}(a_i - b_i)^2}{N-2}} \tag{2.25}$$

where $a_i \in A$, $b_i \in B \ \ \forall\ 1 \leq i \leq N$.

Small stress value indicates that visualization has very little error and higher efficiency in preserving the original information.

Correlation function (γ): this metric computes the linear correlation between original input distances and colour distances in visualization [Mignotte, 2012]. The value of correlation is equal to 1 when all distances are perfect preserved, where positive slope between two vectors with perfect linear. In the other hand, the value equal to -1 if the two vectors have prefect linear relationship with negative slope. The correlation metric is defined as the follow:

$$\gamma = \frac{A^T\ B/|A| - \overline{A}\ \overline{B}}{\sigma_A\ \sigma_B} \tag{2.26}$$

where $|A|$ is the number of components in A, and $\overline{A}$ and σ_A are the mean and standard deviation of A, respectively.

Local Continuity (LC) computes the degree of similarity between two corresponding nearest neighbours sets in projected and original spaces [Chen and Buja, 2009]. The average of all cases represents the efficiency measurement of a projected space. Formally, let k nearest neighbours set in original space to data point i is $\mathcal{N}_k^X(i) = \{j_1, j_2, ..., j_k\}$, and the k nearest neighbours set to i in projected space is $\mathcal{N}_k^Y(i) = \{l_1, l_2, ..., l_k\}$. The measurement of overlapping between two sets is evaluated by

$$N_k(i) = |\mathcal{N}_k^X(i) \cap \mathcal{N}_k^Y(i)| \tag{2.27}$$

$N_k(i)$ is normalized to the [0,1] interval in order to compute the faithfulness measure of i:

$$Faithfulness_k(i) = \frac{1}{k} N_k(i) \tag{2.28}$$

In this case the faithfulness value of projected space will be :

$$Faithfulness_k = \frac{1}{N}\sum_{i=1}^{N} Faithfulness_k(i) \tag{2.29}$$

where N is total number of data points in data sets.

2.5 Conclusion

The objective of using DR is to transfer the high-dimensional data sets to the low-dimensional space so that it is maintaining the essential original information. The simplest approach is to use linear methods, but the complexity of the recent data sets make these methods useless. Thus, the nonlinear versions of those linear methods were introduced to overcome their limitations. Nonlinear transformations can be used to project a high-dimensional data sets in a low-dimensional space. Although this method succeeded in solving some problems, some nonlinear DR methods perform poorly on complex data sets, such as LLE. Nonlinear DR methods, such as Isomap and CDA, which are used in geodesic distance have proven their ability to deal with different complex data sets. The consuming time and storage when using geodesic distance reduces their efficiency. The SPE is a good method that can be used with large data sets because of its ability to converge the results in a short time. Most of the DR methods suffer from false neighbourhood and continuity errors, especially with large dimensionality of data sets where stopping the loss of original information becomes impossible. Thus, the future methods for DR should overcome these errors as much as possible, as we will show in the next chapter.

Chapter 3

Visualization of Remote Sensing Imagery Data Sets Using Faithful Stochastic Proximity Embedding

In this chapter we will address the following topics:

1. Limitation of DR methods with large data sets, as remote sensing imagery.
2. Methodology of Faithful Stochastic Proximity Embedding (*FSPE*).
3. Point-wise metric to show efficiency of visualisation as gray image.
4. Experimental results of *FSPE* application on remote sensing imagery.
5. Conclusion.

Here we present a new nonlinear DR method which is called Faithful Stochastic Proximity Embedding (*FSPE*) to visualize remote sensing imagery data sets. *FSPE* overcomes the main shortcomings of the DR by sending the false neighbour points away, and preserving the neighbourhood relation to the true neighbours points, which are inside the local neighbourhood. The visualization of our proposed method displays the faithful, useful and meaningful colours, where the objects of the image can be easily distinguished. Moreover, a point-wise metric is introduced to measure the quality of each pixel in the visualization. The performances of *FSPE* and PCA, SPE,

Isomap, CDA and LLE are compared, and the efficiency of the proposed method in both visualization accuracy and computational cost is shown.

3.1 Introduction

A variety of DR methods are used in the visualization of remote sensing imagery data sets. Although linear methods, such as PCA [Cui et al., 2009] [Tyo et al., 2003] and MDS [Steyvers, 2002], are widely used, their ability has been reported only for linear data sets and they perform poorly on nonlinear data sets because of their nonlinear structure [Dianat and Kasaei, 2010] [Bachmann et al., 2006]. The problem of modelling a nonlinear data sets have been addressed by nonlinear methods, as Isomap [Tenenbaum et al., 2000] [Bachmann et al., 2005] [Bachmann et al., 2006]), CDA [Lee et al., 2004], LLE [Mohan et al., 2007] [Roweis and Saul, 2000] and SPE [Rassokhin and Agrafiotis, 2003], in which they attempt to preserve the intrinsic geometry of the nonlinear data sets.

The relationship between the amount of reduction and efficiency is inverse, where the efficiency is reduced when the dimensionality of low-dimensional space is very low, and vice versa. Inefficiency indicates the inability of the DR method to preserve the original information in low-dimensional space. Although, in general, visualization of a remote sensing imagery data sets requires that the dimensionality of low-dimensional space is equal to three [Cui et al., 2009], [Du et al., 2008], the efficiency of visualization is reduce because the relationship between the neighbours are loose, as in Figure 3.1. Thus, some of the colours in the visualization of a remote sensing imagery data sets do not represent the correct relations derived from the original system, and, therefore, those colours are false colours. On the other hand, in Figure 3.1, we noted that the preservation of the original information is increased when the dimensionality of low-dimensional space is increased.

The goal of this chapter is to introduce a faithful version of SPE which shows faithful behaviour; we call it *FSPE*, which maintains the main advantages of original SPE and overcomes false neighbourhoods as much as possible. *FSPE* is a general method and can be used to visualize the remote sensing imagery data sets in a higher quality visualization. The main idea of our approach is to use projected space as a key element in guiding projection by specifying the local neighbourhoods rather than a global

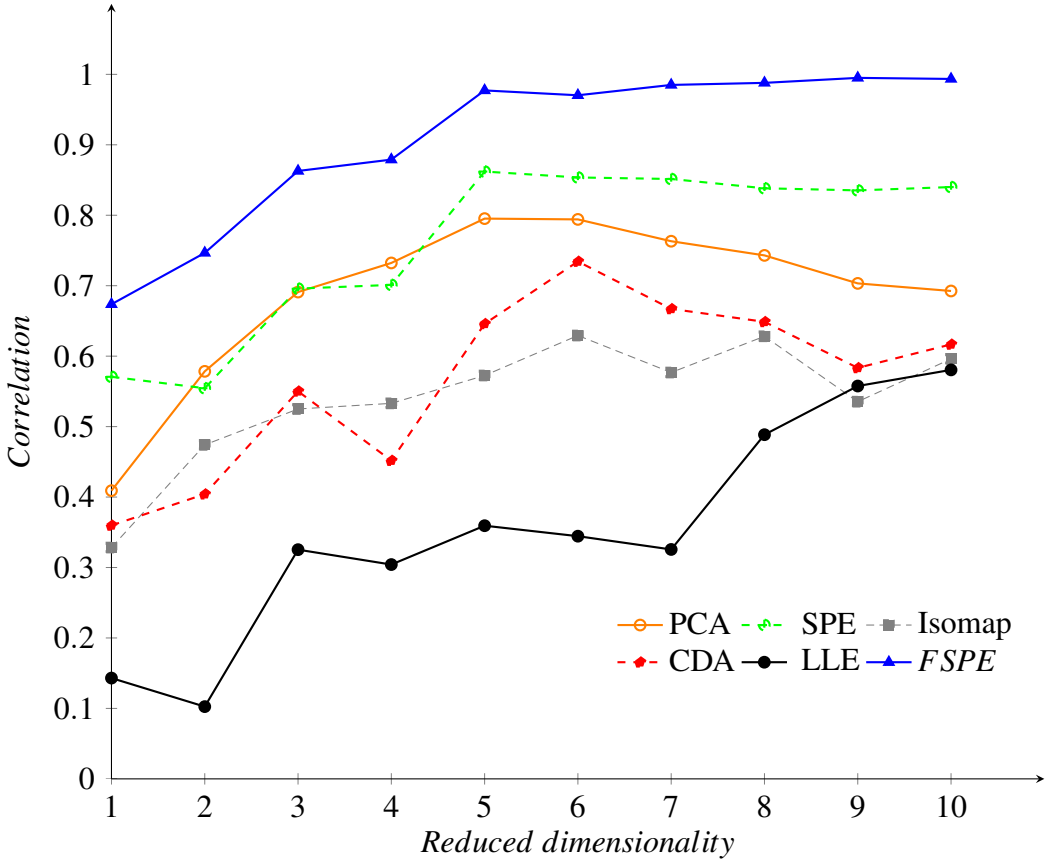


Figure 3.1: *Distortions cannot be overcome when the amount of reduction is equal to 3 or less. Six dimension reduction methods are used to reduce the size of data from 224 to 10, 9, ... , 3, 2, or 1. By using correlation measurement [Cui et al., 2009] [Jacobson and Gupta, 2005], we note that when the amount of reduction is few (3 or less), the efficiency of the solution will be much lower than when the reduction is greater than 3. This indicates that the preserving of the relation among bands of remote sensing imagery data sets (of AVIRIS Moffet Field data sets from the southern end of San Francisco Bay, California, done in 1997) becomes very difficult when the dimensionality of low dimension space is very low.*

neighbourhood. We use a decreasing local neighbourhood radius to incorporate the convergence of projected space, where the calculating distances are reduced through projection. In addition, a new point-wise metric is introduced to measure the faithfulness of each pixel in the visualization, rather than other metrics that give one measurement value [Cui et al., 2009] [Jacobson and Gupta, 2005]. Our measurement will be helpful to the user in locating the unfaithful locations in the visualization, thereby identifying the pixels whose colours are not representative of the corresponding points' relationships in the original dimension space.

3.2 Visualization of Remote Sensing Imagery Data Sets by Dimensionality Reduction

The goal of DR is to find a visualization that has pixel distances that are identical to their corresponding points in the multi-bands of the remote sensing imagery data sets. Let us suppose there is an image with h bands denoted as $X = \{x_1, x_2, ..., x_h\}$; each band has m pixels, and DR goal is to find a projected space has l bands $l \ll h$ and $l = 3$. DR attempts to place y_j onto the projected space in such away that their euclidean distance $d_{ij} = ||y_i - y_j||$ is closed to the corresponding distance $r_{ij} = ||x_i - x_j||$ in original high dimensions space. That means, DR will minimize the following stress function:

$$f = \sum_i \sum_{j \neq i} (r_{ij} - d_{ij})^2 \ W(\varphi) \tag{3.1}$$

where φ is r_{ij} or d_{ij}. In general, there are two assumptions of a weight function $W(\varphi)$ [France and Carroll, 2011]. The first assumption is supposed by the continuity method, which is $W(r_{ij}) = 1$. Linear continuity DR methods, such as PCA and MDS, use Euclidean distance r_{ij} to measure the dissimilarity between two data points in a space [Cox and Cox, 2000]. While Euclidean distance cannot measure the correct distance on a nonlinear data sets, nonlinear continuity methods, such as Isomap, use the concept of geodesic distance r_{ij}, which is suitable to compute the shortest path between two points that lie on a surface [Tenenbaum et al., 2000].

On the other hand, the trustworthiness methods suppose $W(d_{ij}) = 1$ if $d_{ij} \leq d_c$ and $W(d_{ij}) = 0$ if $d_{ij} > d_c$ for a decreasing neighbour radius d_c. In CCA [Demartines and Hrault, 1997], the distance r_{ij} is the Euclidean distance between data points i and j in original high-dimensional and low-dimensional spaces. CDA uses the same idea, but r_{ij} is the geodesic distance between points [Lee et al., 2004].

3.3 Faithful Stochastic Proximity Embedding

3.3.1 Stochastic Proximity Embedding (SPE)

Geodesic distance is better than Euclidean distance in recognizing the difference between near and far away neighbours, and it has good approximation to their distances.

Isomap attempts to preserve the geodesic distance r_{ij} in the original space with their corresponding linear distance in the projected space [Bachmann et al., 2006] [Silva and Tenenbaum, 2003]. However, geodesic distance is not always a benefit; a long consuming time and storage requirement make Isomap inappropriate for large data sets [Cui et al., 2009].

SPE is a nonlinear method that proceeds by calculating Euclidean distance for global neighbourhood points r_{ij} within a fixed neighbourhood radius r_c. Although the original version of SPE is faster than Isomap [Agrafiotis and Xu, 2002], it takes time to get acceptable results. It requires to call the random number generator twice for every stage of improving the coordinates [Rassokhin and Agrafiotis, 2003]. The modified version of SPE has been able to speed up the projection process by reducing the random selection. This new structure of SPE has become more convenient for implementation within the GPU.

The modified version of SPE algorithm iteratively refines the embedding by repeatedly selecting a point i randomly and using it to update the coordinates of every point $j \neq i$ in a manner similar to a stochastic gradient descent [Rassokhin and Agrafiotis, 2003] as follows:

$$y_j \leftarrow y_j + \lambda(t)\, S(r_{ij})\, \frac{r_{ij} - d_{ij}}{d_{ij} + \varepsilon}\, (y_j - y_i) \tag{3.2}$$

$$S(r_{ij}) = \begin{cases} 1 & if\ (r_{ij} \leq r_c)\ \vee\ ((r_{ij} > r_c)\ \wedge\ (d_{ij} < r_{ij})) \\ 0 & Otherwise \end{cases} \tag{3.3}$$

where $\lambda(t)$ is learning rate at time t, which is decreased over the time, ε is a tiny number used to avoid division by zero. As it happens in the other DR methods, the visualization by SPE might encounter DR problems, which make it distorted [Bachmann et al., 2005].

3.3.2 *FSPE*

Our method uses the faithfulness idea with SPE instead of its continuity to generate a faithful SPE (*FSPE*) method, which is able to deal with such complex problems as

remote sensing imagery data sets. The points in projected space are updated depending on their relation, as follows:

$$y_j \leftarrow y_j + \lambda(t)\, T(d_{ij})\, \frac{r_{ij} - d_{ij}}{d_{ij} + \varepsilon}\, (y_j - y_i) \tag{3.4}$$

$$T(d_{ij}) = \begin{cases} 1 & if\ (d_{ij}\ \leq\ d_c(t))\ \vee\ ((d_{ij} > d_c(t))\ \wedge\ (d_{ij} < r_{ij})) \\ 0 & Otherwise \end{cases} \tag{3.5}$$

where $d_c(t)$ is a decreased neighbourhood radius over time. *FSPE* starts iteratively, on the projected space, with selecting a random point in time t, which updates all the local neighbourhood points in a sufficient region within local neighbourhood radius $d_c(t)$, so the coherent structure will be constructed by sending false neighbours away; according to Equation 3.4. The local neighbourhood radius d_c starts with large value at t_0 to include all points, and then gradually this value is decreased through times by $d_c(t_0)/(t+1)$ to keep neighbourhood points with the improvement of projected space.

The differences between SPE and *FSPE* have concentrated on the inputs to their weight functions. The input of the $S(r_{ij})$, in the Equation 3.2, indicates the original space is used to compute the distance r_{ij} within a fixed neighbourhood radius r_c. On the other hand, *FSPE* uses projected space and a decreasing neighbourhood radius $d_c(t)$ in the definition of $T(d_{ij})$, in Equation 3.4, which makes the proposed method overcomes the DR problems, as we will see when we answer the following question:

How does the *FSPE* method work?

Let us suppose the point $p_1 \in Y$ is selected at step t of the projection iteration, and the radius of local neighbourhood of p_1 at this step is $d_c(t)$, as in Figure 3.2. While the false neighbourhoods are the farther away data points in original space are projected nearby in the projected space, according to the Equation 3.4, there are two main tasks of p_1 to overcome these errors. It sends the false neighbour points away, and preserves the neighbourhood relation to the true neighbours, which are inside the local neighbourhood. If some true points are missed, p_1 will not try to search for them because they will be returned back during the projection process.

To illustrate further, let us suppose there are another three points $p_2,\ p_3,\ p_4 \in Y$,

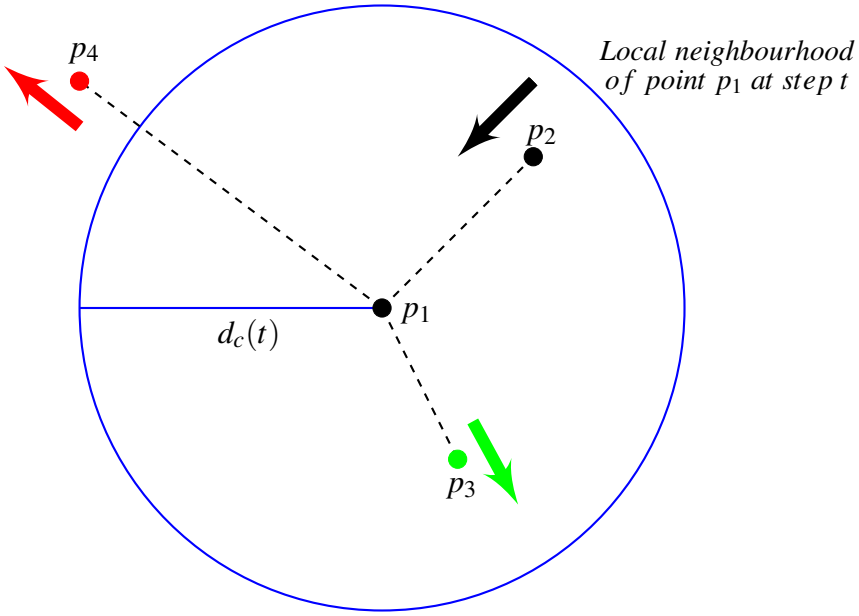


Figure 3.2: *The main idea of FSPE. At step t of iteration process. p_1 preserves its distance with true neighbours, as with p_2, and pushes away the false neighbour points, as with p_3. The points which are outside the local neighbourhood, which their ($d_{ij} < r_{ij}$), as with p_4, are pushed further away.*

where d_{12} & $d_{13} < d_c(t)$, and $d_{14} > d_c(t)$. On the first hand, if $d_{12} \geq r_{12}$, p_2 is a true neighbour point because $\frac{r_{12}-d_{12}}{d_{12}+\varepsilon}$ will be less or equal to zero. The coordinates of p_2 are updated by pulling p_2 to p_1, or leaving it without change, in order to preserve d_{ij} with r_{ij} as much as possible. Thus, if $(d_{ij} < d_c(t)) \wedge (d_{ij} \geq r_{ij})$, the *FSPE* does the pulling process.

On the second hand, if $d_{13} < r_{13}$, the value of $\frac{r_{13}-d_{13}}{d_{13}+\varepsilon}$ will be positive, which causes to push p_3 away because it is a false neighbour. While p_3 might be missing a true neighbour of another point, the updating of its coordinate causes to send it to the correct location. Therefore, if $(d_{ij} < d_c(t)) \wedge (d_{ij} < r_{ij})$, the *FSPE* does the pushing process.

The last possibility, which could face *FSPE*, is if $d_{14} > r_{14}$, the point p_4 is pushed further away because $\frac{r_{14}-d_{14}}{d_{14}+\varepsilon}$ will be greater than zero. Thus, if $(d_{ij} > d_c(t)) \wedge (d_{ij} < r_{ij})$, the *FSPE* does the pushing process.

3.4 Point-wise Quality Metric

While measuring the quality of visualization is important, the formal measurements should evaluate the amount of preserving neighbourhood colour distances in the visualization with their corresponding ones in the original data sets. The correlation coefficient metric, in Equation 2.26 is a well-known method used to compute the linear correlation between remote sensing imagery data sets distances and colour distances in visualization [Cui et al., 2009] [Jacobson and Gupta, 2005]. While this metric gives one measurement value $\in [-1,1]$, it is not enough to know the efficiency of the colour of each pixel in the visualization.

For the purpose overcome this weakness, our suggestion is to compute the correlation for each point through the following equation:

$$\gamma(i) = \frac{A_i^T\ A_i/|A_i| - \overline{A}_i\ \overline{B}_i}{\sigma_{A_i}\ \sigma_{B_i}} \tag{3.6}$$

where A_i is a vector of all pairwise distance of the data point i to the all other points in original space and B_i is a vector of its corresponding pairwise colour distances in the visualization. The result of this point-wise metric, in Equation 3.6, is a faithful image, which represents the degree of matching the colour distances in visualization with their corresponding distances in the bands of a remote sensing imagery data sets. It identifies the weak and strong faithful pixels, where the very dark pixels have the lowest faithful values, and the white pixels have the highest faithful values. The number of faithful colours will be very high in faithful images when the colour distances among pixels in visualization and their corresponding distance in original spaces are preserved. On the other hand, the faithful image has a lot of points that have low faithful values to indicate when the visualization has false colours, because their colour distances among pixels are not preserved with their corresponding pixels in the original space.

3.5 Experimental Results

All the methods are run on Intel(R) i7-930 2.80 GHz CPU with 12 GB memory on Windows 7. We ran our proposed method in Microsoft Visual Studio C++ 2008 with CUDA 4.2 and NVIDIA GeForce GTX 480 graphics card with a buffer size of 1 GB.

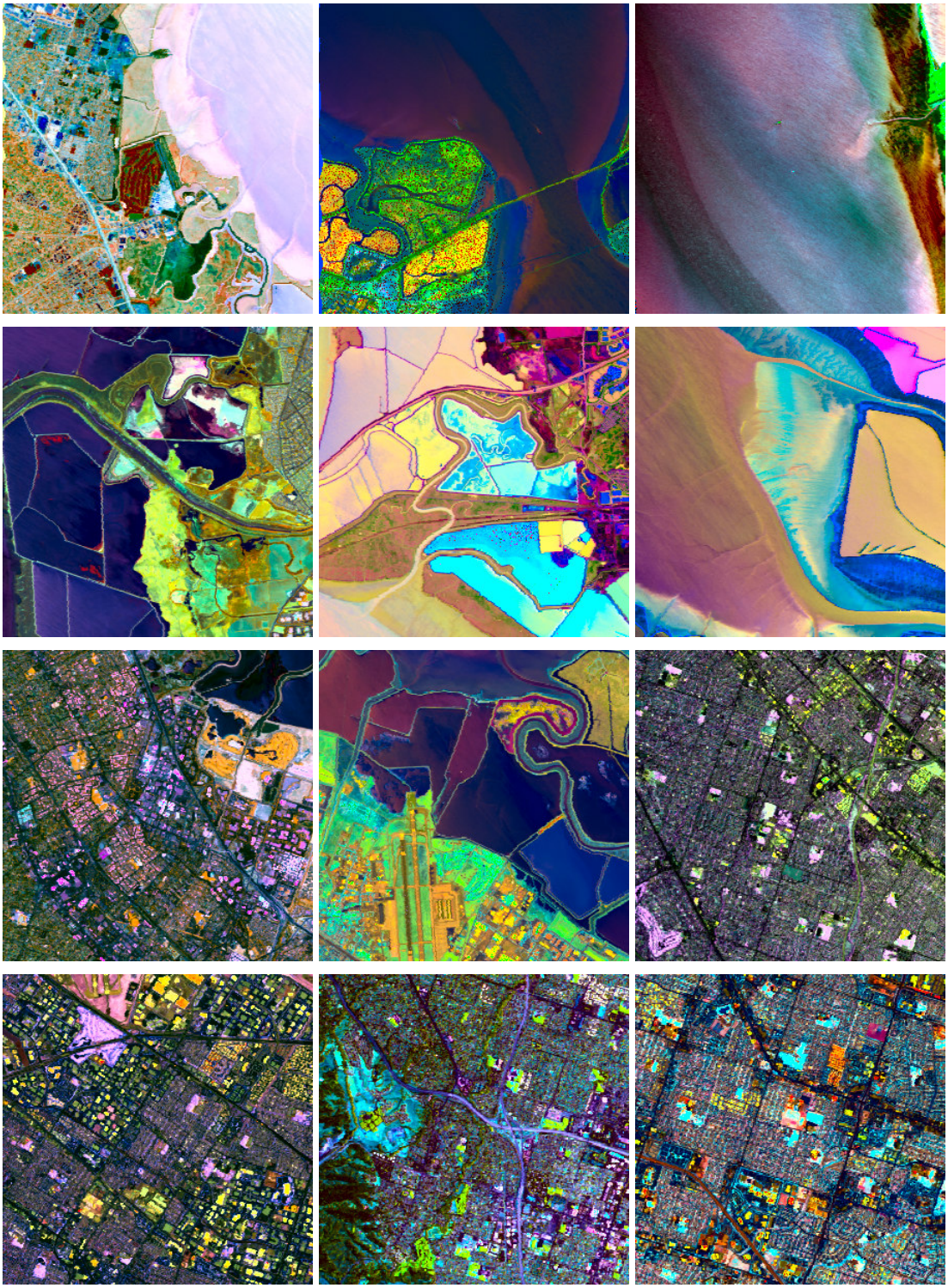


Figure 3.3: *The visualizations of 12 regions of VIRIS Moffet Field data sets from the southern end of San Francisco Bay, California, done in 1997 [AVIRIS, 2013]. The visualizations were done by FSPE (the proposed method).*

We used the AVIRIS Moffet Field data sets from the southern end of San Francisco Bay, California, done in 1997 [AVIRIS, 2013].

Because some methods cannot work with large data sets, we divided this data sets into twelve small regions, each one with 300x300x224 pixels (see Figure 3.3). For *FSPE*, the low dimension space is initialized randomly within [0,1] and $\varepsilon = 1\text{x}10^{-8}$. Learning rates λ and d_c are decreasing linearly and nonlinearly, respectively, through projection process, as in the following Equations:

$$\lambda(i) = \lambda(0) - \lambda(0)\frac{i}{ncycles} \tag{3.7}$$

$$d_c(i) = \frac{d_c(0)}{1+i} \tag{3.8}$$

where $ncycles$=1x10^3 is the maximum number of iterations, $i = 1, 2, ..., ncycles$, $\lambda(0) =$ 1.0 and $d_c(0)$=1 x10^3.

We compared *FSPE* with five methods: PCA, SPE, Isomap, CDA and LLE for all 12 regions. In the following experiments, the three channels of projected spaces, which are carried by PCA, SPE, Isomap, CDA, LLE and *FSPE*, are mapped in *CIE Lab* colour space. The generated colour image represents all information in 3D projected space in a manner that is suitable with the human vision system, and we will explain that in Section 3.5.2. The comparisons of *FSPE* versus PCA, SPE, Isomap, CDA and LLE are evaluated in quantitative and visual manners.

3.5.1 Quantitative Comparison

We used two measurement methods, which are correlation metric and LC, to measure the efficiency of the projected spaces of PCA, SPE, Isomap, CDA, LLE and *FSPE*. To give a clear analysis, the results of each method will be explained individually. Some methods depend on the parameters to get their results; for example, Isomap and LLE use the number of neighbours k, and SPE uses the neighbour radius r_c. Figure 3.4 shows the efficiency of methods by using different values of neighbourhood points, where $k = r_c$ for all values, and, in general, the best results when the size of neighbourhood points is equal to 50.

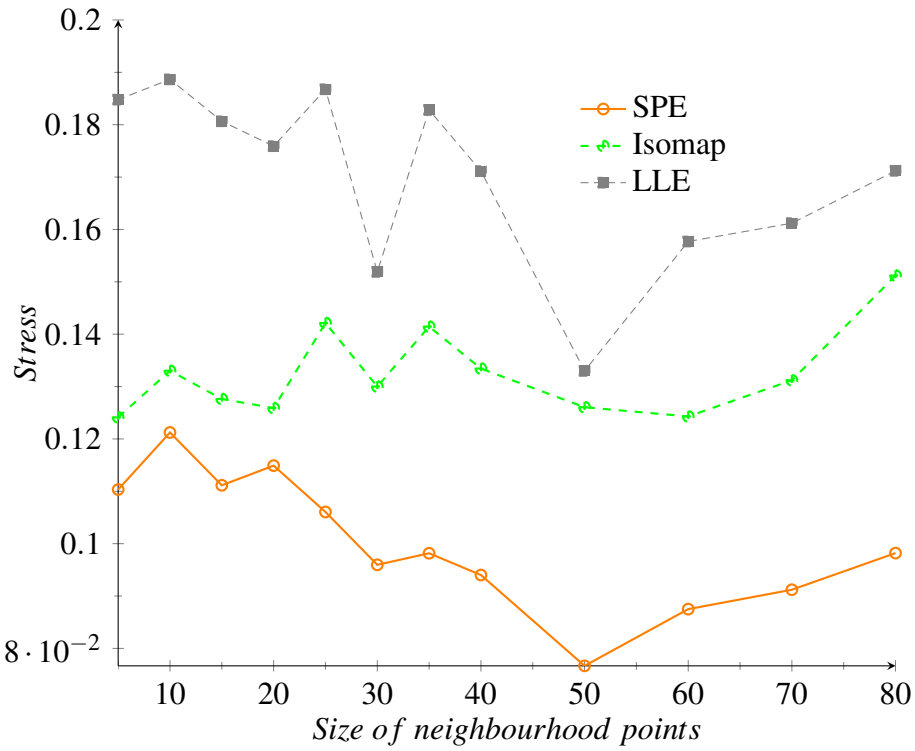


Figure 3.4: *The DR methods have different efficiency when using different values of neighbourhood points for Region_0.*

Correlation Metric:

We measured the efficiency of the visualizations by *FSPE* and five other methods by using the correlation metric. *FSPE* achieves better preserving the colour distance of visualization when compared with other methods. The order of the efficiency of the proposed method is the first in 5 regions and the second in 3 regions, as in Table 3.1. Moreover, the average efficiency of all 12 regions indicates that our proposed method is the better. When using the geodesic distance to compute the distance between points in the original space, *FSPE* has maintained its ability in preserving the original information, as in Table 3.2.

LC metric:

Table 3.3 shows the measurement of the results of PCA, SPE, Isomap, CDA, LLE and *FSPE*, where each value in it represents the best among 10 runs. *FSPE* achieved a better preserving of neighbourhood distances, where its visualizations were the better

Table 3.1: *By using Euclidean distance, correlation measurement values of the comparison between FSPE and PCA, SPE, Isomap, CDA and LLE for 12 regions. The higher, the better.*

	PCA	SPE	Isomap	CDA	LLE	*FSPE*
Mean	0.599	0.641	0.602	0.604	0.409	**0.717**
Region_0	0.691	0.696	0.525	0.550	0.325	**0.863**
Region_1	**0.901**	0.809	0.550	0.557	0.712	0.862
Region_2	0.469	**0.994**	0.817	0.900	0.136	0.904
Region_3	0.579	0.620	**0.800**	0.766	0.707	0.584
Region_4	0.673	0.641	0.646	0.648	0.368	**0.794**
Region_5	0.591	0.813	0.809	0.753	**0.885**	0.532
Region_6	0.653	0.504	0.562	**0.715**	0.516	0.620
Region_7	0.760	0.662	0.549	0.588	0.142	**0.809**
Region_8	0.442	**0.671**	0.414	0.128	0.277	0.619
Region_9	0.217	0.141	0.546	0.608	0.141	**0.867**
Region_10	0.520	**0.684**	0.560	0.487	0.324	0.360
Region_11	0.695	0.462	0.448	0.551	0.374	**0.786**

Table 3.2: *By using geodesic distance, correlation measurement values of the comparison between FSPE and PCA, SPE, Isomap, CDA and LLE for 12 regions. The higher, the better.*

	PCA	SPE	Isomap	CDA	LLE	*FSPE*
Mean	0.506	0.609	0.540	0.517	0.402	**0.719**
Region_0	0.535	0.575	0.640	0.643	0.780	**0.994**
Region_1	**0.900**	0.811	0.544	0.630	0.714	0.863
Region_2	0.410	**0.953**	0.740	0.428	0.147	0.857
Region_3	0.486	0.640	**0.894**	0.777	0.721	0.583
Region_4	0.672	0.636	0.642	0.374	0.390	**0.785**
Region_5	0.211	0.486	0.225	0.426	**0.539**	0.464
Region_6	0.648	0.514	0.405	**0.672**	0.167	0.634
Region_7	0.583	0.670	0.560	0.595	0.130	**0.815**
Region_8	0.480	**0.655**	0.477	0.427	0.331	0.637
Region_9	0.119	0.153	0.375	0.318	0.153	**0.890**
Region_10	0.500	**0.700**	0.389	0.375	0.414	0.396
Region_11	0.528	0.515	0.592	0.536	0.337	**0.716**

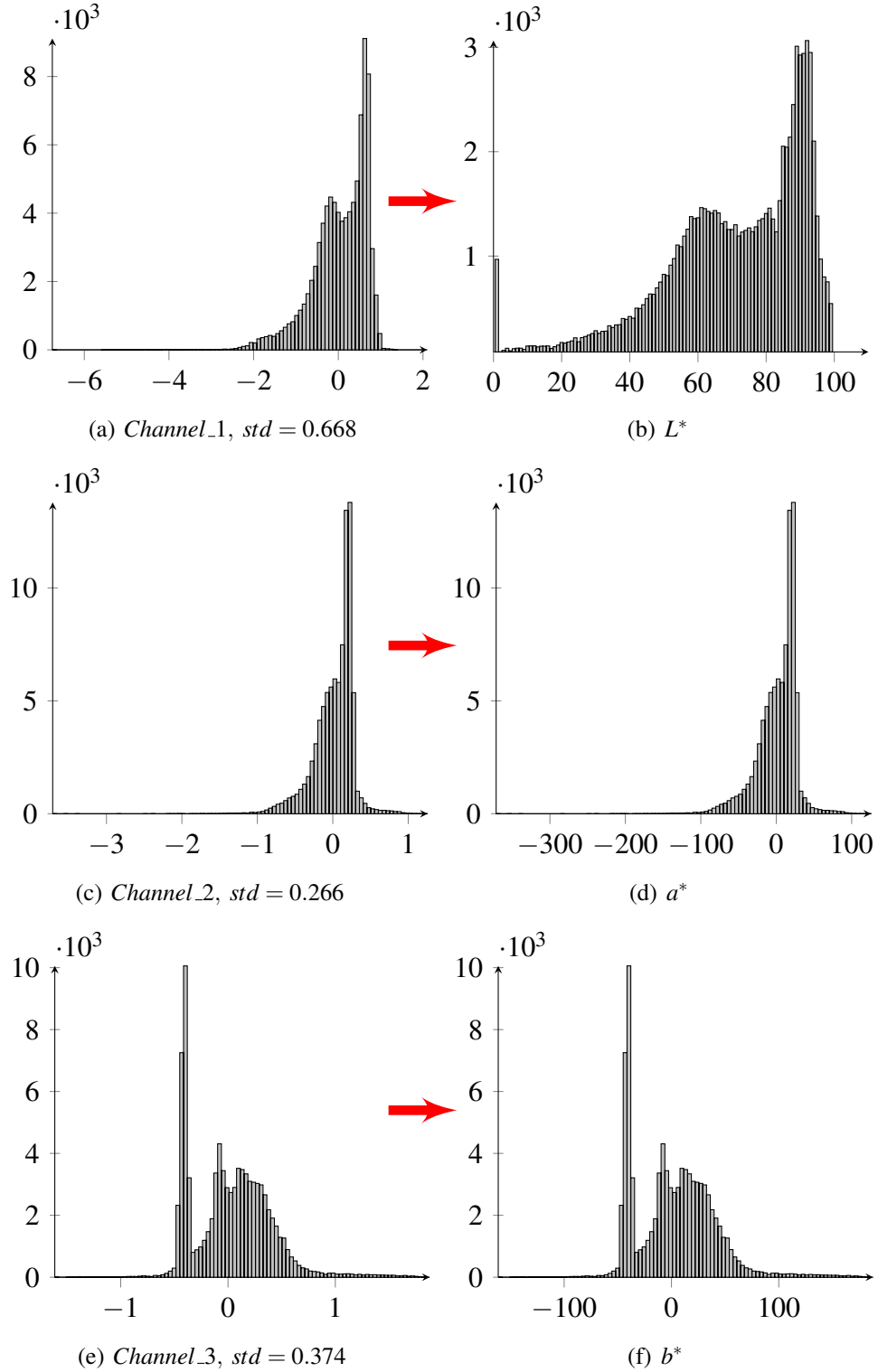
Table 3.3: *Performance of DR projections measured according to the LC. The higher, the better.*

	PCA	SPE	Isomap	CDA	LLE	*FSPE*
Mean	0.249	0.293	0.205	0.214	0.095	**0.347**
Region_0	0.203	0.288	0.262	0.206	0.108	**0.319**
Region_1	0.121	0.214	0.172	0.186	0.128	**0.311**
Region_2	**0.591**	0.580	0.180	0.160	0.110	0.295
Region_3	0.106	0.115	0.161	**0.242**	0.100	0.239
Region_4	0.381	**0.445**	0.144	0.230	0.118	0.440
Region_5	0.071	0.140	0.166	0.193	0.110	**0.334**
Region_6	0.176	0.196	0.182	0.233	0.126	**0.438**
Region_7	0.193	0.193	0.255	**0.285**	0.040	0.227
Region_8	0.061	0.082	0.182	0.196	0.046	**0.370**
Region_9	0.082	0.191	0.146	0.258	0.131	**0.497**
Region_10	0.500	**0.564**	0.300	0.187	0.052	0.342
Region_11	0.503	**0.504**	0.311	0.186	0.068	0.347

in 6 regions, and it got the highest mean of all 12 regions. For Region_0, Table 3.4 showed *FSPE* has fewer false neighbourhood points and a greater number of the true neighbourhood points, which are 2959 and 28,725, respectively. Table 3.5 explains the summation of the false neighbourhood points and also the true neighbourhood points in all regions. In this table, we can see the same scenario, where the better in all cases is *FSPE*.

FSPE has few number of false neighbourhood points, which have zero measurements; that is, PCA, SPE, Isomap, CDA and LLE in Region_0 because most of the data points by *FSPE* preserve their neighbourhood relation. Thus, *FSPE* gets a higher number of true neighbourhood and their mean of trustworthy measurement.

LC measurement showed that the *FSPE* achieves better visualization, when compared with the reset of other methods, because it preserves the distance among pixels in the original high-dimensional data sets with their corresponding colour distance in visualization as well. In general, the drawback of other methods is that having a high number of false colours leads to a reduction in their efficiency. In brief, we can see that our method generates a more robust coherence structure by preserving the colour distances of original space, whereas PCA, SPE, Isomap, CDA and LLE lose some of that distance preserving in their visualizations.

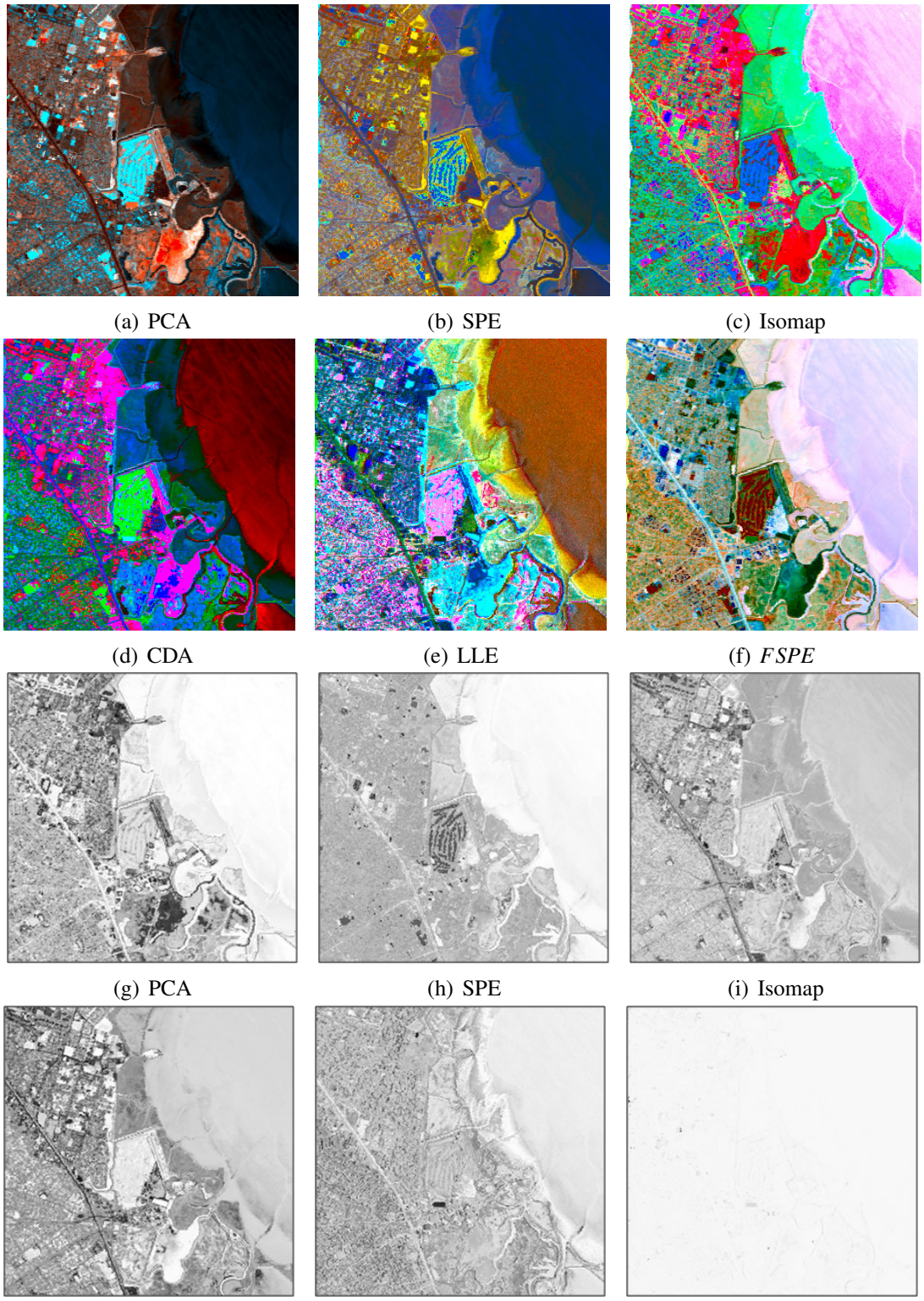


(a) *Channel_1*, $std = 0.668$ (b) L^*

(c) *Channel_2*, $std = 0.266$ (d) a^*

(e) *Channel_3*, $std = 0.374$ (f) b^*

Figure 3.5: *FSPE's projected space of Region_0 is projected into CIE Lab colour space. The largest standard deviation among three channels of FSPE's projected space is chosen to be L^* component, and the other channels are assigned to the a^* and b^* components.*



(a) PCA (b) SPE (c) Isomap

(d) CDA (e) LLE (f) *FSPE*

(g) PCA (h) SPE (i) Isomap

(j) CDA (k) LLE (l) *FSPE*

Figure 3.6: *Gray-scale images of point-wise correlation values. The top two rows show the colour display of a remote sensing imagery by PCA, SPE, Isomap, CDA, LLE and FSPE, respectively. The bottom two rows presents the corresponding gray-scale display of point-wise correlation values; the brighter, the better (i.e., higher).*

Table 3.4: *LC metric. FSPE has a few number of false neighbourhood points (the fewer is the better), which have zero measurements; that is, PCA, SPE, Isomap, CDA and LLE in Region_0 because most of the data points by FSPE preserve their neighbourhood relation. Thus, FSPE gets a higher number of true neighbourhood.*

Method	Mean of faithfulness (Standard deviation)	No. of true neighbour points	No. of false neighbour point
PCA	0.109 (0.130)	7164	45405
SPE	0.203 (0.187)	17927	12134
Isomap	0.240 (0.114)	12801	9643
CDA	0.219 (0.119)	9552	12080
LLE	0.107 (0.069)	13831	7779
FSPE	**0.259** (0.167)	**28725**	**2959**

Table 3.5: *For all Regions: by using LC metric, comparison among methods by using the number of false neighbourhood points (the fewer is the better), and the number of true neighbour points. Percentages of these points to other points are also shown, where the total number of points is 12 90000.*

Method	True neighbour		False neighbour	
	No.	Percent	No.	Percent
PCA	130821	12%	443713	41%
SPE	444179	41%	151133	13%
Isomap	130882	12%	239832	22%
CDA	105031	9%	191800	17%
LLE	18336	1%	273766	25%
FSPE	**488657**	**45%**	**17662**	**1%**

3.5.2 Faithful Visualization

For visualization of data sets, we use the *CIE Lab* colour space that is perceptually uniform [Mignotte, 2012] [Lawrence et al., 2011]. However, there are many possibilities for choosing the channels corresponding to the L^*, a^* and b^* components. Therefore, we chose the most varying one for the L^* component and the remaining two for the a^* and b^* components, followed by a simple linear stretching such that $L^* \in [0, 100]$ and both a^* and b^* have a zero mean. In order to result in a high contrast, the L^* component shifts its values to fill the entire brightness range, where a small percentage (1%) of the brightest and darkest pixels are ignored. The relations among channels are kept through scaling a^* and b^* by using the scaling parameter which were used to scale

L^*. Figure 3.5 shows the channels' histogram of *FSPE*'s projected space with their corresponding components of $L^*a^*b^*$ colour space.

As well as providing a quantitative measure of the projected space, the faithful grey scale image by point-wise correlation metric is important to examine the reliability of the colours in the visualization of the remote sensing imagery data sets. Figure 3.6 shows the grey scale images of the six methods. A bright pixel means that pixel i has higher correlation $\gamma(i)$, which helps to distinguish the reliable pixels of its corresponding pixels in the visualization. The resulting grey scale efficiency images show that by *FSPE* being brighter than those by other methods. This proves the reliability of the colours in the resulting visualization for Region_0 of remote sensing imagery by our proposed method.

Detection the visualization errors through a rigid body transformation:

To complete the comparison, we need to know the amount of similarities and differences between the aforementioned methods. Rigid body transformation [Golub and Loan, 1989] takes into consideration any differences between two visualizations. We assumed the visualization by *FSPE* is a reference image (remaining stationary), whereas visualizations by PCA, SPE, Isomap, CDA and LLE are transformed to match the stationary image. The displacements of each point in the transformed visualization is relative to its corresponding points in visualization by *FSPE*. In this case, the transformed visualizations which have the best possible colour values fit to visualization by *FSPE*.

Figure 3.7 shows the rigid body transformation from the SPE's visualization to the *FSPE*'s visualization for Region_0. The visual comparison between them showed that there are several pixels in SPE's visualization which did not match their corresponding in visualization of *FSPE*.

For example, p_1 and p_2 have different colours in SPE's visualization, although their corresponding pixels in *FSPE*'s visualization have approximately the same colours. In addition, p_2 and p_3 have approximately the same colours in SPE's visualization, but their corresponding pixels in *FSPE*'s visualization have different colours. The reasons for that can be analysed by computing and comparing among their distances in the original and projected spaces.

Table 3.6 shows that p_1 and p_2 are neighbours in original space and they are far

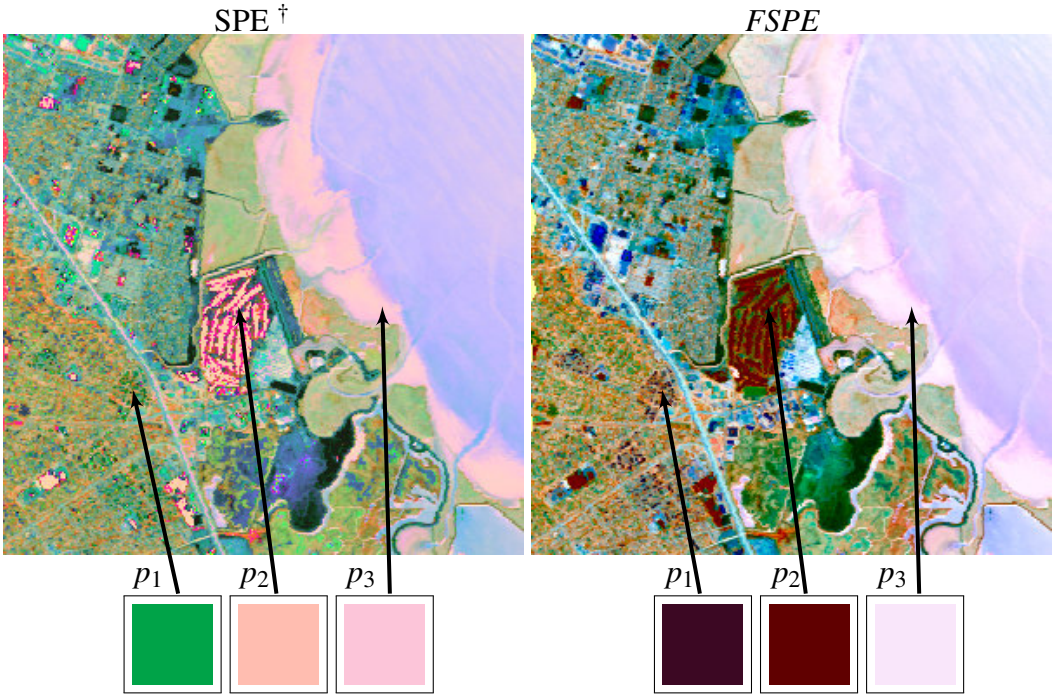


Figure 3.7: *p_1 and p_2 are neighbours in the original space and they are far from p_3 (see Table 3.6. SPE cannot preserve the distance between p_1 and p_2 in the visualization, where p_2 has false colours by getting the same colour of p_3, and their colours are different than those in p_1, as in the second column in Table 3.6. FSPE preserves the distances between points by preserving the neighbourhood relation between p_1 and p_2, where they obtained approximately the same colours. p_3 obtained different colours because they are projected from far away from them.*

†Result of rigid body transformation from the SPE's visualization to the *FSPE*'s visualization

Table 3.6: *The distances among the three points p_1, p_2 and p_3 in the original space and their corresponding distance in the projected spaces of SPE and FSPE.*

	Original Space (r_{ij})	SPE's Projected Space (d_{ij})	*FSPE*'s Projected Space (d_{ij})
distance_12	0.7887	1.4278	0.3701
distance_23	1.7565	0.3133	0.5381
distance_13	1.0316	1.1276	0.8605

from p_3, where $(r_{12} < r_{23})$ and $(r_{12} < r_{13})$. These relationships are lost in the projected space of SPE, which projected p_2 and p_3 to be neighbours, and p_1 was far from them. In other words, SPE satisfied the $(d_{12} > d_{23})$ and $(d_{12} > d_{13})$, which means p_2 is a false neighbour to p_3, and continuity error causes p_1 and p_2 were far away in projected

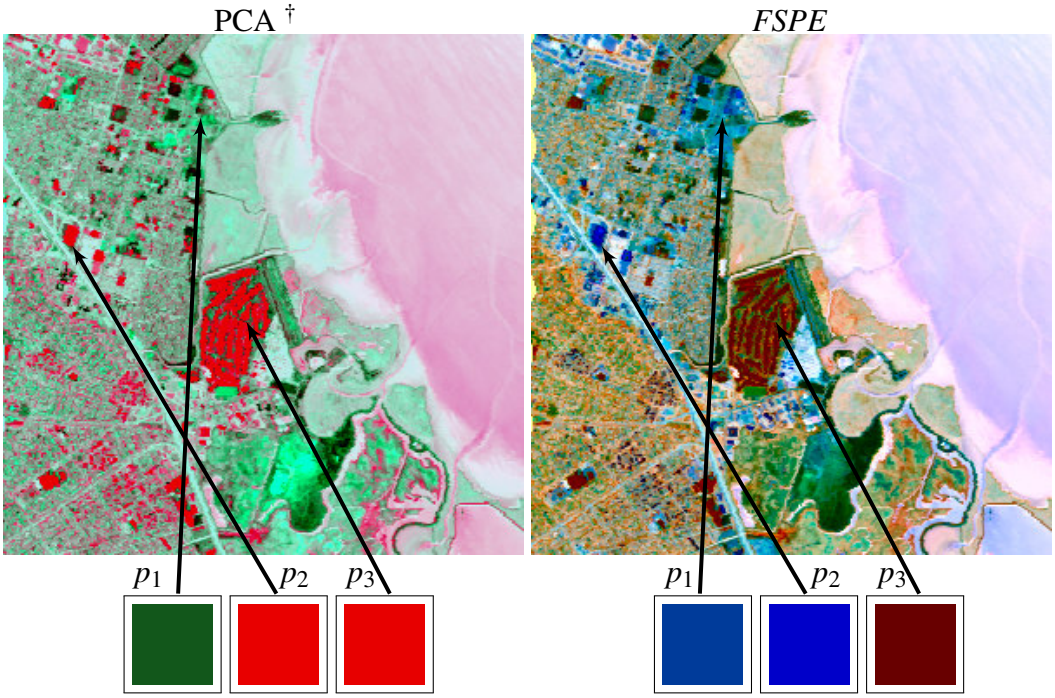


Figure 3.8: *PCA versus FSPE in visualizing the Region_0. The distance between p_1 and p_2 was not preserved in the PCA's visualization, (see Table 3.7). p_2 has false colour by being neighbour to p_3, although they are far away in the original space. p_3 is projected far away of p_1 and p_2 in the FSPE's visualization, and the projected space's distance between p_1 and p_2 is preserved with its corresponding distance in the original space.*

† Result of rigid body transformation from the PCA's visualization to the *FSPE*'s visualization

Table 3.7: *The distances among the three points p_1, p_2 and p_3 in the original space and their corresponding distance in the projected spaces of PCA and FSPE.*

	Original Space (r_{ij})	PCA's Projected Space (d_{ij})	*FSPE*'s Projected Space (d_{ij})
distance_12	0.4242	1.5473	0.6165
distance_23	1.5087	0.0066	1.2677
distance_13	1.3727	1.5407	0.8049

space.

On the other hand, *FSPE* is able to overcome the false neighbourhood and continuity errors. The colours in *FSPE*'s visualization reflect the correct relationships between the points. For example, p_1 and p_2 have approximately the same colours to indicate that they are neighbours in original space and projected space. In addi-

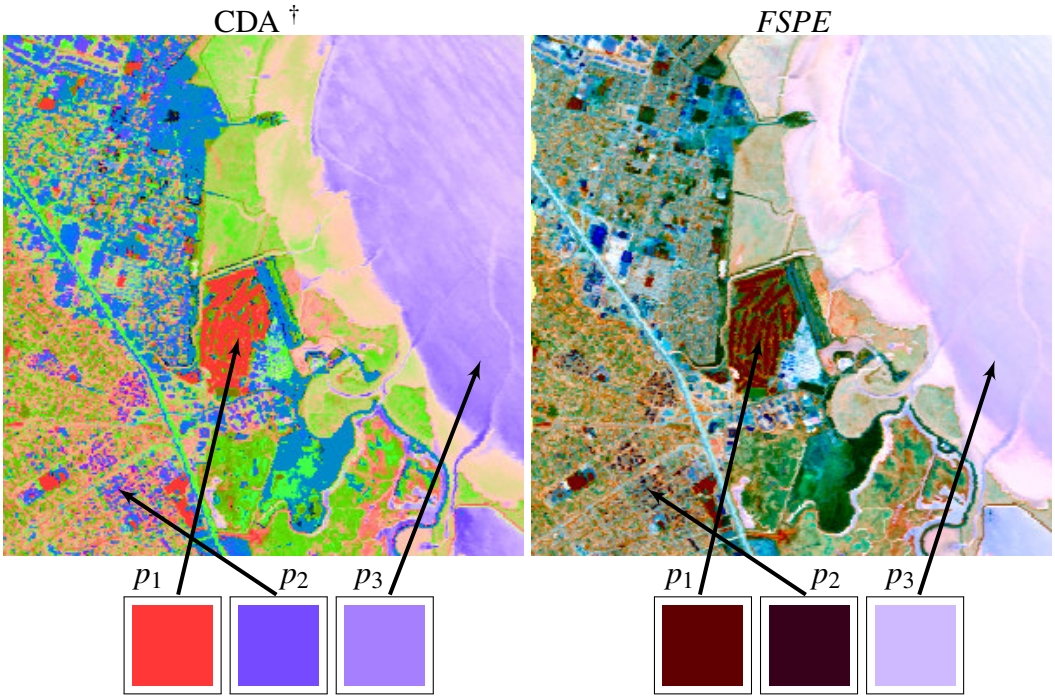


Figure 3.9: *Colour distance between p_1 and p_2 is not preserved in the CDA's visualization, although they are neighbours in the original space (see Table 3.8). Moreover, p_2 and p_3 have the convergent colours, although they are far away in the original space, therefore, p_2 has false colour. FSPE succeeded in preserving the neighbour relation between p_1 and p_2 with their original space distance, and p_3 are projected far away of them as in the original space.*

† Result of rigid body transformation from the CDA's visualization to the *FSPE*'s visualization

Table 3.8: *The distances among the three points p_1, p_2 and p_3 in the original space and their corresponding distance in the projected spaces of CDA and FSPE.*

	Original Space (r_{ij})	CDA's Projected Space (d_{ij})	*FSPE*'s Projected Space (d_{ij})
distance_12	0.3757	1.2342	0.2834
distance_23	1.7420	0.8527	0.8630
distance_13	1.9742	1.4842	0.6834

tion, their colours are completely different than this in p_3, which satisfies the relation $(d_{12} < d_{23})$ and $(d_{12} < d_{13})$.

The visual comparison between PCA's visualization and *FSPE*'s visualization of Region_0, in Figure 3.8, illustrate that many pixels in PCA's visualization did not match the corresponding one in the visualization of *FSPE*. For example, p_2 and p_3 have the

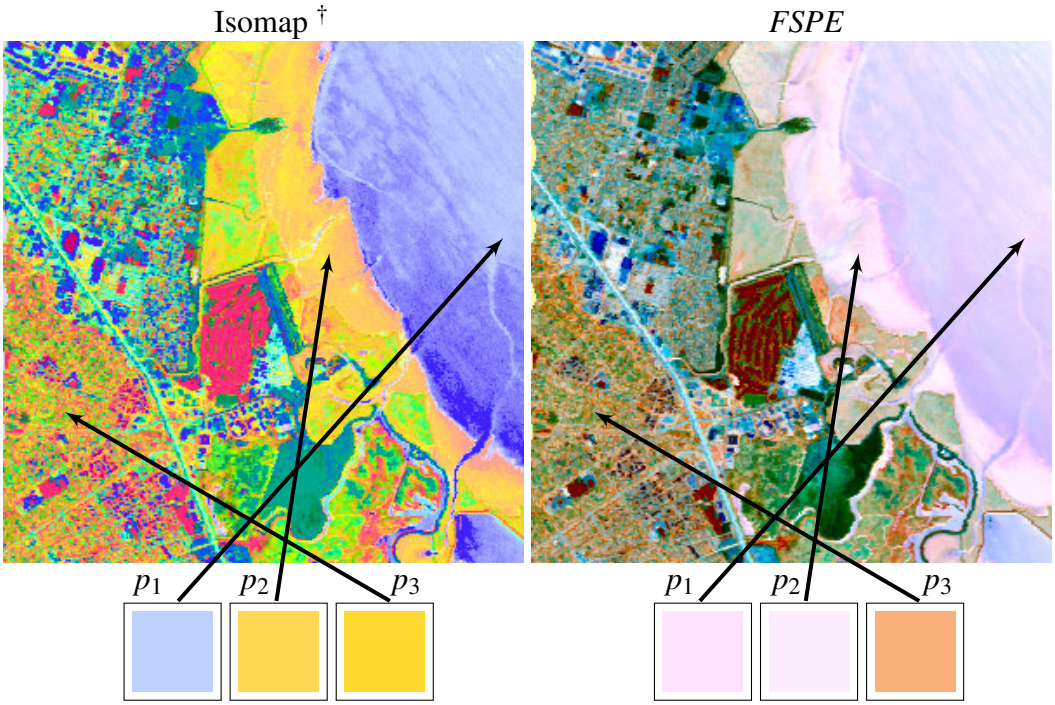


Figure 3.10: *Colour distance between p_1 and p_2 is not preserved in the Isomap's visualization, although they are neighbours in the original space (see Table 3.9). Moreover, p_2 and p_3 have the same colours, although they are far away in the original space. Thus, p_2 has false colour. FSPE succeeded in preserving the neighbour relation between p_1 and p_2 with their original space distance, and p_3 are projected far away of them as in the original space.*

† Result of rigid body transformation from the Isomap's visualization to the FSPE's visualization

Table 3.9: *The distances among the three points p_1, p_2 and p_3 in the original space and their corresponding distance in the projected spaces of Isomap and FSPE.*

	Original Space (r_{ij})	Isomap's Projected Space (d_{ij})	*FSPE*'s Projected Space (d_{ij})
distance_12	0.1157	1.8365	0.0315
distance_23	1.4496	0.3601	1.7107
distance_13	1.4346	2.1777	1.7059

same colours in PCA's visualization, which are different to those of p_3. Although the colours of p_1 and p_3 are true, p_2 obtained false colour. Table 3.7 shows the distances among points p_1, p_2 and p_3 in the original space were not preserved as well as in the projected space of PCA. The distance between p_2 and p_3 in the projected space (d_{23}=0.0066) is less than their distance in the original space (r_{23}=1.5087), which leads

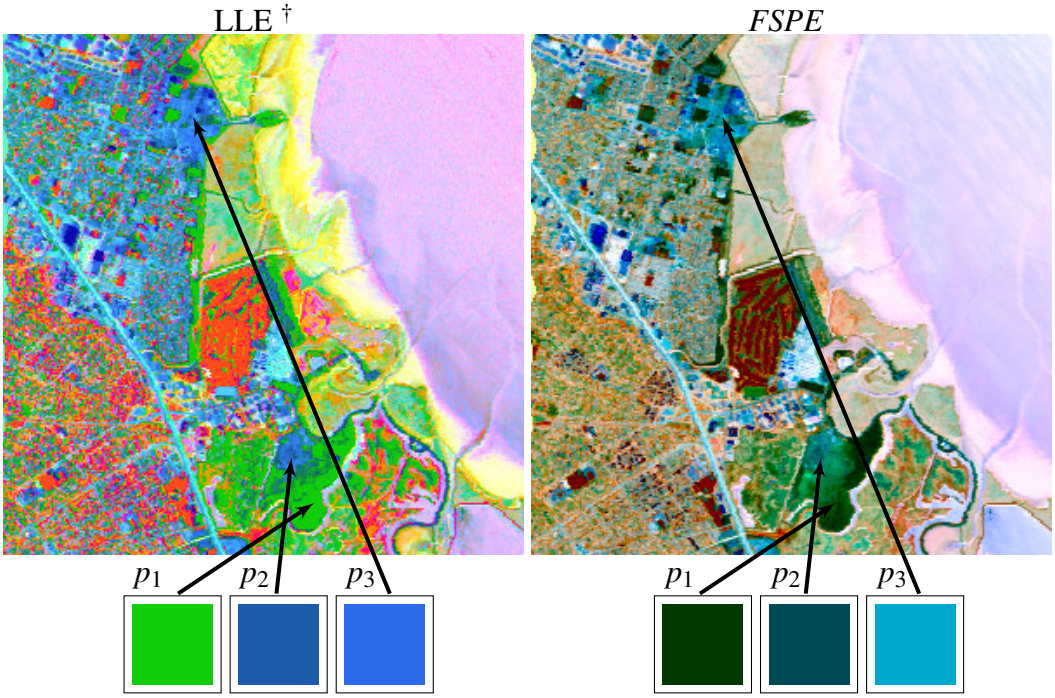


Figure 3.11: *LLE lets non-neighbour regions (as in p_2 and p_3) in the original space to be neighbours in the visualization, and neighbour regions (as in p_1 and p_2) in the original space to be non-neighbour in the visualization (see Table 3.10). FSPE preserves the neighbourhood relation between p_1 and p_2 in its visualization, where they have the same colours. The non-neighbourhood relation between them and p_3 is also preserved in the visualization by giving p_3 different colour.*

†Result of rigid body transformation from the LLE's visualization to the *FSPE*'s visualization

Table 3.10: *The distances among the three points p_1, p_2 and p_3 in the original space and their corresponding distance in the projected spaces of LLE and FSPE.*

	Original Space (r_{ij})	LLE's Projected Space (d_{ij})	*FSPE*'s Projected Space (d_{ij})
distance_12	0.8688	1.9729	0.0518
distance_23	1.0139	0.6765	0.8539
distance_13	1.8755	2.4668	0.8846

p_2 to be a false neighbour to p_3.

In addition, the projected space of PCA was affected by a continuity error, which caused the neighbourhood points p_1 and p_2 to be projected far away ($r_{12} < d_{12}$). On the other hand, the colours in *FSPE*'s visualization reflect the correct relationships

among the points. The converged colours of p_1 and p_2 are completely different to this in p_3, which proves that the preserving neighbourhood distances are done as well by *FSPE*.

Using CDA causes the projection of the neighbourhood points in original space as non-neighbourhood points in projected space, and vice versa. Table 3.8 shows p_1 and p_2 are neighbours in the original space and they are projected by CDA far away in the projected space. *FSPE* is better in comparison with CDA because it is able to preserve the distance between points in the projected space. Figure 3.9 shows that neighbourhood points p_1 and p_2 have approximately the same colours in *FSPE*'s visualization, and their colours are different to p_3's colour. Thus, we can say the colours in *FSPE* represent the correct relationships between points. In the visualization of CDA, p_2 acquired false colour, which is similar to p_3, and the colour differences between p_1 and p_2 indicate the loss of the original neighbourhood relation between them.

Figure 3.10 showed p_2 has colour very close to p_3, in the Isomap's visualization, although they are not neighbours in the original space. The difference in their distances in the original space and the projected space, as in Table 3.9, indicates p_2 was projected to be a false neighbour to p_3. In addition, the continuity error causes the neighbourhood points p_1 and p_2 in the original space to be far away in the projected space, and they obtained different colours in the visualization according to that. The DR problems are overcome in *FSPE*, as in the third column in Table 3.9. In the *FSPE*'s visualization, the colours of p_1, p_2 and p_3 indicate p_1 and p_2 are neighbours in the original space and the projected space, and p_3 is far from them in both spaces.

LLE has false colours in some regions, for example, in Figure 3.11 where the colour of p_2 in the visualization is very close to the colour value of p_3, although their distance in the original space indicated that they did not have this type of relationship. p_1 and p_2 should be neighbours in the projected space, but LLE did not preserve that. Therefore, different colours in the visualization were obtained. Table 3.10 showed that the LLE do not preserve the original distances in its projected space, whilst *FSPE* achieved that by preserving the original distance r_{12}, r_{23} and r_{13} as well as with their corresponding distance d_{12}, d_{23} and d_{13} in the projected space. The colours of p_1, p_2, and p_3 also confirm the ability of *FSPE* is preserving the original information.

The visual comparison demonstrates that although the aforementioned methods might generate satisfactory colour display for remote sensing imagery, *FSPE* visual-

ization is preferable for revealing that colour display overcomes the false neighbourhood errors. In Chapter 6, we will apply *FSPE* on the GPU in order to speed up the projection process.

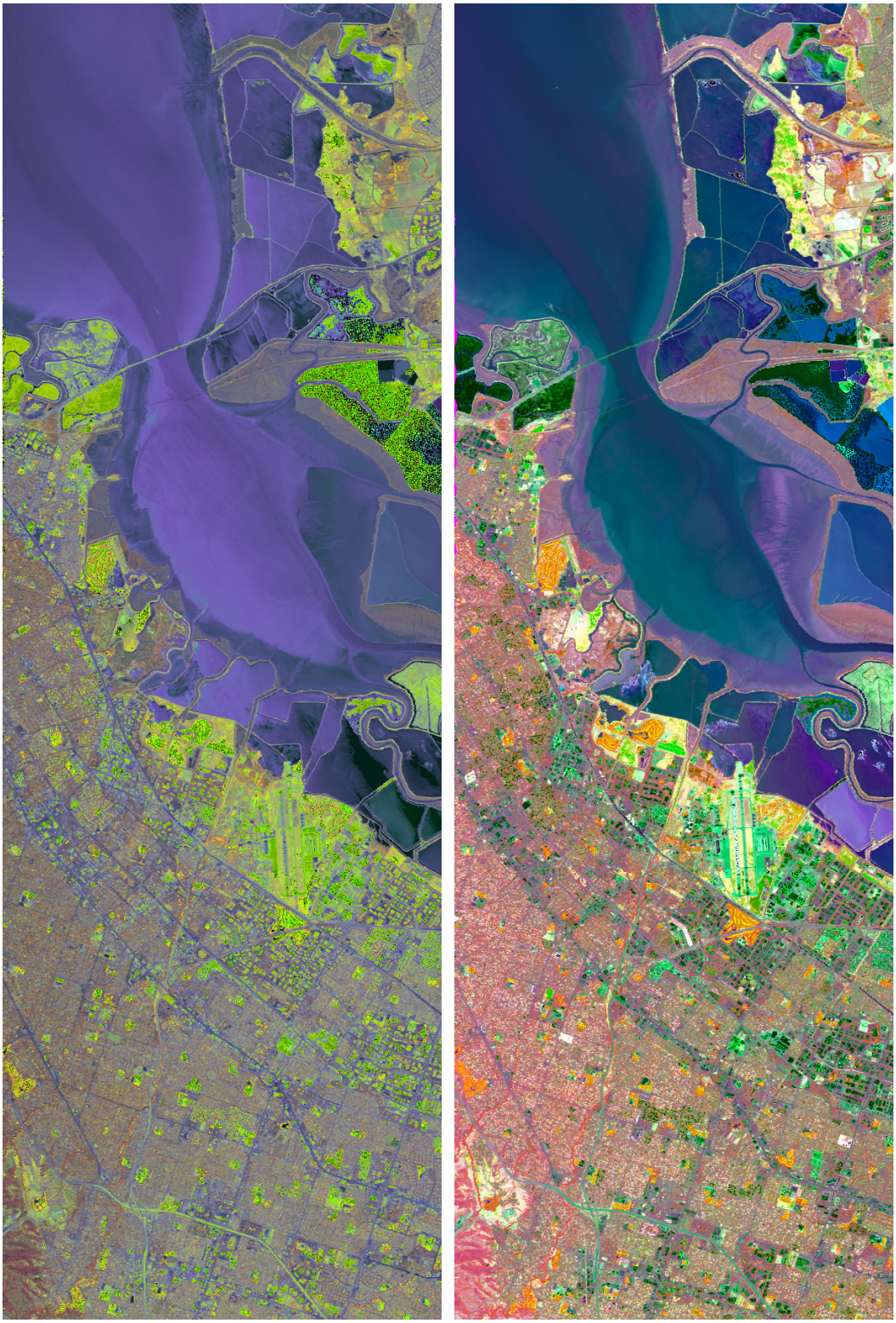
Visualization of Large data sets

Figures 3.12, 3.13 and 3.14 show the comparison between the qualitative *CIE Lab* visualizations of large data sets, which has dimension 1800x600x224, by PCA, CCA, SPE and *FSPE*. Our method is the better among them where it is able to reveal faithful colour variations.

3.6 Discussion

In this chapter, we showed that *FSPE* has the ability to visualize difficult data sets. Therefore, in terms of visualization, our proposed method gives satisfactory results. This ability comes from dealing adequately with the projected space. In general, a projected space is improved through projection process, consequently, we decided to focus on this state by using decreasing neighbourhood size in order to continue this improvement. In each step of the projection process, the neighbourhood size is reduced in order to keep pace with improvements in the projected space. In addition, the space optimization is reduced gradually, with *FSPE* focusing, to begin with, on maintaining a distant relationship between the points and then maintaining the nearby relationships.

Although the results in this chapter are convincing, our method cannot fully overcome the presence of false neighbourhoods. The source of this limitation is the big difference between the topological structures of high-dimensional data sets and of low-dimensional space. The direct projection between them makes it impossible to preserve all the distances between the points between two spaces. Thus, in order to overcome this limitation, we will re-define the way of applying the DR in Chapter 4. In Chapter 5, we will show another limitation of the *FSPE* when using it in the classification task, in that its results are not the best when compared with other DR methods. Thus, we can say the best application with regard to *FSPE* is related to visualization.

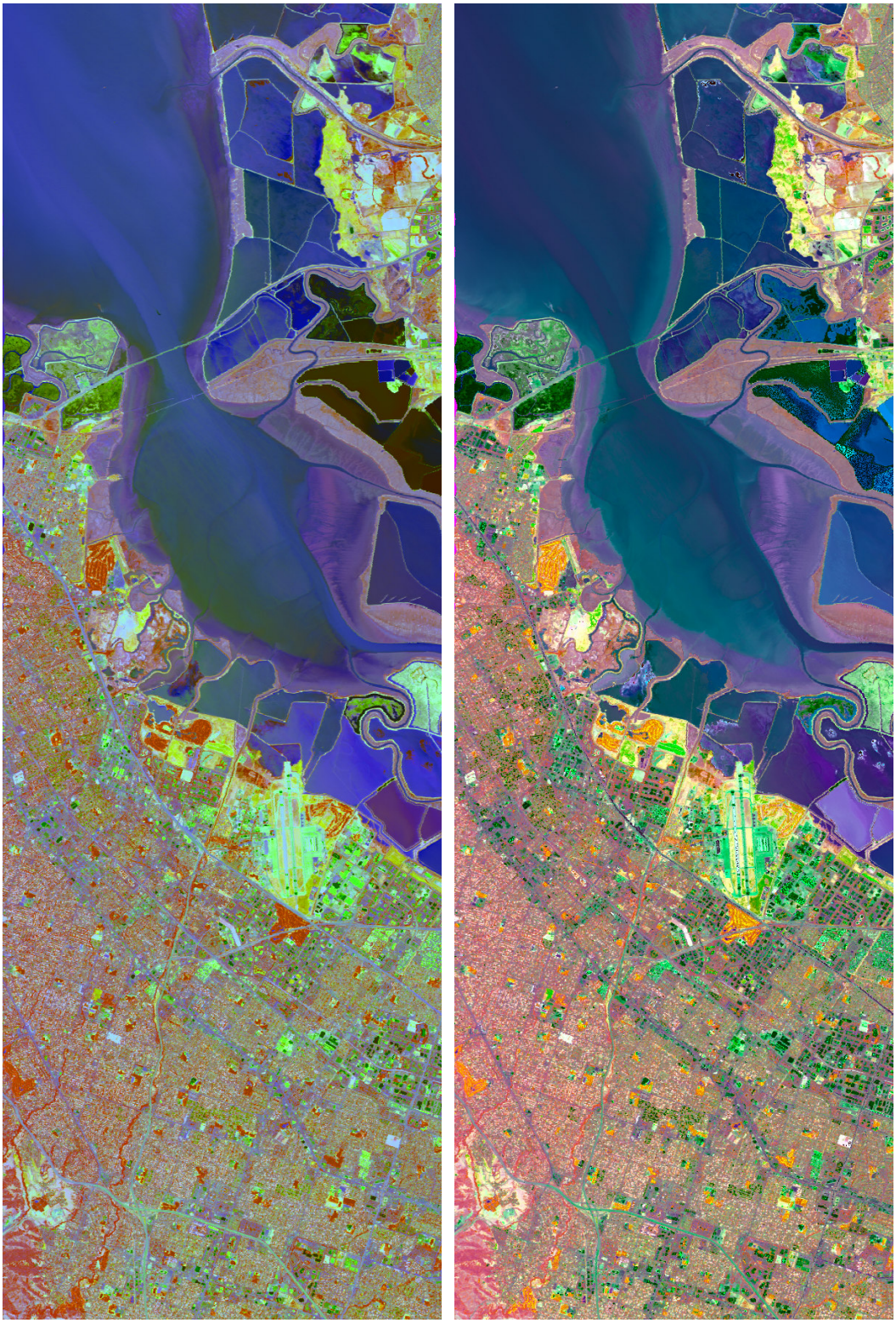


(a) SPE, $\gamma = 0.801$ (b) *FSPE*, $\gamma = 0.923$

Figure 3.12: *Qualitative CIE Lab visualization of data sets with 1800x600x224 dimension. FSPE's visualization reveals different colour variations, and its correlation value is higher than SPE's visualization.*

(a) CCA, $\gamma = 0.823$ (b) *FSPE*, $\gamma = 0.923$

Figure 3.13: *Visualizing the large data sets (1800x600x224) by FSPE is better when compared with CCA, where the correction measurement of FSPE's visualization is higher than CCA's visualization.*



(a) PCA, $\gamma = 0.795$ (b) *FSPE*, $\gamma = 0.923$

Figure 3.14: *Linear projection for large data sets (1800x600x224) by PCA caused to lose more information, and nonlinear projection by FSPE preserved most of the remote sensing imagery data sets information.*

3.7 Conclusion

A new faithful dimension reduction method, called *FSPE*, was proposed in this chapter to visualize a remote sensing imagery data sets. The benefit of visualizing by *FSPE* is that it is able to recognize the features by preserving their colour distances between visualization and the bands of the original data sets. The results showed *FSPE* can overcome many of the problems introduced by false colours by deriving higher quality colours in its visualization. Experiments indicated that the proposed method is better than continuity methods such as PCA, SPE, and Isomap. In addition, the experiments demonstrated that the proposed trustworthy method is better than traditional trustworthy methods such as CDA, because the *FSPE* prevents false neighbourhood errors to occur in the results. Moreover, the results proved the *FSPE* is better than SPE, where the error projection is minimized with focusing on the low-dimensional space rather than the original space. Although, *FSPE* succeeds in overcoming the false neighbourhood in many places in its visualization, it might suffer from this problem. In the next chapter, we will redefine the DR in order to increase the efficiency of the visualization.

The main ideas presented in this chapter is accepted for publication in: Najim S., Lim I, Wittek P. and Jones M. FSPE: Visualisation of Hyperspectral Imagery Using Faithful Stochastic Proximity Embedding. IEEE Geoscience and Remote Science Letters. 25/3/2014.

Chapter 4

Visualization of High-dimensional Data Sets by Sequential Dimensionality Reduction

In this chapter we will address the following topics:

1. Problems of the dimensionality reduction.
2. Methodology of Sequential Dimensionality Reduction (*SDR*),
3. Experimental results of *SDR* application on hyperspectral imagery.

This chapter introduces a new technique called Sequential Dimensionality Reduction (*SDR*), to visualize high-dimensional data sets, as remote sensing imagery. The *SDR* is introduced to directly project the high-dimensional data sets into a low-dimensional space. Although this idea works very well when the dimensionality of the original data sets is small, its visualization is not efficient enough with large input dimensions. Unlike DR, *SDR* redefines the problem of DR as a sequence of multiple DR problems, each of which reduces the dimensionality by a small amount. The *SDR* can be considered as a generalized idea that can be applied to any method. The SPE, CCA and FSPE methods are chosen in this chapter because of their speed and efficiency compared to other methods. The superiority of *SDR* over DR is demonstrated experimentally by using correlation, LC and stress metrics. Moreover, as most DR methods also employ DR ideas in their projection, the performance of *SDR* and 20 DR

methods are compared. The GPU is the best way to speed up the *SDR* method, where the speed of execution has been increased by 74 times in comparison to when it was run on the CPU.

4.1 Introduction

Due to the difference between the topological structures of data points in the high-dimensional data sets with the topological structure of the projected space, it is difficult to preserve the neighbourhood relations between data points. Loss of any neighbour point could lead to an increase in the amount of error because the missing neighbour point will occupy a place of another point in the lower space. Therefore, the amount of error could be growing explosively when increasing the false neighbourhood. Overlap between points is the dominate thing when direct projection from high-dimensional data sets to low-dimensional space is still used.

This work proposes *SDR* to redefine the DR by a sequence of multiple DR problems, each of which reduces the dimensionality by a small amount. In contrast to the DR method, *SDR* preserves the neighbourhood relations between data points of multiple reduced consecutive spaces. Thus, false neighbourhoods are reduced as much as possible and the quality of the visualization is optimized.

4.2 Methodology: Sequential Dimensionality Reduction

The goal of the proposed method is to solve the DR by finding a representation of N points in d space, where its neighbourhood relationships are preserved with their corresponding relations in original space. To do that, we redefine the problem of DR as a sequence of multiple DR problems, each of which reduces the dimensionality by a small amount. We call this *SDR*.

More specifically, given $X = x_1, x_2, ..., x_n$ be a data with instances $X \subset \mathcal{R}^D$. *SDR* attempts to do the transformation in the following equation :

$$G : \mathcal{R}^D \xrightarrow{G_1} \mathcal{R}^{D-S} \xrightarrow{G_2} \mathcal{R}^{D-2S} \xrightarrow{G_3} \mathcal{R}^{D-3S} \ ... \xrightarrow{G_k} \mathcal{R}^d \tag{4.1}$$

where $d \ll D$, and S is the step of dimensionality reduction can be in the range $[1, D-1]$. The transformation (G_i) attempts to project N points of $(D-(i-1)S)$ space

into $(D-i\ S)$ space. The transformation between two spaces is reasonably as close as possible because of the similarity of dimensionality between two spaces. Thus, the problem for $(D-i\ S)$ space is solved. Efficiency of transformation permits to recursively apply it until the target dimension is obtained. Therefore, the neighbourhood relations between original points are kept carefully through a sequence of transformations until it obtains d space.

To explain the general idea of the *SDR* when $S=1$, let's suppose the point p_j in the v space. It is able to preserve its neighbourhood relationships when it is projected to the $v-1$ space. Thus, the amount of false neighbourhood when $G_v : \mathcal{R}^v \rightarrow \mathcal{R}^{v-1}$ is very small because the difference between the two spaces is just one dimension. A complete transformation from D space to d spaces is obtained by applying Equation 4.1.

The point that should be discussed is how to choose S. As we defined before, S can be in the range $[1,\ D-1]$. For higher efficiency $S=1$, where the intermediate transformation problem will be between very close topology spaces. The error is minimized, and the problem is defined as a $D-1$ DR problem. The efficiency is reduced when $S>1$, and the worst case when $S=D-1$, which is defined as one DR problem.

SDR accepts as input set of N points in D space, a transformation function G, and the amount of dimensionality reduction S. d is the dimension of lower space which should be given. *SDR* algorithm recursively reduces the dimensionality of a space until obtaining the target space, and the following steps describe that:

1. Let $v\ =\ D$.
2. If $v\ \leq\ d$
   {Stop}
3. If $(v-S)\ <\ d$ then $S=v-d$
4. N points of v space is projected in next lower space ($v-S$) space by applying $G:\mathcal{R}^v \rightarrow \mathcal{R}^{v-S}$.
5. let $v\ =\ v-S$, and go to step 2.

Briefly, *SDR* put the projected points in the correct location in low-dimensional space, which leads to a reduction in the amount of error and an increase in the degree

of compliance with original space. The efficiency of *SDR* is clearer when the input dimension is very high and the target dimension is very low. The length of the execution time is one of the limitations that is encountered with the *SDR*; therefore, using the GPU is the most appropriate way to overcome this problem. While SPE, CCA and FSPE methods have more advantages than other methods, we chose them to be used with the *SDR* idea to generate sequential SPE (*SSPE*), sequential CCA (*SCCA*) and sequential FSPE (*SFSPE*), respectively.

Sequential Stochastic Proximity Embedding (*SSPE*)

SPE is a nonlinear method and proceeds by calculating Euclidean distance for global neighbourhood points. *SDR* with SPE generates a new method, which is *SSPE*, which uses the idea of *SDR* and SPE's cost function. While SPE's cost function is as a minimized function [Agrafiotis et al., 2010] when transforming D space to d space ($d \ll D$), *SSPE* will be minimizing for the same reasons because nothing is changed in relation between D and d spaces. Thus, through *SSPE*'s projection process, the points in the next space are updated according to Equation 4.2 when they are projected from v space:

$$y_j(v) \leftarrow y_j(v) + \lambda(t)\, S(r_{ij}(v))\, \frac{r_{ij}(v) - d_{ij}(v)}{d_{ij}(v) + \varepsilon}\, (y_j(v) - y_i(v)) \tag{4.2}$$

$$S(r_{ij}(v)) = \begin{cases} 1 & if\ (r_{ij}(v) \leq r_c) \vee ((r_{ij}(v) > r_c) \wedge (d_{ij}(v) < r_{ij}(v))) \\ 0 & Otherwise \end{cases} \tag{4.3}$$

where $\lambda(t)$ is learning rate at t time, and ε a tiny number used to avoid division by zero (we used $\varepsilon = 1\text{x}10^{-8}$). $r_{ij}(v) = ||x_i(v) - x_j(v)||$ and $d_{ij}(v) = ||y_i(v) - y_j(v)||$.

Sequential Curvilinear Component Analysis (*SCCA*)

CCA attempts to preserve the pairwise distances in low-dimensional space with their corresponding pairwise distance in original high-dimensional data sets. *SCCA* is generated when CCA is applied to the *SDR* idea. *SCCA* uses the same equation of CCA

when v space is projected to next space, as in equation below:

$$\phi(Y(v)) = \sum_{i<j} (r_{ij}(v) - d_{ij}(v))^2 \, F(d_{ij}(v), \lambda_t) \tag{4.4}$$

where $r_{ij}(v) = ||x_i(v) - x_j(v)||$ and $d_{ij}(v) = ||y_i(v) - y_j(v)||$ are the Euclidean distances between data points i and j in v and next spaces, respectively. F is a bounded decreasing function, and allows *SCCA* to preserve the distances on different scales depending on the time-dependent value of λ_t, which is started with a large value to cover all data points, and then gradually decreased throughout processing. It is defined as:

$$F(d_{ij}(v), \lambda_t) = \begin{cases} 1 & if \; d_{ij}(v) \leq \lambda_t \\ 0 & Otherwise \end{cases} \tag{4.5}$$

Sequential Faithful Stochastic Proximity Embedding (*SFSPE*)

The benefit of visualizing by FSPE, which is proposed in the previous chapter, is that it is able to recognize the features by preserving their colour distances between visualization and the original data sets. FSPE can overcome the false colours as much as possible to derive reliable colours in its visualization, as we conducted in the previous chapter. Although FSPE is a good method, applying it with *SDR* to generate *SFSPE* gives the method extra power to overcome the false neighbourhood errors. The *SFSPE* method uses the same function of FSPE, as in Equation 4.6, when projecting next space from v space.

$$y_j(v) \leftarrow y_j(v) + \lambda(t) \, T(d_{ij}(v)) \, \frac{r_{ij}(v) - d_{ij}(v)}{d_{ij}(v) + \varepsilon} \, (y_j(v) - y_i(v)) \tag{4.6}$$

$$T(d_{ij}(v)) = \begin{cases} 1 & if \; (d_{ij}(v) \leq d_c(t)) \vee ((d_{ij}(v) > d_c(t)) \wedge \\ & \quad (d_{ij}(v) < r_{ij}(v))) \\ 0 & Otherwise \end{cases} \tag{4.7}$$

where $d_c(t)$ is a neighbour radius that decreases over time. $r_{ij}(v) = ||x_i(v) - x_j(v)||$ and $d_{ij}(v) = ||y_i(v) - y_j(v)||$.

Table 4.1: *Methods employed in comparison. The first column contains the name of methods, second column shows the type of method (linear (L) or nonlinear (NL)), third column shows the source of reference and the symbol ✓ in the fourth column indicates the original codes have been used.*

Method	Type	Source	Originality
PCA	L	[Jolliffe, 2002]	✓
CCA	L	[Demartines and Hrault, 1997]	
CDA	NL	[Lee et al., 2004]	✓
Factor_analysis	NL	[Darlington, 1999]	
Fast_MVU	NL	[Weinberger and Saul, 2006]	
Hessian_LLE	NL	[Donoho and Grimes, 2005]	
Isomap	NL	[Tenenbaum et al., 2000]	✓
Kernel_PCA	NL	[Taylor and Christianini, 2004]	
Laplacian	NL	[Belkin and Niyogi, 2003]	
LLC	NL	[Shi and Malik, 2000]	
LLE	NL	[Roweis and Saul, 2000]	✓
LLTSA	NL	[Zhang et al., 2007]	
LPP	L	[Zhi and Ruan, 2008]	
LTSA	NL	[Zhang and Zha, 2004]	
NPE	NL	[He et al., 2005]	✓
Prob_PCA	NL	[Tipping and Bishop, 1999]	
SPE	NL	[Rassokhin and Agrafiotis, 2003]	✓
SNE	NL	[Bunte et al., 2012]	✓
tSNE	NL	[Maaten and Hinton, 2008]	✓
FSPE	NL	In the previous chapter	✓

4.3 Experimental Results

In this section, *SDR* and DR are evaluated in quantitative and qualitative manners. The performance of the proposed method is analysed using the remote sensing imagery data sets. The *SDR* idea is implemented on SPE, CCA and FSPE, which are called *SSPE*, *SCCA* and *SFSPE*, respectively, and compared with their original DR version. The comparison *SDR* with 20 DR methods is also included, and Table 4.1 shows information about those methods.

We implemented our method in Microsoft Visual Studio C++ 2008 with CUDA 4.2 in Windows 7. The hardware included Intel(R) i7-930 CPU clocked at 2.80 GHz, with 12 GBytes of main memory. The graphics processor was an NVIDIA GeForce GTX 480 with a buffer size of 1 GByte. We used the AVIRIS Moffet Field data sets from

the southern end of San Francisco Bay, California, done in 1997 [AVIRIS, 2013]. We divided this data sets into small regions, each one with 300x300x224 pixels.

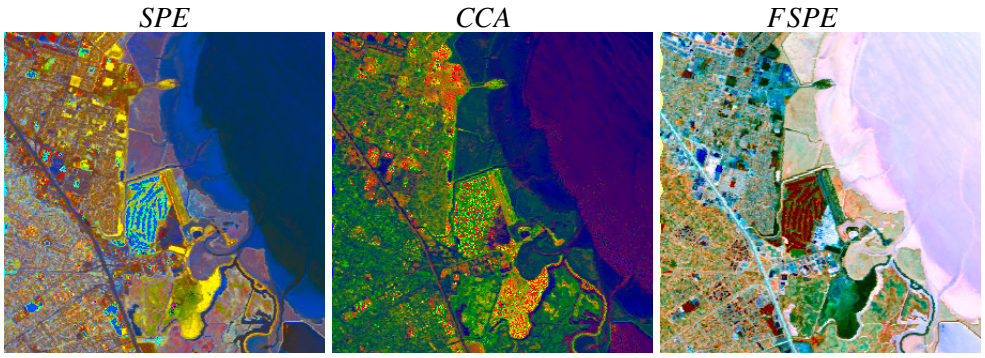
Tables 4.2, 4.3 and 4.4 shows that *SSPE*, *SCCA* and *SFSPE*, respectively, achieve good values with correlation, LC and stress metrics, where, for all regions, the correlation and LC values by our proposed method are highest, and the same is also true for stress values in which it has given lesser values.

Table 4.2: *Results of comparison the SDR method (represented by SSPE), when $S = 1$, with the DR method (represented by SPE). The proposed method got the highest correlation and LC, and less stress values in all regions. (Correlation and LC: the highest is a bester, Stress: the lowest is a better).*

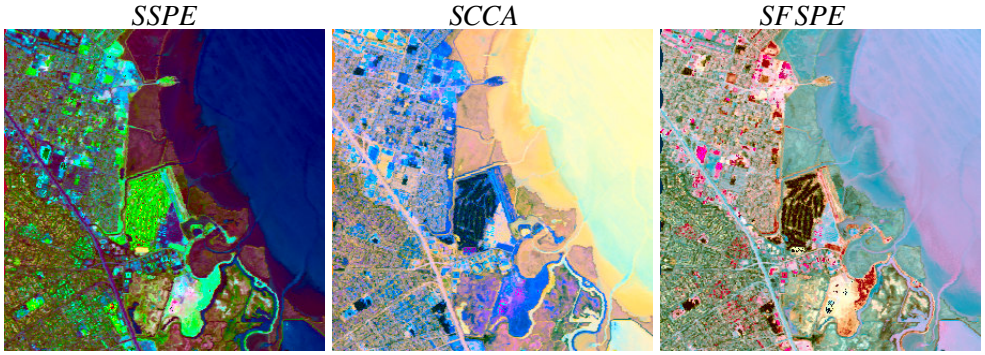
	Correlation		LC		Stress	
	SPE	*SSPE*	SPE	*SSPE*	SPE	*SSPE*
Mean	0.641	0.948	0.293	0.440	0.099	0.018
Region_0	0.696	0.998	0.288	0.421	0.158	0.002
Region_1	0.809	0.837	0.214	0.238	0.254	0.035
Region_2	0.994	0.936	0.580	0.613	0.011	0.007
Region_3	0.620	0.788	0.115	0.312	0.060	0.006
Region_4	0.641	0.966	0.445	0.485	0.150	0.066
Region_5	0.813	0.989	0.140	0.277	0.110	0.016
Region_6	0.504	0.998	0.196	0.450	0.001	0.001
Region_7	0.662	0.984	0.193	0.382	0.080	0.006
Region_8	0.671	0.994	0.082	0.408	0.006	0.004
Region_9	0.141	0.991	0.191	0.544	0.289	0.045
Region_10	0.684	0.893	0.564	0.578	0.030	0.016
Region_11	0.462	0.996	0.504	0.576	0.190	0.014

To compare the *SDR* and DR methods in a qualitative manner, the visualizations of the DR methods are compared with their corresponding visualizations of the *SDR* methods. In this chapter, Region_0, Region_4 and Region_5 are chosen to show the comparison SPE with *SSPE*, CCA with *SCCA* and *FSPE* with *SFSPE*.. In Figure 4.1, we can see the performances of the proposed methods in previous tables for Region_0 are confirmed here. The *SSPE*, *SCCA* and *SFSPE* show more details than SPE, CCA and FSPE, where the *SDR* methods overcome the false colours as well as possible.

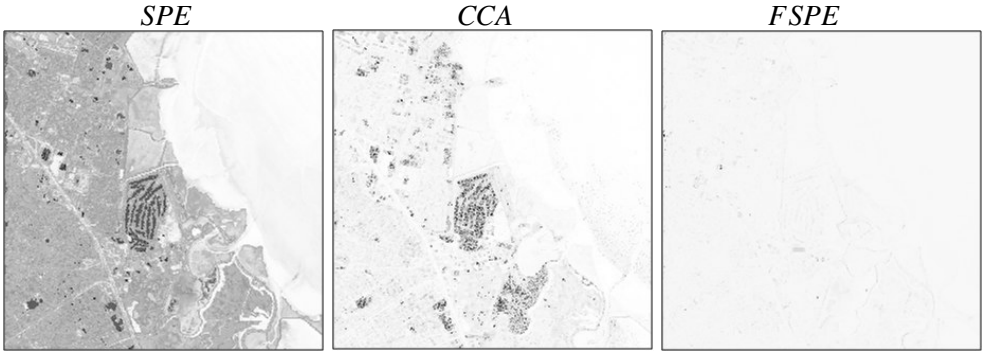
We can explain the quality of *SDR* by using point-wise correlation images, which was proposed in the previous chapter. For Region_0, in Figure 4.1, the correlation images of visualizations by SPE and CCA show they have many false colours which



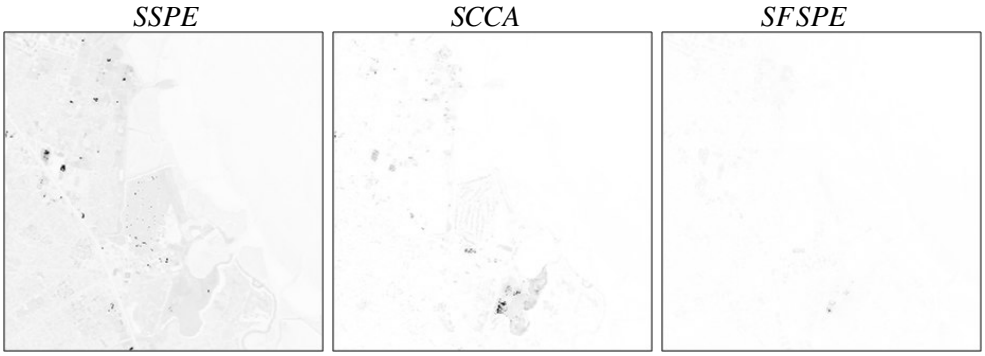
(a) Visualizations by Dimensionality Reduction Methods



(b) Visualizations by Sequential Dimensionality Reduction Methods



(c) Correlation of Dimensionality Reduction Methods' Visualization



(d) Correlation of Sequential Dimensionality Reduction Methods' Visualization

Figure 4.1: *a) Visualizations of regions_0 by using DR methods (SPE, CCA and FSPE); b) Visualizations by SDR methods (SSPE, SCCA and SFSPE); c) Correlation images of DR methods' visualizations; and d) Correlation images of SDR's methods. The SDR's visualizations show more details than the DR's visualizations and their correlation images show the quality of the points in the corresponding visualization colours.*

Table 4.3: *Results of comparison the SDR method (represented by SCCA), when $S = 1$, with the DR method (represented by CCA). The proposed method got the highest correlation and LC, and less stress values in all regions. (Correlation and LC: the highest is a bester, Stress: the lowest is a better).*

	Correlation		LC		Stress	
	CCA	*SCCA*	CCA	*SCCA*	CCA	*SCCA*
Mean	0.654	0.904	0.308	0.379	0.195	0.014
Region_0	0.873	0.896	0.308	0.419	0.011	0.002
Region_1	0.754	0.909	0.215	0.226	0.334	0.021
Region_2	0.501	0.894	0.083	0.253	0.869	0.012
Region_3	0.679	0.994	0.329	0.412	0.038	0.002
Region_4	0.551	0.865	0.384	0.406	0.012	0.002
Region_5	0.514	0.765	0.192	0.262	0.097	0.032
Region_6	0.545	0.897	0.349	0.460	0.002	0.002
Region_7	0.642	0.894	0.474	0.413	0.103	0.008
Region_8	0.699	0.898	0.311	0.414	0.057	0.018
Region_9	0.798	0.921	0.421	0.536	0.213	0.006
Region_10	0.598	0.989	0.306	0.376	0.284	0.019
Region_11	0.698	0.921	0.318	0.375	0.319	0.047

Table 4.4: *Results of comparison the SDR method (represented by SFSPE), when $S = 1$, with the DR method (represented by FSPE). The proposed method got the highest correlation and LC, and less stress values in all regions. (Correlation and LC: the highest is a bester, Stress: the lowest is a better).*

	Correlation		LC		Stress	
	FSPE	*SFSPE*	*FSPE*	*SFSPE*	*FSPE*	*SFSPE*
Mean	0.717	0.981	0.347	0.443	0.031	0.004
Region_0	0.863	0.998	0.319	0.433	0.001	0.001
Region_1	0.862	0.905	0.311	0.313	0.033	0.011
Region_2	0.904	0.999	0.295	0.356	0.028	0.017
Region_3	0.584	0.997	0.239	0.435	0.036	0.003
Region_4	0.794	0.977	0.440	0.437	0.012	0.004
Region_5	0.532	0.958	0.334	0.387	0.010	0.009
Region_6	0.620	0.993	0.438	0.512	0.052	0.001
Region_7	0.809	0.997	0.227	0.451	0.036	0.005
Region_8	0.619	0.995	0.370	0.468	0.036	0.000
Region_9	0.867	0.976	0.497	0.541	0.019	0.001
Region_10	0.360	0.984	0.342	0.437	0.047	0.001
Region_11	0.786	0.997	0.347	0.543	0.058	0.000

they are incorrectly visualized. The point-wise correlation image of visualizations by *SSPE* and *SCCA* show the false colours have been successfully overcome. Although FSPE's point-wise correlation image has higher quality, the *SFSPE*'s point-wise correlation images are better. The same scenarios are achieved for Region_4 and Region_5 in Figures 4.2 and 4.3, respectively, where *SSPE*, *SCCA* and *SFSPE* were proven to show the correct colours.

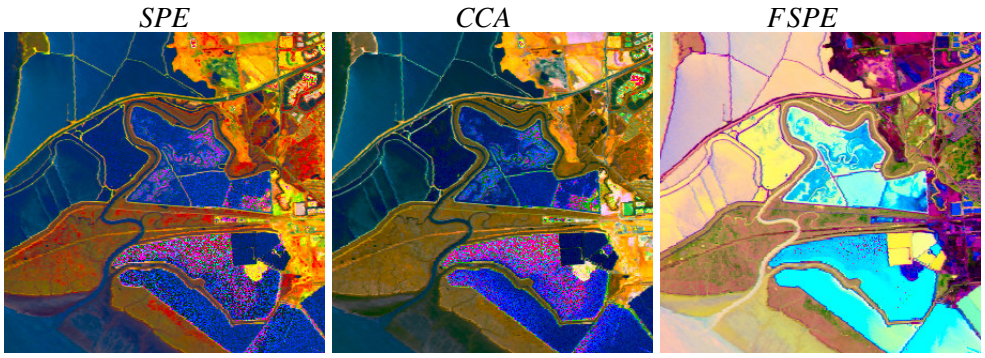
The results of comparisons between 20 DR methods and 3 SDR methods for Region_0, Region_4 and Region_5 as shown in Table 4.5, and Figure 4.4 show the visualizations of Region_0 by these methods. The results confirmed and supported our aforementioned discussion about the quality of the proposed method. Although SPE and CCA are not the best among the other methods, *SSPE* and *SCCA* are much better in correlation, LC and stress measurement values. Moreover, FSPE is one of the better among 20 DR methods in preserving neighbourhood relationships, but *SFSPE* is better than that. Figure 4.5 shows average of correction and stress metrics values, for all regions. The highest correlation and less stress values are achieved by *SDR* methods (*SSPE*, *SCCA* and *SFSPE*).

Speed is the next important item that should be addressed. The DR idea is much faster than our proposed method. Thus, the GPU is the best way to speed up the *SDR* method; that is, where the speed of execution has been increased by 74 times than when it ran on the CPU, as in Table 4.6 and Figure 4.6. Thus, the speed problem that occurs in *SDR* is solved. Figure 4.7 shows that the efficiency of the DR methods (represented by SPE, CCA and FSPE) are not affected by increasing their iteration number to be equal to the *SDR* methods (*SSPE*, *SCCA* and *SFSPE*). The *SDR* methods are decreasing through the projection process.

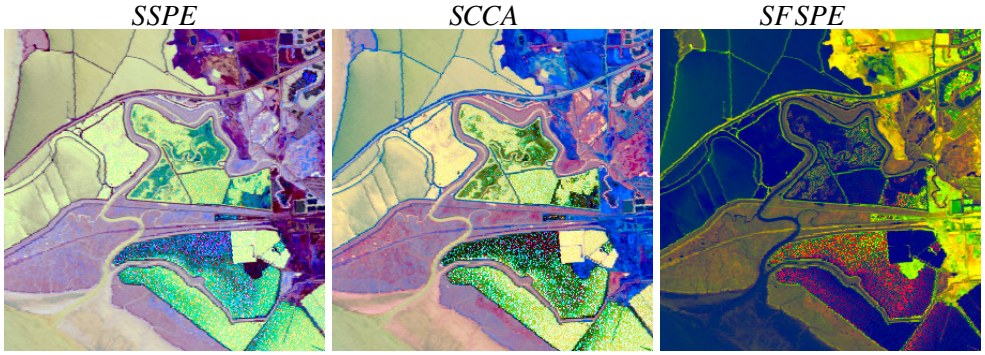
Even though our method gave good results when the sequences of multiple DR were reduced by amount ($S = 1$), this amount can be larger than one. However, the efficiency of *SDR* is reduced when S is increased, as in Figure 4.8.

4.4 Conclusions

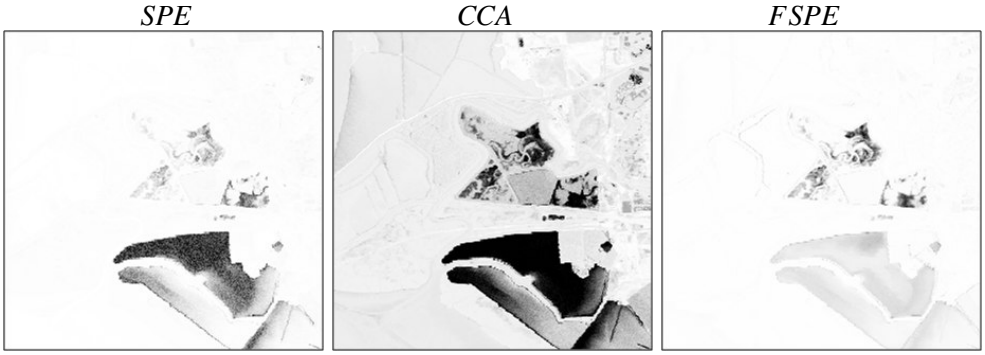
A new method called *SDR* has been proposed in this chapter to visualize remote sensing imagery. Theoretically, we illustrated that *SDR* maintains and preserves the relations among neighbour points in low-dimensional space. The results showed the accuracy of the proposed *SDR*, which leads to a better visualization with minimum



(a) Visualizations by Dimensionality Reduction Methods



(b) Visualizations by Sequential Dimensionality Reduction Methods

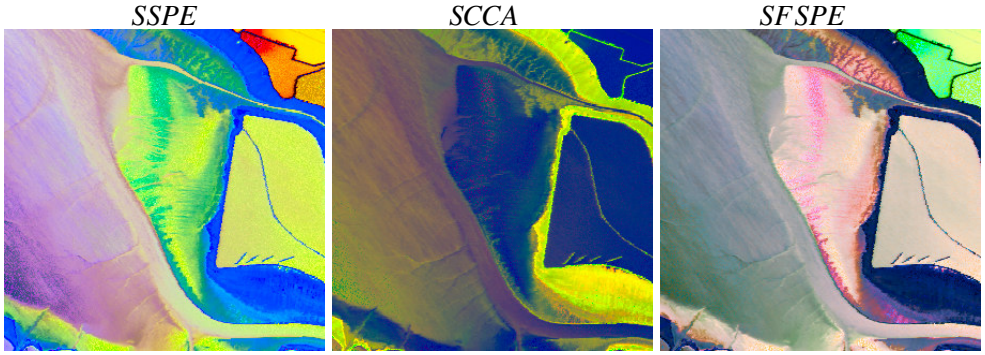


(c) Correlation of Dimensionality Reduction Methods' Visualization

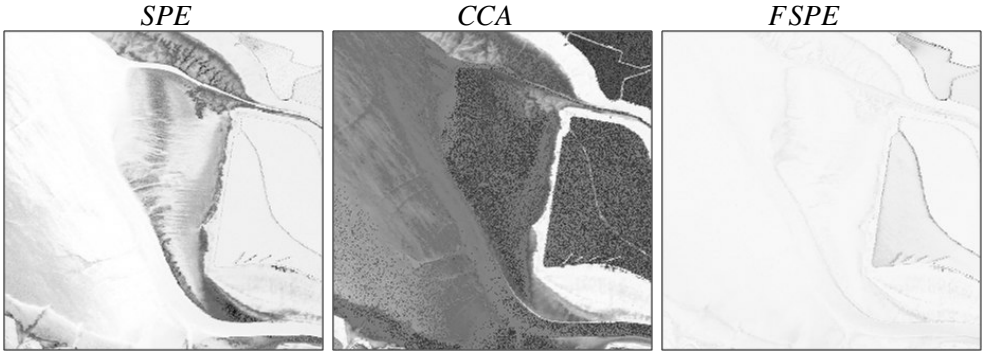
(d) Correlation of Sequential Dimensionality Reduction Methods' Visualization

Figure 4.2: *Visualization of Region_4 in remote sensing imagery data sets by SDR methods are better than their corresponding DR methods. Gray-scale efficiency image by point-wise correlation metric showed that the preserving original information is very high with SDR methods, especially with SFSPE method.*

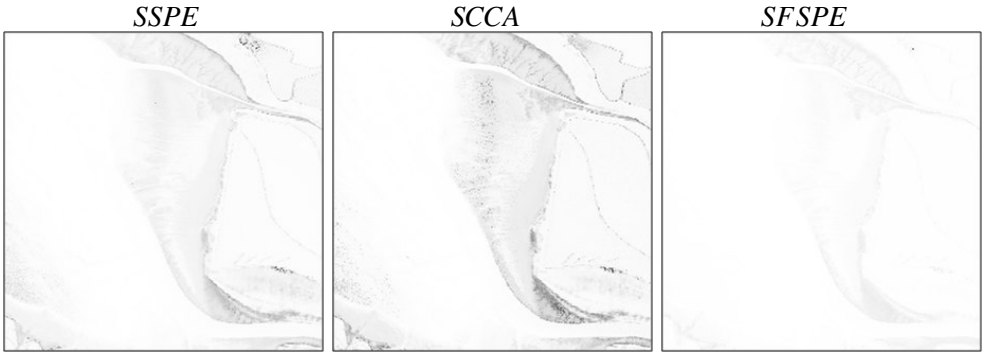
(a) Visualizations by Dimensionality Reduction Methods



(b) Visualizations by Sequential Dimensionality Reduction Methods



(c) Correlation of Dimensionality Reduction Methods' Visualization



(d) Correlation of Sequential Dimensionality Reduction Methods' Visualization

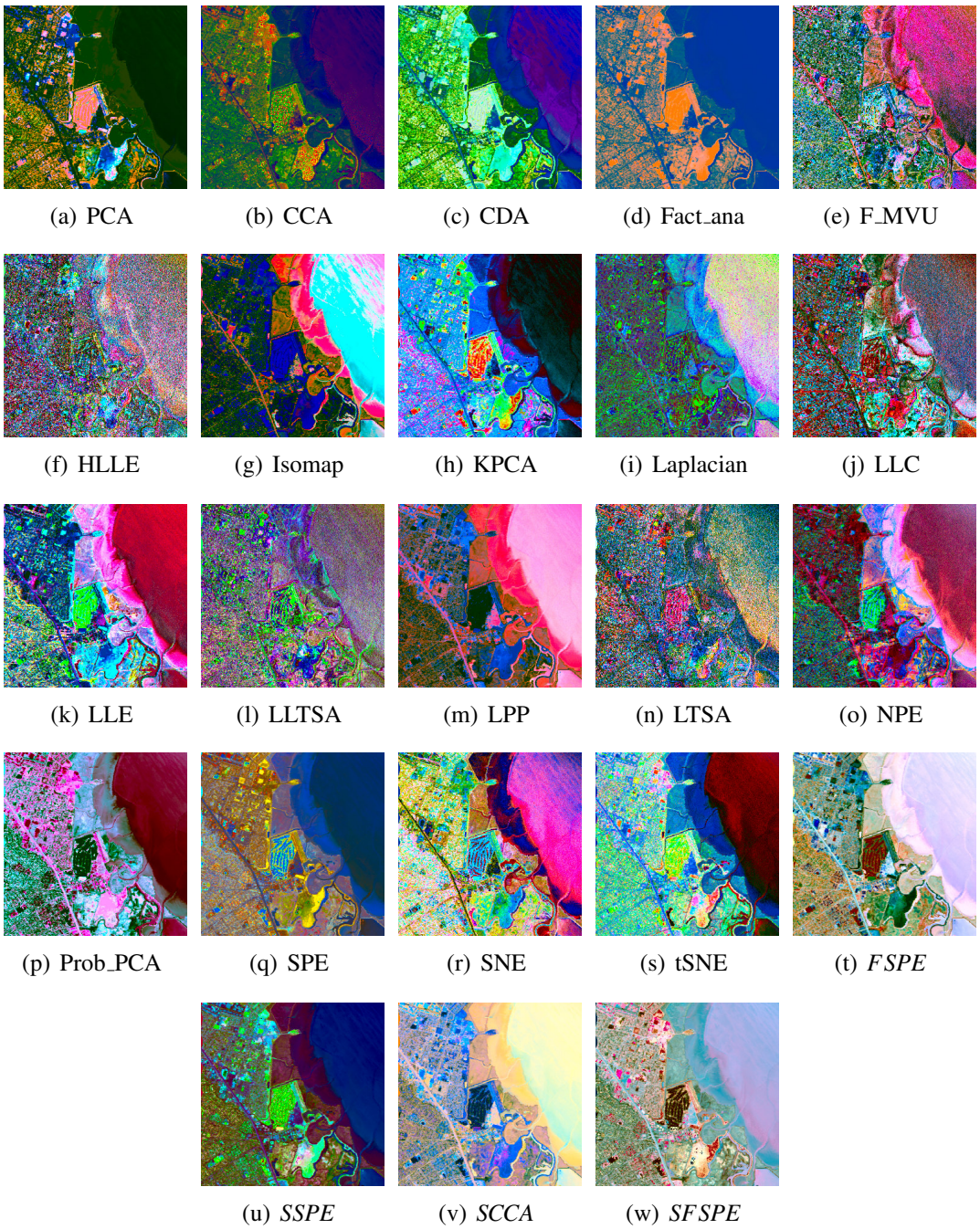
Figure 4.3: *SDR methods generate higher quality visualizations of Region_5, and SFSPE is better than SSPE and SCCA. More details appeared in their visualizations because their ability in preserving original information. The grey-scale efficiency images showed that the DR methods lost more information, which caused some of the colours in their visualizations not to be true.*

Table 4.5: *Correlation, LC and stress values of comparisons among 23 methods for three regions (Region_0, Region_4 and Region_5). SSPE, SCCA and SFSPE , when $S = 1$, are the best in all cases, where they have higher correlation and lesser stress values than other methods. (Correlation: the highest is a bester, LC: the highest is a best, Stress: the lowest is a better).*

	Correlation			LC			Stress		
Method	R_0	R_4	R_5	R_0	R_4	R_5	R_0	R_4	R_5
PCA	0.691	0.673	0.695	0.091	0.125	0.072	0.071	0.137	0.430
CCA	0.873	0.551	0.514	0.308	0.384	0.192	0.011	0.127	0.097
CDA	0.550	0.748	0.751	0.251	0.052	0.057	0.570	0.118	0.221
Fact_analysis	0.827	0.420	0.653	0.232	0.188	0.218	0.090	0.388	0.032
Fast_MVU	0.336	0.399	0.003	0.013	0.028	0.009	0.350	0.568	0.303
Hessian_LLE	0.265	0.291	0.003	0.011	0.010	0.009	0.072	0.543	0.213
Isomap	0.525	0.656	0.448	0.213	0.070	0.055	0.072	0.113	0.144
Kernel_PCA	0.697	0.768	0.594	0.054	0.071	0.039	0.044	0.096	0.046
Laplacian	0.240	0.619	0.581	0.015	0.021	0.036	0.108	0.177	0.097
LLC	0.243	0.339	0.320	0.010	0.017	0.010	0.242	0.184	0.352
LLE	0.325	0.368	0.374	0.078	0.039	0.010	0.071	0.145	0.286
LLTSA	0.324	0.428	0.208	0.013	0.010	0.006	0.108	0.175	0.097
LPP	0.752	0.605	0.660	0.077	0.119	0.077	0.112	0.178	0.097
LTSA	0.282	0.284	0.217	0.012	0.008	0.006	0.112	0.161	0.097
NPE	0.398	0.314	0.334	0.027	0.008	0.007	0.108	0.175	0.094
Prob_PCA	0.474	0.641	0.716	0.137	0.157	0.136	0.076	0.130	0.180
SPE	0.696	0.641	0.810	0.288	0.445	0.140	0.158	0.150	0.110
SNE	0.395	0.642	0.730	0.030	0.031	0.023	0.256	0.156	0.123
tSNE	0.581	0.437	0.648	0.037	0.027	0.024	0.387	0.118	0.256
FSPE	0.863	0.794	0.532	0.319	0.440	0.334	**0.001**	0.012	0.010
SSPE	0.997	0.966	**0.989**	0.421	**0.485**	0.277	0.002	0.066	0.016
SCCA	0.896	0.865	0.765	0.419	0.406	0.262	0.002	**0.002**	0.032
SFSPE	**0.998**	**0.977**	0.958	**0.433**	0.437	**0.387**	**0.001**	0.004	**0.009**

Table 4.6: *The computation time, in seconds, of the SDR methods in the CPU and the GPU. the GPU is the best way to speed up the SDR methods; that is, where the speed of execution has been increased by approximately 74 times than when it ran on the CPU.*

	CCA	*SCCA*	SPE	*SSPE*	*FSPE*	*SFSPE*
CPU	15.519	3857.907	12.416	3107.147	12.416	3107.147
GPU		52.210		42.050		42.050



(a) PCA (b) CCA (c) CDA (d) Fact_ana (e) F_MVU

(f) HLLE (g) Isomap (h) KPCA (i) Laplacian (j) LLC

(k) LLE (l) LLTSA (m) LPP (n) LTSA (o) NPE

(p) Prob_PCA (q) SPE (r) SNE (s) tSNE (t) *FSPE*

(u) *SSPE* (v) *SCCA* (w) *SFSPE*

Figure 4.4: *Visualization of Region_0 by using 20 DR methods and 3 SDR methods. According to Table 4.5, the worst visualization is achieved by the Laplacian method, and the visualizations' quality of SSPE, SCCA and SFSPE are the better among them.*

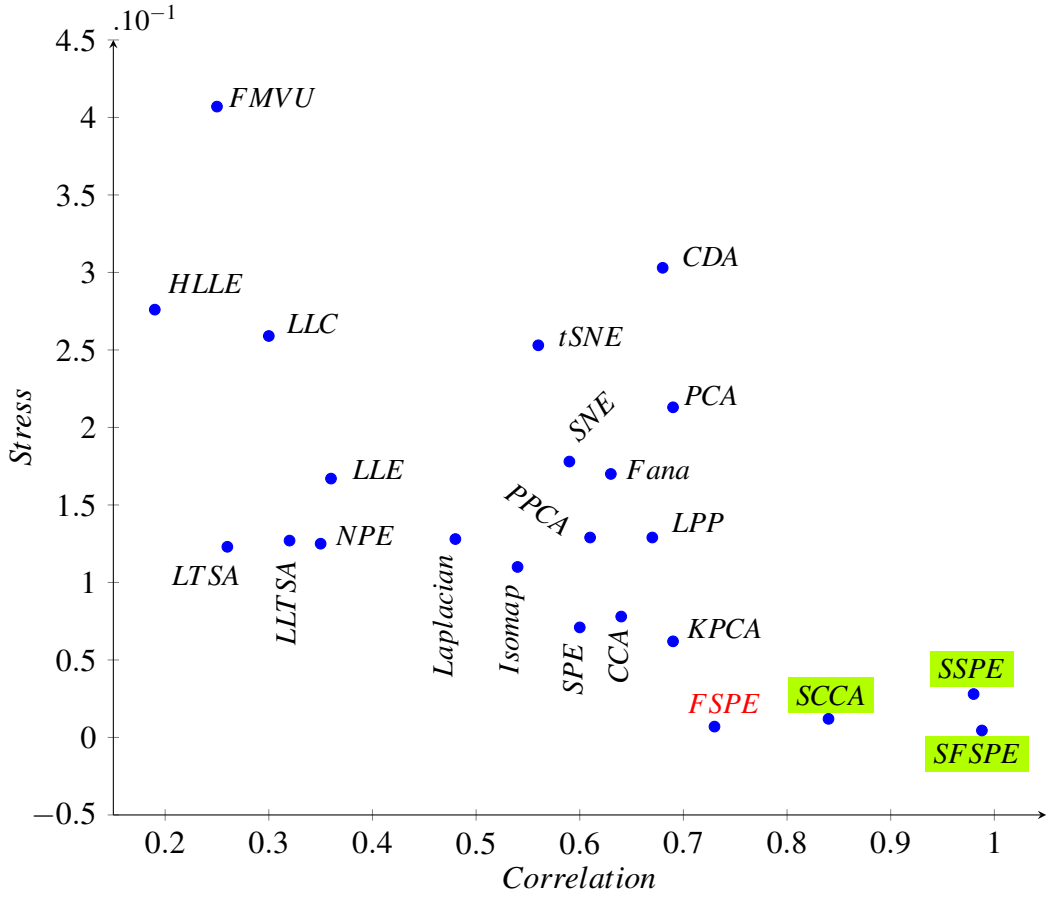


Figure 4.5: *The average of comparisons, in the Table 4.5, of 3 SDR methods (SSPE, SCCA and SFSPE) and 20 DR methods. SDR methods get the highest correlation and the lowest stress values.*

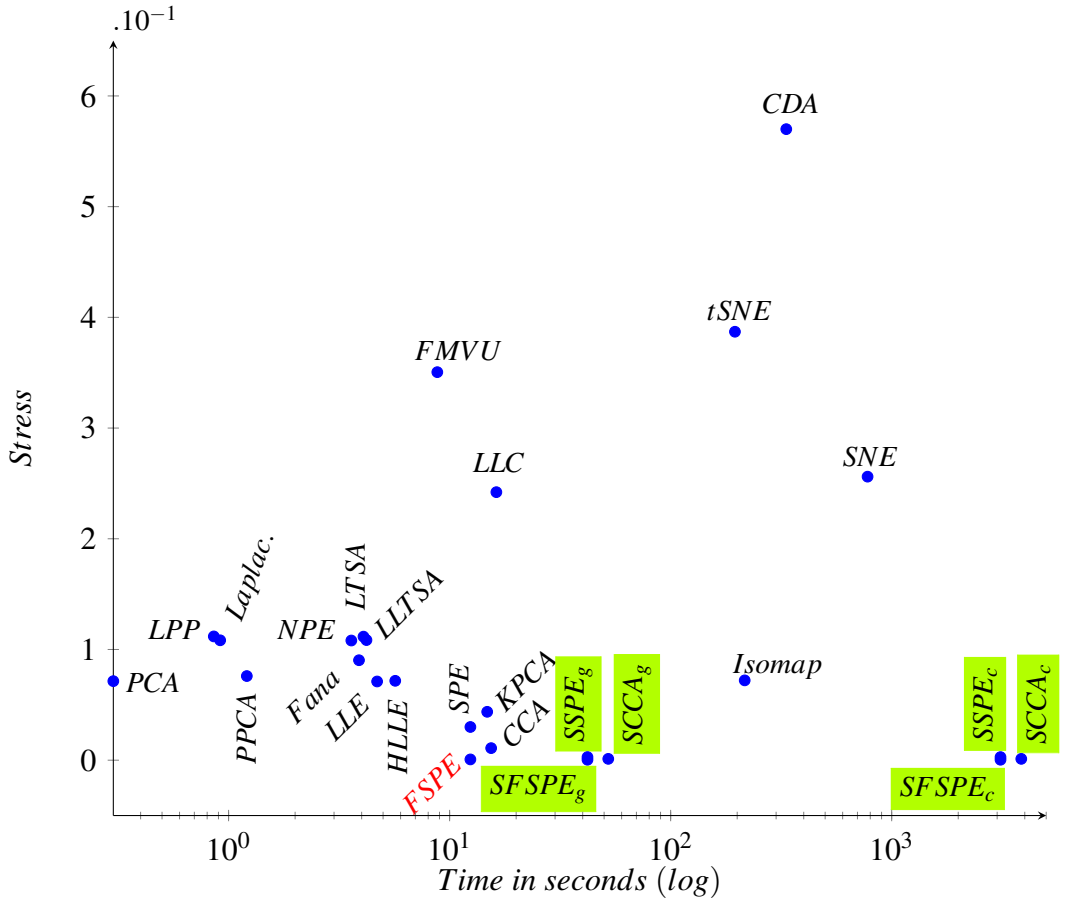


Figure 4.6: *The execution times of SDR methods in CPU (SCCA_c, SSPE_c and SFSPE_c) are very high, and the role of using GPU was very positive in increasing the speed of SDR methods (SCCA_g, SSPE_g and SFSPE_g), as in the Table 4.6. Thus, for Region_0, the execution speed of SDR methods are acceptable when comparing that with those of DR methods.*

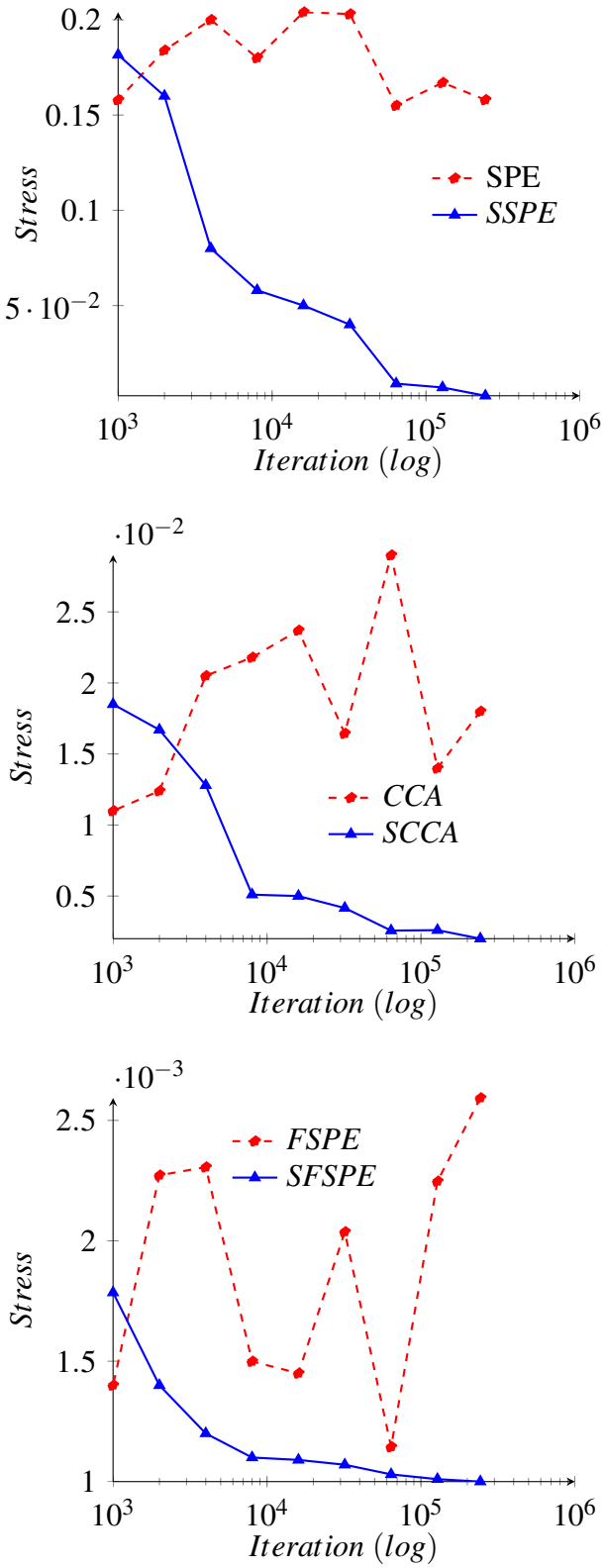


Figure 4.7: *DR and SDR use the same number of iterations. In DR, there is no significant impact on the change in DR's efficiency through iterations, but the stress is gradually reduced with SDR.*

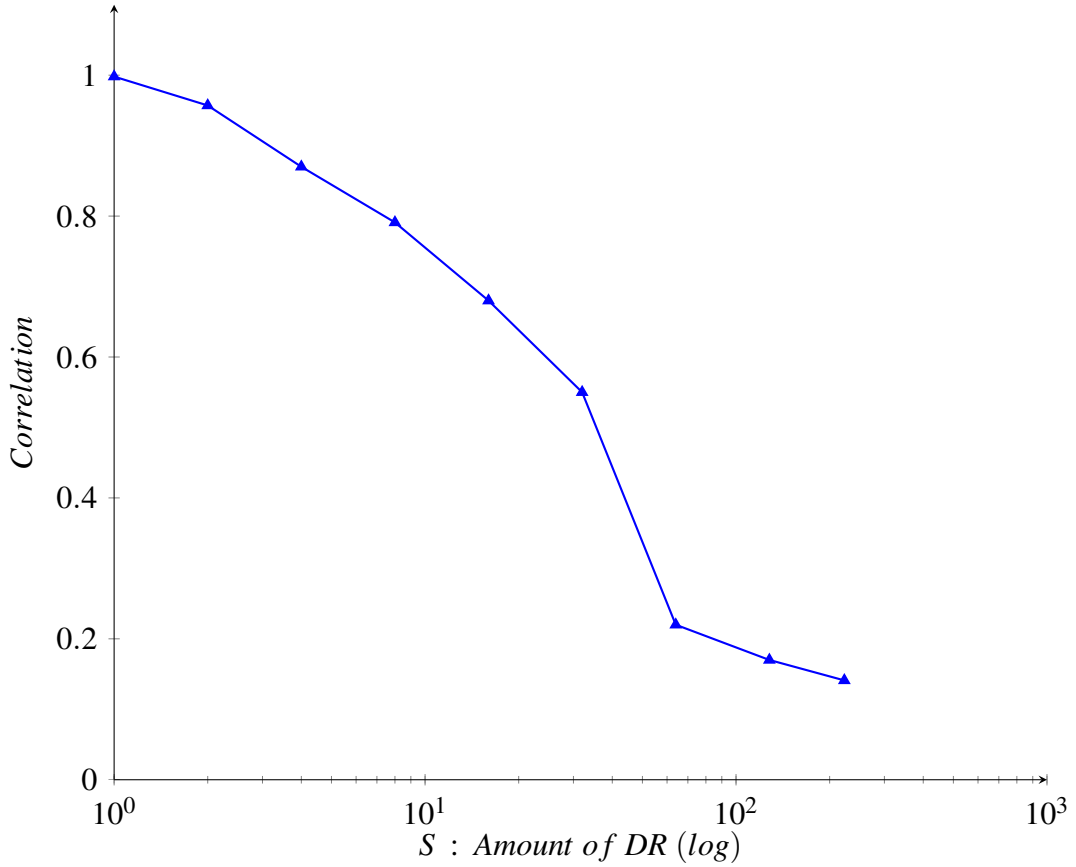


Figure 4.8: *The correlation measurement value of SSPE is very high (which is equal to 0.991 for region_9) when S (amount of DR) is equal to 1. The efficiency of SSPE is reduced when this reduction amount is greater than 1, and the lowest correlation value is 0.141 when the reduction amount is equal to 223.*

false colours compared to the direct projection of DR method, where those results were confirmed by comparisons of our method with 20 other methods. It has been also demonstrated that the speed of *SDR* on the GPU is much faster than it is on the CPU.

The main ideas presented in this chapter have been published in: Najim S. and Lim I. (2014). Visualization of Remote Sensing Imagery by Sequential Dimensionality Reduction on Graphics Processing Unit. In Proceedings of the 5th International Conference on Information Visualization Theory and Applications (IVAPP 2014), , pages 71-79, Lisbon, Portugal, 5-8 January.

Chapter 5

Faithful Visualization of Different Data Sets

In this chapter we will address the following topics:

1. Unfolding the 3 dimensions of a curved cylinder data sets.
2. Projecting the 560 dimensions of a human face data sets into two dimensions.
3. Representing and classifying the two networks into two dimension space.
4. Conclusion.

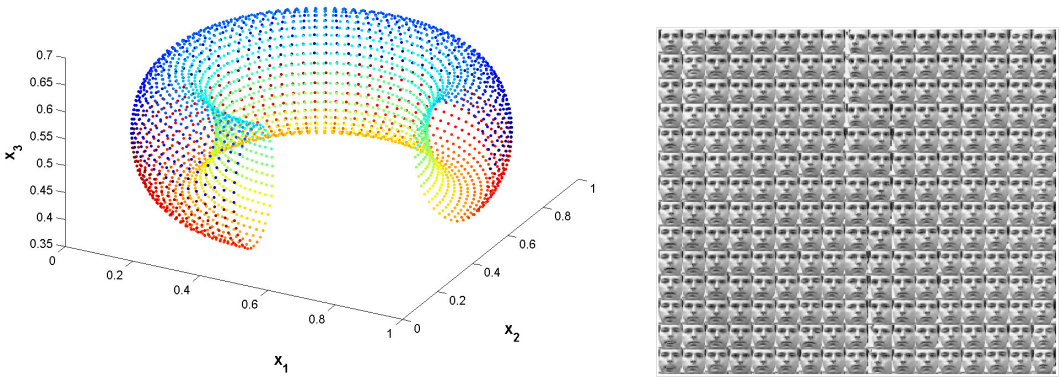
5.1 Introduction

Many well-known and promising DR methods are used in visualisation. However, the results might face the problem of a false neighbourhood, which could lead to incorrect analysis. Our suggested methods, *FSPE* and *SDR*, can be used to overcome these errors as far as possible. *FSPE* and *SDR* are able to produce visualisations by preserving the corresponding point distances between the projected space and the original data sets as much as possible.

In this chapter, we will apply our proposed methods, *FSPE* and *SDR*, to reduce the dimensionality of the experimental and real data sets. The comparison of 20 DR methods will be based on the three measurement metrics: correlation, LC and stress.

To apply *SDR*, we will select *SFSPE*. The GPU application was applied in this chapter to accelerate the projection process. More details about GPU application will be provided in Chapter 6.

The results show the ability of our methods to preserve neighbourhood relationships, and to reveal much interesting information. The large dimensions of the human face data sets were reduced to two dimensions by our method, and they are better when compared with the results of other methods. The comparison with the experimental data sets, as three dimensions of a curved cylinder, show the ability of *FSPE* to unfold these complex data sets and, at the same time, to preserve most of the information of the original data sets. In the network classification, our methods are not always the best, but their classifications are acceptable.



(a) Curved cylinder data sets

(b) Frey face data sets

Figure 5.1: *a) Three dimension of curved cylinder is described by* $x = (7 + 2\ cos(u))\ cos(v)$, $y = (7 + 2\ cos(u))\ sin(v)$, $z = 2\ sin(u)$, *where u and v are two dimension random numbers in the interval [0, 12.5664] and [0, 4.7124], respectively. The total number of points is 3200. b) Sample images of Frey face data sets. These data sets consist of 1965 images (each image in 20 x 28 gray scale pixels) of a single person face extracted from a digital movie. These data sets in the dimension 560.*

5.2 Data Sets

1. Curved cylinder data sets: Figure 5.1(a) shows these data sets. The goal is to unfold these data sets into two dimensions. The colours are added to the points to show their locations in the unfolded space.

2. Human face data sets: Figure 5.1(b) shows a sample of the images in the Fray face data sets. These data sets are a short video for Frey with different facial expressions. They have 1965 images, and the size of each one is 20x28. The images of these data sets will be projected into two dimensions to put them in a sequence close to that in the video clip. these data sets are available at http://www.cs.nyu.edu/ roweis/data.html.

3. Network: Two networks will be classified by DR. The first network, American College Football, is formed by teams in nodes, and the two teams are connected by edge if they have played each other that season. This network has 115 nodes (teams) and 613 edges (games). The goal is to find a two-dimensional representation to classify these data sets. It is available at https://Networkdata.ics.uci.edu/data.php?id=5.
   The Primary School Network is the second data sets and represents the relationship between students in different classes. This network has 10 classes, with nine of them being student classes and one being a teacher class. This network is available at http://www.sociopatterns.org/datasets/primary-school-cumulative-Networks/.

5.3 Unfolding The Curved Cylinder Data Sets

Figure 5.2 shows the unfolded space of the three dimensions of the curved cylinder data sets which are carried by the PCA, CCA, CDA, Isomap, LLE, SPE, tSNE and *FSPE* methods, and the results of the other methods are in Appendix A (in the Figures 1 and 2). Many methods cannot unfold these data sets, for example, Factor_analysis, FastMVU, Hessian_LLE, Laplacian, LLC and LLTSA. Other methods can unfold these data sets, but the results are not as efficient as they should be. For example, the linearity of PCA causes a compression of the projected space rather than its unfolding. Curved

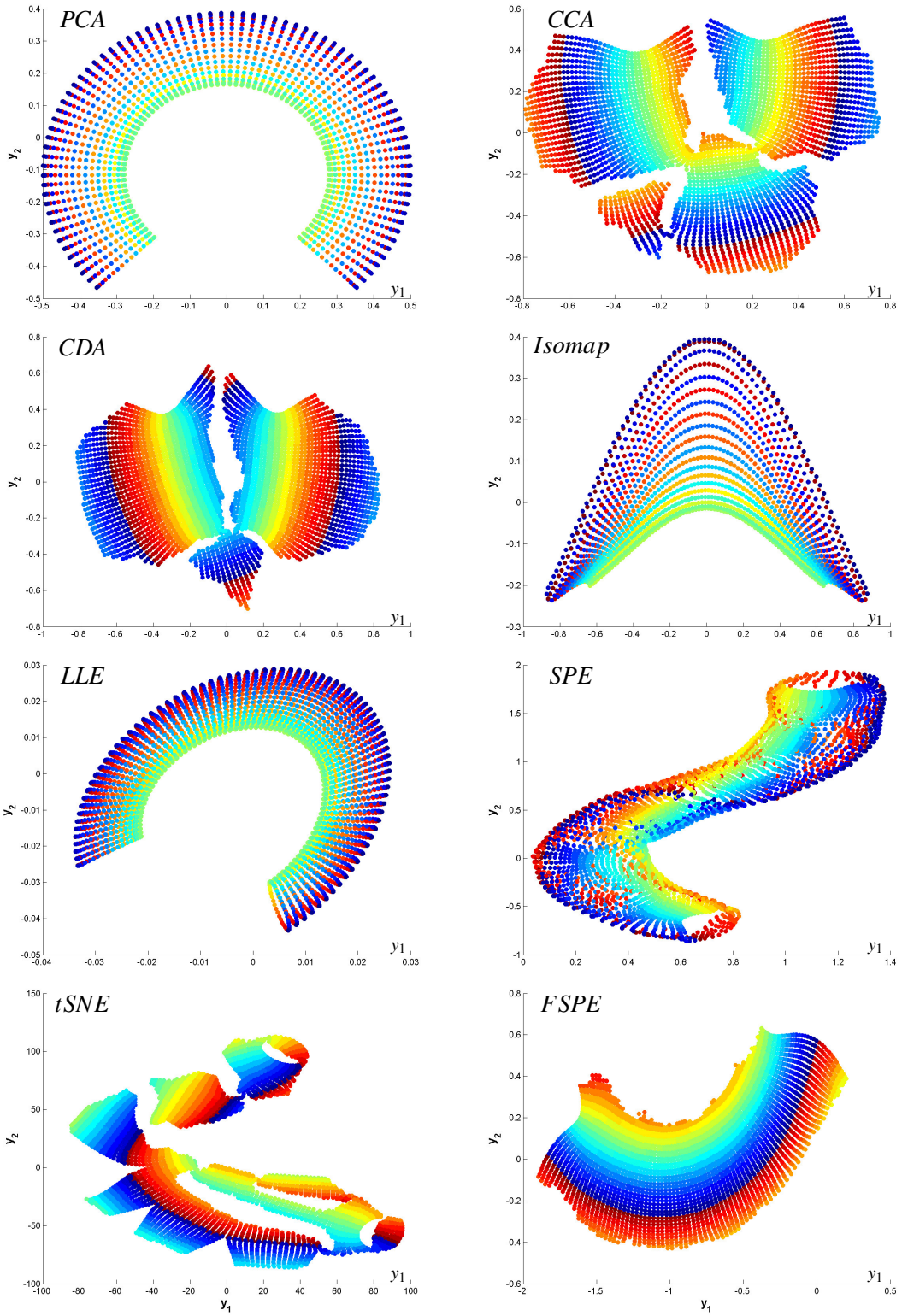


Figure 5.2: *Unfolded space of a three-dimensional curved cylinder by PCA, CCA, CDA, Isomap, LLE, SPE and tSNE and FSPE. The best unfolded space is the one that preserves the neighbourhood relations between points. The colour of each point is the same colour of its corresponding point in the original data sets, as shown in Figure 5.1(a). Therefore, these colours help us to see how FSPE's unfolded spaces are consistent and do not have any type of tearing or flattening.*

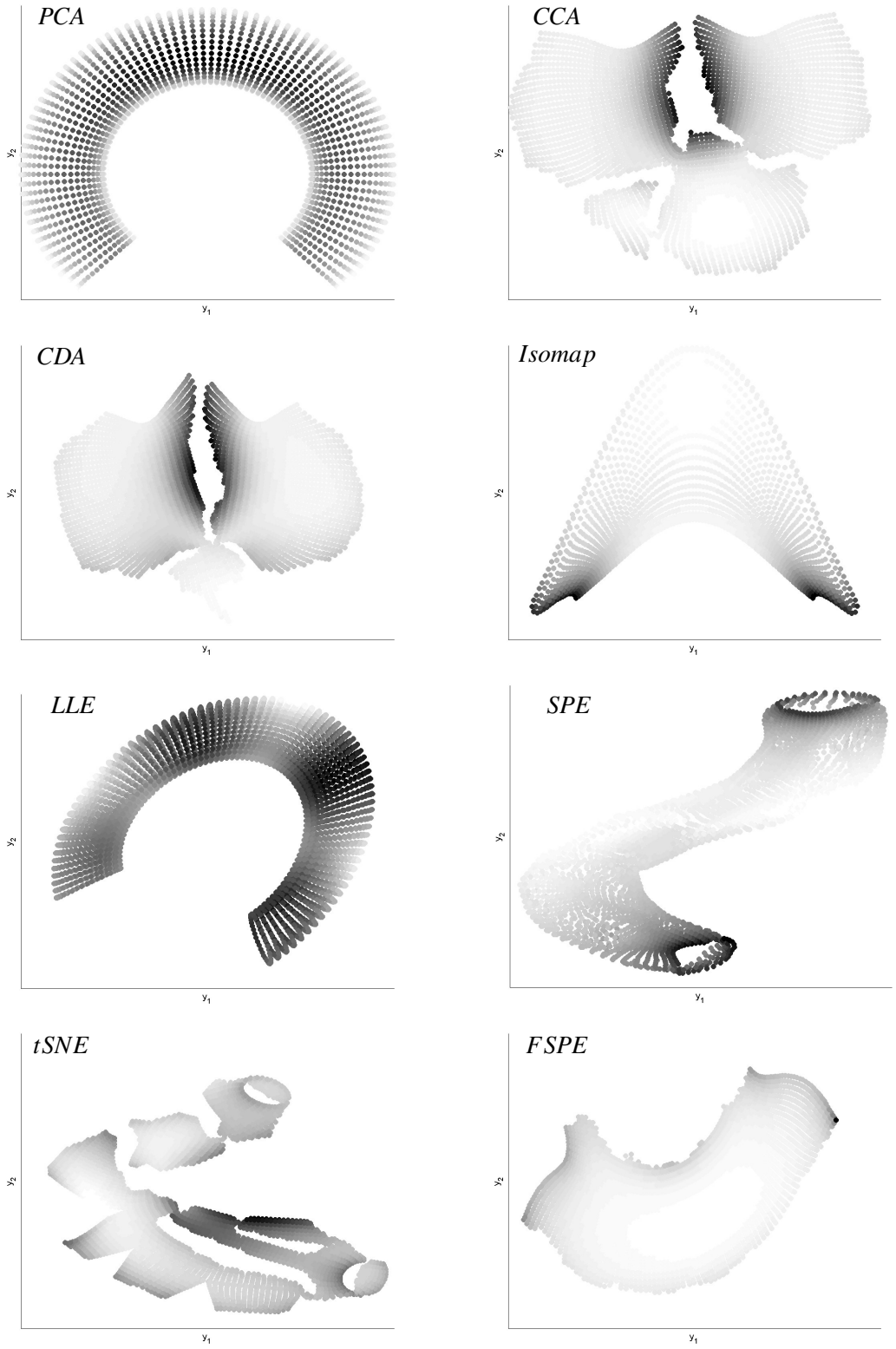


Figure 5.3: *The grey-scale efficiency images by the point-wise correlation metric of the unfolded space of curved cylinder in Figure 5.2. The bright white of most points in the FSPE's result indicates that the neighbourhood relations of these points are preserved.*

Table 5.1: *Unfolded spaces, in Figure 5.2, of the curved cylinder data sets by using 20 methods are measured by correlation, LC and stress metrics. The measurement results show that our method (FSPE) gets satisfactory results, where it got the highest correlation and LC measurement values, and the lowest stress error. (Correlation and LC: the highest is a bester, Stress: the lowest is better)*

Method	Correlation	LC	Stress
PCA	0.9189	0.5311	0.2405
CCA	0.7848	0.8850	0.1935
CDA	0.7190	0.8760	0.2046
Factor_analysis	0.8621	0.1586	0.1893
FastMVU	0.2683	0.1709	0.3260
Hessian_LLE	0.8474	0.2530	0.2170
Isomap	0.9799	0.5092	0.0489
Kernel_PCA	0.9149	0.5254	0.2663
Laplacian	0.8389	0.2355	0.3542
LLC	0.0423	0.1705	0.4866
LLE	0.8665	0.5185	0.2007
LLTSA	0.9189	0.5311	0.2405
LPP	0.5901	0.3892	0.2612
LTSA	0.8472	0.2539	0.2386
NPE	0.4630	0.3900	0.2896
Prob_PCA	0.8457	0.5218	0.2100
SPE	0.9779	0.6308	0.0716
SNE	0.9431	0.5706	0.0977
tSNE	0.8768	0.8230	0.1909
FSPE	**0.9870**	**0.8855**	**0.0161**

cylinder properties can be seen in LPP, NPE and SNE, but they could not unfold the data. Unfolding shapes by SPE is different because neighbourhood relations among points are preserved in some regions, but they overlap in other locations of its unfolded space.

Among the 20 methods, four CCA, CDA, tSNE and *FSPE* can unfold the curved cylinder data sets into two dimensions. The projected spaces by CCA and CDA are torn into four and three parts, respectively. Thus, the cohesion properties of the unfolded spaces are lost. The best unfolded space by tSNE is worse than that of CCA and CDA, because it is torn into several parts. The unfolded space is done by *FSPE*, where the cohesion of its unfolded shape is kept. Preserving neighbourhood relations by *FSPE*

overcomes the problems of tearing and overlapping.

In a quantitative comparison, Table 5.1 shows the results of the comparisons among the 20 methods by using correlation, LC and stress metric measurements. *FSPE* got fewer errors in stress measurement, which is 0.0161, and got higher degrees of preserving faithfulness distance in correlation and LC measurements, which are 0.9870 and 0.8855, respectively. The worst results were obtained by LLC, in correlation and stress metrics, and Factor_analysis, in the LC metric. The grey-scale efficiency images by point-wise correlation in Figure 5.3, and in the Appendix A (in the Figures 3 and 4) for other methods, show the locations of the high- and low-efficiency points in the unfolded space. They prove what has been achieved by our method, where *FSPE* can overcome the false neighbourhood errors as well. Most of its points are bright white to show their ability to preserve the neighbourhood distances of the curved cylinder data sets.

5.4 Projecting The Frey Face Data Sets into Two dimensional Space

In Figures 5.4 and 5.5, the 560 dimensions of the Frey face data sets are projected into a two-dimensional space by using our proposed methods (*FSPE* and *SFSPE*). The results of the other 19 methods are in Appendix A (in the Figures 5 and 6). Table 5.2 shows that the *SFSPE* method is better than the other methods because that method got the highest values in the correlation and LC measurement metrics, which are 0.8296 and 0.5858, respectively, and lowest error in the stress metrics, which is 0.1089. The *FSPE* method is the second-best method. The worst methods used to reduce the Frey face data sets into two dimensions are Kernel_PCA, Hessian_LLE and Laplacian, depending, respectively, on the correlation, LC and stress metrics.

In Figure 5.4, we show that *FSPE* was able to classify images based on the amount of similarity. Images that show a person as he looks to the left side are projected at the top of the projected space. They are, in turn, classified into two classes: The left represents the state of happiness, and the right represents the state of unhappiness. We also note that showing the tongue occurred in the circumference of the state of unhappiness. Images that show a person as he looks to the right side are projected

Table 5.2: *The results of reducing the dimensionality of the Frey face data sets into two dimensions, which are carried by 21 dimension reduction methods, are measured by using the correlation, LC and stress metrics. The measurement results show that our methods (FSPE and SFSPE) got the highest values in the correlation and LC measurements, and the lowest errors in stress measurement. (Correlation and LC: the highest is better, Stress: the lowest is better)*

Method	Correlation	LC	Stress
PCA	0.7656	0.2216	0.2773
CCA	0.7719	0.3854	0.1246
CDA	0.7166	0.4159	0.1241
Factor_analysis	0.7462	0.2304	0.1757
FastMVU	0.3634	0.0772	0.9978
Hessian_LLE	0.0130	0.0125	0.9969
Isomap	0.7805	0.2522	0.2436
Kernel_PCA	0.0030	0.0535	1.0000
Laplacian	0.6956	0.3103	1.0000
LLC	0.1059	0.2527	0.9965
LLE	0.4424	0.1111	0.9999
LLTSA	0.0130	0.0125	0.9999
LPP	0.3422	0.0869	1.0000
LTSA	0.0037	0.0130	0.9999
NPE	0.6990	0.1161	0.9999
Prob_PCA	0.7609	0.2350	0.9937
SPE	0.6242	0.0762	0.9984
SNE	0.0059	0.0138	0.7547
tSNE	0.6591	0.5779	0.8527
FSPE	0.8105	0.5740	0.1193
SFSPE	**0.8296**	**0.5858**	**0.1089**

in the lower part of the projected space. These images, in turn, separate the states of happiness from those of unhappiness, as the images of the first and second states are put in the left and the right of the projected space, respectively. The last thing that can be observed, when a person looks forward, is that the images are placed in the middle of the projected space.

SFSPE, in Figure 5.5, has maintained a relationship between the images, where they are divided into three classes. In the first class, when a person looks to the right, the images have been projected in the left side of the projected space. In the second class, when a person looks to the right, the images have been projected in the left of the projected space. The state of happiness in the images of first and second classes

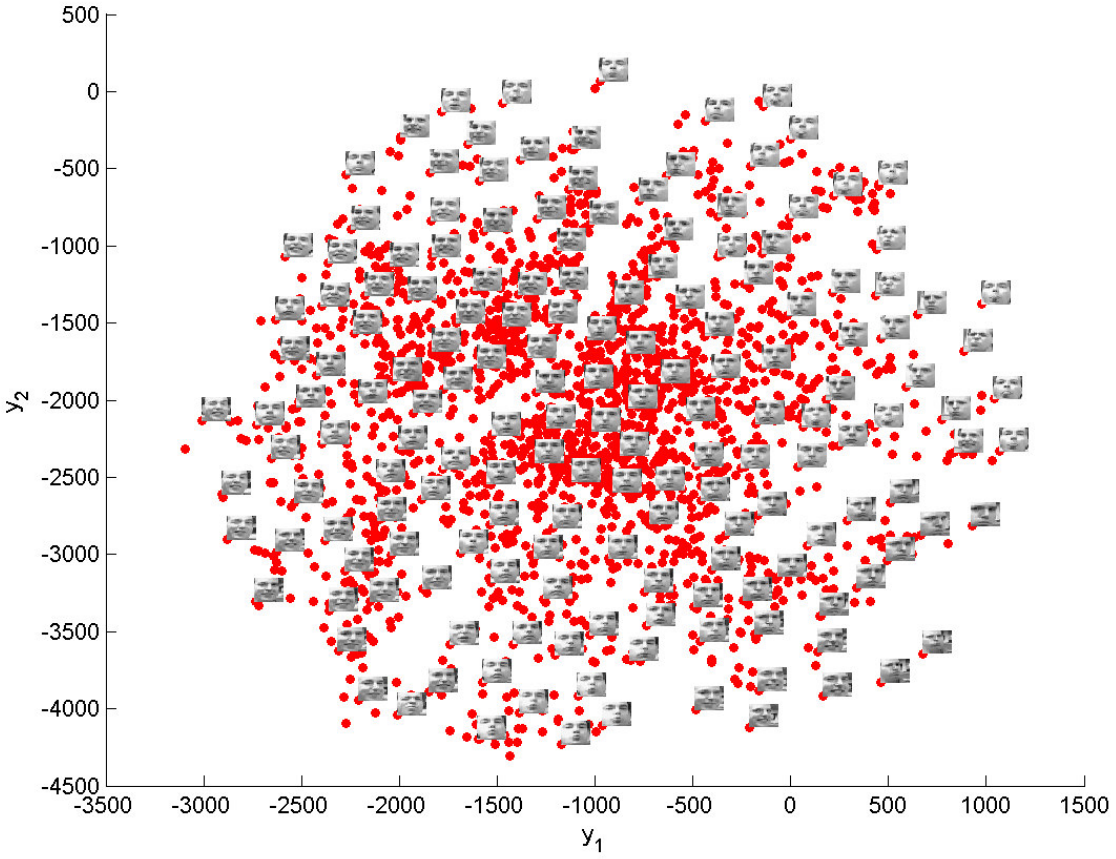


Figure 5.4: *Projecting the Frey face data sets into 2-dimensional space by using FSPE.*

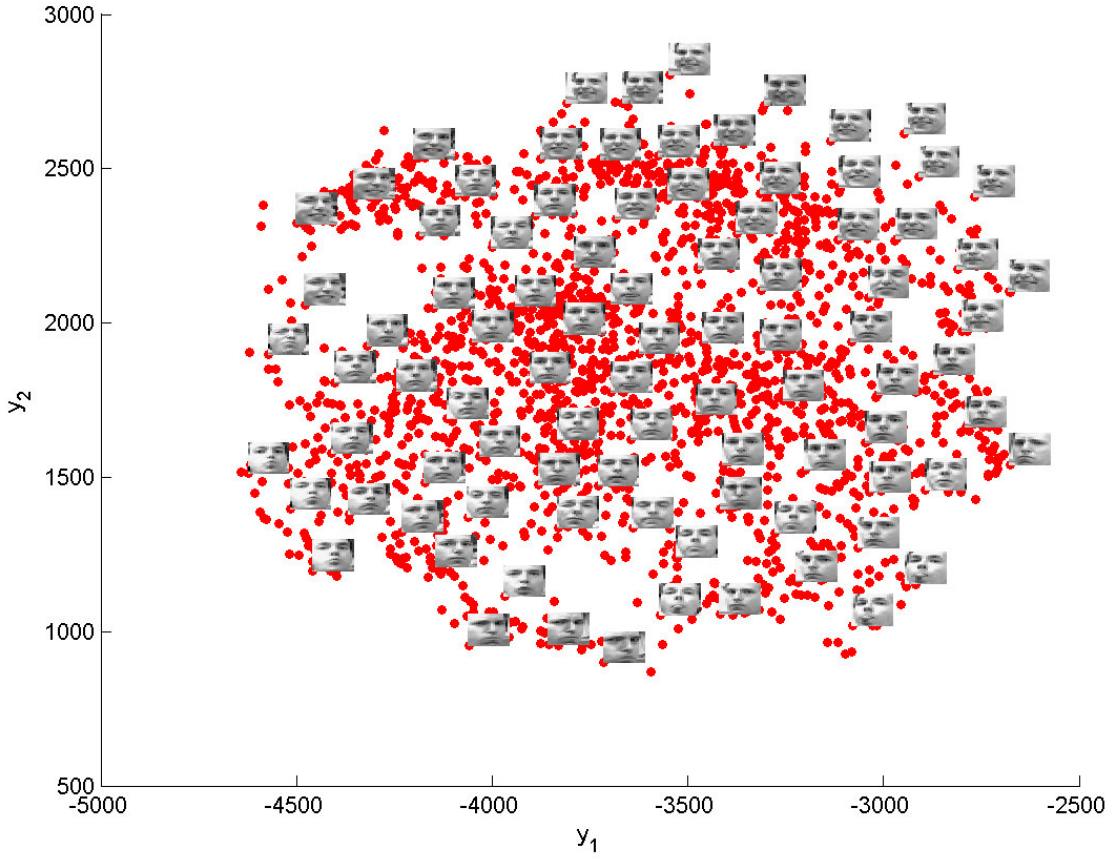


Figure 5.5: *Projecting the Frey face data sets into 2-dimensional space by using SFSPE.*

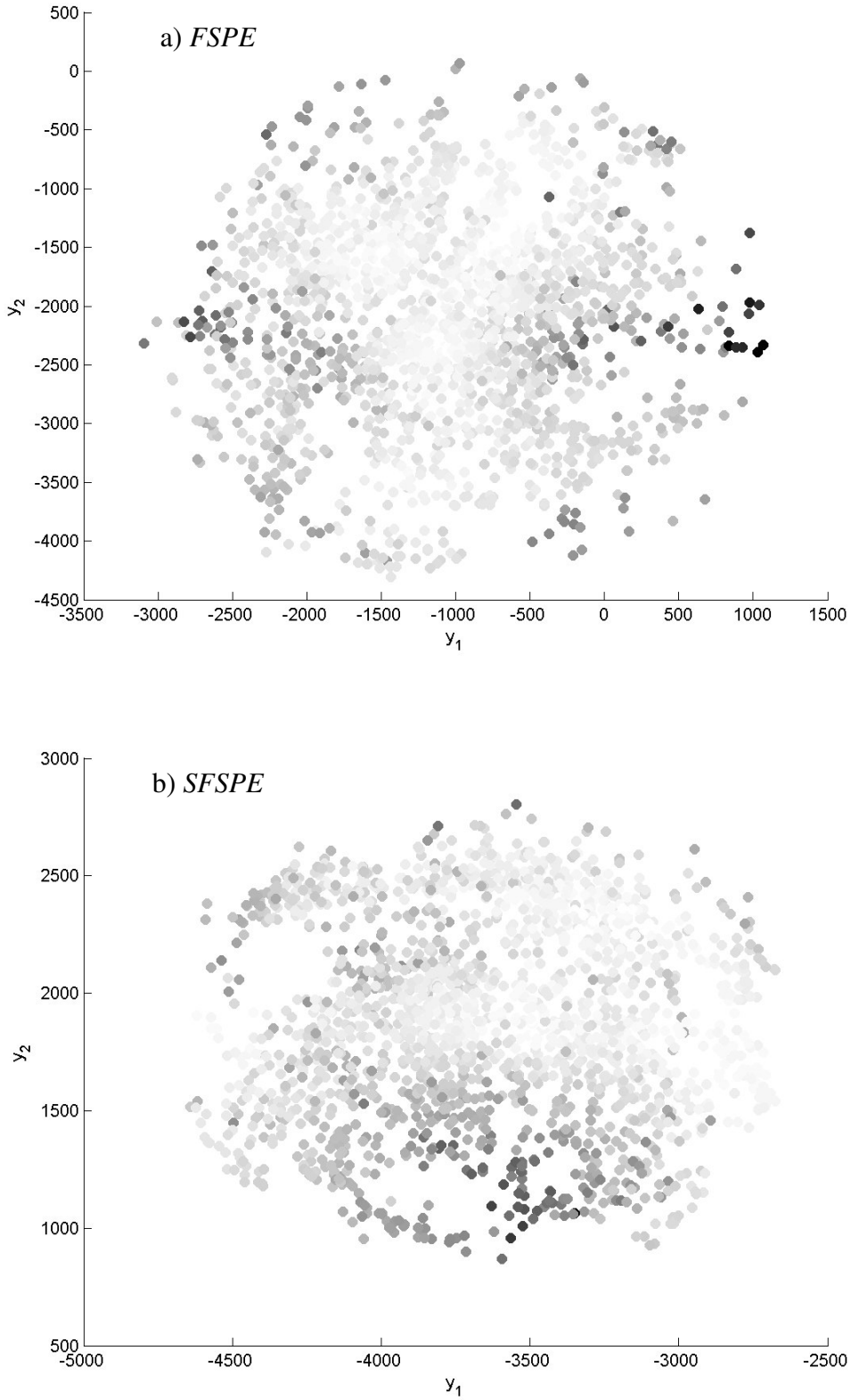


Figure 5.6: *Grey-scale efficiency image by the point-wise correlation metrics of results in Figures 5.4 and 5.5 by our suggested methods. The number of dark points in the projected space of FSPE and SFSPE is fewer than those by other methods, which are in Appendix A (in Figure 3 and 4).*

was at the top, and the state of unhappiness at the bottom. In the centre of the projected space, the images of a person as he looks forward were projected.

Figure 5.6 shows the grey-scale efficiency images by the point-wise correlation metrics of the results in Figures 5.4 and 5.5, and the grey-scale efficiency images of other methods, in the Appendix A (in the Figures 7 and 8). We can see that the number of dark points in the results of *FSPE* and *SFSPE* is less than those in other methods, which indicates that the proposed methods are minimising false neighbourhood error as much as possible.

5.5 Network Classification

Classifying a network is a very important step in developing a better understanding, recognition and clarification of the relationship between datapoints so as to allow a good analysis. In addition, classification might be important as a means of reducing the number of comparisons and focusing on the regions of interest.

First Network

The American College Football Network considers the games played between 115 teams in the year 2000. There is a priori knowledge about gathering these teams into groups. DR methods can classify this network by reducing its dimensionality into a small number of dimensions where most of the information is concentrated. In this section, we will evaluate the performance of *FSPE* and *SFSPE* in classifying this network, and compare them with 19 DR methods.

Figure 5.7 shows six of the low-dimensional representations, which are carried by PCA, Prob_PCA, SNE, tSNE, *FSPE* and *SFSPE*, of the American College Football Network. The classification of this network is clear in this figure. We can see the classification by SNE, tSNE, *FSPE* and *SFSPE* is better than by PCA and Prob_PCA. We measured the efficiency of 21 results by using correlation, LC and stress metrics, as shown in Table 5.3. Although *FSPE* and *SFSPE* have got higher correction and lower stress values, SNE is better when using the LC metric. Thus, in classification, we cannot say that our methods are the best, because some other methods are better. However, our methods obtained an acceptable classification for this network, while

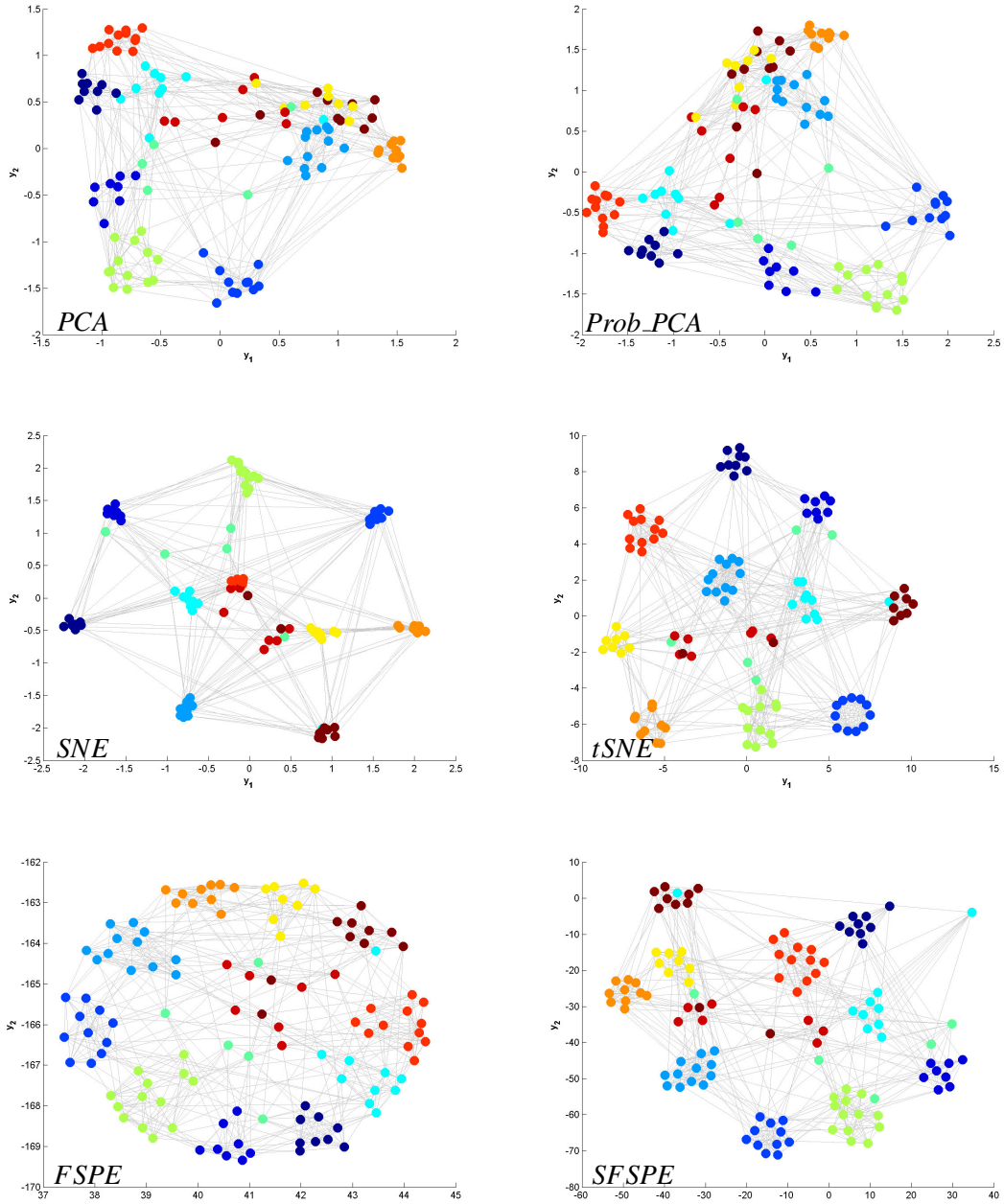


Figure 5.7: *Low-dimensional representation of the American College Football Network by PCA, Prob_PCA, SNE, tSNE, FSPE and SFSPE. The best representation is the one that preserves neighbourhood relations between points. The points which have the same colours are teams belonging to the same group.*

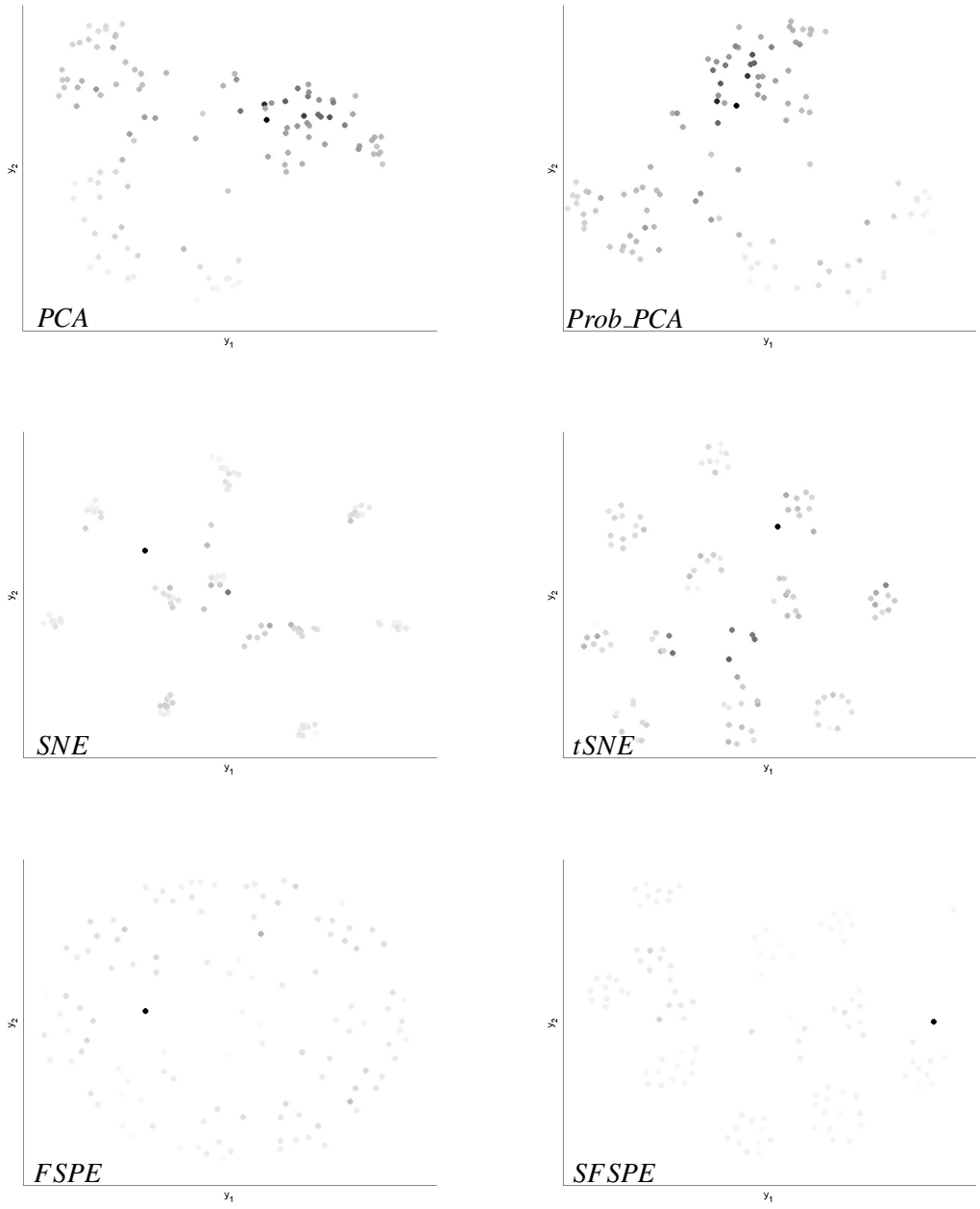


Figure 5.8: *The grey-scale efficiency images by the point-wise correlation metric of the low-dimensional representations of the American College Football Network in Figure 5.7. The bright white points indicate that the neighbourhood relations of these points are preserved.*

Table 5.3: *Measuring the low-dimensional representation of the American College Football Network, which was carried out by 21 methods, by using correlation, LC and stress metrics. (Correlation and LC: the highest is better, Stress: the lowest is better)*

Method	Correlation	LC	Stress
PCA	0.673	0.579	0.429
CCA	0.640	0.529	0.543
CDA	0.591	0.594	0.176
Factor_analysis	0.462	0.515	0.616
FastMVE	0.312	0.362	9.363
Hessian_LLE	0.294	0.417	0.579
Isomap	0.582	0.512	0.836
Kernel_PCA	0.156	0.356	0.948
Laplacian	0.613	0.598	0.976
LLC	0.189	0.246	0.880
LLE	0.568	0.545	0.926
LLTSA	0.306	0.575	0.941
LPP	0.449	0.499	0.983
LTSA	0.306	0.575	0.946
NPE	0.463	0.493	0.952
Prob_PCA	0.670	0.577	0.327
SPE	0.638	0.567	0.290
SNE	0.705	**0.638**	0.285
tSNE	0.667	0.571	1.693
FSPE	0.731	0.602	0.153
SFSPE	**0.766**	0.585	**0.083**

some methods failed in this task. The relationship between correlation and stress metrics is shown in Figure 5.9, which shows that the efficiency of some visualisations is not good, in that these visualisations generate a large amount of error, as in the case of Kernel_PCA, LLC, FastMVU, LTSA, LLTSA and Hess_LLE. In addition, we can show that some methods have a high degree of efficiency using the correlation metric and, at the same time, a very large error, as in the case of the tSNE and the Laplacian method. Other methods have remained in the middle in that they are not as efficient as expected for their results to be relied on, but their results are not so bad that they cannot be rejected at all.

When we compare the representation of these data sets, as shown in Figure 5.7, to determine the amount of false neighbourhood nodes, we show that the grey-scale effi-

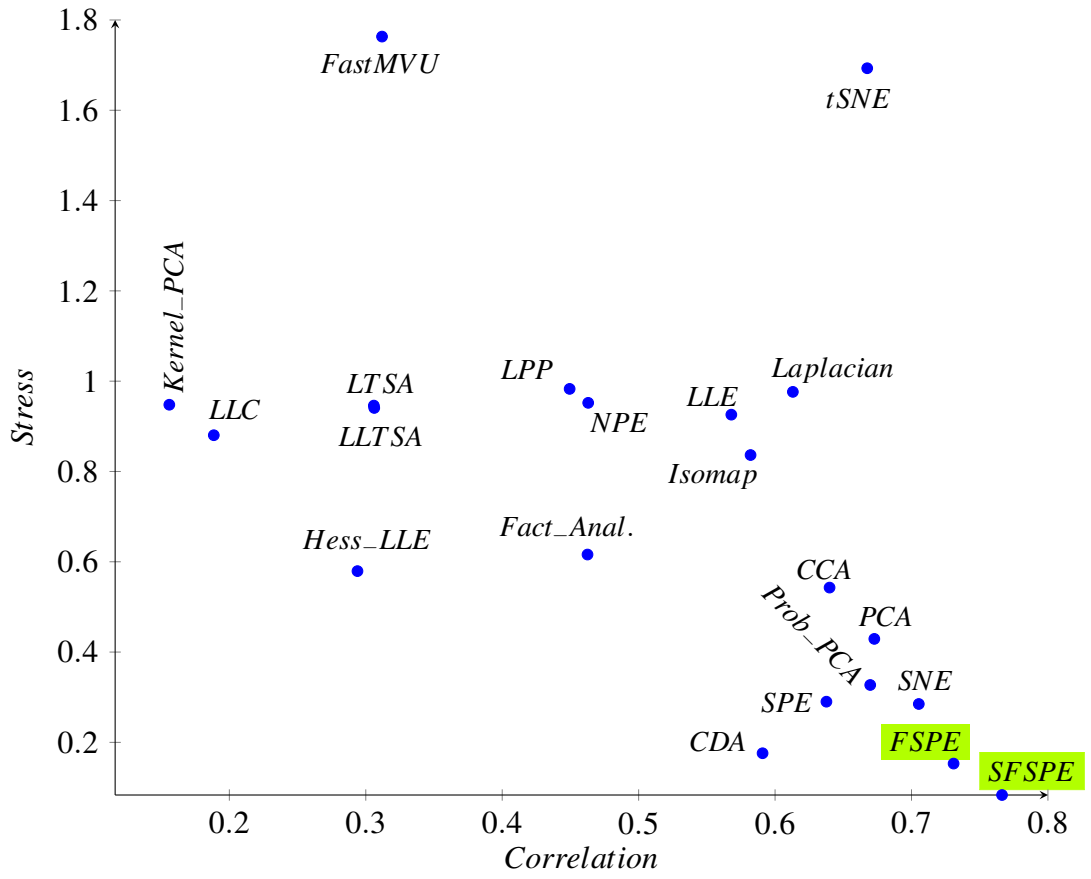


Figure 5.9: *The results of the comparisons, in Table 5.3, of our methods (FSPE and SFSPE) get the highest correlation and the lowest stress values.*

ciency image as measured by the point-wise correlation metric, gives a full explanation of this subject. In Figure 5.8, the dark points refer to the lost neighbourhood relationships with their neighbours. The evaluation of the representation of PCA, Prob_PCA, SNE and tSNE methods indicates that many nodes are false nodes. Grey-scale efficiency images of our proposed methods show that most of the points are the brighter white ones, which prove that the points could maintain their neighbourhood relationships as they were in the original space.

Second Network

The dimension of the Primary School Network is 236. The DR will be used to reduce this high degree of dimensionality into a 2-dimensional space. This network consist of

Table 5.4: *Measuring the low-dimensional representation of the second network, which was carried out by 21 methods, by using correlation, LC and stress metrics. (Correlation and LC: the highest is better, Stress: the lowest is better)*

Method	Correlation	LC	Stress
PCA	0.745	0.596	0.046
CCA	0.821	0.538	0.368
CDA	0.796	0.612	0.487
Factor_analysis	0.712	0.566	1.035
FastMVE	0.386	0.240	0.115
Hessian_LLE	0.107	0.103	0.117
Isomap	0.540	0.702	0.914
Kernel_PCA	0.048	0.108	0.117
Laplacian	0.503	0.716	0.117
LLC	0.268	0.178	0.100
LLE	0.347	0.521	0.118
LLTSA	0.590	0.393	0.118
LPP	0.414	0.708	0.117
LTSA	0.132	0.104	0.117
NPE	0.380	0.433	0.118
Prob_PCA	0.743	0.545	0.117
SPE	0.376	0.200	0.676
SNE	0.103	0.106	0.053
tSNE	0.635	**0.760**	0.112
FSPE	0.839	0.654	0.039
SFSPE	**0.841**	0.662	**0.040**

10 classes, and the low-dimensional space should allow the user to show these classes. Our methods (*FSPE* and *SFSPE*) will be compared with 19 DR methods in the classification of this network.

Figure 5.10 shows the six representations of the Primary-School Network by using PCA, CCA, CDA, Prob_PCA, *FSPE* and *SFSPE*. We used three metrics to measure the efficiency of the results. These are correlation, LC and stress, and Table 5.4 shows the details. The results show that our methods (*FSPE* and *SFSPE*), CCA and CDA, have higher correction values, with our methods the highest. By using LC metric, we find tSNE, Laplacian, LPP and Isomap have good accuracy, and the efficiency of our methods are inferior to them. Using the stress metric, the amount of error resulting from *SFSPE*, *FSPE*, PCA and SNE, are the less than the 21 DR methods.

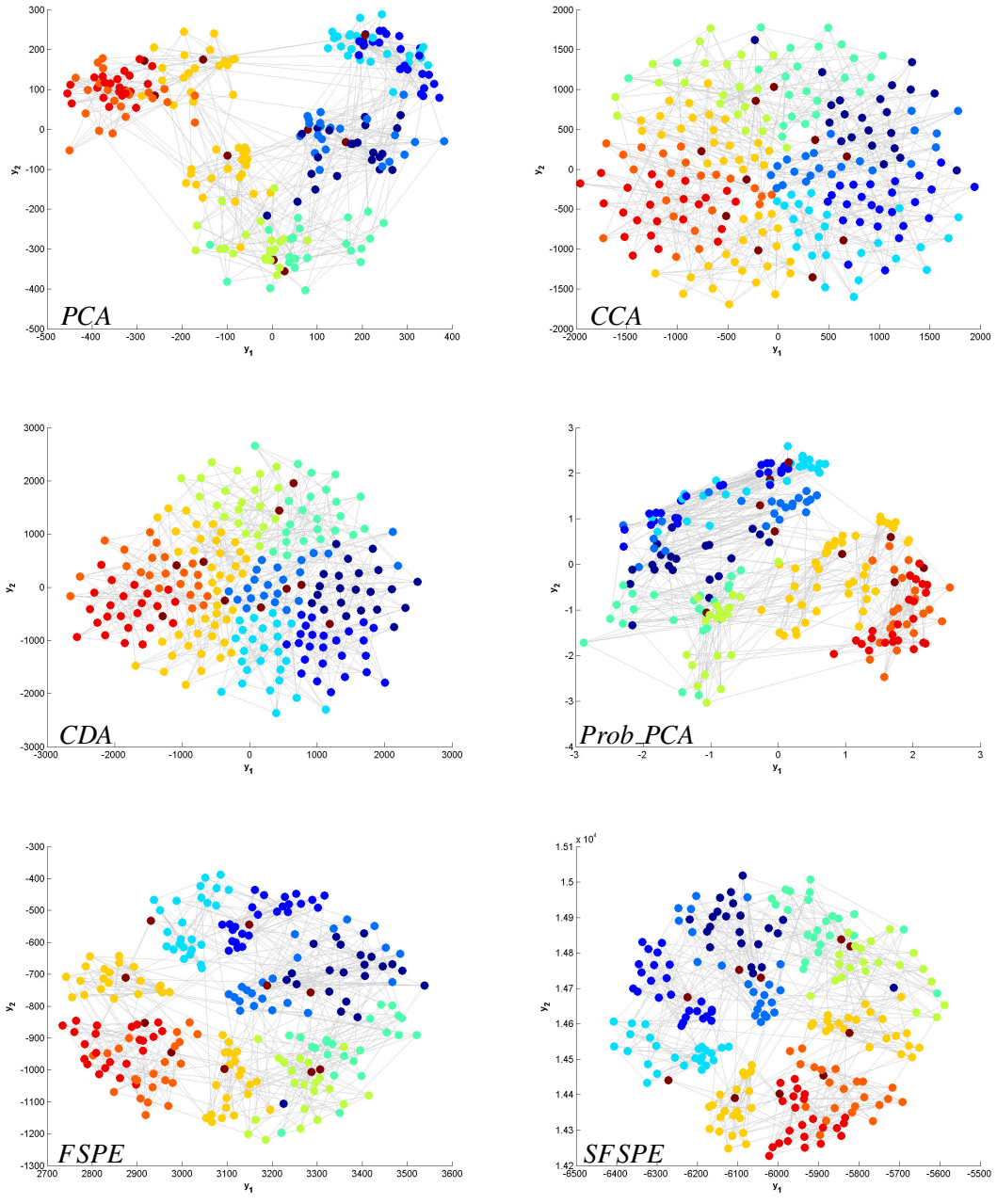


Figure 5.10: *Low-dimensional representation of the second network by PCA, CCA, CDA, Prob_PCA, FSPE and SFSPE. The best representation is the one that preserves neighbourhood relations between points. The points which have the same colours belong to the same class.*

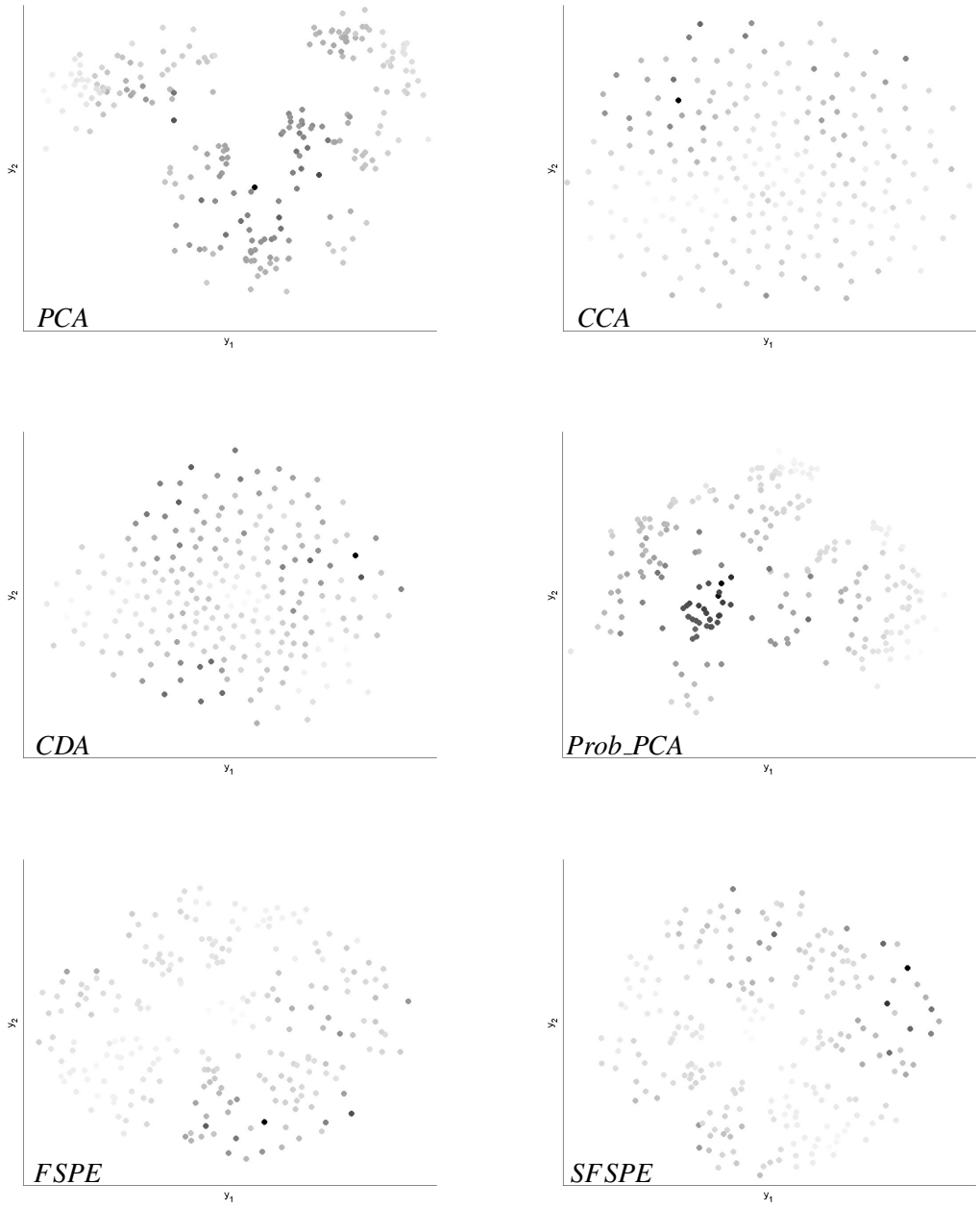


Figure 5.11: *The grey-scale efficiency images by the point-wise correlation metric of the low-dimensional representations of the second network in Figure 5.10. The bright white points indicate that the neighbourhood relations of these points are preserved.*

For the purpose to clarifying the efficiency of each point in the visualization, we use the point-wise correction metric. Figure 5.11 shows that preserving the neighbourhood relationships using *SFSPE* and *FSPE* is better than the rest of the DR methods.

5.6 Conclusion

We compared the results of our proposed methods (*FSPE* and *SFSPE*) and 19 other DR methods when applied to three scenarios: the unfolding Curved Cylinder data sets; projecting human face data sets onto two dimensions; and classifying the two networks. The primary aim of this chapter is to evaluate the efficiency of our methods in terms of neighbourhood preservation. The results showed the ability of *FSPE* to unfold the complex data sets, such as in the case of the curved cylinder. *FSPE* and *SFSPE* led to good, two-dimensional representations of the Frey face data sets. Our methods in terms of network classification are not always the best, in that other methods are sometimes better. However, the measurements found that visualisation by other methods lost more information by having many false neighbourhoods, and the visualisations by *FSPE* and *SFSPE* overcame the DR problem as much as possible. We have demonstrated that the two proposed methods can provide an insight into what was missed by the other methods, and have found that they are better in terms of preserving percentages of the neighbourhood distances. We believe that these results will stimulate others to adopt our visualisation approach.

The main ideas presented in this chapter is published in: S.A. Najim, I.S. Lim, Trustworthy dimension reduction for visualization different data sets, Inform. Sci. (2014), http://dx.doi.org/10.1016/j.ins.2014.03.048.

Chapter 6

Speed-up The Processing by Using Graphics Processing Unit

In this chapter we will address the following topics

1. Introduction of the graphics processing unit (GPU),
2. Our contributions in GPU application,
3. Experimental results,
4. Conclusion.

This chapter illustrates parallel processing in the graphics processing unit (GPU). In addition, we present a parallel method to visualize a remote sensing imagery data sets and measure its efficiency on the GPU. Visualization of a remote sensing imagery data sets is a common challenge task in DR. The requirement to accelerate the projection process and efficiency measurement of the visualization comes from the large size of the data sets. We have implemented the *FSPE* method on GPUs to speed up its projection process. To measure the efficiency of the visualization, the parallel codes of the two well-known metrics in this field namely, correlation and residual variance are introduced. The results showed that the high computational efficiency of the GPU helped to reduce the time spent on processing the results and computing their efficiency.

6.1 Introduction

Nowadays, the implementation time has become a key factor for the success of the software. Not too long ago, the focus was on the CPU in order to speed the computation, where it was required to wait at least 12 months for minor changes to the current speed. Although multiple CPUs have been introduced instead of a single CPU to implement the tasks in parallel, the efficiency of the speed has been limited to simple applications because of the vast amount of data.

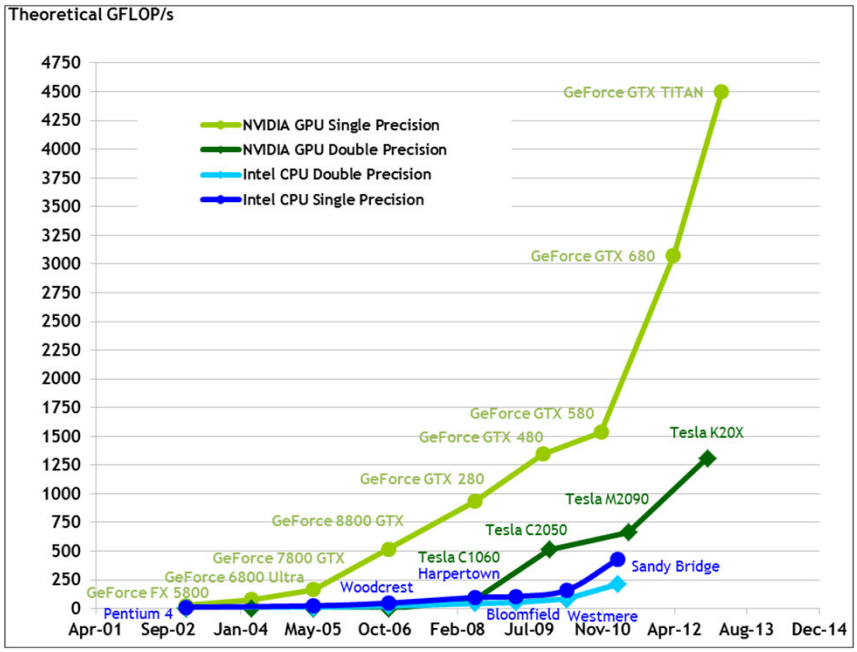
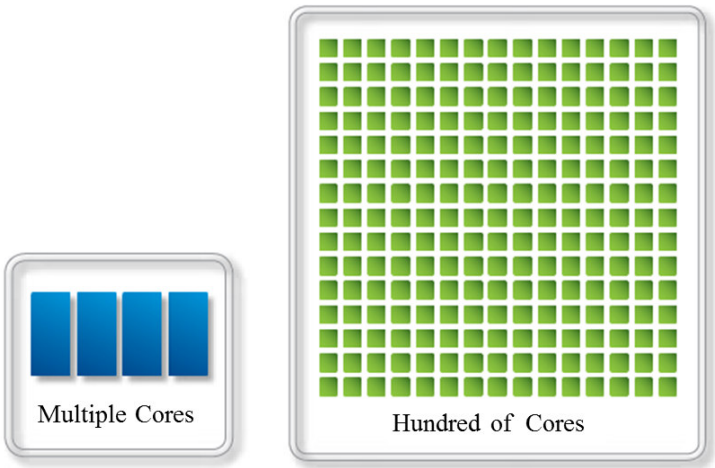


Figure 6.1: *The performance of the GPU has increased exponentially in recent years, which makes it deliver superior performance in parallel computation compared with the CPU [NVIDIA, 2013].*

Over the past few years, the processing power and memory bandwidth of the new generation of graphics cards have become significantly better than the CPU. In addition to their capability of displaying graphics, the GPUs can accelerate general purpose computations. It is useful for information extraction, in visualization, in telecommunications, and in many scientific fields, including biology and chemistry, and because its computational power goes far when compared with the CPU [Buatois et al., 2009] [Sanders and Kandrot, 2011]. Figure 6.1 shows that the exponential increase in the performance of the GPU in the last nine years is better than that of the CPU.

The tendency to use the GPU is due to the following reasons: Firstly, the technology of the CPU is not capable of managing a large scale of computations, and we can say that the improvement in the CPU technology has reached a stable scale. That means that the CPU alone in a computer system cannot provide what people demand. Secondly, the GPU's speed is 10-100 times faster than CPU, which makes it suitable to implement the challenge computational problems. Thirdly, complex computation tasks can be sent to the GPU by dividing them into several smaller subtasks that process at the same time. Finally, the GPU provides a chance to solve the impractical tasks that the CPU cannot because of its limited technology.

The GPU has been developed independently to have hundreds of cores, as in Figure 6.2. The GPU depends on the phenomenon "single instruction multiple data (SIMD)" to execute the same instruction on different shared processors by using different data. The GPU requires a parallel structure of a method in order to implement it, otherwise the execution would be very poor.



(a) CPU (b) GPU

Figure 6.2: *The general structure of the CPU and the GPU. a) The CPU contains very few cores that can work in parallel. b) The GPU contains hundreds of small cores that can work in parallel.*

Compute unified device architecture (CUDA) is the hardware and software NVIDIA parallel computing architecture that integrates with the environment, such as Microsoft Visual Studio C++, to provide a way to write the CUDA C++ (or CUDA C) program.

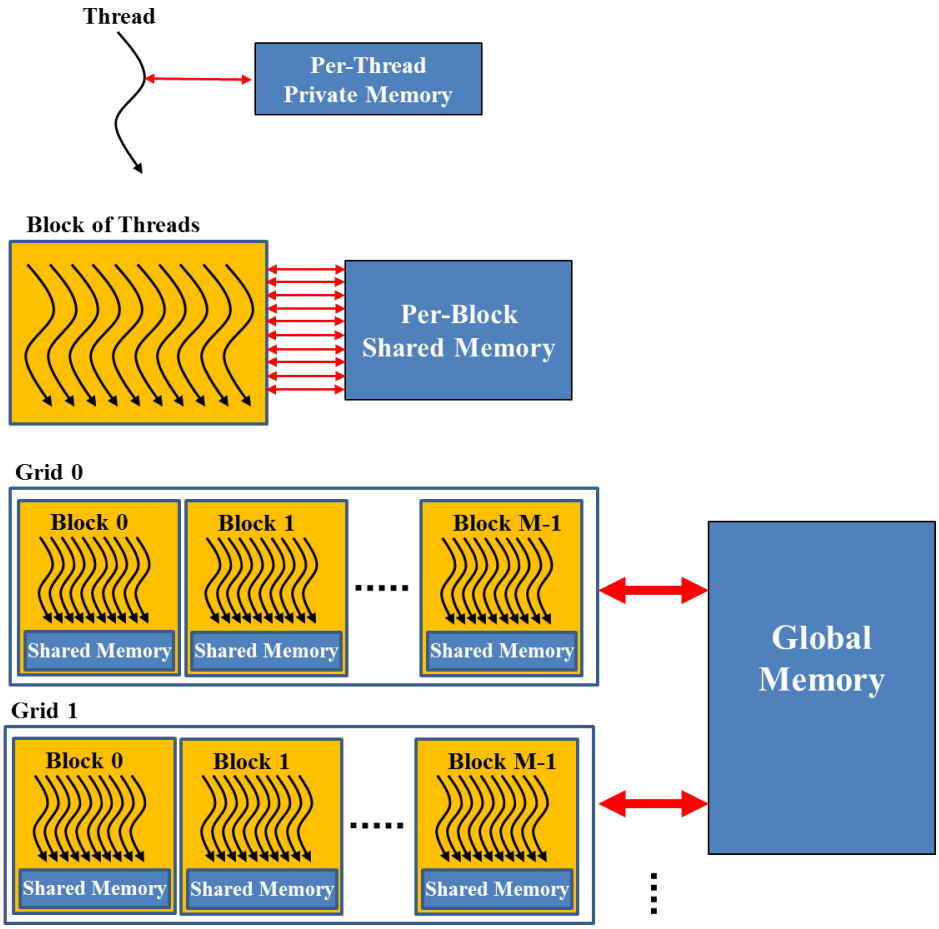


Figure 6.3: *GPU consist of many threads which are grouped into blocks, which are then grouped into grids of blocks. Three types of memory (private memory for a thread, shared memory for a block, and global memory for all threads) with more cores lead GPU to execute the parallel program independently, and in less time.*

It has two types of functions: host and kernel. Host functions are responsible for the execution of the sequential codes on the CPU, and the computing control is given to the GPU by calling the kernel function in order to execute the parallel codes. The GPU consists of a large set of threads that are grouped into blocks, and many blocks can be grouped into grids. Each thread has a small private memory, and each block has memory to share their threads together [NVIDIA, 2009]. Threads in a block synchronize their cooperation in accessing shared memory. Figure 6.3 shows the architecture of CUDA.

According to size of the data sets (N), the number of threads in a block can be

specified [Sanders and Kandrot, 2011]. For example, in the following CUDA C++ codes, the block is declared as one dimension vector has ($M = 256$) threads:

$$int\ \ ThreadsPerBlock = 256;$$
$$int\ \ BlocksNeeded = (N + ThreadsPerBlock - 1)/ThreadsPerBlock;$$
$$dim3\ \ dimBlock(ThreadsPerBlock);$$
$$dim3\ \ dimGrid(BlocksNeeded);$$

Another possibility is the block can be declared as two dimension matrix has $16 * 16$ threads, as in the following codes:

$$int\ \ Bx = 16;$$
$$int\ \ By = 16;$$
$$dim3\ \ block(Bx,\ By);$$
$$dim3\ \ grid(N/Bx,\ N/By);$$

Each thread in a block has unique index, *threadIdx*, and index of each block in a grid is *blockIdx*. Thus, thread index in a grid is done by:

$$Index\_x = blockIdx.x\ *\ blockDim.x\ +\ threadIdx.x$$
$$or$$
$$Index\_y = blockIdx.y *\ blockDim.y\ +\ threadIdx.y$$

where $blockDim.x\ \ or\ \ blockDim.y$ is used to provide number of blocks in a grid.

The GPU and the CPU are working together to achieve higher performance [Kirk and Hwu, 2010]. However, the relation between them is not direct because they cannot access each other's memories directly, as in Figure 6.4. To call the GPU, that data should be copied from the CPU memory to the GPU memory by using the *cudaMemcpy* function under *cudaMemcpyHostToDevice* mode. The same thing is happening when results are returned back from the GPU to the CPU: the data are copied from the GPU to the CPU by using the *cudaMemcpy* function under *cudaMemcpyDeviceToHost* mode. The number of the GPU threads should be greater or equal to the size of the moved data sets to it. The GPU parallelism benefits more by a large scale data sets, because the CPU computation speed is enough to execute simple computations.

The main strong point of the GPU is its highly parallel computations, which solve

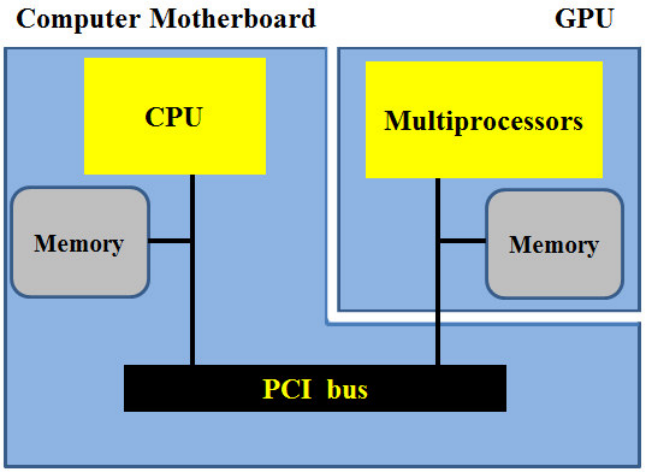


Figure 6.4: *CPU and GPU are cooperative together in executing CUDA program. Data should be copied from CPU memory to GPU memory for parallel execution because the GPU cannot access directly the CPU memory, and vice versa.*

the common problem encountered by many different applications. The GPU is used in different applications to achieve high performance. Some of the applications are [NVIDIA, 2013]:

- *Visualization*: The GPU is a significant advancement for the visualization field because an enormous amount of computations are often required. It can be used to speed up processing large-scale data sets to find their colour image.
- *Medical image*: Using the GPU in this field is important in order to speed up results to do extra processing or treatments according to the analysis [Scherl et al., 2007] [Stone et al., 2007].
- *Video enhancement*: The techniques used in video enhancement are always slow. Thus, the GPU has a more powerful computation engine that can be applied to these techniques rather than the CPU [Bachoo, 2010].
- *Bioinformation and life science*: Analysing protein and DNA sequences are more general subjects that use the GPU because of its better performance [Schatz et al., 2007] [Trapnell and Schatz, 2009].
- *Data mining and neural networks*: The GPU is used to find useful information among large size of databases and train the artificial neural network on a lot of

information in order to reduce computation time [Ma and Agrawal, 2009].

- *Softwares development*: The GPU is very advantageous in developing software. For example, using the GPU with Matlab and Labview increases the computation speed [Buatois et al., 2009].

In this chapter, we will visualize a remote sensing imagery data sets by the DR method on the GPU to speed up the projection process. In addition, the measurement of the efficiency by correlation and residual variance metrics of the visualization are also done on the GPU. The comparison with the CPU implementation will be done in order to see the efficiency of the parallel codes.

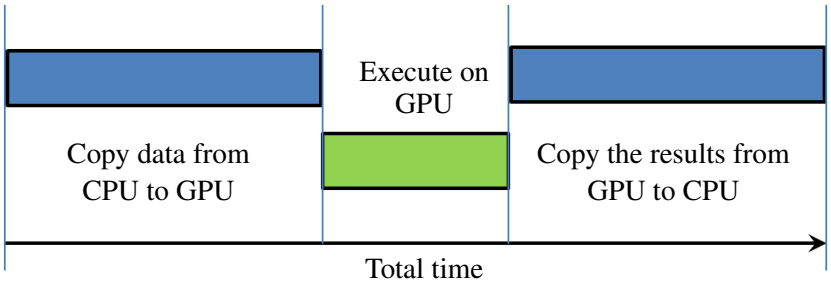


Figure 6.5: *Evaluation of one point requires to move N points from CPU to GPU. These points are processing in parallel to evaluate their relation with point in the hand.*

6.2 Our Contributions in GPU Application

Evaluating the efficiency of each point with the remaining points in the data sets in a general way has been used by most methods, where the final efficiency of the data sets represents the average of efficiencies from all those points. According to this premise, the principle of divide and conquer is the best way to speed up processing. The evaluation of each point is computed by computing the value of its neighbourhood relationship with all other points in parallel. If we suppose that there are N points in the data sets, the processing requires the sending of that point with all other points (N) to the GPU, as in Figure 6.5. The N points in the GPU are processed in parallel to measure the efficiency of that point. Parallel implementation helps a lot in reducing the execution time, especially if the size of the data sets is very large. We will use this idea to implement *FSPE*, correlation and residual variance metrics on the GPU.

6.2.1 Parallel *FSPE*

The *FSPE* can be implemented directly on the GPU because the method equation, in Section 3.3.2, is a parallel equation. Figure 6.6 showed that the kernel *FSPE_GPU* updates the coordinates of all points in the data sets in parallel depending on the selected point i. In the host, a point is selected each time and sent to the GPU through kernel *FSPE_GPU* to do a parallel task of updating the other points according to the *FSPE* equation. The result of applying *FSPE*, which is projected space and has three dimensions, is projected into *CIE Lab* colour space, as we explained in section 3.5.2.

```
__global__ void FSPE_GPU(float i, float *Y, float *X, float lambda, float d_c, int
Dimension, int LDimension, int N){
  int j = blockIdx.x * blockDim.x + threadIdx.x;
  if (j < N)
   { float sum = 0.0;
     for (int k=0; k<LDimension; k++)
        sum += (float) ((Y[i+ N*k] - Y[j+ N*k]) * (Y[i+N*k] - Y[j+N*k]));
     float dij = sqrt(sum);
     sum = 0.0;
     for (int k=0; k<Dimension; k++)
        sum += (float) ((X[i+ N*k] - X[j+ N*k]) * (X[i+N*k] - X[j+N*k]));
     float rij = sqrt(sum);
     if ((dij<=d_c) || ((dij > d_c) && (dij<rij)))
       {  float T = (float) (lambda * (rij - dij) / (dij + 1e-8));
          for (int k=0; k<LDimension; k++)
            Y[j+N*k] += (float) (T * (Y[j+N*k] - Y[i+N*k]));
       }
   }
   __syncthreads();
}
```

Figure 6.6: *FSPE is implemented on the GPU by using CUDA C++ codes. This kernel attempts to update the coordinates of all points in the data sets according to their relationship of the point i. Dimension is the dimension of the original space X, LDimension is the dimension of the projected space Y. N is the size of the data sets.*

```
__global__ void step_one(float i, float *B, float *A, int Dimension, int LDimension,
float *S_a, float *S_b, float *S_ab, int N){
  int j = blockIdx.x * blockDim.x + threadIdx.x;
  if (j < N)
   { float sum = 0.0;
     for (int k=0; k<LDimension; k++)
        sum += (float) ((B[i+ N*k] - B[j+ N*k]) * (B[i+N*k] - B[j+N*k]));
     float dij = sqrt(sum);
     sum = 0.0;
     for (int k=0; k<Dimension; k++)
        sum += (float) ((A[i+ N*k] - A[j+ N*k]) * (A[i+N*k] - A[j+N*k]));
     float rij = sqrt(sum);

     S_a[j]  += rij / N;
     S_b[j]  += dij / N;
     S_ab[j] += (dij * rij) / N;
    }
    __syncthreads();
}
```

Figure 6.7: *First step to compute correlation metric on the GPU (CUDA C++). This kernel attempts to compute the $\overline{A}$, $\overline{B}$ and $A^T\ B/|A|$ for the point i. The results are assigned to the variables S_a, S_b and S_ab, respectively. Dimension is the dimension of the vector A, LDimension is the dimension of the vector B. The projected and original spaces have the same size of data points, which are N.*

6.2.2 Parallel Correlation Metric

The visualization of remote sensing imagery data sets by *FSPE* should be measured to see how it preserves the original information. Although the implementation time of the correlation metric needs to be long in the CPU because of the large volume of remote sensing imagery data sets, parallel implementation of it can overcome this limitation. To implement Equation 2.26 on the GPU, one kernel is not enough because the standard deviations (σ_A and σ_B) depend on the mean values ($\overline{A}$ and $\overline{B}$, respectively).

To implement it, we will use two stages to construct CUDA C++ codes. In the first stage, we compute the values of the variables $\overline{A}$, $\overline{B}$ and $A^T\ B/|A|$ of each data point in the data sets. Figure 6.7 shows that the kernel (*step_one*) computes the vectors of these values in parallel on the GPU. The final values for these variables represent the average of the total sum of their values of all points, which have been computed on the CPU.

In the second stage, we compute the variables for each point in the data sets, σ_A and σ_B. The kernel (*step_two*), in Figure 6.8, is responsible for implementing this task on the GPU. In the CPU, the overall average is computed to get the sum of all their values and to know the final value of the data sets. Therefore, all variables of Equation 2.26 are ready for application directly to get the final correlation value of the visualization by *FSPE*.

```
__global__ void step_two(float i, float *B, float *A, int Dimension, int LDimension,
int N, float Mean_a, float Mean_b, float *Std_a, float *Std_b){
  int j = blockIdx.x * blockDim.x + threadIdx.x;
  if (j < N)
   { float sum = 0.0;
     for (int k=0; k<LDimension; k++)
        sum += (float) ((B[i+ N*k] - B[j+ N*k]) * (B[i+N*k] - B[j+N*k]));
     float dij = sqrt(sum);
     sum = 0.0;
     for (int k=0; k<Dimension; k++)
        sum += (float) ((A[i+ N*k] - A[j+ N*k]) * (A[i+N*k] - A[j+N*k]));
     float rij = sqrt(sum);

     Std_a[j] += (rij - Mean_a) * (rij - Mean_a) / N;
     Std_b[j] += (dij - Mean_b) * (dij - Mean_b) / N;
    }
    __syncthreads();
}
```

Figure 6.8: *Second step to compute correlation metric on the GPU (CUDA C++). This kernel attempt to compute the σ_A and σ_B for the point i. The results are assigned to the variables Std_a and Std_b, respectively. Dimension is the dimension of the vector A, LDimension is the dimension of the vector B. The projected and original spaces have the same size of datapoints, which are N. Mean_a and Mean_b represent the $\overline{A}$ and $\overline{B}$, respectively, of all data sets.*

6.2.3 Parallel Residual Variance Metric

Residual variance is also important to measure the stress error of the visualization by *FSPE*, but it needs to be implemented on the CPU for a long time. Equation 2.25 is much simpler than Equation 2.26, where implementation of the residual variance on the GPU requires just one kernel, as in Figure 6.9. This kernel attempts to compute

```
__global__ void Residual_variance(float i, float *B, float *A, int Dimension, int
LDimension, int N, float *Stress_i){
  int j = blockIdx.x * blockDim.x + threadIdx.x;
  if (j < N)
   { float sum = 0.0;
     for (int k=0; k<LDimension; k++)
        sum += (float) ((B[i+ N*k] - B[j+ N*k]) * (B[i+N*k] - B[j+N*k]));
     float dij = sqrt(sum);
     sum = 0.0;
     for (int k=0; k<Dimension; k++)
        sum += (float) ((A[i+ N*k] - A[j+ N*k]) * (A[i+N*k] - A[j+N*k]));
     float rij = sqrt(sum);

     Stress_i[j] += (rij - dij) * (rij - dij) / (N-2);
    }
    __syncthreads();
}
```

Figure 6.9: *CUDA C++ codes of residual variance. This kernel computes the stress error (*Stress_i*) for the point "i".*

the stress error (*Stress_i*) for a point *i*. *Dimension* is the dimension of the vector *A*, *LDimension* is the dimension of the vector *B* and *N* is the size of data sets.

6.3 Experimental Results

In this section, the implementation on Intel(R) i7-930 2.80 GHz CPU with 12 GB memory on Windows 7. We ran our proposed method in Microsoft Visual Studio C++ 2008 with CUDA 4.2 and NVIDIA GeForce GTX 480 graphics card with a buffer size of 1 GB. We used the AVIRIS Moffet Field data sets from the southern end of San Francisco Bay, California, done in 1997 [AVIRIS, 2013]. The data sets contain 224 bands, where each band has 1800x600 points.

In Chapter 3, we showed the ability of *FSPE* in visualizing remote sensing imagery data sets. Although the results of parallel implementation on GPU have exactly the same results as sequential results on the CPU, execution time by the GPU achieves a higher speed-up than the CPU. Table 6.1 and Figure 6.10 show the comparison between CUDA C++ (GPU) and the CPU codes in time execution for different data sizes.

Table 6.1: *FSPE speed performance comparisons between the CPU codes and the CUDA C++ codes with different size of remote sensing imagery data sets. (Time unit is minutes.)*

Data Size	CPU	GPU	Times GPU faster than CPU
1x300x300	26.80	0.021	1276
2x300x300	104.044	0.040	2601
3x300x300	238.06	0.060	3967
4x300x300	468.751	0.076	6167
5x300x300	716.511	0.096	7463
6x300x300	1205.302	0.113	10666
7x300x300	1961.032	1.300	1508
8x300x300	2090.713	3.389	616
9x300x300	2497.094	6.054	412
10x300x300	3755.885	9.037	415
11x300x300	5633.828	12.864	437
12x300x300	8450.741	15.034	562

The computation time of *FSPE* is further reduced because the CPU and the GPU are working together, where the CPU uses one (or very few) core(s) to execute host function, and the GPU is called to execute kernel function in parallel.

Table 6.2: *Performance comparisons of the CPU codes and CUDA C++ codes in small data sets (300x300x224). Correlation or Stress computation has about 10 times speed-up whereas FSPE has over 1000 times speed-up because FSPE is not required to run on all points, where random selected points are enough to update the other points. (Time unit is minutes.)*

	GPU	CPU	Times GPU faster than CPU
FSPE	0.021	26.80	1276
Correlation	7.698	111.126	14
Stress	3.771	47.374	13

In order to measure the efficiency of the visualization, we have to use correlation and residual variance measurements metrics. These metrics are implemented on the GPU by using CUDA C++. Table 6.2 shows the comparison between CUDA C++ and the CPU codes in time execution for data sets have N=300x300 points. The CPU and the GPU are working together, where the CPU uses very few cores to execute the host function, and the GPU is called to execute the kernel function in parallel. It should be noted that the moving of information from host to device, and from device to host, takes a lot of time, as in Figure 6.11. Therefore, we move long the vector of high-

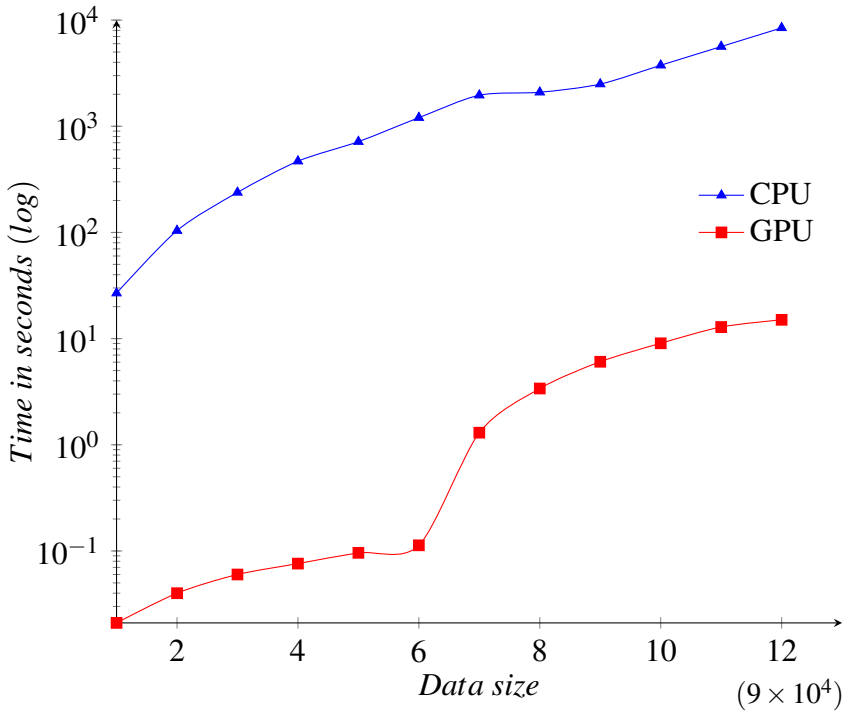


Figure 6.10: *The computation time of the FSPE algorithm for different data sets size. The GPU implementation is able to speed up the execution of FSPE compared to its implementation on CPU.*

dimensional data sets and the vector of low-dimensional space just one time from host to device through the projection before starting the *FSPE* in order to avoid losing more time.

When the GPU receives the computation control, N threads work parallel. Thus, the GPU parallelism is more beneficial with a large remote sensing imagery data sets because the CPU computation speed is enough to execute simple computation.

6.4 Conclusion

In this chapter, DR was implemented on the GPU to visualize remote sensing imagery data sets. The benefit of a parallel implementation is to obtain the results in as short a time as possible. The results showed that CUDA implementation of *FSPE* is faster than their sequential codes on the CPU in calculating floating-point operations, especially for a large data sets, such as remote sensing imagery. The GPU is more suitable

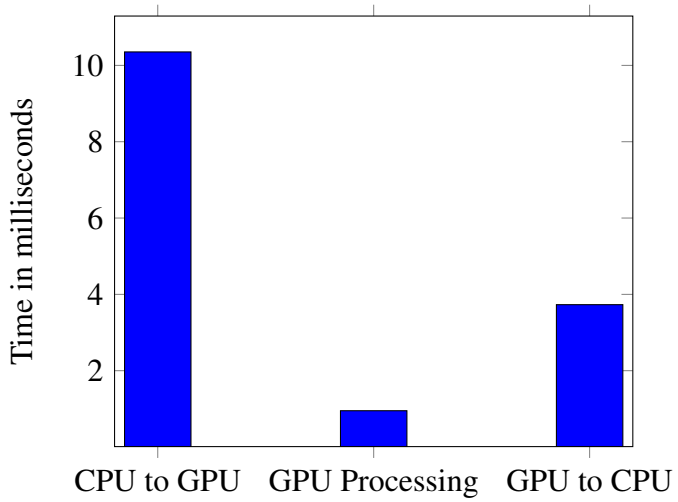


Figure 6.11: *Time of moving data from the CPU to the GPU is greater than pure execution in the GPU. Moving the results back to the CPU is also greater than pure execution in the GPU. In this experiment, the time of moving data from the CPU to the GPU is greater than moving them from the GPU to the CPU because of the large size of input data.*

to the implementation of the correlation and residual variance measurement methods because they do a large computation. We illustrated that this massive speed-up requires a parallel structure to be suitable for running on the GPU. Large data sets, such as remote sensing imagery, is a better candidate data sets to be implemented on the GPU to accelerate the computation time, which will be one hundred times speed-up over the CPU.

Chapter 7

Enhancing The Visualization by Using Faithful Stochastic Proximity Embedding

In this chapter we will address two ideas, which are:

1. Introduction
2. Enhancement Methodology.
3. Experimental Results.
4. Conclusion.

7.1 Introduction

In this chapter, we will prove the validity of our hypothesis "If we suppose that such information does exist in low-dimensional space, the missing neighbourhood points can be retrieved". In the previous chapters, we showed the ability of our proposed method *FSPE* in overcoming the false neighbourhood when it is implemented alone. Therefore, *FSPE* can be used as a supplementary stage for enhancing the final results of other methods.

In this chapter, a new method will be proposed to investigate a good visualization of remote sensing imagery data sets on the GPUs. The main shortcoming of classical DR

methods is that their visualizations inevitably lose a significant amount of information in remote sensing imagery. Our proposed method preserves the colour distance as noticeably as possible for high-dimensional data sets by defining extra steps to enhance the quality of visualization. It classifies the visualization colours into false and true, and focuses on the worst false colours to improve them. Although the method can be implemented as unsupervised, it is supervised when a user selects the interested points to enhance. The proposed method is a general enhancement step, which can be used to enhance any visualization; for example, visualization of remote sensing imagery data sets, which are carried by CDA, SPE and Isomap methods. The results showed the ability of our method to overcome the false colours and return the more reliable true colours. The colours of visualization showed more details, which represent the real neighbourhood relation in the original data sets.

7.2 Enhancement Method

The main idea of our method is to select the worst false neiughbourhood points in the low-dimensional space and enhance them by applying a good DR method, such as *FSPE*, to retrieve the true neighbourhood points. There are two possibilities to select these points:

1. The enhancement starts by selecting the worst area in the visualization, $I = \{i_1, i_2, ..., i_p\}$, where $p < N$ and N is a size of whole visualization pixels. *FSPE* on the GPU is used to enhance the efficiency of the selecting points by re-projecting them. While this process enhances the selected points and might update the other points, the whole visualization is enhanced. The final visualization shows the selected area is enhanced and most of the false colours are overcome. Selecting the area this way might have different colours, and *FSPE* makes those colours more faithful.

2. Another way to apply enhancement is to select the worst area in the colour space of the visualization. Any two-dimensional colour space can be chosen to apply this way, and we selected CIE Yxy because it is well known and easy to understand [Kerr and E., 2005] [Schanda, 2007]. The selection depends on the colour relationship between points instead of their locations in the visualization. The steps are:

(a) The input visualization is mapped into CIE Yxy colour space, where the locations of the visualization colours are shown in an understandable way which makes selecting from it very easy. To do this, the visualization is mapped into CIE XYZ and then the chromaticity coordinates x and y of CIE Yxy are computed by using the following equations:

$$x = \frac{X}{X+Y+Z} \tag{7.1}$$

$$y = \frac{Y}{X+Y+Z} \tag{7.2}$$

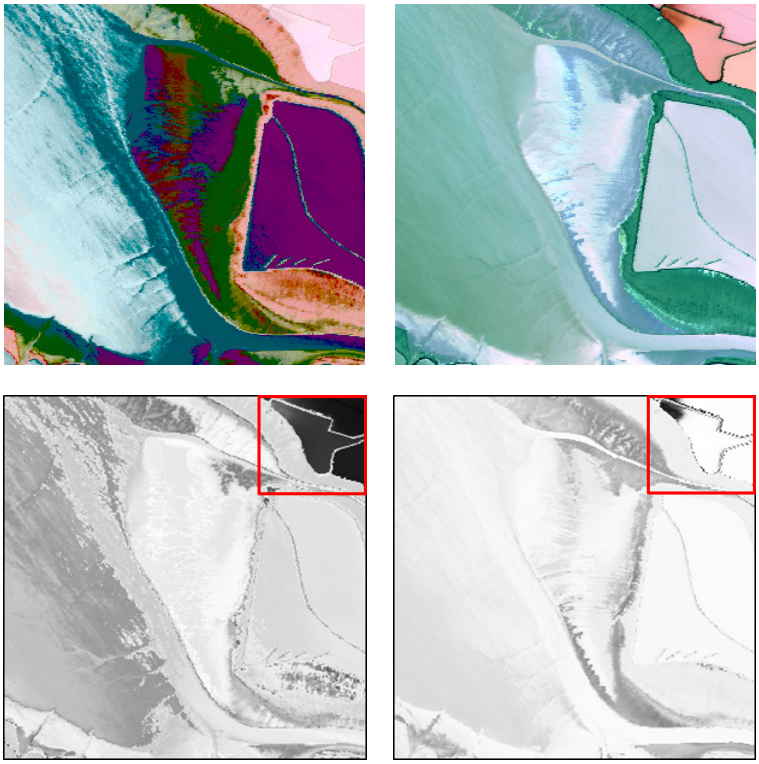
(b) As long as the location of each colour in the CIE Yxy colour space is determined and the efficiency of each point can be computed by using point-wise correlation metric, the points' efficiencies are displayed in CIE Yxy colour space to clarify the true and false colours. The points' efficiencies in CIE Yxy colour space are represented by values in the range [0,1], where 0's value indicates that this point has worst efficiency in preserving its neighbourhood, and 1's value indicates the colour at this point is true.

(c) In CIE Yxy colour space, the worst efficiency area is selected to enhance its points' colours.

(d) *FSPE* on the GPU is applied on the selected points in the previous step. *FSPE* attempts to place falsely colour point y_j in such a way that its Euclidean distance $d_{ij} = ||y_i - y_j||$ is close to corresponding distance $r_{ij} = ||x_i - x_j||$ in the original high-dimensional data sets. Finally, the colours of the visualization are enhanced.

7.3 Experimental Results and Discussion

In Figure 7.1, the first way of enhancement method was applied by selecting the upper right area of the visualization, which was carried by CDA. The corresponding grey-scale efficiency image of this visualization, which is computed by a point-wise correlation metric, shows the selected area has very low efficiency. After the application of *FSPE* to this area, we note the efficiency of this area has improved and lot of information has been explored. The colours of this area have become more faithful to

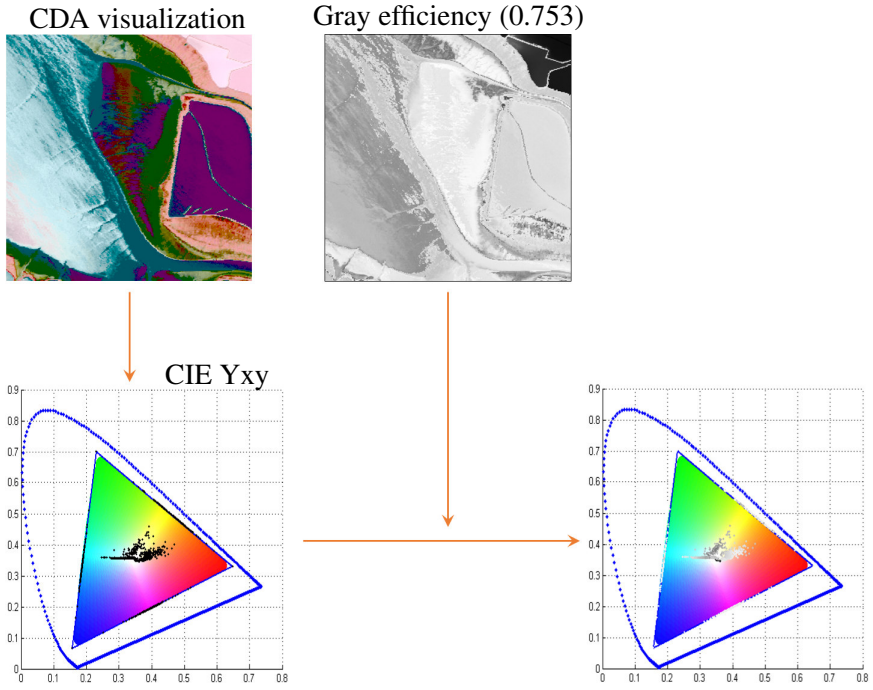


(a) Before, 0.036 (b) After, 0.890

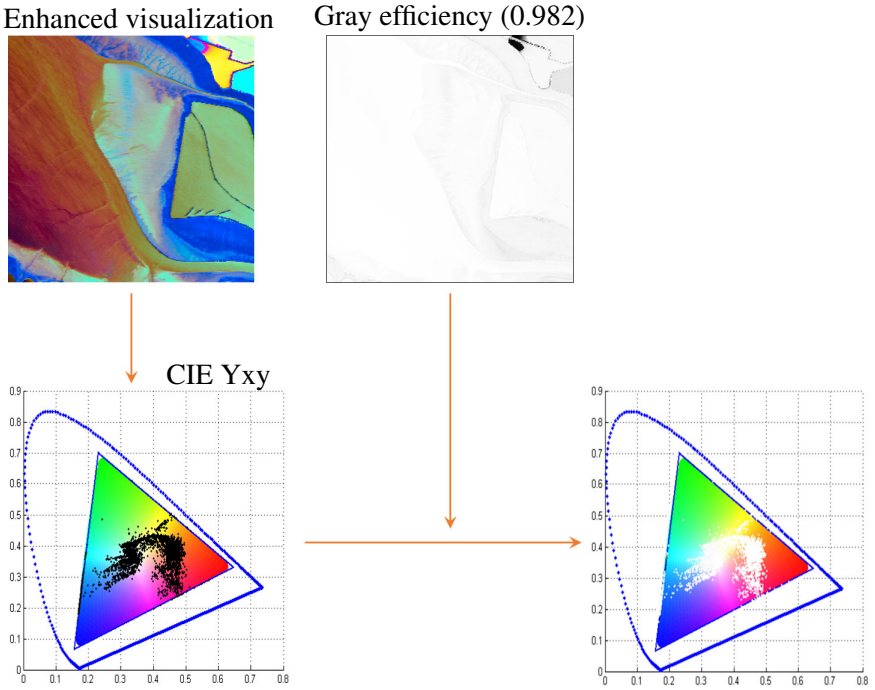
Figure 7.1: *a) The gray-scale image shows the upper right area of the visualization of remote sensing imagery, which was carried by CDA, has very little point-wise correlation* ($\gamma = 0.036$). *b) Application local enhancement by FSPE helped to increase the correlation of this area into* ($\gamma = 0.890$). *The local enhancement process has helped in the discovery of more effective colours. Moreover, the efficiency of the whole visualization is increased from 0.753 to 0.911.*

their corresponding ones in original bands of the remote sensing imagery data sets. In addition, overcoming the problems of false colours also helped to improve the rest of the colours in the visualization. The efficiency of the whole visualization is increased from 0.753 to 0.911.

In Figure 7.2(a), the second way of the enhancement method was applied on the CDA's visualization. The grey efficiency image showed there were many false colours. The colours of visualization are mapped into CIE Yxy colour space. While the location of each point was specified, the efficiency of each point is displayed in the CIE Yxy colour space. The low efficiency points can be recognized in the colour space, and the worst area can be selected. *FSPE* was applied to enhance the faithfulness of



(a) Before applying FSPE



(b) After applying FSPE

Figure 7.2: *Visual comparison between traditional and enhanced visualizations. a) The gray efficiency image shows there are many points in the CDA visualization that have low efficiency in preserving their neighbourhood. CIE Yxy colour space shows the locations of the truth and false colours in the visualization. b) Most of the false colours are overcome in the enhanced visualization by using FSPE, and CIE Yxy colour space shows that, too. The efficiency of the visualization is increased from 0.753 to 0.982.*

the selected area, where the efficiency is increased from 0.753 to 0.982, as in Figure 7.2(b). This enhancement means the false colours are overcome by getting their true colours. The differences in the visualization before and after the enhancement are clear, where we note a lot of information has been shown, which is hidden in the original visualization.

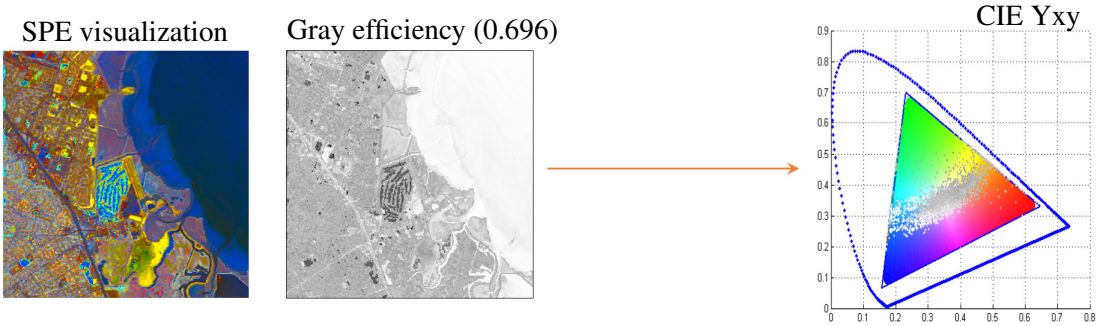
Both ways of the enhancement method are good in reducing the effect of the false colours, but the second way is the best. Therefore, we will use the second way in the rest of the experiments in enhancing the visualizations by SPE and Isomap.

Table 7.1: *The enhanced visualization got the highest correlation and least stress values. (Correlation: the highest is better, Stress: the lowest is better).*

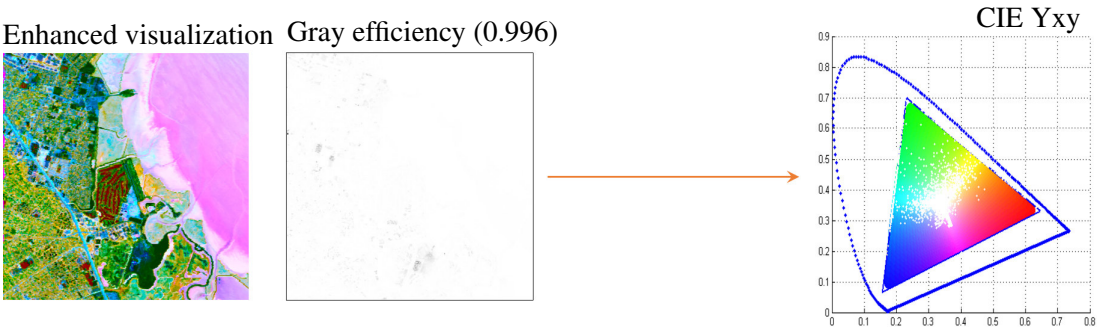
	Correlation		Stress	
	Before Enhan.	After Enhan.	Before Enhan.	After Enhan.
CDA	0.7530	0.9820	0.0662	0.0210
SPE	0.6958	0.9960	0.1583	0.0026
Isomap	0.5251	0.9955	0.0723	0.0016

Figure 7.3(a) showed that the visualization by SPE has false colours, where its efficiency is 0.696. The lowest efficiency colours are shown clearly when the visualization and its grey efficiency are mapped into CIE Yxy colour space. By applying *FSPE* on the worst efficiency colours point, most of the false colours are overcome, and the efficiency of the visualization is increased to 0.996. Figure 7.3(b) showed the visualization has faithful colours, where its grey efficiency image and CIE Yxy colour space displayed high efficiency values of the most points.

Isomap can visualize the remote sensing imagery data sets, but many false colours show in the visualization, as in Figure 7.4(a). The grey efficiency image and the CIE Yxy colour space of the visualization clearly illustrate that. When the worst area in CIE Yxy colour space is selected and *FSPE* is applied to them, the efficiency of the enhanced visualization is increased from 0.525 to 0.995. In enhanced visualization, as in Figure 7.4(b), most of the false colours are overcome, and more accurate colours appeared to represent the retrieved information.



(a) Before applying FSPE



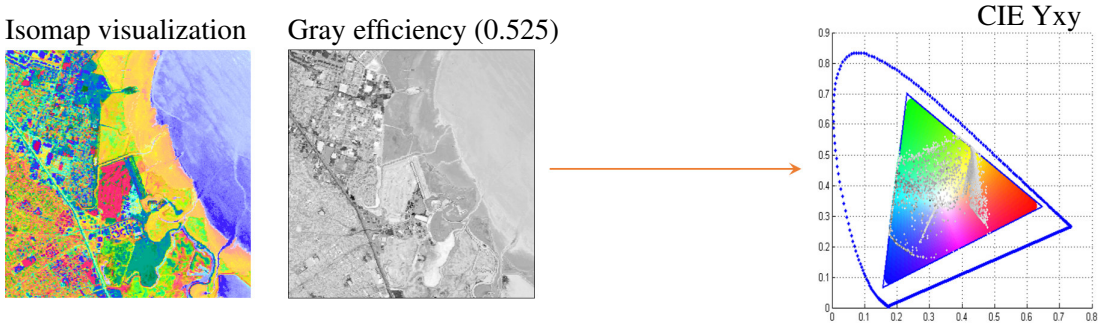
(b) After applying FSPE

Figure 7.3: . *a) The grey efficiency image shows that many colours have low efficiency in the SPE's visualization. CIE Yxy colour space of this visualization showed which colours are false and which were true. b) Applying FSPE on the points that have the lowest colour efficiency causes an increase in the efficiency of all the visualization colours from 0.696 to 0.996. CIE Yxy colour space of enhanced visualization shows that most colours have high efficiency.*

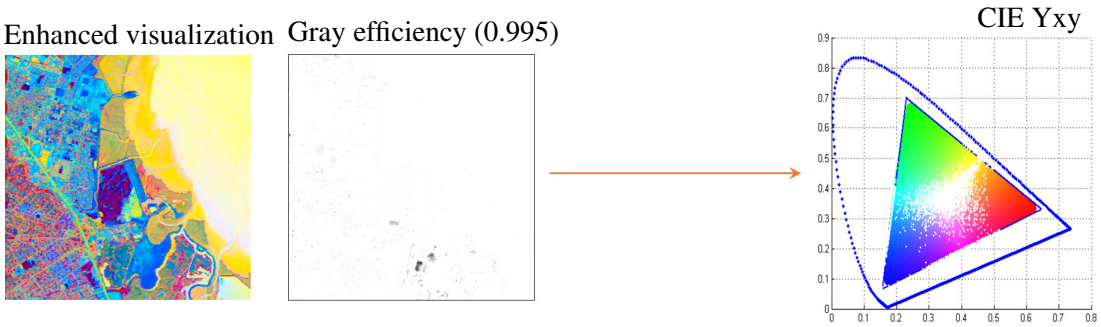
7.4 Conclusion

We can conclude by saying that this new enhancement method is proposed to enhance the visualization of remote sensing imagery. Our approach can work automatically to overcome the false colours in visualization and reveal the reliable colours, but we prefer to take a user opinion in this matter because he can choose the most interested region according to his analysis. The results showed, by using our method, the trustworthiness of the visualization is increased by overcoming most of the false neighbourhood points that are generated during the projection process.

In this chapter, we noticed that the *FSPE* has an important role in overcoming the false neighbourhood points in the results of other methods. The enormous dimensionality size and complexity of the remote sensing imagery data sets were a good application in demonstrating the ability of the proposed method in preserving original



(a) Before applying FSPE



(b) After applying FSPE

Figure 7.4: *a) The visualization by Isomap has several false colours. b) The enhancement visualization when FSPE is applied on the selected worst efficiency colours.*

information. Choosing the worst efficient points, regardless of the method of selection, to update the rest of the points has helped in improving the visualization information. Thus, the validity of the hypothesis, in the beginning of the section 7.1, has been proved.

The main ideas presented in this chapter have been published in: Najim S., Lim I. and Saeed M. (2013). Trustworthy Enhancing the Visualization of Remote Sensing Imagery Dataset on GPU. In Proceedings of 6th International Conference on Developments in eSystems Engineering (DESE 2013), Abu Dhabi, UAE, 16-18 December.

Chapter 8

Final General Conclusion

8.1 Conclusion

In this dissertation, some hypotheses have been introduced to support the idea of DR to get faithful results. They focused on the most important problem, which is false neighbourhood errors in a low-dimensional manifold, and attempted to overcome them as much as possible. Here, we will discuss our hypotheses, which was proposed in the first chapter, to assess the validity of the proposal:

Useful information exists in low-dimensional space despite transforming the data from a higher dimensional space. Thus, the DR problems can be overcome. In Chapter 3, we proposed the "Faithful Stochastic Proximity Embedding (*FSPE*)" method, which focuses on low-dimensional space in reducing the dimensionality of high-dimensional data sets. *FSPE* can recognize the difference between false and true neighbourhood points through the projection process. The false neighbourhoods are sent away and the true neighbourhoods are strengthened with their neighbourhood. *FSPE* makes its low-dimensional space in the case of continuous enhancement because the false neighbourhood points are under control. The results showed that *FSPE* is better than other dimension reduction methods when it was used to visualize three real data sets, namely remote sensing imagery and human faces. Its visualization is better than 19 DR methods, where the points of visualization are reliable and almost free from false neighbourhood. In addition, it is able to unfold complex data sets, such as curved cylinders. *FSPE* can be used in the classification task, such network

classification, but its results are not the best when compared with other DR methods. Therefore we can say that our proposed method (*FSPE*) has demonstrated the validity of this hypothesis.

If we suppose that such information does exist in low-dimensional manifold, the missing neighbourhood points can be retrieved. According to the previous hypothesis, *FSPE* has the ability to overcome the false neighbourhood in low-dimensional manifold, which is carried by any DR method. In Chapter 7, *FSPE* was used as a supplementary stage for enhancing the final results of other methods. The focus was on the false neighbourhoods, where their enhancement had a positive impact on the other points. The dependence on a low-dimensional manifold by using *FSPE* maintains the original information. Therefore the validity of this hypothesis has been proved.

If the projected space has been transferred to the appropriate competent colour space, the derived information will be accurate and richly represented. Although many papers have used *CIE Lab* colour space in the visualization, the quality of the colours needs to be more accurate in order to satisfy what a user wants. Our method chose the most varying one for the L^* component and the remaining two for the a^* and b^* components, followed by a simple linear stretching such that $L^* \in [0, 100]$ and both of a^* and b^* have a zero mean. In order to attain a high contrast result, the L^* component shifts its values to fill the entire brightness range, where a small percentage (1%) of the brightest and darkest pixels are ignored. The relations among channels are kept through scaling a^* and b^* by using the scaling parameter that was used to scale L^*. The results showed that this way generates higher quality visualization, where its colours show more details. Therefore, the validity of this hypothesis is satisfied.

The way of applying DR can be redefined to increase the efficiency of the projected space. We pointed out in Chapter 4, that the direct projection of the high-dimensional data sets to the low-dimensional space does not guarantee the preservation of the original neighbourhood relation. The difference between the topological structures of two spaces causes a loss of information. Our proposed method of *SDR* redefines the problem of DR as a sequence of multiple DR problems, each of which reduces the dimensionality by a small amount. The transformation from one dimension to the next lower one gives a great opportunity to preserve the neighbourhood relations between the

points because of the similarity of the topological structures between two consecutive spaces. The applications of *SDR* on the remote sensing imagery and human face data sets gave better results than 20 traditional dimension reduction methods, as in Chapter 5. We can conclude that the success of the *SDR* in preserving original information represents the proof of the validity of this hypothesis.

The users role is important in evaluating the efficiency of the visualization. If there is a mechanism to give him a chance to see the site of strengths and weaknesses, that can help in the analysis of those data sets. In Chapter 7, we allowed the user to participate in the enhancement method to satisfy this hypothesis. He can select the points of interest to be enhanced rather than allowing the method to work automatically

The role of the GPU was important to speed up the results, especially when we were dealing with large-scale data sets. The parallel processing idea on the GPU was used in the implementation of the proposed hypotheses and measurement methods.

Although we used the four measurement metrics (stress, local continuity, correlation and point-wise correlation) for evaluating the results, the point-wise correlation metric method (which is proposed in Chapter 3) was better. Evaluating each point in the visualization, the grey-scale efficiency image is formed, which shows the user the location of the false and true points. A grey-scale efficiency image gives more detail instead of representing the evaluation in one single value.

8.2 Future Work

1. Visualization by DR is important in studying the data sets, but the colours are not enough to get an integrated analysis, especially in a complex data sets, such as remote sensing imagery. Therefore, our future work will concentrate on getting a three-dimensional visualization. This method helps to distinguish between the mountains and the plains, water and land, and high and low buildings.

2. One of the weakness in DR is to think in one direction; namely, the transformation of high-dimensional data sets to low-dimensional space. This is an important thing, but it faces definite problems, one of which is the inability of some

of the data-points to rely on a new space. In order to validate the efficiency of the result, transferring a low-dimensional space to an original space should be possible and correct. In the future, we will try to achieve this goal.

3. Many DR methods initialize the low-dimensional space with random values. There is no relationship between the generated values and the data sets that is to be processed. Because of the importance of this step in obtaining good progress during the projection process, one of our future goals is to generate ideal initial values that depend on the quality of the data sets.

4. The general idea of the LLE method is to use the same basic principles of neural networks. This method is unable to deal with complex problems due to the weakness of its structure, which prefers linear neural networks. Our future goal is to develop this algorithm by using nonlinear structures of neural networks to be a powerful method.

5. Although the role of the GPU was important in speeding up the work, sometimes the speed did not meet the intention of getting the results in a shorter time. Thus, we will try to develop the GPU CUDA codes by using the principle of CUDA streams, which will speed up the process.

Appendix A

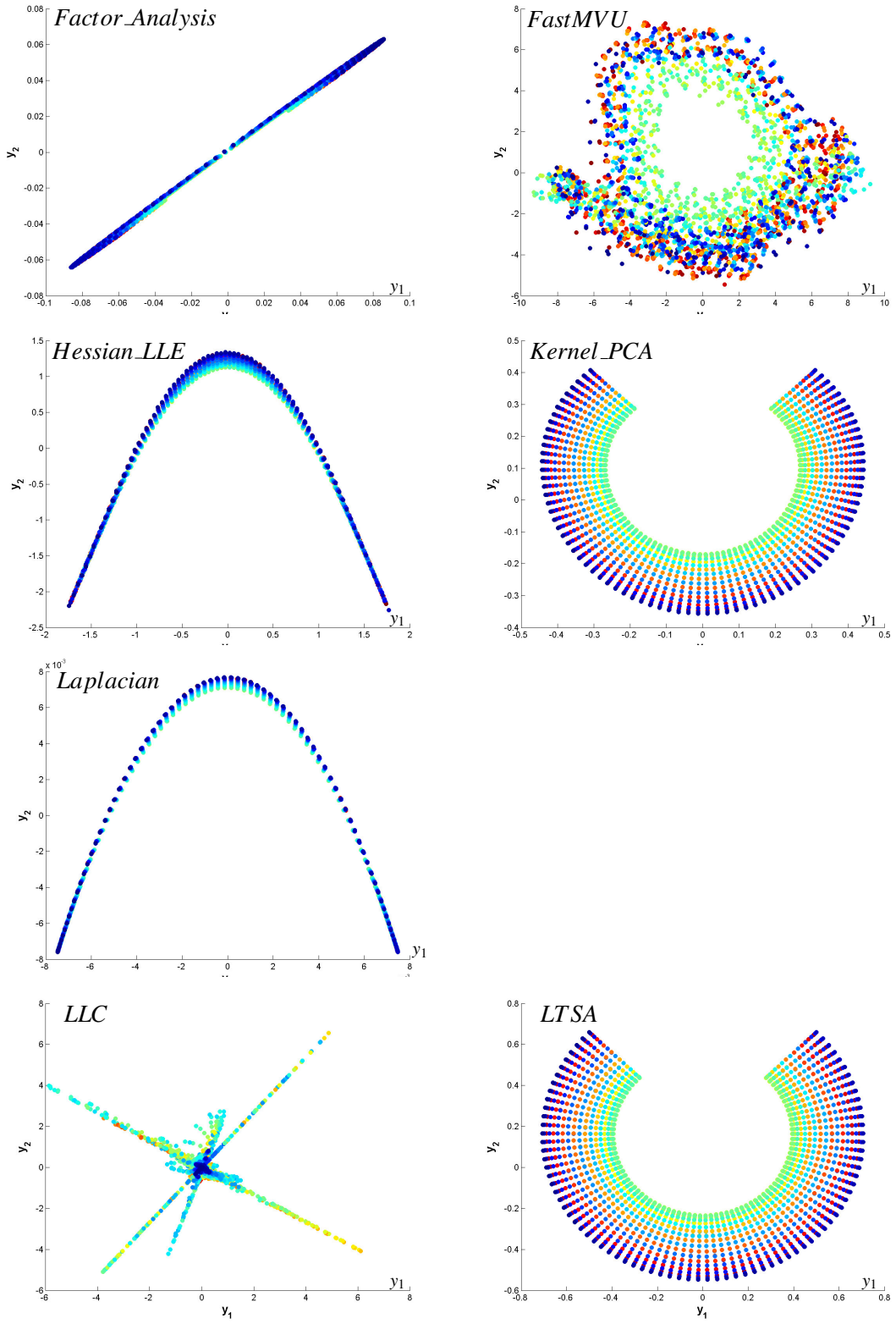


Figure 1: *Unfolded space of three dimension curved cylinder by many methods. The colour of each point in the unfolded space is the same colour of its corresponding point in original dataset, as in the Figure 5.1(a).*

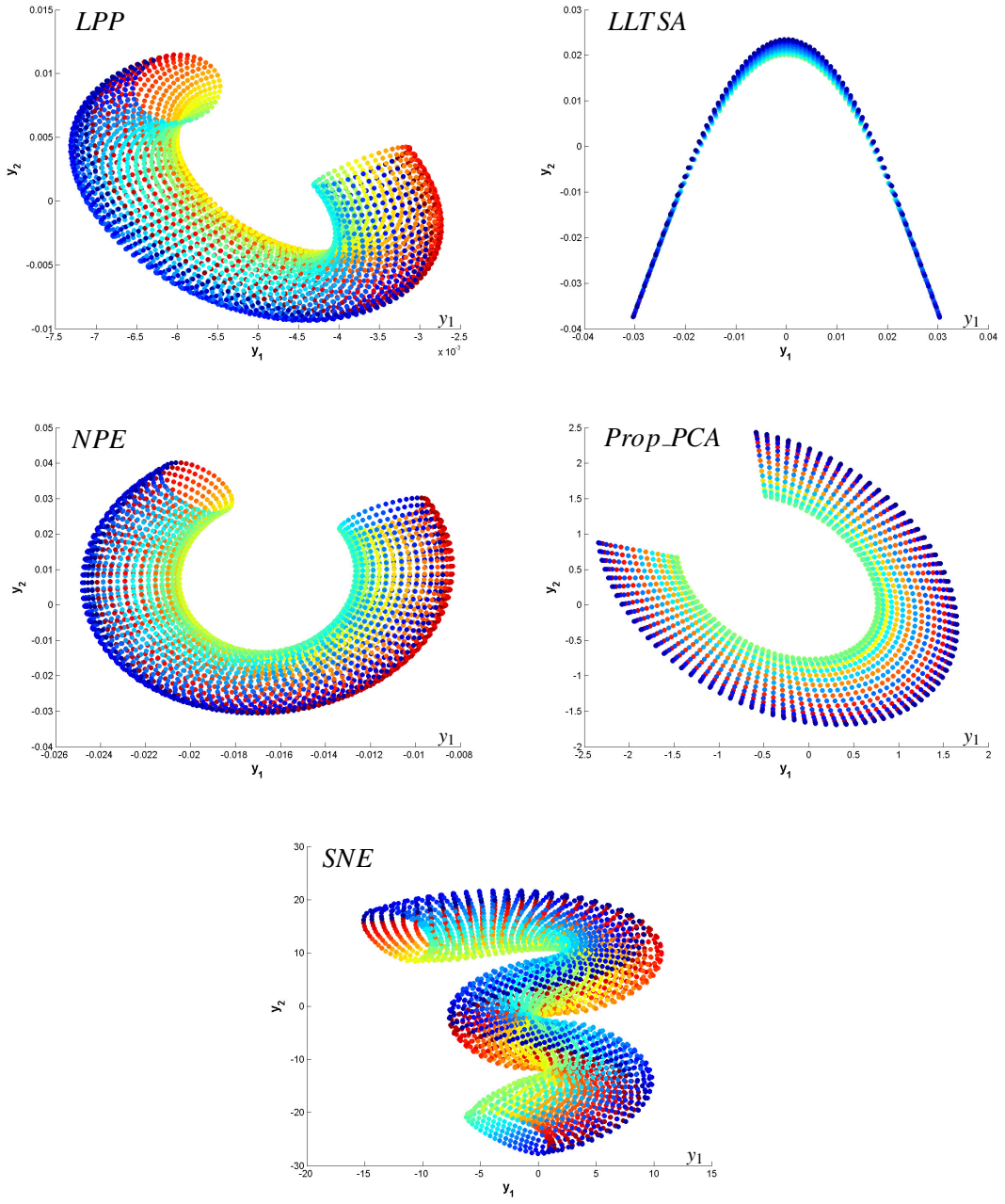


Figure 2: *Unfolded space of three dimension curved cylinder by many methods. The colour of each point in the unfolded space is the same colour of its corresponding point in original dataset, as in the Figure 5.1(a).*

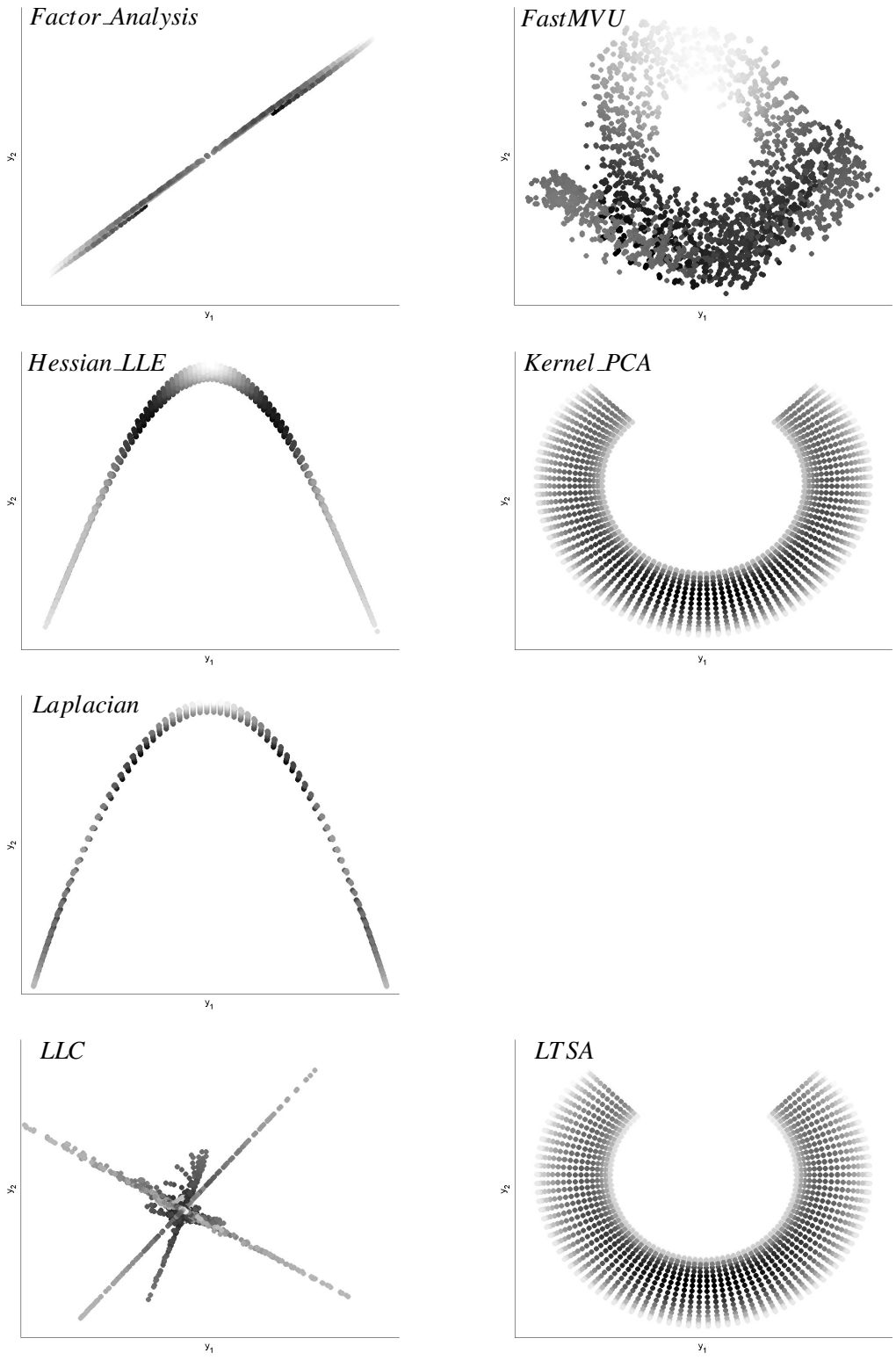


Figure 3: *The gray-scale efficiency images by point-wise correlation metric of the unfolded shapes of curved cylinder in the Figure 1. The bright white points indicate they have efficiency in preserving its neighbourhood relation, and the dark points refer they have low efficiency in preserving its neighbourhood relation.*

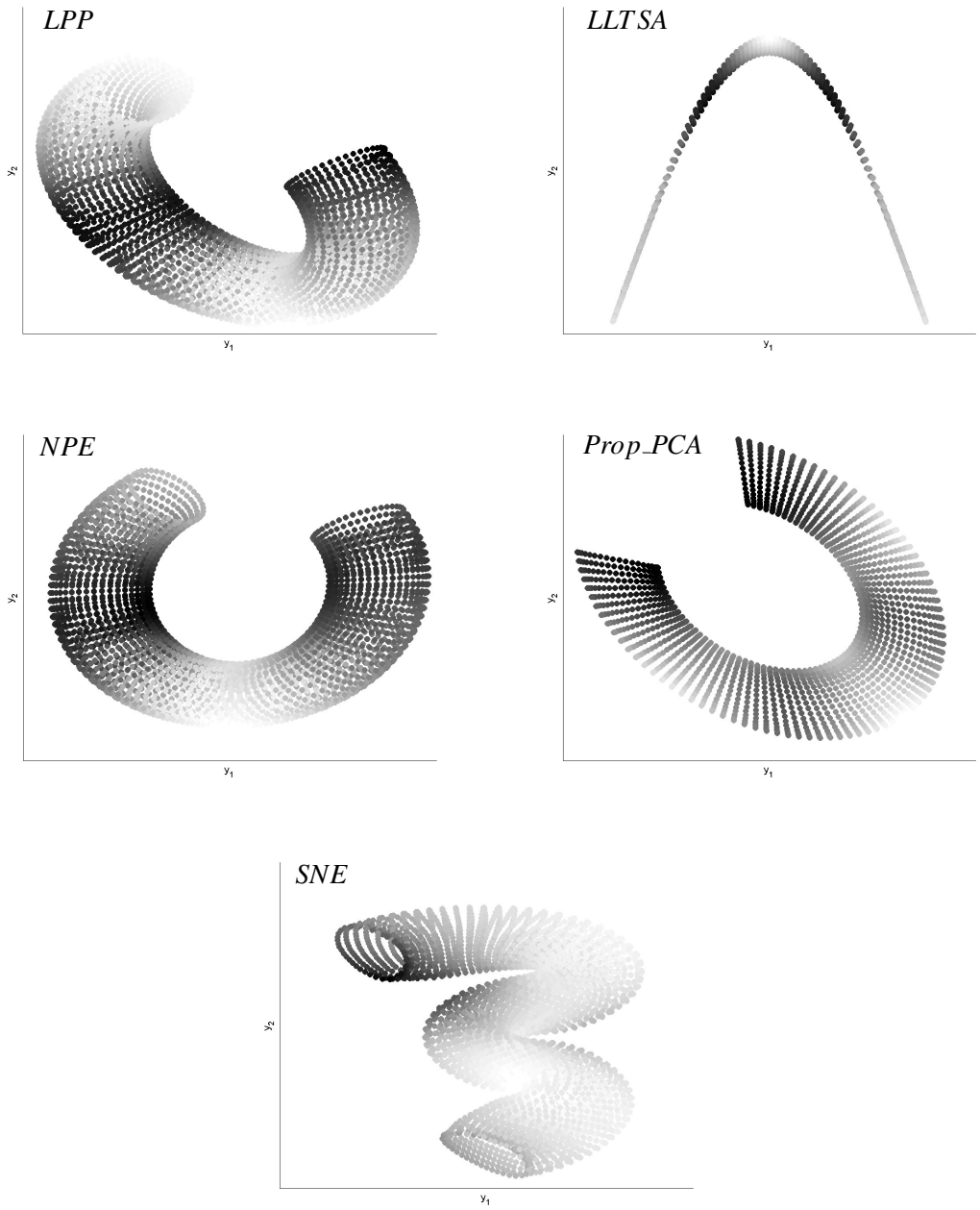


Figure 4: *The gray-scale efficiency images by point-wise correlation metric of the unfolded shapes of curved cylinder in the Figure 2. The bright white points indicate they have efficiency in preserving its neighbourhood relation, and the dark points refer they have low efficiency in preserving its neighbourhood relation.*

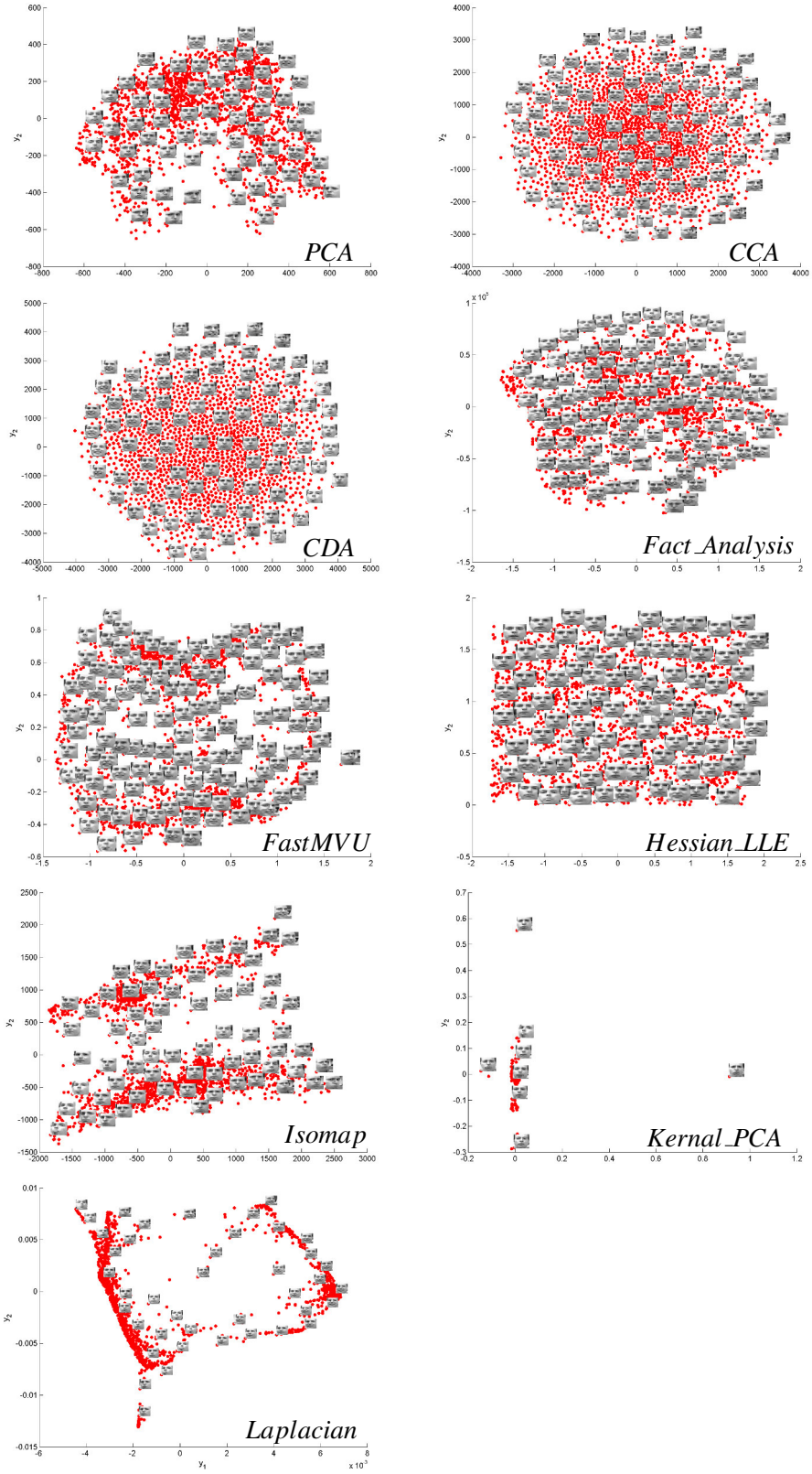


Figure 5: *Projecting the Frey face dataset into 2-dimensional space by many dimension reduction methods.*

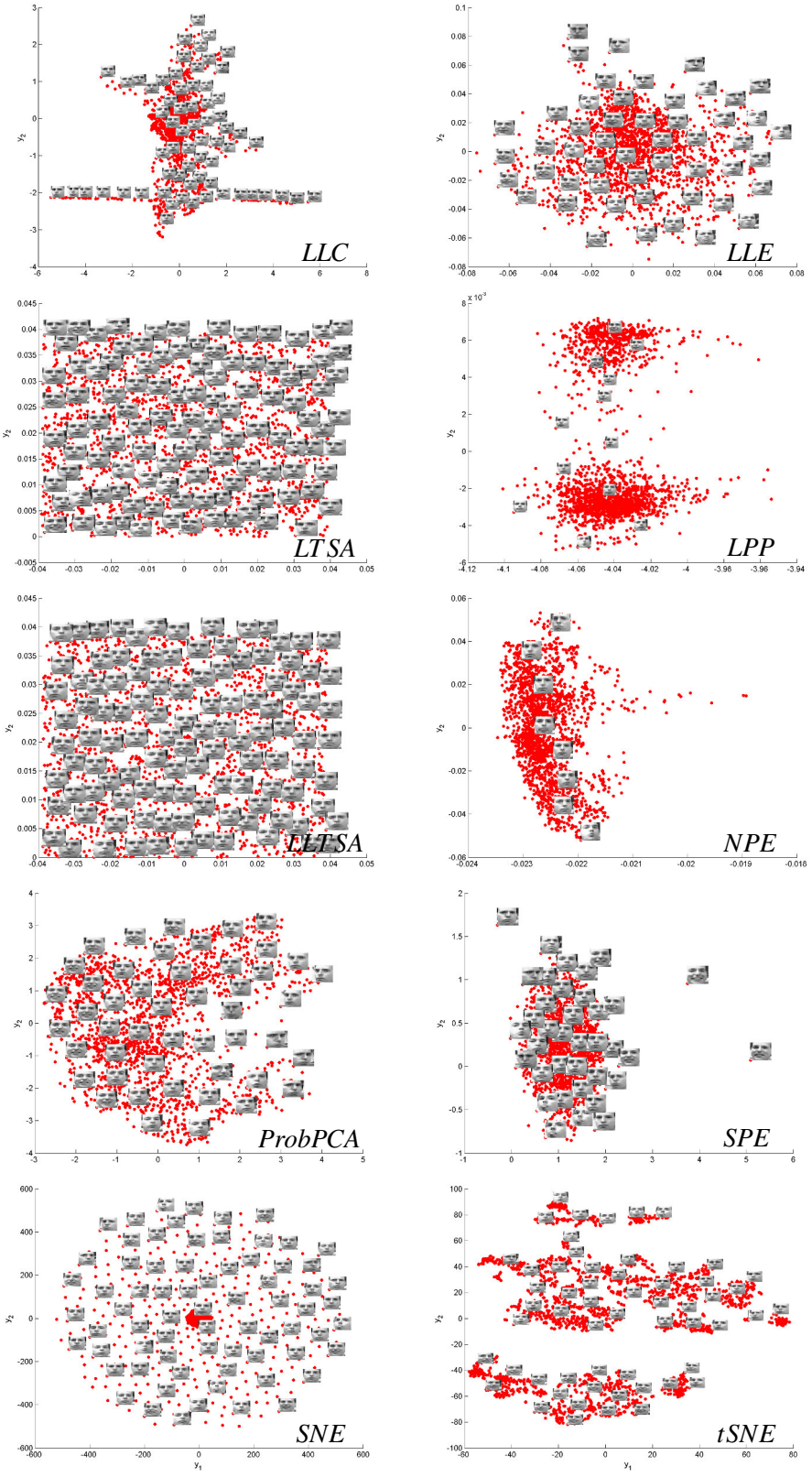


Figure 6: *Projecting the Frey face dataset into 2-dimensional space by many dimension reduction methods.*

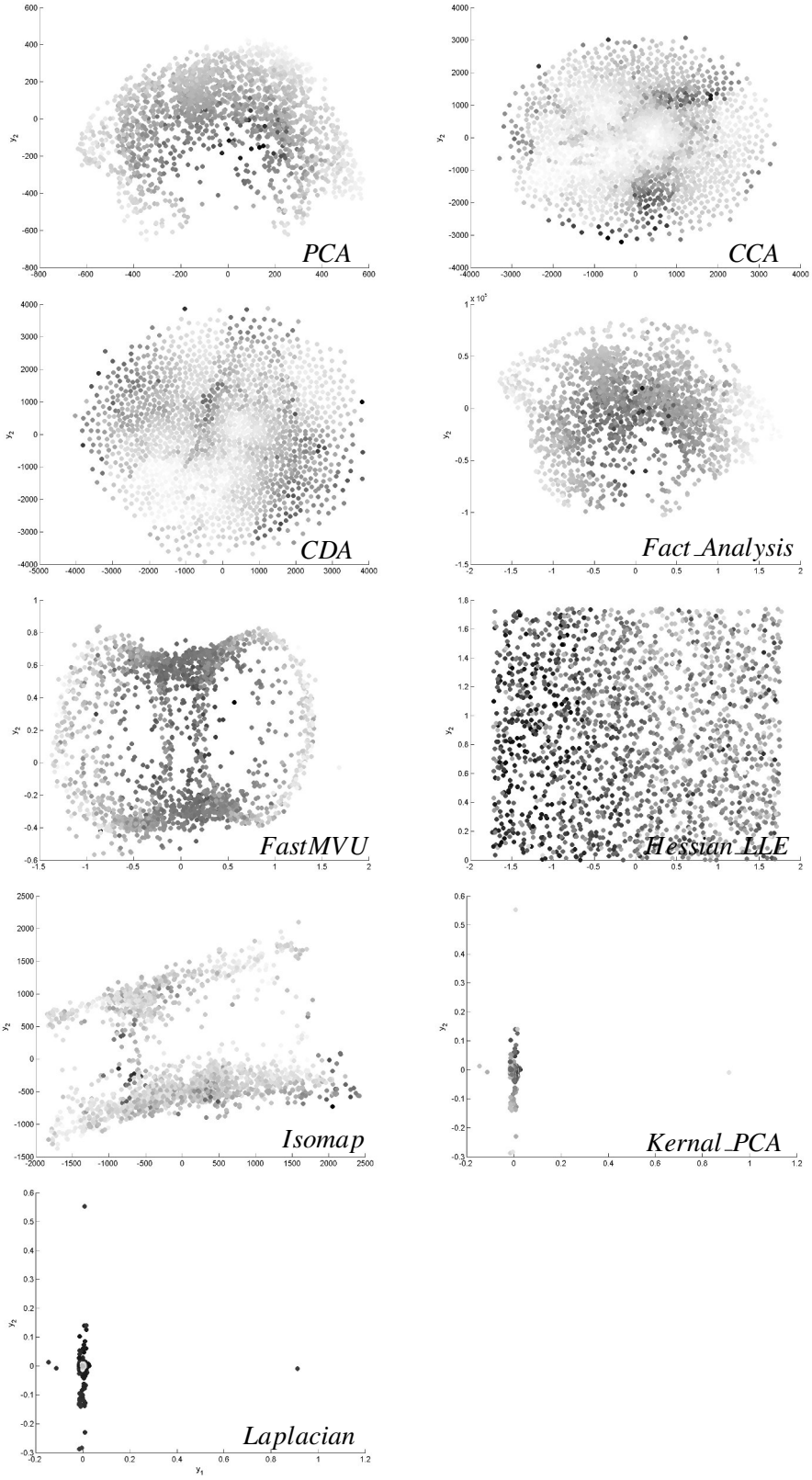


Figure 7: *Gray-scale effiency image by point-wise correlation metric of results in the Figure 5.*

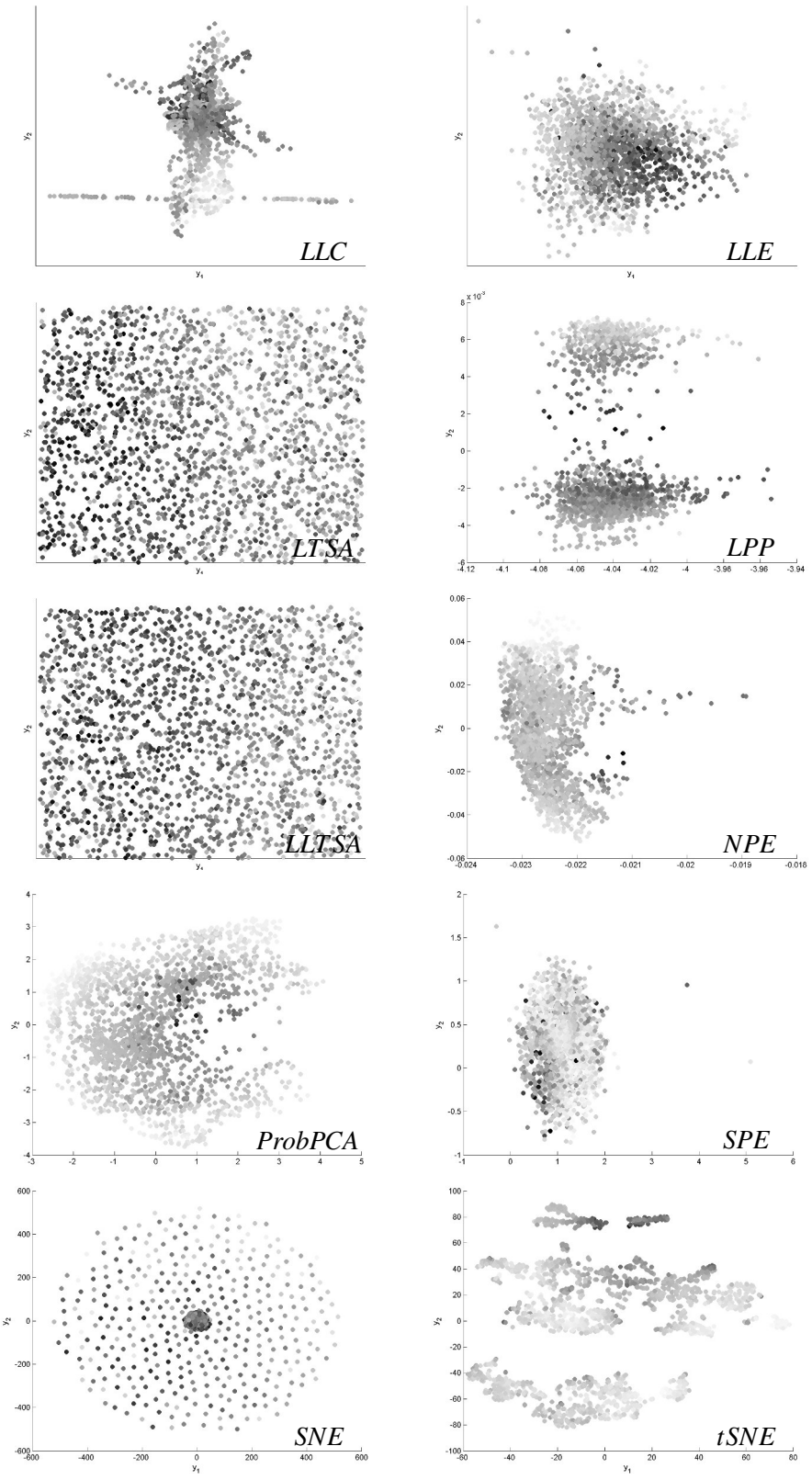


Figure 8: *Gray-scale effiency image by point-wise correlation metrics of results in the Figure 6.*

References

Agarwal, S., Wills, J., Cayton, L., Lanckriet, G., Kriegman, D., and Belongie, S. (2007). Generalized non-metric multidimensional scaling. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07). San Juan, Puerto Rico*. 8

Agrafiotis, D. K. (2003). Stochastic proximity embedding. *Computational Chemistry*, 24:1215–1221. 8, 16

Agrafiotis, D. K. and Xu, H. (2002). A self-organizing principle for learning nonlinear manifolds. *Proceedings of National Academy of Sciences of the United States of America*, 99:15869–15872. 16, 33

Agrafiotis, D. K., Xu, H., Zhu, F., Bandyopadhyay, D., and Liu, P. (2010). Stochastic proximity embedding: Methods and applications. *Molecular Informatics*, 29:758–770. 16, 60

Akkucuk, U. and Carroll, J. D. (2006). Paramap vs. isomap: A comparison of two nonlinear mapping algorithms. *Journal of Classification*, 23:221–254. 16

Akkucuk, U. and Carroll, J. D. (2010). *Nonlinear Mapping Using a Hybrid of PARAMAP and Isomap Approaches*, pages 371–380. Springer: Berlin-Heidelberg-New York. 16

Ameer, P. M. and Jacob, L. (2012). Localization using stochastic proximity embedding for underwater acoustic sensor networks. In *National Conference on Communications (NCC)*. 16

AVIRIS (2013). *http://aviris.jpl.nasa.gov/index.html*. 37, 38, 63, 105

Bachmann, C. M., Ainsworth, T. L., and Fusina, R. A. (2005). Exploiting manifold geometry in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 43:441–454. 25, 30, 33

Bachmann, C. M., Ainsworth, T. L., and Fusina, R. A. (2006). Improved manifold coordinate representations of large-scale hyperspectral scenes. *IEEE Transactions on Geoscience and Remote Sensing*, 44:2786–2803. 25, 30, 33

Bachoo, A. (2010). Using the cpu and gpu for real-time video enhancement on a mobile computer. In *IEEE 10th International Conference on Signal Processing (ICSP)*. 100

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396. 18, 20, 62

Borg, I. and Groenen, P. (2005). *Modern Multidimensional Scaling: Theory and Applications.* Springer Verlag. 6, 8, 11

Buatois, L., Caumon, G., and Lvy, B. (2009). Concurrent number cruncher - a GPU implementation of a general sparse linear solver. *International Journal of Parallel, Emergent and Distributed Systems*, 24:205–223. 96, 101

Bunte, K., Haase, S., Biehl, M., and Villmann, T. (2012). Stochastic neighbor embedding (SNE) for dimension reduction and visualization using arbitrary divergences. *Neurocomputing*, 90:23–45. 62

Chen, L. and Buja, A. (2009). Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104:209–219. 27

Chen, S. and Zhang, D. (2011). Semisupervised dimensionality reduction with pairwise constraints for hyperspectral image classifcation. *IEEE Geoscience and Remote Sensing Letter*, 8:369–373. 25

Clark, J. and Holton, D. A. (2005). *A First Look At Graph Theory*. World Scientific Publishing Co. Pte. Ltd. 15, 22

Cox, T. F. and Cox, M. A. A. (2000). *Multidimensional Scaling*. CRC Press, 2nd ed. 32

Cui, M., Razdan, A., Hu, J., and Wonka, P. (2009). Interactive hyperspectral image visualization using convex optimization. *IEEE Transaction on Geoscience and Remote Sensing*, 47:1673–1684. 25, 30, 31, 33, 36

Darlington, R. B. (1999). Factor analysis. Technical report, Cornell University. 62

Demartines, P. and Hrault, J. (1997). Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8:148–154. 32, 62

Dianat, R. and Kasaei, S. (2010). Dimension reduction of optical remote sensing images via minimum change rate deviation method. *IEEE Transactions on Geoscience and Remote Sensing*, 48:198–206. 30

Donoho, D. L. and Grimes, C. (2005). Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data. In *The National Academy of Sciences, 102(21):7426-7431.* 62

Du, Q., Raksuntorn, N., Cai, S., and Moorhead, R. J. (2008). Color display for hyperspectral imagery. *IEEE Transaction on Geoscience and Remote Sensing*, 46:1858–1866. 25, 30

Dwyer, T., Marriott, K., and Wybrow, M. (2007). Integrating edge routing into force-directed layout. *Lecture Notes in Computer Science*, 4372:8–19. 26

Forero, P. A. and Giannakis, G. B. (2012). Sparsity-exploiting robust multidimensional scaling. *IEEE Transaction on Signal Processing*, 60:4118–4134. 11

France, S. L. and Carroll, J. D. (2011). Two-way multidimensional scaling: A review. *IEEE Transactions on Systems, Man and Cyberneticspart C: Applications And Reviews*, 41:644–661. 6, 32

Geng, X., Zhan, D.-C., and Zhou, Z.-H. (2005). Supervised nonlinear dimensionality reduction for visualization and classifcation. *IEEE Transactions on Systems, Man and Cyberneticspart B: Cybernetics*, 35:1098–1107. 15

Golub, G. H. and Loan, C. F. V. (1989). *Matrix Computations: 2nd Edition*. Baltmore: Jons Hopkins. 45

Hamarneh, G., Member, S., McIntosh, C., and Drew, M. S. (2011). Perception-based visualization of manifold-valued medical images using distance-preserving dimensionality reduction. *IEEE Transactions on Medical Imaging.*, 30:1314–1327. 15

Harel, D. and Koren, Y. (2002). Graph drawing by high-dimensional embedding. *Graph Drawing*, pages 207–219. 26

He, X., Cai, D., Yan, S., and Zhang, H.-J. (2005). Neighborhood preserving embedding. In *Tenth IEEE International Conference on Computer Vision (ICCV) (Volume:2)*. 62

Hu, Y. (2006). Efficient, high-quality force-directed graph drawing. *The Mathematica Journal*, 10:37–71. 26

Jacobson, N. P. and Gupta, M. R. (2005). Design goals and solutions for display of hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 43:2684–2692. 31, 36

Ji, S. and Ye, J. (2008). Generalized linear discriminant analysis: A united framework and efficient model selection. *IEEE Transactions on Neural Networks*, 19:1768–1782. 12

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer Verlag, New York, Inc. 10, 62

Kaski, S., Nikkil, J., Oja, M., Venna, J., Trnen, P., and Castrn, E. (2003). Trustworthiness and metrics in visualizing similarity of gene expression. *BMC Bioinformatics*, 4:1–13. 24

Kaski, S. and Peltonen, J. (2011). Dimensionality reduction for data visualization. *IEEE Signal Processing Magazine*, 28:100–104. 23

Kerr, D. A. and E., P. (2005). Color and color spaces. Technical report, Stanford University. 110

Kim, T.-K., Stenger, B., Kittler, J., and Cipolla, R. (2011). Incremental linear discriminant analysis using sufficient spanning sets and its applications. *International Journal of Computer Vision*, 91:216–232. 11

Kirk, D. B. and Hwu, W.-M. W. (2010). *Programming Massively Parallel Processors*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA. 99

Kotwal, K. and Chaudhuri, S. (2010). Visualization of hyperspectral images using bilateral filtering. *IEEE Transaction on Geoscience and Remote Sensing*, 48:2308–2316. 25

Lawrence, J., Arietta, S., Kazhdan, M., Lepage, D., and OHagan, C. (2011). A user-assisted approach to visualizing multidimensional images. *IEEE Transactions on Visualization and Computer Graphics*, 17:1487–1498. 44

Lee, J. A., Lendasse, A., and Verleysen, M. (2004). Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing*, 57:49–76. 22, 30, 32, 62

Lee, J. A. and Verleysen, M. (2007). *Nonlinear Dimensionality Reduction*. Springer. 8, 11

Lespinats, S. and Aupetit, M. (2009). False neighborhoods and tears are the main mapping defaults. how to avoid it ? how to exhibit remaining ones? quality issues, measures of interestingness and evaluation of data mining models. In *The QIMIE 2009. Bangkok, Thailand*. 24

Lespinats, S. and Aupetit, M. (2011). CheckViz: Sanity check and topological clues for linear and non-linear mappings. *Computer Graphics Forum*, 30:113–125. 24

Lespinats, S., Verleysen, M., Giron, A., and Fertil, B. (2007). DD-HDS: A method for visualization and exploration of high-dimensional data. *IEEE Transactions on Neural Networks*, 18:1265–1279. 23

Li, J. X. (2004). Visualization of high-dimensional data with relational perspective map. *Information Visualization*, 3:49–59. 22

Ma, W. and Agrawal, G. (2009). A compiler and runtime system for enabling data mining applications on GPU. *ACM SIGPLAN Notices - PPoPP '09*, 44:287–288. 101

Maaten, L. J. P. V. and Hinton, G. (2008). Visualizing high-dimensional data using t-SNE. *Machine Learning Research*, 9:2579–2605. 62

Maaten, L. J. V. D. (2013). Matlab toolbox for dimensionality reduction. In *http://homepage.tudelft.nl/19j49/Software.html*. 9

Mahmood, Z. and Scheunders, P. (2011). Enhanced visualization of hyperspectral images. *IEEE Geoscience and Remote Sensing Letter*, 8:869–873. 25

Martinez, A. M. and Zhu, M. (2005). Where are linear feature extraction methods applicable? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1934–1944. 11

Meng, D., YeeLeung, and ZongbenXu (2011). A new quality assessment criterion for nonlinear dimensionality reduction. *Neurocomputing*, 74:941–958. 8

Mignotte, M. (2012). A bicriteria optimization approach based dimensionality reduction model for the color display of hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 50:501–513. 27, 44

Mohan, A., Sapiro, G., and Bosch, E. (2007). Spatially coherent nonlinear dimensionality reduction and segmentation of hyperspectral images. *IEEE Geoscience and Remote Sensing Letter*, 4:206–210. 25, 30

Nishisato, S. (2006). *Multidimensional Nonlinear Descriptive Analysis*. Boca Raton, FL: Chapman & Hall. 6

NVIDIA (2009). NVIDIAs next generation CUDA compute architecture: Fermi. In *http://www.nvidia.com/object/fermi-architecture.html*. NVIDIA Corporation. 98

NVIDIA (2013). GPU computing applications. NVIDIA corporation. 96, 100

Onclinx, V., Wertz, V., and Verleysen, M. (2009). Nonlinear data projection on non-euclidean manifolds with controlled trade-off between trustworthiness and continuity. *Neurocomputing*, 72:1444–1454. 25

Peltonen, J. (2009). Visualization by linear projections as information retrieval. *Advances in Self-organizing Maps Lecture Notes in Computer Science*, 5629:237–245. 23

Rassokhin, D. N. and Agrafiotis, D. K. (2003). A modified update rule for stochastic proximity embedding. *Molecular Graphics and Modelling*, 22:133–140. 16, 30, 33, 62

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326. 30, 62

Samko, O., Marshall, A., and Rosin, P. (2006). Selection of the optimal parameter value for the isomap algorithm. *Pattern Recognition Letters*, 27:968–979. 15

Sanders, J. and Kandrot, E. (2011). *CUDA By Example*. Addison-Wesley Professional. 96, 99

Schanda, J., editor (2007). *Colorimetry: Understanding the Cie System*. John Wiley & Sons, Inc. 110

Schatz, M. C., Trapnell, C., Delcher, A. L., and Varshney, A. (2007). High-throughput sequence alignment using graphics processing units. *BMC Bioinformatics*, 8. 100

Scherl, H., Keck, B., Kowarschik, M., and Hornegger, J. (2007). Fast GPU-based ct reconstruction using the common united device architecture (CUDA). In *IEEE Nuclear Science Symposium Conference Record*. 100

Schreck, T., von Landesberger, T., and Bremma, S. (2010). Techniques for precision-based visual analysis of projected data. In *IS & T/SPIE Conference on Visualization and Data Analysis (VDA 2010) . San Jose, California.* 24

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22:888–905. 62

Silva, V. D. and Tenenbaum, J. B. (2003). Global versus local methods in nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems*, 15:705–712. 33

Smith, R. B. (2012). *Introduction to Hyperspectral Imaging*. MicroImages, Inc. 25

Steyvers, M. (2002). Multidimensional scaling. *Encyclopedia of Cognitive Science*. 30

Stone, S. S., Haldar, J. P., mei W. Hwu, W., Sutton, B. P., and pei Liang, Z. (2007). How GPUs can improve the quality of magnetic resonance imaging. In *Proc GPGPU*. 100

Taylor, J. S. and Christianini, N. (2004). Kernel methods for pattern analysis. *Cambridge University Press*. 62

Telea, A. (2008). *Data Visualization Principles and Practice*. A K peters, Ltd. 23

Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323. 15, 26, 30, 32, 62

Tipping, M. E. and Bishop, C. M. (1999). Mixtures of probabilistic principal component analysers. *Neural Computation*, 11:443–482. 62

Trapnell, C. and Schatz, M. C. (2009). Optimizing data intensive GPGPU computations for DNA sequence alignment. *Parallel Computing*, 35:429–440. 100

Tu, S. T., Chen, J. Y., Yang, W., and Sun, H. (2012). Laplacian eigenmaps-based polarimetric dimensionality reduction for SAR image classifcation. *IEEE Transaction on Geoscience and Remote Sensing*, 50:170–179. 18

Tyo, J. S., Konsolakis, A., Diersen, D. I., and Olsen, R. C. (2003). Principal components based display strategy for spectral imagery. *IEEE Transaction on Geoscience and Remote Sensing*, 41:708–718. 25, 30

Venna, J. and Kaski, S. (2001). *Neighborhood Preservation In Nonlinear Projection Methods: An Experimental Study*. Springer, Berlin. 25

Venna, J. and Kaski, S. (2005). Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity. In *WSOM Conference 05. Paris, France*. 24

Verma, R., Khurd, P., and Davatzikos, C. (2007). On analyzing diffusion tensor images by identifying manifold structure using isomaps. *IEEE Transactions on Medical Imaging*, 26:772–778. 15

Weinberger, K. and Saul, L. K. (2006). An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *Proceedings of the National Conference on Artificial Intelligence, Boston, MA*. 62

Yang, L. (2011). Distance-preserving dimensionality reduction. *Wiley Interdisc. Rew.: Data Mining And Knowledge Discovery*, 1:369–380. 8

Zeng, X. and Luo, S. (2008). Generalized locally linear embedding based on local reconstruction similarity. In *Fifth International Conference on Fuzzy Systems and Knowledge Discovery*. 18

Zhang, J. (2008). *Visualization For Information Retrieval*. Springer-Verlag Berlin Heidelberg. 6

Zhang, T., Yang, J., Zhao, D., and Ge, X. (2007). Linear local tangent space alignment and application to face recognition. *Neurocomputing*, 70:1547–1553. 62

Zhang, Z. and Zha, H. (2004). Principal manifolds and nonlinear dimensionality reduction via local tangent space alignment. *SIAM Journal of Scientific Computing*, 26:313–338. 62

Zhi, R. and Ruan, Q. (2008). Fractional supervised orthogonal local linear projection. In *CISP '08. Congress on (Volume:2) Image and Signal Processing.* 62