Fall 12-1-2015

Face Recognition with Multi-stage Matching Algorithms

Xianming Chen
University of Southern Mississippi

The University of Southern Mississippi

FACE RECOGNITION WITH MULTI-STAGE MATCHING ALGORITHMS

by

Xianming Chen

Abstract of a Dissertation
Submitted to the Graduate School
of The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

December 2015

ABSTRACT

FACE RECOGNITION WITH MULTI-STAGE MATCHING ALGORITHMS

by Xianming Chen

December 2015

For every face recognition method, the primary goal is to achieve higher recognition accuracy and spend less computational costs. However, as the gallery size increases, especially when one probe image corresponds to only one training image, face recognition becomes more and more challenging. First, a larger gallery size requires more computational costs and memory usage. Meanwhile, that the large gallery sizes degrade the recognition accuracy becomes an even more significant problem to be solved.

A coarse parallel algorithm that equally divides training images and probe images into multiple processors is proposed to deal with the large computational costs and huge memory usage of the Non-Graph Matching (NGM) feature-based method. First, each processor finishes its own training workload and stores the extracted feature information, respectively. And then, each processor simultaneously carries out the matching process for their own probe images by communicating their own stored feature information with each other. Finally, one processor collects the recognition result from the other processors. Due to the well-balanced workload, the speedup increases with the number of processors and thus the efficiency is excellently maintained. Moreover, the memory usage on each processor also evidently reduces as the number of processors increases. In sum, the parallel algorithm simultaneously brings less running time and memory usage for one processor.

To solve the recognition degradation problem, a set of multi-stage matching algorithms that determine the recognition result step-by-step are proposed. Each step picks a small proportion of the best similar candidates for the next step and removes the others. The behavior of picking and removing repeats until the number of remaining candidates is small enough to produce the final recognition result. Three multi-stage matching algorithms—n-ary elimination, divide and conquer, and two-stage hybrid—are introduced to the matching

process of traditional face recognition methods, including Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Non-graph Matching (NGM). N-ary elimination accomplishes the multi-stage matching from the global perspective by ranking the similarities and picking the best candidates. Divide and conquer implements the multi-stage matching from the local perspective by dividing the candidates into groups and selecting the best one of each group. For two-stage hybrid, it uses a holistic method to choose a small amount of candidates and then utilizes a feature-based method to find out the final recognition result from them. From the experimental results, three conclusions can be drawn. First, with the multi-stage matching algorithms, higher recognition accuracy can be achieved. Second, the larger the gallery size, the greater the improved accuracy brought by the multi-stage matching algorithms. Finally, the multi-stage matching algorithms achieve little extra computational costs.

FACE RECOGNITION WITH MULTI-STAGE MATCHING ALGORITHMS

by

Xianming Chen

A Dissertation
Submitted to the Graduate School
and the School of Computing
at The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Approved:

_____
Dr. Zhaoxian Zhou, Committee Chair
Associate Professor, School of Computing

_____
Dr. Chaoyang Zhang, Committee Member
Professor, School of Computing

_____
Dr. Zheng Sun, Committee Member
Associate Professor, School of Computing

_____
Dr. Zheng Wang, Committee Member
Assistant Professor, School of Computing

_____
Dr. Ras Pandey, Committee Member
Professor, Physics and Astronomy

_____
Dr. Karen S. Coats
Dean of the Graduate School

December 2015

ACKNOWLEDGMENTS

I would like to express my immense gratitude to my advisor Dr. Zhaoxian Zhou, for his continuous support, patient guidance, and positive encouragement through my four years' study experience at USM. Many thanks also go to the members of my dissertation committee, Dr. Chaoyang Zhang, Dr. Jonathan Sun, Dr. Zheng Wang, and Dr. Ras Pandey, for their precious time and valuable advice on my dissertation.

To my father Jinghua Chen, who taught me to be a thinking man seeking the truth of the world: I give thanks to him for my birth and for raising me. Special thanks to my beloved wife, Dongxin Lin, for her countless sacrifices and endless patience and understanding.

TABLE OF CONTENTS

ABSTRACT iii

ACKNOWLEDGMENTS iv

LIST OF ILLUSTRATIONS vii

LIST OF TABLES ix

LIST OF ABBREVIATIONS x

NOTATION AND GLOSSARY xi

1 INTRODUCTION **1**
1.1 Introduction of Face Recognition 1
1.2 Brief History of Face Recognition 3
1.3 Recognition Degradation of Face Recognition 5
1.4 Multi-stage Matching Algorithms 6
1.5 Organization of the dissertation 8

2 TRADITIONAL RESEARCH ON FACE RECOGNITION **9**
2.1 Principal Component Analysis (PCA) Method 9
2.2 Linear Discriminant Analysis (LDA) Method 14
2.3 Non-Graph Matching (NGM) Method 18
2.4 Comparison of Holistic Method and Feature-based Method 33

3 MULTI-STAGE MATCHING ALGORITHMS **35**
3.1 Recognition Degradation Problem 35
3.2 The Core of Multi-stage Matching Strategy 37
3.3 N-ary Elimination Matching Algorithm 39
3.4 Divide and Conquer Matching Algorithm 41
3.5 Two-stage Hybrid Matching Algorithm 43
3.6 Summary of the Three Multi-stage Matching Algorithms 48

4 EXPERIMENTS **49**
4.1 Experiment Configurations 49
4.2 Experimental Results 52
4.3 Summary for Experiments 61

5 CONCLUSIONS AND FUTURE WORKS 63
5.1 Conclusions 63
5.2 Future Works 64

BIBLIOGRAPHY 66

LIST OF ILLUSTRATIONS

Figure

1.1 Access control in application of face recognition . . . 1
1.2 A typical flow path face recognition system . . . 2
1.3 Face recognition scenario . . . 3
1.4 The recognition degradation problem of NGM . . . 7
1.5 The core of multi-stage matching strategy . . . 8

2.1 A simple example for KLT . . . 10
2.2 PCA training matrix . . . 11
2.3 PCA average face . . . 12
2.4 Remove the average component . . . 12
2.5 Re-expression of an image . . . 12
2.6 Principal Components . . . 13
2.7 PCA matching with Minimum Distance . . . 14
2.8 A simple example for FLD projection . . . 15
2.9 Definitions of S_B and S_w . . . 17
2.10 A case of windowed Fourier Transform . . . 19
2.11 GWT response with four orientations . . . 20
2.12 The 40 Gabor filters . . . 21
2.13 GWT response on an image with 40 filters . . . 22
2.14 The workflow of NGM . . . 23
2.15 Dividing an image into blocks . . . 24
2.16 Feature points distribution . . . 24
2.17 Data structure of a feature point . . . 25
2.18 An illustration of NGM matching . . . 26
2.19 The visualized parallel process in time axis . . . 29
2.20 Speedup with different number of processors . . . 31
2.21 Efficiency with different number of processors . . . 32
2.22 General expression of holistic methods . . . 33
2.23 Facial details extracted by feature-based methods . . . 34

3.1 Demonstration of matching dispersion of NGM . . . 37
3.2 Recognition degradation of face recognition . . . 38
3.3 The core of multi-stage matching strategy . . . 39
3.4 NGM with n-ary elimination matching algorithm . . . 41
3.5 NGM with divide and conquer matching algorithm . . . 42
3.6 The demonstration of matching with divide and conquer ($M = 2$) . . . 43
3.7 Recognition degradation comparison of two types of methods . . . 45

3.8 The two-stage hybrid matching algorithm work flow 47

4.1 Original fa and fb sample images 50
4.2 Normalization process 51
4.3 Sample normalized training and probing images 51
4.4 NGM accuracy comparison with n-ary elimination 54
4.5 Computation times (in seconds) comparison with binary elimination 55
4.6 Recognition accuracy with divide and conquer 58
4.7 Recognition performance of two-stage hybrid 60
4.8 Computational costs of hybridizing PCA with NGM 62

LIST OF TABLES

Table

1.1 Computational costs (in seconds) comparison of PCA and NGM 4

2.1 Computation costs (in seconds) comparison of PCA and NGM 27
2.2 Memory usage (in MB) comparison of PCA and NGM 27
2.3 Computational time (in sec) with different number of processors 30
2.4 Speedup with different number of processors 30
2.5 Efficiency with different number of processors 30
2.6 Memory usage (in MB) with different number of processors 31
2.7 Memory usage rate with different number of processors 32

3.1 Matching Dispersion of a probe image indexed 10 36
3.2 A typical case of n-ary elimination ($n = 2$) matching algorithm 40
3.3 Gallery size=100, grouping number M=5 44
3.4 A rectified examples by the two-stage hybrid matching algorithm 46

4.1 Recognition accuracy of n-ary elimination 53
4.2 Recognition accuracy differences of n-ary elimination 53
4.3 Computation costs (in seconds) comparison with binary elimination 56
4.4 Recognition accuracy of divide and conquer 57
4.5 Recognition difference of divide and conquer 57
4.6 Recognition accuracy of two-stage hybrid 59
4.7 Improvement of hybridizing PCA and NGM 59
4.8 Improvement of hybridizing LDA and NGM 61
4.9 Computation costs (in seconds) of hybridizing PCA with NGM 61

LIST OF ABBREVIATIONS

EBGM - Elastic Bunch Graph Matching
FLD - Fisher's Linear Discriminant
FR - Face Recognition
GWT - Gabor Wavelet Transform
KLT - Karhunen-Loeve Transform
LDA - Linear Discriminant Analysis
NGM - Non-graph Matching
PCA - Principal Component Analysis

NOTATION AND GLOSSARY

General Usage and Terminology

The notation used in this text represents fairly standard mathematical and computational usage. In many cases these fields tend to use different preferred notation to indicate the same concept, and these have been reconciled to the extent possible, given the interdisciplinary nature of the material. In particular, the notation for partial derivatives varies extensively, and the notation used is chosen for stylistic convenience based on the application. While it would be convenient to utilize a standard nomenclature for this important symbol, the many alternatives currently in the published literature will continue to be utilized.

The blackboard fonts are used to denote standard sets of numbers: $\mathbb{R}$ for the field of real numbers, $\mathbb{C}$ for the complex field, $\mathbb{Z}$ for the integers, and $\mathbb{Q}$ for the rationals. The capital letters, $A, B, \cdots$ are used to denote matrices, including capital greek letters, e.g., Λ for a diagonal matrix. Functions which are denoted in boldface type typically represent vector valued functions, and real valued functions usually are set in lower case roman or greek letters. Calligraphic letters, e.g., $\mathcal{V}$, are used to denote spaces such as $\mathcal{V}$ denoting a vector space, $\mathcal{H}$ denoting a Hilbert space, or $\mathcal{F}$ denoting a general function space. Lower case letters such as i, j, k, l, m, n and sometimes p and d are used to denote indices.

Vectors are typeset in square brackets, e.g., $[\cdot]$, and matrices are typeset in parentheses, e.g., $(\cdot)$. In general the norms are typeset using double pairs of lines, e.g., $||\cdot||$, and the absolute value of numbers is denoted using a single pairs of lines, e.g., $|\cdot|$. Single pairs of lines around matrices indicates the determinant of the matrix.

Chapter 1

INTRODUCTION

1.1 Introduction of Face Recognition

As an indispensable part of pattern recognition and a successful application in artificial intelligence, face recognition has been receiving increasingly significant attention in the past few decades. Emergence of academic conferences, periodicals and varieties of commercial applications in face recognition proves its rapid development. Annual conferences such as IEEE Conference on Computer Vision and Pattern Recognition (CVPR) since from 1983 and International Conference on Computer Vision (ICCV) since from 1987 represent the top level of face recognition academy exchanges, while IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI) and International Journal of Computer Vision (IJCV) report the latest progress about face recognition research. On the other hand, commercial applications of face recognition have pervaded our daily life, such as facial capture of social media, access control systems, anti-terrorism, and so on. Figure 1.1 demonstrates a case of access control in an application of face recognition [24].

Figure 1.1: Access control in application of face recognition

Generally, the input of a face recognition system is still images or video stream, while the output is the recognition result. Inside the system, there are three processes, which are face detection, face training, and face matching [56]. Figure 1.2 illustrates a typical flow path of

a face recognition system. Actually, each of the three processes can be another individual research area in computer vision and artificial intelligence. For instance, face detection leads to other applications like face tracking, pose estimation, and so on [22, 14, 23, 45]. Face training leads to other research topics such as emotion recognition and facial feature tracking [1, 25, 4, 35, 36, 41]. For face matching, it refers to determining the recognition result by matching the training features, so it leads to the study of choosing an effective mathematics function to finish the final step of face recognition [2, 37, 3, 17, 31].

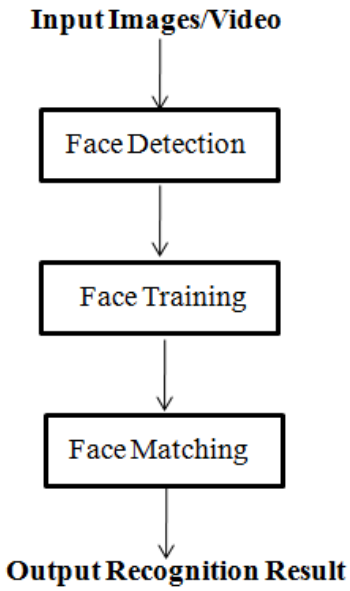

Figure 1.2: A typical flow path face recognition system

From the perspective of research, the scenario of face recognition can be narrower: given a still or video image of a human face, match it from a stored database of faces, of which identities are known to researchers. The images to be recognized are defined as probe images, and the images stored in the database are termed as training images. In the experimentation of face recognition research, both the probe images and training images are selected from some standard databases, such as FERET, MIT, Yale, AR, and so on. And many evaluation methods have been proposed for face recognition research [42, 44, 46, 26, 20]. In our work, we use the same amount of probe images and training images, which are one-to-one corresponding, as depicted as in Figure 1.3, since it is more convenient to evaluate the recognition accuracy.

Figure 1.3: Face recognition scenario

1.2 Brief History of Face Recognition

Artificial intelligence has been a dream of human beings for some time, yet it has not attained any real achievement until the arising of computer science technology. In the past four decades, various methods have been conducted by psycho-physicists, neuro-scientists, and computer engineers to solve face recognition problems with higher accuracy and less computation cost. According to different emphases of methodology, face recognition methods can be roughly classified as holistic, feature-based, and hybrid methods [56].

The earliest research of face recognition can be traced back to the 1950s [7] and to the 1960s [6] in psychology area. In 1972, Darwin et al. [11] published some work on facial expression of the emotion. The first work on automatic face recognition was done by Kanade et al. [28] in 1973. In the mid-1970s, researchers used measured attributes of facial features to classify patterns. Then research on face recognition had a dormant period in the 1980s. Later, due to growing interest in commercial opportunities, the availability of hardware resources and higher security requirements, face recognition stepped into its prime in the 1990s. Kirby et al. [30] and Turk et al. [49] established the foundation of eigenfaces, an appearance-based holistic method, which is also called Principal Component Analysis (PCA). Belhumeur et al. [40], Etemad et al. et al. [12] and Zhao et al. [57] built up another holistic method, Fisherfaces, i.e. Linear Discriminant Analysis (LDA). Meanwhile, another type of method, the feature-based method, has also achieved great success. The representations are Elastic Bunch Graph Matching (EBGM) by Wiskott et al. [51] and Active Appearance Models (AAM) by Edwards et al. [40]. From the beginning of the 21st-

century, the main test subject of face recognition has turned from still images to video-based images due to the higher requirements of anti-terrorism systems, access control systems, surveillance, and so on. New methods, such as "Face-ARG Matching" [39], "Probabilistic Models" [16], have also raised concerns about pattern recognition area. Therefore, face recognition research continues to play an important role in the area of computer vision and artificial intelligence.

1.2.1 Face Recognition and Parallelization

As the amount of training images (i.e. the gallery size) increases, the response time of a face recognition system inevitably rises. Moreover, the required memory usage of the system also goes up. On the other hand, the gallery size that a face recognition system is able to work with manifests its capacity and scalability. Obviously, the simplest way to simultaneously reduce the response time and memory usage for a face recognition system is parallelization.

Compared with holistic methods that have fast execution speed, there is more significance to apply parallelization on feature-based methods, since feature-based methods require more computational costs to extract local features and more memory usage to store them. Table 1.1 shows the difference of the two types of method in computational costs with the same experimental platform. For instance, to recognize 100 probe images from 100 training images, PCA, a typical holistic method, spends 2.4 seconds, while NGM (Non-Graph Matching), a representation of feature-based method, takes 1718.4 seconds, with three more orders. Thus, it is more beneficial to parallelize feature-based methods.

Table 1.1: Computational costs (in seconds) comparison of PCA and NGM

Gallery Size	PCA Time	NGM Time
100	2.4	1718.4
200	10.6	3541.7
300	22.0	5417.6
400	38.1	7638.2
500	60.0	9685.7

In our work, we propose a coarse parallel method for NGM. Rather than parallelizing the NGM itself, the parallel method equally distributes training images and probe images to the multiple processors. First, each processor finishes its own training workload and stores the extracted feature information respectively. And then, the matching process is executed in each processor simultaneously by communicating its own stored feature information with each other. Finally, one processor collects the recognition result from the other processors. Due to the well-balanced workload by equal distribution, the speedup increases with the

number of processors, and thus the efficiency is excellently maintained from the experimental results. Moreover, the memory usage on each processor also gets reduced as the number of processors increases. The parallelization of NGM brings two merits. First, it almost reduces the running time linearly as the number of processors increases, so it provides great significance for a real-time face recognition systems. Second, since the parallelization allows the needed memory to be distributed on multiple processors, the gallery capacity of a face recognition system can be greatly enlarged.Therefore, parallelization brings more practicality for a face recognition system.

1.3 Recognition Degradation of Face Recognition

Although some other methods of pattern recognition—such as iris recognition, fingerprint recognition—are more reliable in terms of recognition accuracy, the noninvasive feature of face recognition has its own incomparable advantage, especially in security systems that require unaware cooperation from the target participants [38]. However, face recognition becomes more and more challenging so that the recognition accuracy gradually decreases, especially under two circumstances: "single sample per person" [55, 19] and large gallery size [43]. This fact is termed as the recognition degradation problem in this dissertation.

"Single sample per person" refers to a condition that one probe image corresponds to only one training image. Many reported face recognition algorithms require a gallery that multiple training samples correspond to one probe image so as to deal with different poses, lighting conditions and other situations. If working with "single sample per person", the recognition accuracy of many reported algorithms gradually decreases so that they may fail to work effectively due to the extremely limited information provided by a single sample [48]. On the other hand, successful face recognition using "single sample per person" is critical due to two factors. First, it is arduous to collect multiple samples for one target person, especially for a sake of surveillance or anti-terrorism. Second, fewer samples per person also mean lower cost for storing and processing them, which is critical for a real-time system. Therefore, working effectively with "single sample per person" will be of great benefit for face recognition systems.

Besides "single sample per person", a face recognition system is also required to be able to work well with a large gallery size, which manifest its capacity and reliability. However, a larger gallery size certainly brings more challenges to face recognition. For holistic methods, say PCA, the recognition result is decided by the training image that has the shortest projection distance with the target probe image in the "eigenfaces" space. Since the relative projection of any one training image and the probe image is certain in

the sub-faces space, larger gallery size means more possibilities for an unexpected training image to have a shorter distance than the corresponding training image has. In that case, a false recognition happens. For feature-based methods, say NGM, the matching process is to match the feature points of the training images and the target probe image. As the number of training images increases, it is more possible for one single feature point of a probe image to match to a wrong training image. If the corresponding training image fails to get enough matched feature points, a false recognition also occurs. On the other hand, with the development of the face recognition system, the working gallery size is consequently required to increase. Therefore, it is also of great importance to increase the capacity and reliability of a face recognition system.

To more explicitly illustrate the recognition degradation problem, the recognition performance of NGM is tested by varying the gallery size from 100 to 1000 with an increment of 100, and one probe image references to only one training image. The experimental accuracies along with different gallery sizes are collected and plotted in Figure 1.4, from which we can see that the accuracy continuously decreases with the number of training images increasing. With a gallery size of 100, the accuracy reaches to a high level above 90%, whereas with 1000, it drops to below 60%, which is unacceptable for face recognition.

1.4 Multi-stage Matching Algorithms

Generally, there are two entries to solve the recognition degradation problem presented in Section 1.3, either the training process or the matching process. If choosing the training process as the entry to solve the recognition degradation problem, the process of features extraction would be an emphasis to be optimized. If choosing the matching process, the issue would become how to boost the performance of face recognition by optimizing the matching algorithm to find out the recognition result. In our work, it is the matching process that we focus on to solve the recognition degradation problem.

The core of computer vision is to teach computers to view the world as human beings do. Other than one round of matching to decide the recognition result in traditional methods, the matching process can be compared to deciding the championship in a sports tournament, which normally requires several rounds, such as preliminary contest, intermediary heat, semi-final, and final. Each training image is a competitor, and the recognition result is the champion, which is required to pass through multiple steps. Each step of matching selects a small portion of the competitors that have the best similarities with the probe image as the new candidates for the next step, and eliminates the rest. Once the number of remaining candidates is small enough, the final round of matching executes and produces

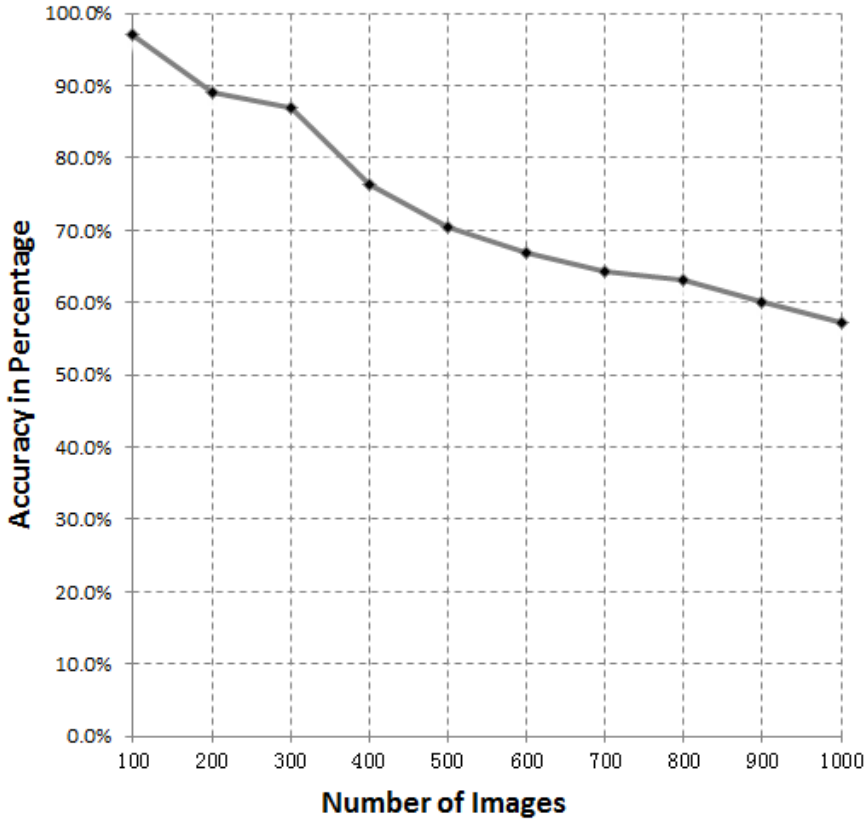

Figure 1.4: The recognition degradation problem of NGM

the recognition result. So, the multi-stage matching strategy has two properties: narrowing searching range in multiple steps and making the decision in a small range. This is the core of the multi-stage matching strategy, as illustrated in Figure 1.5.

In our work, three multi-stage matching algorithms are proposed to solve the recognition degradation problem: n-ary elimination, divide and conquer, and two-stage hybrid. For n-ary elimination, each step of matching picks $1/n$ of the training images as the new candidates and removes the others. The behavior of picking and eliminating repeats until the number of remaining training images is small enough to produce the recognition result. For divide and conquer, the training images are divided into groups with a grouping number, and the best similar one of each group is selected as a new candidate for the next step of matching. The behavior of dividing and conquering continues until the number of remaining training images is small enough to produce the recognition result. For two-stage hybrid, it hybridizes a holistic method and a feature-based method. A holistic method is applied to preprocess

Figure 1.5: The core of multi-stage matching strategy

the gallery and select a small amount of candidates that have the best similarities. And then a feature-based method is used to find out the recognition result.

The experimental results show that the recognition performance has been boosted considerably by the multi-stage matching algorithms. Moreover, it seems that the larger the gallery size, the greater the improvement. Additionally, the multi-stage matching algorithms have little affects to running times.

1.5 Organization of the dissertation

The rest of the dissertation is organized as follows: Chapter 2 introduces several representations of face recognition methods and discusses the origin of the recognition degradation problem. Additionally, a parallel algorithm for NGM is also demonstrated. In Chapter 3, three multi-stage matching algorithms are presented in detail and some preliminary results are also demonstrated. Chapter 4 reports the whole experimental process of the three multi-stage matching algorithms. Finally, Chapter 5 summarizes the dissertation and discusses future works.

Chapter 2
TRADITIONAL RESEARCH ON FACE RECOGNITION

In this chapter, two holistic methods (PCA and LDA) and one feature-based method (NGM) are presented. The recognition degradation problem of them is also discussed subsequently. Finally, a parallel algorithm of NGM is demonstrated.

2.1 Principal Component Analysis (PCA) Method

PCA is an excellent representation of holistic methods, which identify human faces by emphasizing on matching global features, such as facial contours, organ positions, and so on. PCA, a statistically data-processing and dimension-reduction technique, was firstly applied to face recognition by Matthew Turk and Alex Pentland in [49, 50], where the concept of "eigenfaces" was proposed.

2.1.1 Karhunen-Loeve Transform

The basic theory of PCA in mathematics is Karhunen-Loeve Transform (KLT), which is a commonly used orthogonal transformation. Assume $\mathbf{X}$ is n-dimensional random variable, it can be expressed by the weighted sum of n basis vectors:

$$\mathbf{X} = \sum_{i=1}^{n} \alpha_i \phi_i, \tag{2.1}$$

where α_i is the i^{th} weighted coefficient, and ϕ_i is the i^{th} basis vector. This equation is equivalent to:

$$\mathbf{X} = (\phi_1, \phi_2, ..., \phi_n)(\alpha_1, \alpha_2, ..., \alpha_n)^{\mathrm{T}} = \mathbf{\Phi}\boldsymbol{\alpha}. \tag{2.2}$$

Choosing orthonormal basis vectors,

$$\mathbf{\Phi}_i^{\mathrm{T}}\mathbf{\Phi}_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \Longrightarrow \mathbf{\Phi}_i^{\mathrm{T}}\mathbf{\Phi}_j = I, \tag{2.3}$$

so the coefficient vector is:

$$\boldsymbol{\alpha} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{X}. \tag{2.4}$$

Based on the statement above, the expansion coefficients of KLT can be obtained by following steps:

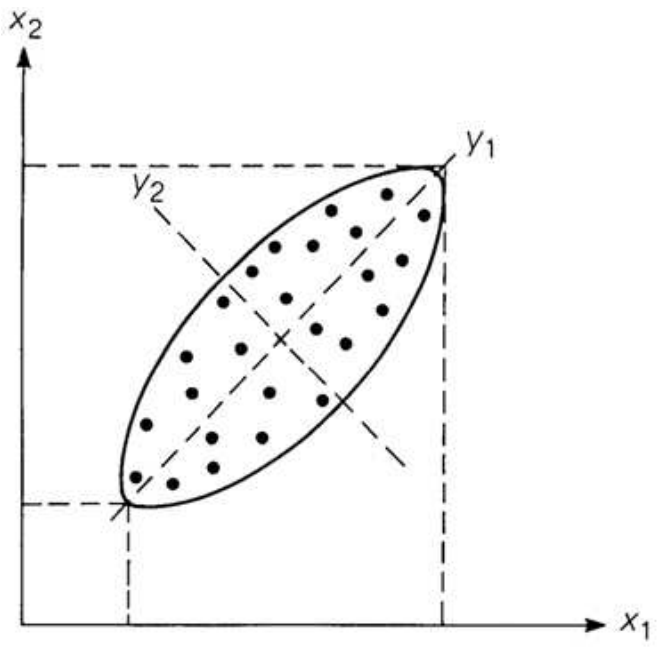

Figure 2.1: A simple example for KLT

1. Solve the autocorrelation matrix $\mathbf{R}$ of the random vector $\mathbf{X}$ by $\mathbf{R} = E[\mathbf{X}^{\mathrm{T}}\mathbf{X}]$. Since there is no practical significance without the mean vector μ involved that indicates the classification information, the covariance matrix of $\mathbf{X} - \mu$ always becomes the generating matrix of KLT coefficients via $\sum = E[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^{\mathrm{T}}]$, where μ is the overall mean vector.

2. Calculate the eigenvalue λ_i and eigenvector ϕ_i of the autocorrelation matrix $\mathbf{R}$ or the covariance matrix, and obtain the complete eigenvectors $\boldsymbol{\phi} = (\phi_1, \phi_2, ..., \phi_n)$

3. The expansion coefficients can be achieved by calculating $\boldsymbol{\alpha} = \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{X}$.

Figure 2.1 illustrates a simple example of KLT. Given a heap of scattered data on a 2-dimensional space as showed in Figure 2.1, other than selecting x_1 and x_2 as the coordinates, y_1 and y_2 that point to the greatest variance of the orientations are a better choice by KLT. The essence of KLT is to establish a new coordinate system so that the greatest variance by any projection of the input data aligns to the first coordinate, the second greatest variance to the second coordinate, and so on. In other words, the KLT removes the correlations of each component of the original data vector, and gets rid of those coordinates that include little information so as to reduce the dimensions of eigenspace. Those chosen coordinates are so-call principal components.

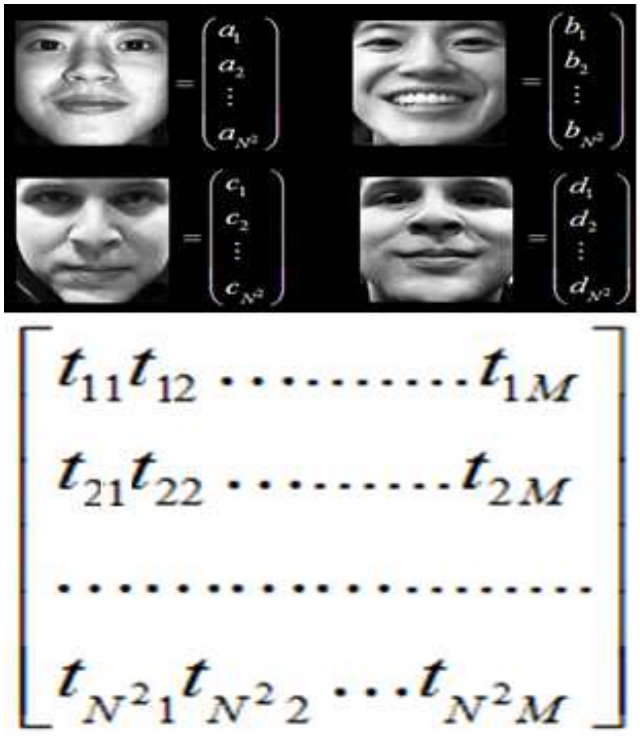

Figure 2.2: PCA training matrix

2.1.2 Eigenfaces for Recognition

For an image with $N \times N$ pixels, it can be considered as a vector with a length of N^2 or a point in an N^2 dimensional space. Analyzing with this type of vectors with KLT comes to the method of Principal Component Analysis. The follows are the general steps of PCA implementation [49, 50]:

1. Load M training set Γ in sequence as a matrix by turning the two dimensional image data into one dimensional vector by column. As illustrated in Figure 2.2, each column of the matrix represents the raw data of an input training image.

2. Calculate the generating matrix for KLT.

   (a) Figure out the average face of the training set $\mathbf{T} = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n$. Figure 2.3 shows how to calculate the average matrix. And it can be also seen that the average face actually does not look like a real human face.

   (b) Calculate the difference faces from average face by $\mathbf{\Phi}_i = \Gamma_i - \mathbf{T}$, as demonstrated in Figure 2.4. This process actually is to remove the average component of the matrix so as to reduce the calculation cost.

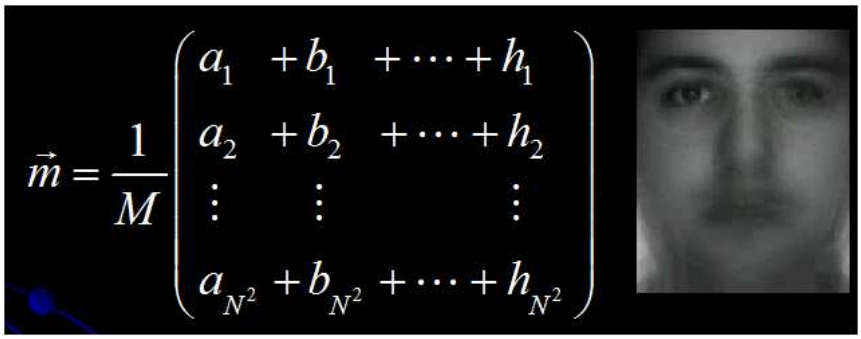

Figure 2.3: PCA average face

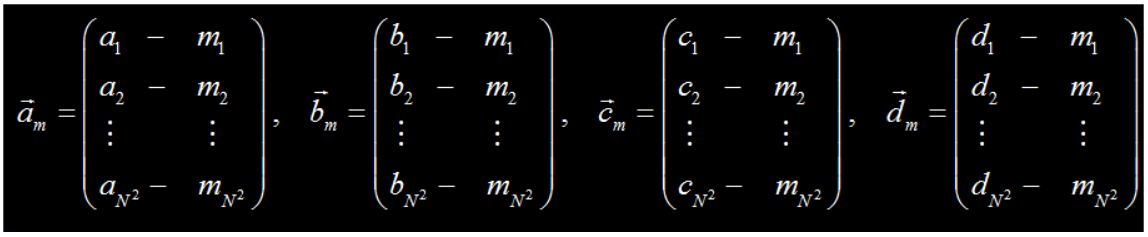

Figure 2.4: Remove the average component

(c) Calculate the covariance matrix of the difference faces by $D = \frac{1}{M}A^{\mathrm{T}}A$, where $A = [\mathbf{\Phi}_1\mathbf{\Phi}_2\mathbf{\Phi}_3...\mathbf{\Phi}_M]$.

3. Calculate the eigenvectors $\mathbf{v}$ and eigenvalues λ of the generating matrix D by Singular Value Decomposition (SVD) so as to get an equation $Cov = A\mathbf{v} = \lambda\mathbf{v}$.

4. Project the training images and probe images on to the eigenvector space and calculate the eigenfaces. Assuming there are K eigenvectors, each image can be represented by the linear combination of eigenvectors and eigenvalues plus the average face as showed in Figure 2.5. In other word, one face can be decomposed as one average face and a series of eigenfaces.

5. Choose a small amount of principal eigenfaces to reduce the dimensionality of sub-

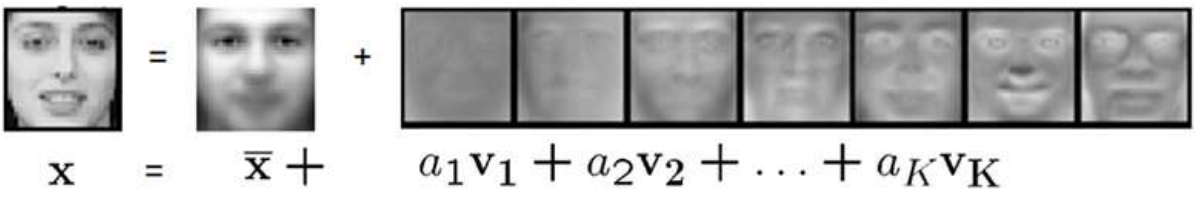

Figure 2.5: Re-expression of an image

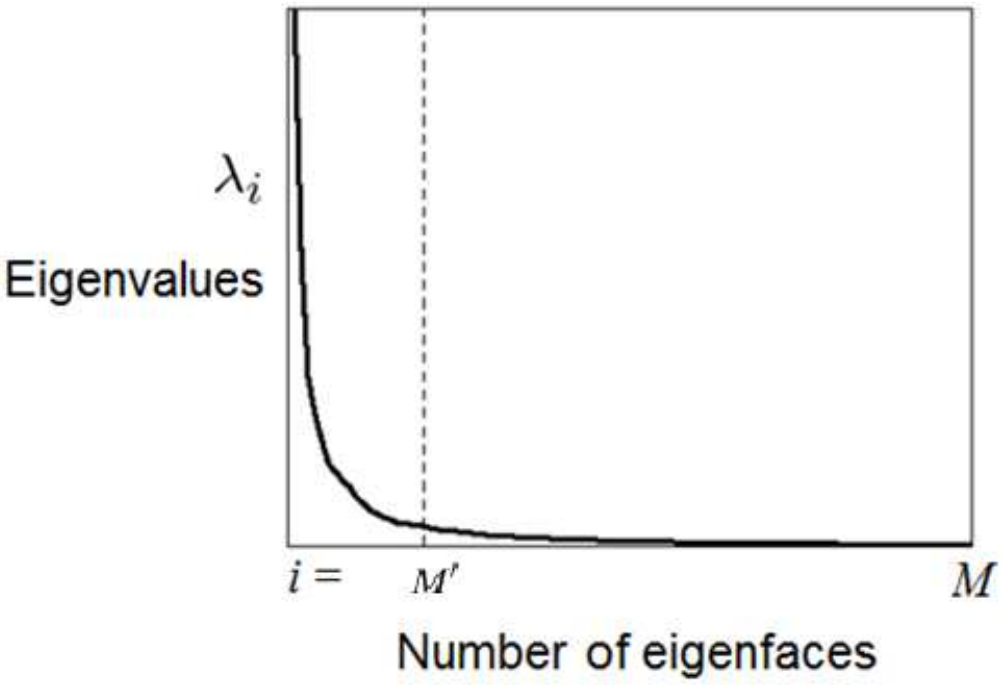

Figure 2.6: Principal Components

faces space. For M training images, generally there are M "eigenfaces" generated. Yet the first few $M'(M' < M)$ eigenfaces possess most of the eigenvalues as showed in Figure 2.6. Those principal "eigenfaces" represent low frequency information, i.e. the main features of an image, while the rest of them generally refer to the high frequency information, i.e. trivial information that might disturb the following recognition.

6. For each probe image, compare the projections of training images with the chosen $M'(M' < M)$ eigenvectors, and determine the classification of samples by applying an appropriate categorizer, such as L_1 Paradigm, L_2 Paradigm, Minimum Distance or Mahalanobis Distance. Figure 2.7 demonstrates an example of using Minimum Distance to determine the recognition result. Since $D1 < D2$, the training image that has $D1$ would be chosen as the recognition result among the two candidates.

Steps 1 to 5 together can be viewed as the training process, and step 6 is the matching process. Since from the birth of PCA, many improved methods based on PCA have been proposed. They either modify the training process or revise the matching process. Linear Discriminant Analysis (LDA) that chooses a different way of projecting is another representation of holistic method, which will be discussed in the next section.

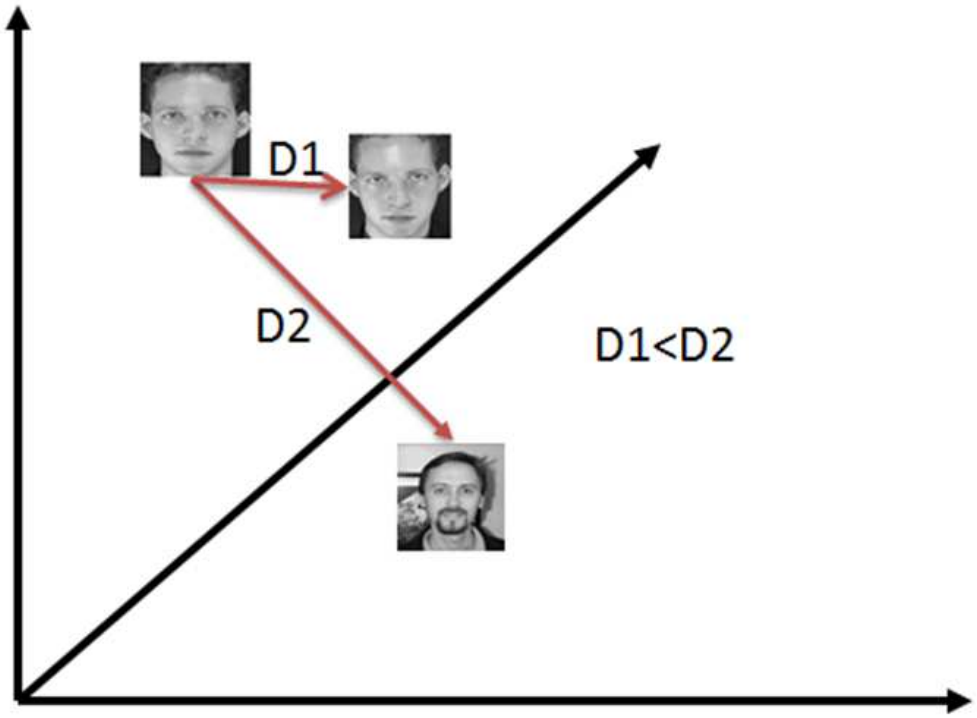

Figure 2.7: PCA matching with Minimum Distance

2.2 Linear Discriminant Analysis (LDA) Method

Similari with PCA, the LDA method proposed firstly by Etemad et al. [13] and Belhumeur et al. [40], projects the image space to a low dimensional feature space by applying Fisher's Linear Discriminant (FLD) [15]. Thus, LDA is also called "Fisherfaces" method.

2.2.1 Fisher's Linear Discriminant

KLT and FLD share one common property of reducing the dimensionality of the feature space, but the implementation strategy is different from each other. KLT yields projection directions that maximize the total scatter across all classes, so PCA retains unwanted variations such as lighting and facial expression [40]. On the other hand, FLD finds projection that maximizes scatter between classes and minimizes scatter within classes, so LDA searches a linear combination of feature vectors that yields the best discrimination among classes. Figure 2.8 illustrates how FLD projects. For the same data input, projection on $x1$ is not a wise choice, since the data cannot be classified due to the overlapping. Yet, it should be the direction of the red line on where the data projects, because it completely distinguishes the two groups of the data. In fact, the most significant difference between

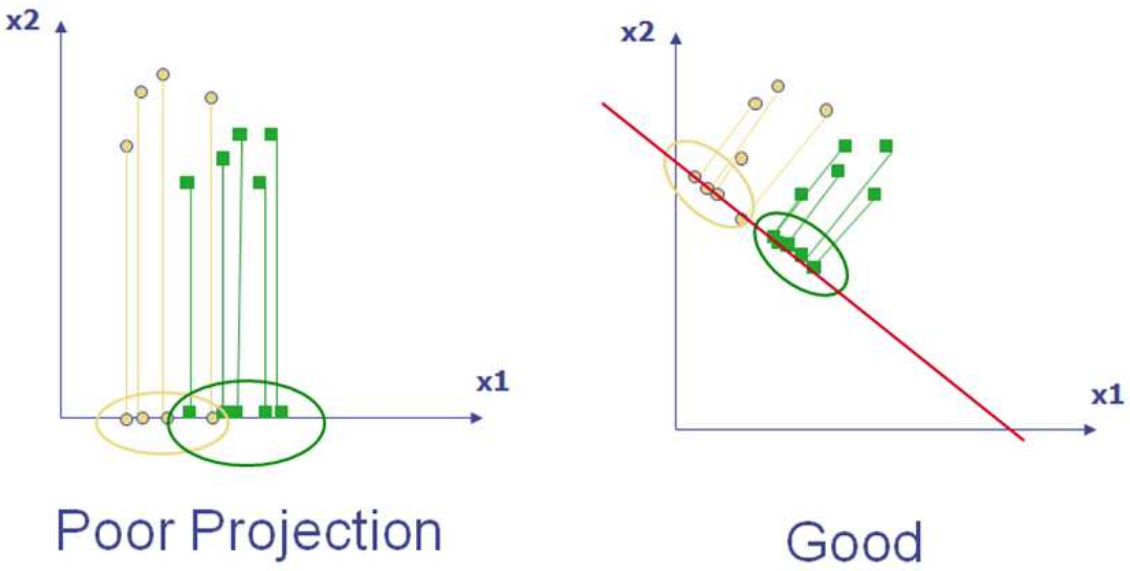

Figure 2.8: A simple example for FLD projection

KLT and FLD is the orthogonality of projection vectors. Each feature vector of KLT should be mutually orthogonal, while it is not necessary for FLD.

Assuming that we are to find a vector w to project the data x onto this vector and get a new coordinate y with two classes, i.e.,

$$y = w^T x. \tag{2.5}$$

After the projection, the distance between the same classes should be as small as possible, while at the same time the distance between different classes should be as large as possible. Define the mean value of each class

$$m_i = \frac{1}{n_i} \sum_{x \in D_i} x. \tag{2.6}$$

After the projection, the mean is

$$\bar{m}_i = \frac{1}{n_i} \sum_{y \in Y_i} y = \frac{1}{n_i} \sum_{x \in D_i} w^T x = \frac{1}{n_i} w^T \sum_{x \in D_i} x = w^T m_i. \tag{2.7}$$

where n_i is the amount of the i^{th} class, D_i is the set of the i^{th} class data, Y_i is the set of the i^{th} class data after the projection.

The average distance between the two classes after the projection is

$$|\bar{m}_1 - \bar{m}_2| = |w^T (m_1 - m_2)|. \tag{2.8}$$

Define the scatter between each data class after the projection is

$$s_i^2 = \sum_{y \in Y_i} (y - \bar{m}_i)^2. \tag{2.9}$$

Now the mission of FLD is converted to finding a w that maximizes the value of

$$J(w) = \frac{|\bar{m_1} - \bar{m_2}|}{s_1^2 + s_2^2}. \tag{2.10}$$

Define the scatter matrix to describe the scatter of each class before projection

$$S_i = \sum_{x \in D_i} (x - m_i)(x - m_i)^T, \tag{2.11}$$

then every term of the denominator can be expressed with S_i and w:

$$\bar{S_i^2} = \sum_{x \in D_i} (w^T x - w^T m_i)^2 = \sum_{x \in D_i} w^T (x - m_i)(x - m_i)^T w = w^T S_i w. \tag{2.12}$$

For two classes, it can be further expressed as

$$\bar{S_1^2} + \bar{S_2^2} = w^T (S_1 + S_2) w = w^T S_W w, \tag{2.13}$$

and the numerator can be expressed as

$$(\bar{m_1} - \bar{m_2})^2 = (w^T m_1 - w^T m_2)^2 = w^T (m_1 - m_2)(m_1 - m_2)^T w = w^T S_B w. \tag{2.14}$$

Therefore, $J(w)$ can be rewritten as

$$J(w) = \frac{w^T S_B w}{w^T S_w w}. \tag{2.15}$$

If the class number c is larger than 2, the between-matrix and within-matrix can be expressed as

$$S_B = \sum_{i=1}^{c} n_i (m_i - m)(m_i - m)^T, m = \frac{1}{n} \sum_x x; S_w = \sum_{i=1}^{c} S_i. \tag{2.16}$$

Figure 2.9 explicitly shows the definitions of S_B and S_w.

In summary, the LDA is to find a set of M feature basis vectors, denoted as $\{\Psi_m\}_{m=1}^{M}$, in such a way that the ratio of the between and within class scatters of the training sample is maximized. The maximization process is formulated as

$$\Psi = \arg \max \frac{|\Psi^T S_B \Psi|}{|\Psi^T S_w \Psi|}, \Psi = [\psi_1, \cdots, \psi_M]. \tag{2.17}$$

This is equivalent to an eigenvalue problem

$$S_B \psi_m = \lambda_m S_w \psi_m, m = 1, 2, \cdots, M. \tag{2.18}$$

Thus, the basis vectors correspond to the first M most significant eigenvectors of $S_w^{-1} S_B$.

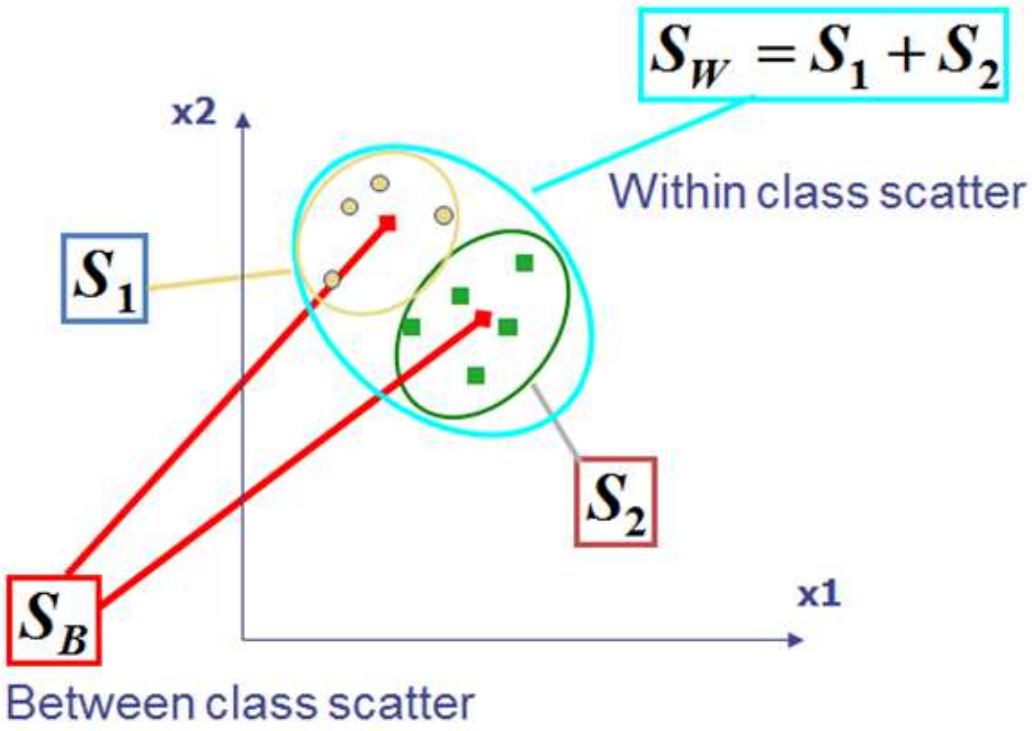

Figure 2.9: Definitions of S_B and S_w

2.2.2 Fisherfaces for Recognition

Assume to recognize C probe images from C training images with "single sample per person", the number of classes is C: $\Gamma = \{\Gamma_i\}_{i=1}^{C}$, each class containing $\Gamma_i = \{\Gamma_{ij}\}_{j=1}^{C_i}$ face images, i.e. $\Gamma_i = 1$. Supposedly, an M-dimensional Fisher subspace spanned by Ψ is generated, a $J \times M$ matrix with $M << J$. The detailed Fisherfaces for recognition is described as follows: [32, 33]

1. Calculate $\overline{\Gamma_i} = \frac{1}{C_i}\sum_{j=1}^{C_i}\Gamma_{ij}$, $\overline{\Gamma} = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{C_i}\Gamma_{ij}$, $\Phi_{b,i} = \sqrt{\frac{C_i}{N}}(\overline{\Gamma}_i - \overline{\Gamma})$, $\Phi_b = [\Phi_{b,1}, \Phi_{b,2}, \cdots, \Phi_{b,c}]$, and $\vec{S}_b = \Phi_b\Phi_b^T$.

2. Find the m eigenvectors of $\Phi_b\Phi_b^T$, with non-zero eigenvalues, and denote them as $\vec{E}_m = [\vec{e}_1, \vec{e}_2, \cdots, \vec{e}_m]$.

3. Calculate the first m most significant eigenvectors ($\vec{U}_m$) of $\vec{S}_b$ and their corresponding eigenvalues (Λ_b) by $\vec{U}_m = \Phi_b\vec{E}_m$ and $\Lambda_b = \vec{U}_m^T\vec{S}_b\vec{U}_m$.

4. Let $\vec{H} = \vec{U}_m\Lambda_b^{-\frac{1}{2}}$. Find eigenvectors of $\vec{H}^T\vec{S}_w\vec{H}$, $\vec{P} = [\vec{p}_1, \vec{p}_2, \cdots, \vec{p}_m]$ sorted in increas-

ing eigenvalues order. The within-class scatter matrix is defined as

$$\vec{S_w} = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{C_i}(\Gamma_i - \overline{\Gamma}_i)(\Gamma_i - \overline{\Gamma}_i)^T. \tag{2.19}$$

5. Choose the first $M(\leq m)$ eigenvectors in $\vec{P}$. Let $\vec{P}_M$ and Λ_w be the chosen eigenvectors and their corresponding eigenvalues respectively.

6. Return $\vec{\Psi} = \vec{H}\vec{P}_M(\eta\vec{I} + \Lambda_w)^{-\frac{1}{2}}$.

7. Compare probe image and training images by using the Euclidean distance in the feature space.

Both PCA and LDA belong to holistic methods that solve face recognition from the perspective of global appearance. In next section, a feature-based method, NGM, will be introduced.

2.3 Non-Graph Matching (NGM) Method

The concept of "jets" was introduced in EBGM method that firstly proposed by Wiskott et. al. in 1997 [51]. NGM [29] is another type of method that uses "jets" to represent a facial image. Different with EBGM that involves manual manipulation to locate some landmarks, NGM executes automatically. To some extent, NGM can be viewed as an automatic version of EBGM.

2.3.1 Gabor Wavelet Transform (GWT)

As a typical feature-based face recognition method, the theory foundation of NGM is Gabor Wavelets Transform (GWT), which is able to extract local features from input signals. To fully understand GWT, it is necessary to look back the relationship between Fourier Transform (FT) and GWT. As one of the most common methods to analyze the frequency properties of a time signal, yet FT can not tell the correspondence of time and frequency exactly, i.e., the time information is lost through the transform. To indicate the information of time and frequency simultaneously, a windowed function should be added into FT so as to extract local frequency feature. Figure 2.10 shows a case of windowed FT that is able to tell the frequency feature varying with the time axis.

In 1946, Denis Gabor proposed Gabor function as a tool for signal detection in noise [18]. As a windowed function, Gabor function was introduced to FT and deduced it into GWT, which is able to extract local features of two-dimensional signals. For two-dimensional

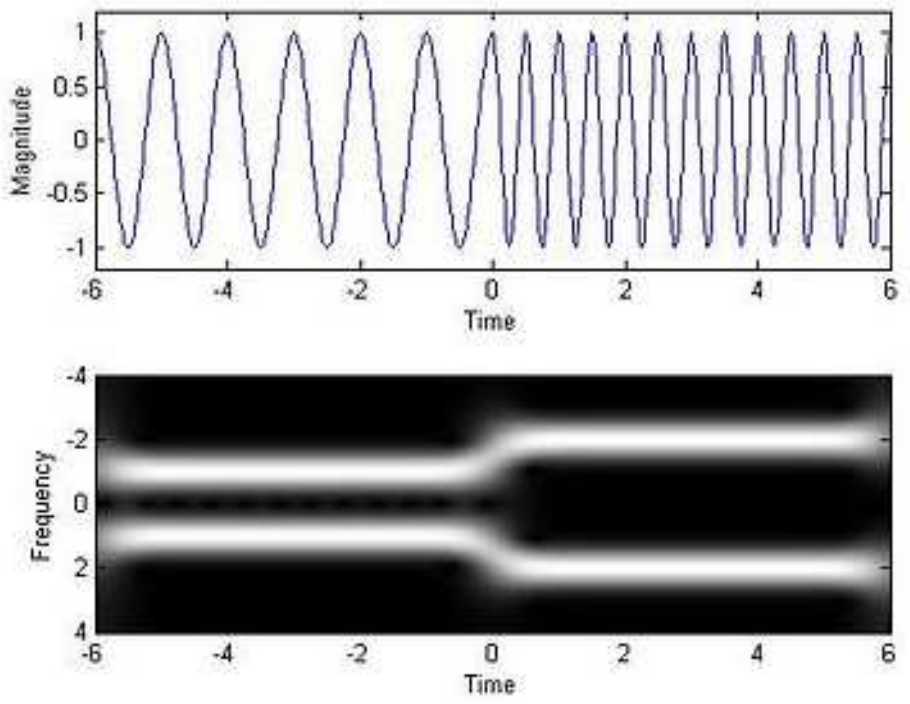

Figure 2.10: A case of windowed Fourier Transform

signals, the local features refer to the variance with wavelength and orientations. Figure 2.11 illustrates an example of GWT response on a Chinese character with four orientations.

For human faces, the sizes, angles, and positions of facial organs, such as nose, eyes, mouth, and so on, are distinct local features. The GWT of an image is to calculate the convolution of the gray value matrix and a set of Gabor filters with different spatial frequencies and orientations. The concept of a "jet" $R_j(\vec{x})$ that refers to a small patch of gray values around a given pixel, is led by the coefficients of convolution result for different orientations and frequencies[51]:

$$R_j(\vec{x}) = \int I(\vec{x}')\Psi_j(\vec{x}-\vec{x}')d^2\vec{x}', \tag{2.20}$$

with a family of Gabor kernels

$$\Psi_j(\vec{x}) = \frac{||\vec{k}_j||^2}{\sigma^2} e^{-\frac{||\vec{k}_j||^2||\vec{x}||^2}{2\sigma^2}} \left[e^{i\vec{k}_j\cdot\vec{x}} - e^{-\frac{\sigma^2}{2}} \right], \tag{2.21}$$

where $I(\vec{x})$ is the gray value of the image at $\vec{x} = (x, y)$. Each Ψ_j is a plane wave characterized by the vector $\vec{k}_j$ enveloped by a Gaussian function, where σ is the standard deviation of this Gaussian. In our work, 5 discrete frequencies with index $v = 0, 1, \cdots, 4$, and 8 orientations with indexes $\mu = 0, 1, \cdots, 7$, 40 kernels in all are utilized. The center frequency of the j^{th}

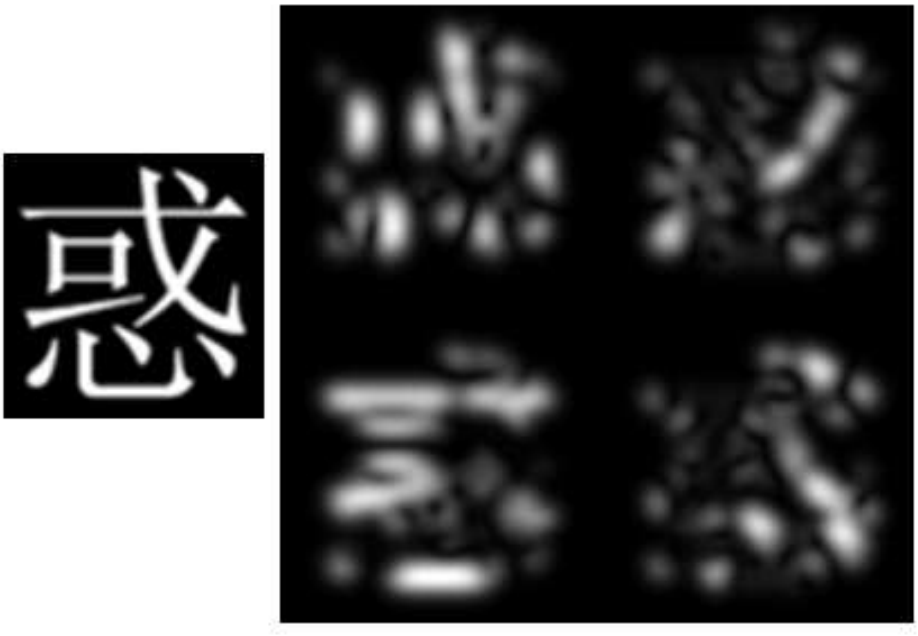

Figure 2.11: GWT response with four orientations

filter is given by the characteristic wave vector, and $j = \mu + 8v$.

$$\vec{k}_j = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_v \cos\theta_\mu \\ k_v \sin\theta_\mu \end{pmatrix}; k_v = 2^{\frac{v+2}{2}}\pi;$$

$$\theta_\mu = \mu\frac{\pi}{2}; v = 0,\cdots,4; u = 0,\cdots,7; j = \mu + 8v. \tag{2.22}$$

Figure 2.12 illustrates the 40 Gabor filters with 5 spatial frequencies and 8 orientations. Figure 2.13 shows the amplitude response of a face image to the 40 Gabor filters.

2.3.2 Speed up the convolution of GWT

Gabor filters can be used to extract local features of an image, but the high computational costs of the convolution process raises an obstacle for feature-based methods to be applied in real time system [47]. For instance, for an image of size $N \times N$, the size of Gabor kernel is $N \times N$, thus the complexity of the convolution of the two is $\Theta(N^4)$, which is unacceptably time-consuming. The process was tested for an image of the size with 311×232 pixels, running on a machine with Intel Core Quad 2.66GHz CPU and 8G RAM. Each convolution for one single filter costs approximately 10 minutes, which means 400 minutes for the whole 40 filters of an image.

To reduce the computational time of the convolution, many methods have been proposed. Conventionally, using Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform

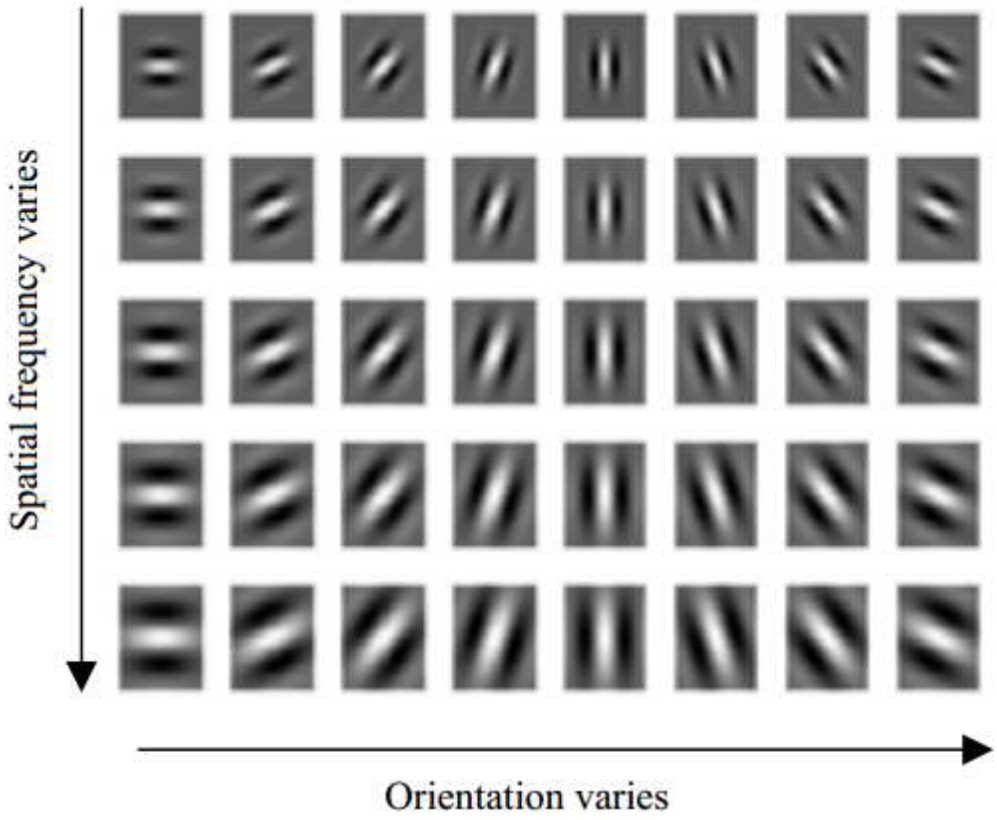

Figure 2.12: The 40 Gabor filters

(IFFT) is the general method to accelerate the convolution process. However, the two stages together are still time consuming to face recognition [47]. Other researchers have tried to optimize the Gabor representations by selecting locations and Gabor wavelets, but the recognition accuracy would be harmed.

The Gabor kernel matrix in the convolution represents the contribution from other pixels to the target pixel, which diminishes quickly with the distance between the target pixel and surrounding pixels. It is feasible to calculate kernel values from pixels that have limited distance to the target. So in our work, we ignore the negligible contribution from 10 pixels' distance away. With the same platform mentioned above, it has been tested that the computational time for GWT can be reduced by about 600 times, or less than one minute to complete the convolution for an image, without significant loss of recognition accuracy.

2.3.3 NGM Implementation

After applying GWT on input images, the following steps are to extract feature points and match similarities. The flowchart of NGM that includes a training process and a matching process is demonstrated in Figure 2.14. The detailed workflow shows as follows [29]:

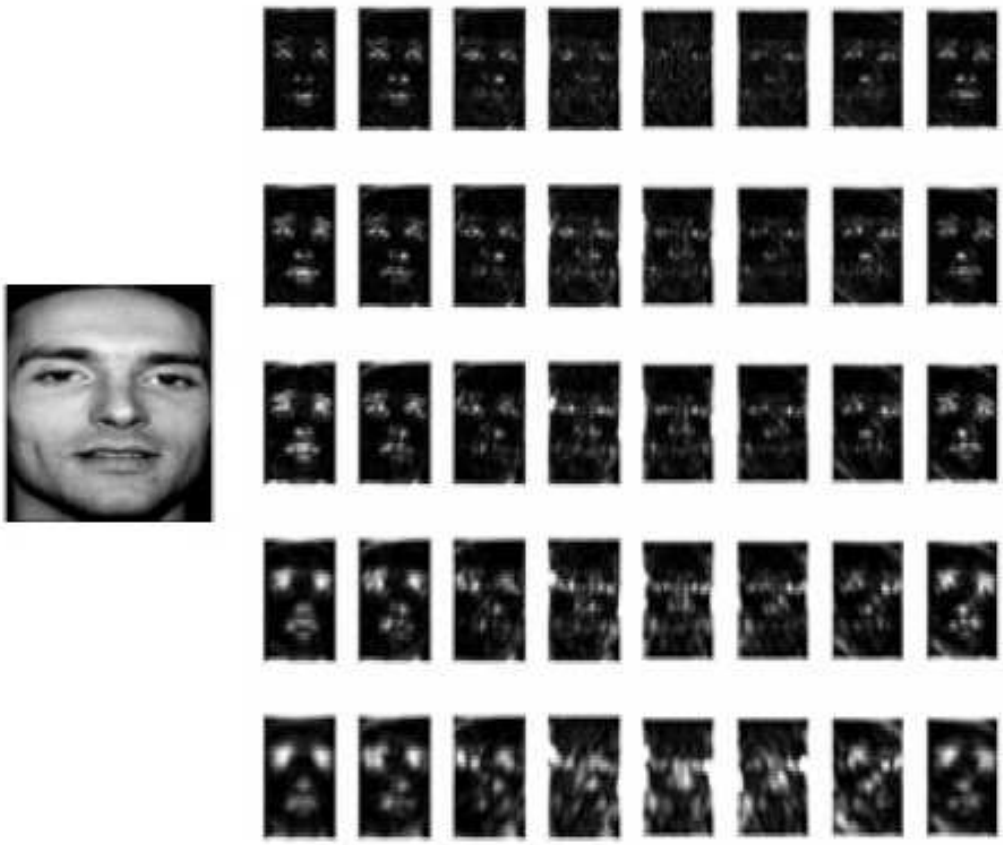

Figure 2.13: GWT response on an image with 40 filters

1. GWT calculating.

   For each image, calculate the GWT response value of each pixel with the 40 Gabor filters.

2. Feature points searching.

   Divide images into blocks with size of $w \times w$ and find a feature point for each block as demonstrated in Figure 2.15. A feature point located at (x_0, y_0) in a window W_0 should satisfy

   $$R_j(x_0, y_0) = \max_{(x,y)\in W_0} \left(R_j(x,y)\right), \tag{2.23}$$

   and

   $$R_j(x_0, y_0) > \frac{1}{N_1 N_2} \sum_{x=1}^{N_1} \sum_{y=1}^{N_2} R_j(x,y), \tag{2.24}$$

   where $(x,y) \in W_0, j = 1, \cdots, 40$, and N_1, N_2 is the width and height of the image. Equation 2.24 is applied to get rid of those points that have the peaks of the responses

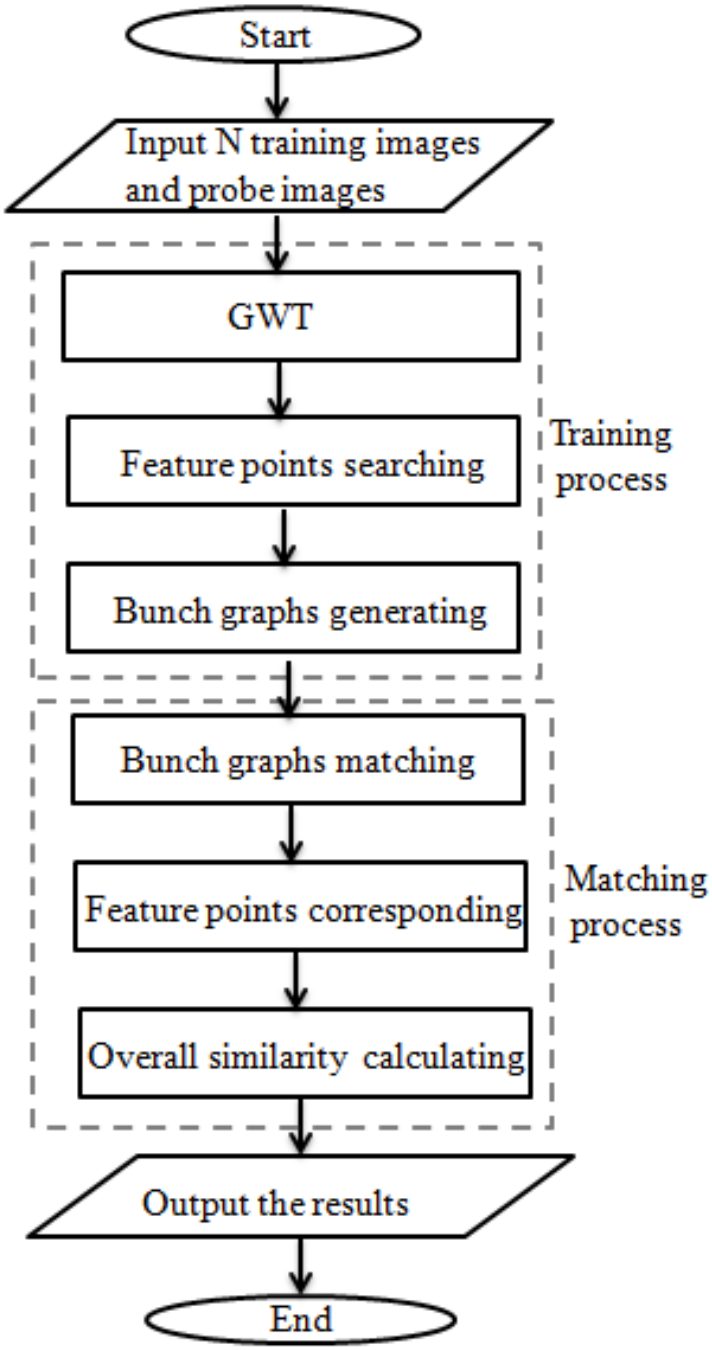

Figure 2.14: The workflow of NGM

in local yet actually do not include valuable information in overall, since their response values are even less than the overall average. It is worth mentioning that the result of comparing two Gabor response values that are in plural form depends on their real-parts. In our work, all the images are normalized with $N_1 = 311, N_2 = 232$. And the window size is set with 20×20 to capture the significant features and avoid

Figure 2.15: Dividing an image into blocks

Figure 2.16: Feature points distribution

redundancy. Figure 2.16 shows the result of feature points searching. It can be seen that most of the searched feature points locate around eyes, nose, mouth, and other positions where contains the main features of a human face.

3. Feature vectors generating.

   Generate the feature vectors by combining the 40 GWT coefficients and 2 coordinates of those feature points. The n^{th} feature vector of i^{th} reference face is defined as:

$$v_{i,n} = \{x_n, y_n, R_{i,j}(x_n, y_n), j = 1, 2, \cdots, 40\}, \tag{2.25}$$

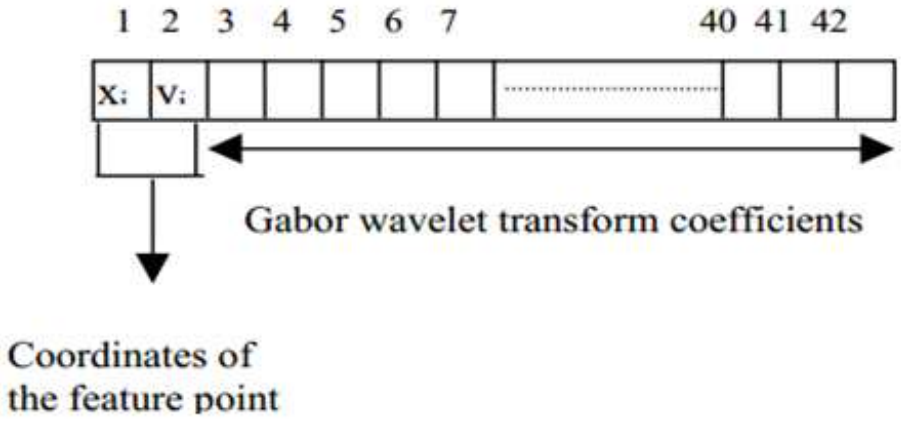

Figure 2.17: Data structure of a feature point

while the first two elements represent the location information of coordinates, and the other 40 are the Gabor filter response at that point. Figure 2.17 illustrates the composition of the data structure of a feature point.

4. Similarity calculating for feature points.

   For each feature point of each probe image, measure the similarities with the feature points of all training images by:

$$S_{p,t}(m,n) = \frac{\sum_l |v_{p,m}(l)||v_{t,n}(l)|}{\sqrt{\sum_l |v_{p,m}(l)|^2 |v_{t,n}(l)|^2}}, l = 3, ..., 42. \tag{2.26}$$

   where $S_{p,t}(m,n)$ represents the similarity of the m^{th} feature vector of the p^{th} probe image, $(v_{p,m})$, to the n^{th} feature vector of the t^{th} reference training image, $(v_{t,n})$. Since the first two elements of a feature vector represent the location information, as the number of vector elements, l starts from 3 and ends at 42. For the similarity of two feature points to be calculated, they should be close enough by calculating their relative location coordinates, or the similarity of them is meaningless to recognition. In this paper, only those feature points within 8 pixels from training images can be involved.

5. Feature points matching.

   For each feature point of each probe image, find out the training image that has the maximum similarity and mark the training image with one score.

6. Overall similarity calculating.

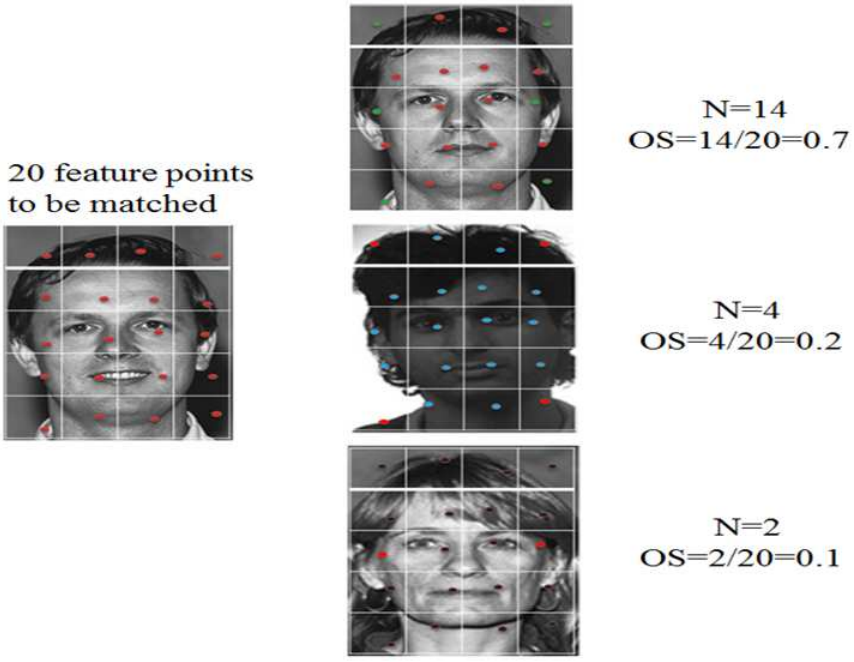

Figure 2.18: An illustration of NGM matching

For each probe image, calculate the overall similarities of training images with the formula:

$$OS_t = \frac{C_t}{N_t}. \tag{2.27}$$

where C_t is the number of feature vectors the t^{th} reference training image that have maximum similarity to a feature vector of the probe image, and N_t is the number of feature vectors that the t^{th} training image has.

7. Recognition result output.

   For each probe image, find out the training image that has the greatest overall similarity, i.e. OS.

Figure 2.18 demonstrates the matching process of NGM to recognize a probe image from three training images, and assume the probe image and three training images all have 20 feature points. The 20 feature points of the probe image are marked in red, and the matched feature points of training images are also in red, or in other colors. Thus, the first training image that has the greatest overall similarity is the recognition result.

2.3.4 A Parallelization Algorithm to NGM

The work of NGM parallelization has been done and published in [8].

Why does NGM need parallelization?

Compared with PCA and LDA algorithms, NGM algorithm has the advantage of higher recognition accuracy [53], especially when the gallery size is relatively small, since the local features extracted from the human faces are more robust against distortions and other external elements. However, NGM is demanding in two aspects. First, it is time-consuming due to the high computational cost of GWT [47]. Table 2.1 shows the computational costs comparison of PCA and NGM with the same experimental platform. It can be seen that NGM is two or three higher time order than PCA. For instance, to recognize 1000 probe images from 1000 training images, PCA spends 232.9 seconds, while NGM takes 21922.6 seconds, with two more orders. So, the high computational times brings an obstacle for NGM to be applied in real-time systems.

Table 2.1: Computation costs (in seconds) comparison of PCA and NGM

Gallery Size	PCA Time	NGM Time
100	2.4	1718.4
200	10.6	3541.7
300	22.0	5417.6
400	38.1	7638.2
500	60.0	9685.7
600	85.0	11822.4
700	114.2	14388.8
800	152.2	16440.2
900	189.3	19107.1
1000	232.9	21922.6

Second, as a feature-based method, NGM needs a huge usage of memory to store the local features extracted by GWT. Table 2.2 collects the memory usage of NGM, which linearly increases with the gallery size. For 100 images, it needs 195 megabits while goes to 323 megabits for 300 images. Certainly, the expanding memory usage implies an obstruction for the scalability of NGM face recognition. On the other hand, the memory usage of PCA or LDA stays relatively stable in a low level.

Table 2.2: Memory usage (in MB) comparison of PCA and NGM

Gallery Size	PCA memory	LDA memory	NGM memory
100	26	30	195
200	47	51	259
300	61	70	323

Parallelization algorithm of NGM

In this section, a coarse parallel method is proposed for NGM to deal with its high computational costs and memory usage. To obtain a high speedup and efficiency for a parallel algorithm, the task scheduling should be managed based on two principles [21]. The first principle is to let the processors do the same work, so the time costs of idling can be reduced to the greatest extent. The other one is to let them do the useful work, so the extra running time can be saved.

In [54], the authors applied geometric features decomposing algorithm to divide a face image into several sub-regions and assign the matching processes onto multimedia service grid to speed up the execution. Based on the two parallel principles, rather than parallelizing the NGM itself, the parallel method equally distributes training images and probe images to the multiple processors. First, each processor finishes its own training workload and stores the extracted feature information respectively. And then, the matching process is executed in each processor simultaneously by communicating their own stored feature information with each other. Finally, one processor collects the recognition result from the other processors.

To illustrate the parallel algorithm more explicitly, we start with an example of p processors, with N training images and N probe database, assuming that N can be divided exactly by p. The parallel algorithm is implemented on Message Passing Interface (MPI) platform with C language as follows: [8]

1. Divide the N training images into p processors equally and finish the training process respectively. So each processor acquires feature matrices of $\frac{N}{p}$ training images.

2. Divide the N probe images into p processors equally and finish the training process respectively. So each processor acquires feature matrices of $\frac{N}{p}$ probe images.

3. Each processor completes the matching process for $\frac{N}{p}$ probe images in turn, by broadcasting the $\frac{N}{p}$ feature matrices with each other.

Figure 2.19 drawn by Jumpshot with the log file generated from MPICH2, illustrates the behavior of paralleling 50 training images and 50 probe images into 5 processors. It can be seen that, on the first half of the time axis, there is almost no MPI action, since all processors do training work respectively. In the second half of the time axis, the 10 bars of each processor shows the communication time for the recognition process of 10 probe images. The five bars circled in a dotted line show that there are five images to be probed simultaneously for one time in five processors. In addition, the narrow bar width indicates that the speedup and efficiency of the parallelization would be excellent.

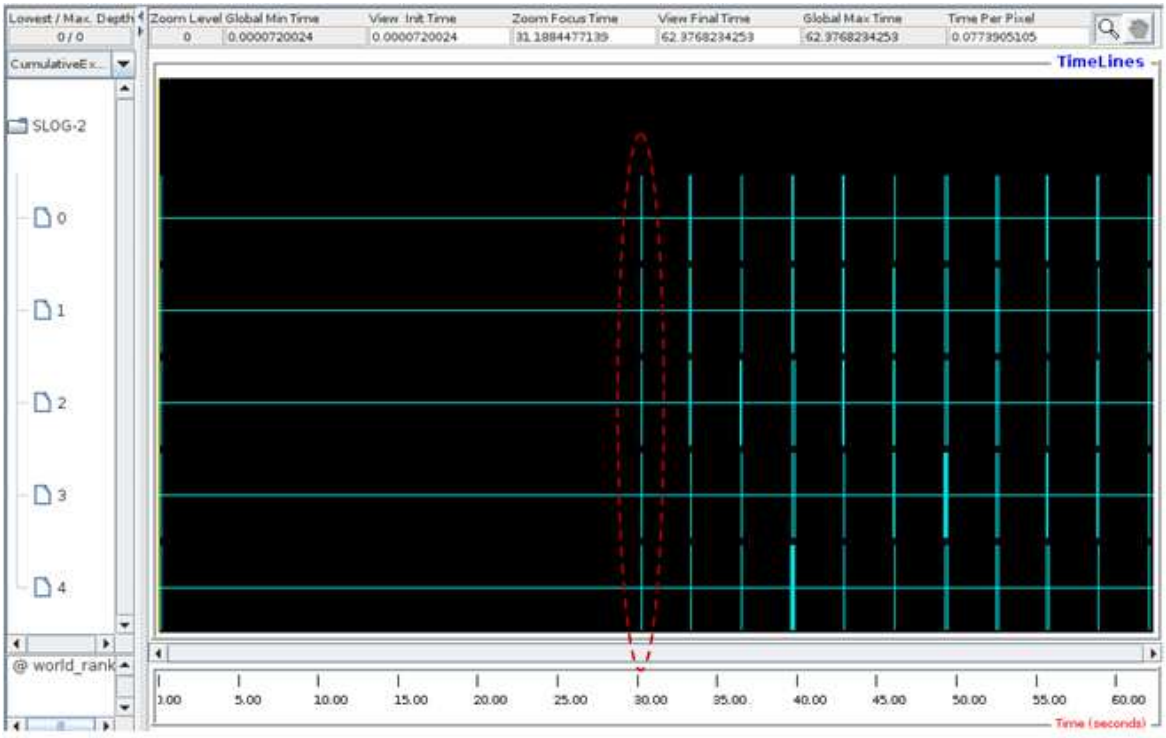

Figure 2.19: The visualized parallel process in time axis

Performance analysis

Both the sequential and parallel algorithms were implemented with C language on MPI platform. The programs were executed on Sequoia, one of the super computers with 124 node clusters in the Mississippi Center for Supercomputing Research (MCSR). Each node consists of Dual Intel Xeon Quad-core E5420 Harpertown processors, a 1333 MHz Front Side Bus, a 12 MB L2 Cache, and a 250 GB Disk Drive. Each Harpertown processor consists of 4 2.5 GHz CPUs and 8 GB memory. Overall, Sequoia has 496 CPUs and 3.44TB memory.

To evaluate the parallel algorithm thoroughly, 100, 200, and 300 images were chosen in the training database and probing database respectively, as different problem size. The codes ran on different number of processors, ascending with 1, 2, 5, 10, 25, and 50 in the end. Testing items include execution time, speedup, efficiency, and memory usage. The accuracy is not discussed here as it provides trivial information about the parallelization.

The sequential computational time ($p = 1$) and parallel computational time, denoted as T_s andT_p, are recorded in Table 2.3. We can see that the execution time decreases as the number of processors increases, which meets our expectation. The speedup is given in Table 2.4, which is defined as $S = \frac{T_s}{T_p}$. The efficiency is provided in Table 2.5, which is defined as

$E = \frac{S}{p}$, where p is the number of processors. The performance is also illustrated in Figure 2.20, where the speedup is shown to approach to the ideal speedup line (the dotted diagonal line), though it begins to deviate away after 10 processors. Likewise, in Figure 2.21, it can be seen that the efficiency, which describes how much time is spent on the meaningful computation, maintains over 0.8 even with 50 processors. In all, Figure 2.20 and Figure 2.21 agree well with the fact that, the larger the problem, the better the performance.

Table 2.3: Computational time (in sec) with different number of processors

Processors	100 images	200 images	300 images
p=1	702.501	1446.598	2347.725
p=2	362.370	750.716	1214.845
p=5	147.064	302.602	488.947
p=10	75.718	153.228	247.051
p=25	31.799	63.939	102.449
p=50	17.203	33.711	53.557

Table 2.4: Speedup with different number of processors

Processors	100 images	200 images	300 images
p=1	1.000	1.000	1.000
p=2	1.939	1.927	1.933
p=5	4.777	4.781	4.802
p=10	9.278	9.441	9.503
p=25	22.092	22.625	22.916
p=50	40.837	42.926	43.836

Table 2.5: Efficiency with different number of processors

Processors	100 images	200 images	300 images
p=1	1.000	1.000	1.000
p=2	0.969	0.963	0.966
p=5	0.955	0.956	0.960
p=10	0.928	0.944	0.950
p=25	0.884	0.905	0.917
p=50	0.817	0.859	0.877

The other crucial element to evaluate a parallel algorithm is the memory usage. Table 2.6 reveals the decreasing memory usage on each processor with the increasing number of processors, while Table 2.7, the memory usage rate table, displays the fact that the memory usage goes down to about $40-50\%$ for 50 processors, compared with the sequential program. Due to the fact that the feature matrices of images, which can be paralleled, are just part of

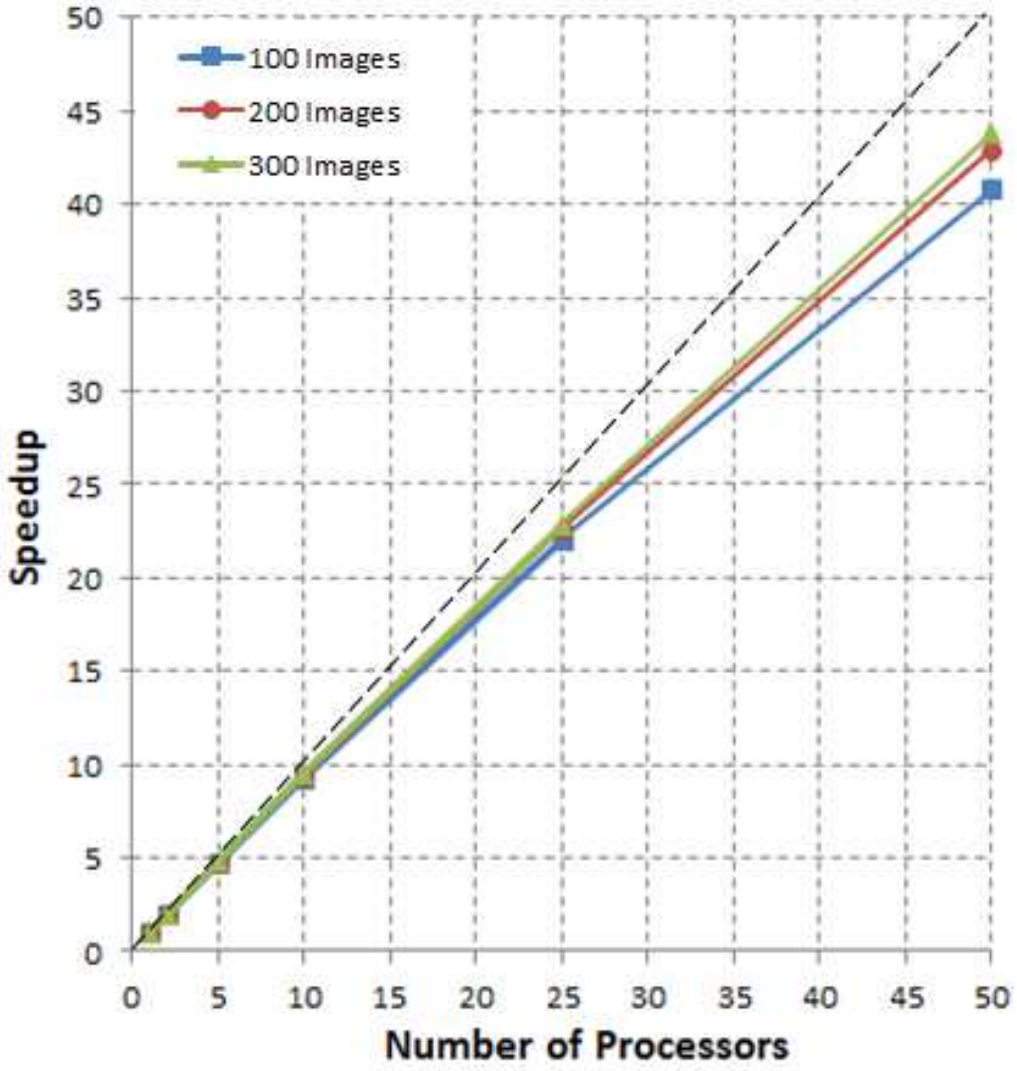

Figure 2.20: Speedup with different number of processors

the participation in the program variables, the decreasing speed of memory usage on each processor slows down gradually with the number of processors.

Table 2.6: Memory usage (in MB) with different number of processors

Processors	100 images	200 images	300 images
p=1	195	259	323
p=2	160	204	236
p=5	139	171	183
p=10	124	159	176
p=25	109	147	167
p=50	82	132	159

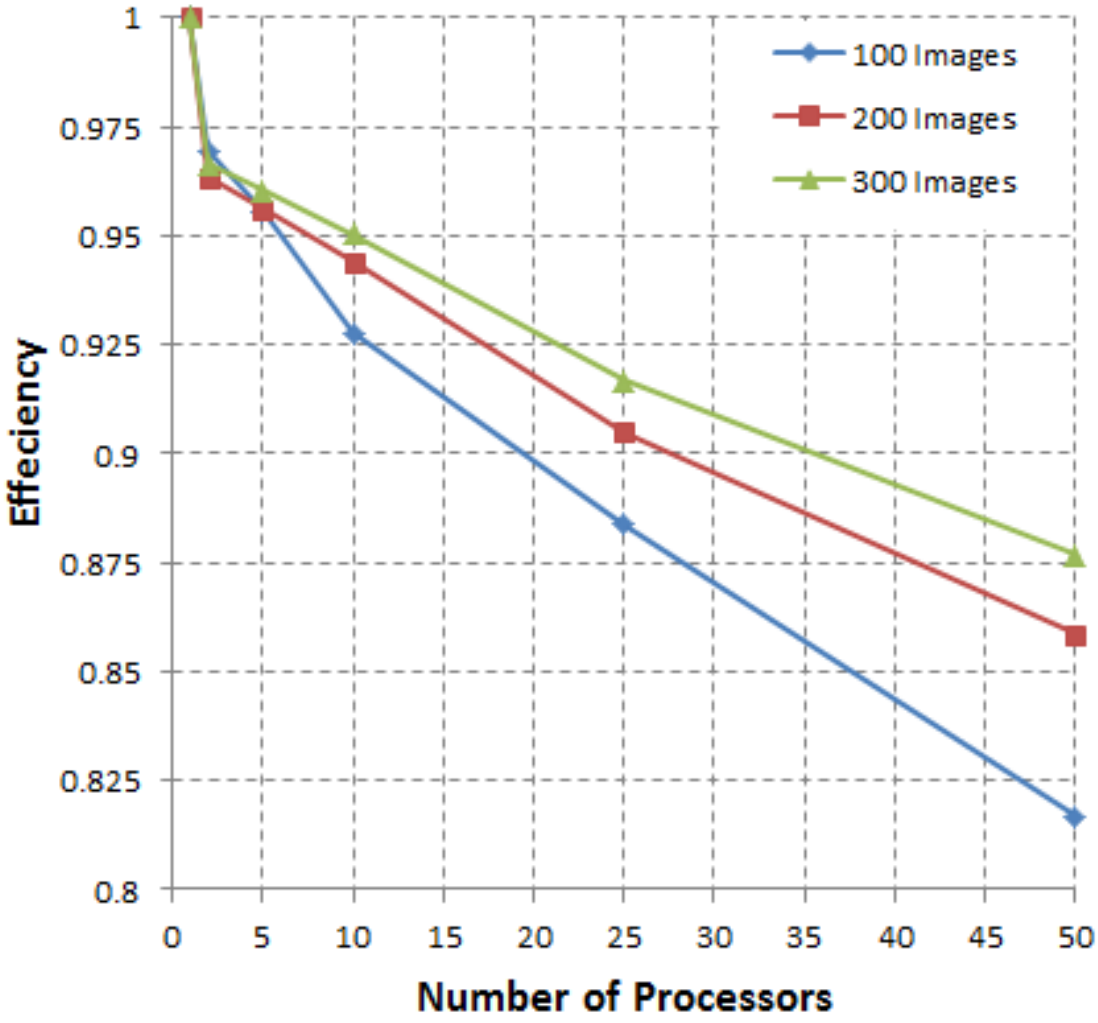

Figure 2.21: Efficiency with different number of processors

Table 2.7: Memory usage rate with different number of processors

Processors	100 images	200 images	300 images
p=1	100.0%	100.0%	100.0%
p=2	81.7%	78.6%	73.0%
p=5	71.3%	65.7%	56.7%
p=10	63.5%	61.3%	54.4%
p=25	55.7%	56.7%	51.6%
p=50	42.0%	50.9%	49.3%

Summary for NGM parallelization

A parallel algorithm for NGM face recognition was implemented to reduce CPU time and memory usages. To achieve the goal, the training process was firstly distributed onto multiple processors. Also, the probe images were then allotted equally. Finally, all the

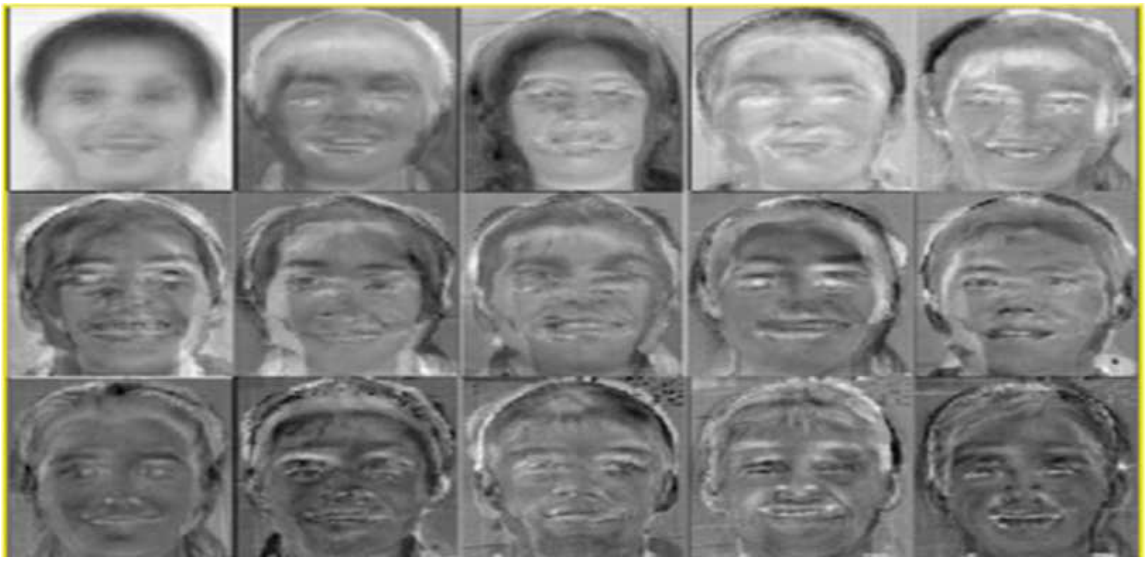

Figure 2.22: General expression of holistic methods

processors carried out the recognition process simultaneously by communicating with each other. Due to well-balanced workload on each processor, the speedup increases with the number of processors and thus the efficiency was maintained excellently, even if the number of processors reached to 50. Furthermore, memory usage on each processor also decreased as the number of processors increased. To summary, the parallel algorithm brings more practicality for NGM face recognition.

2.4 Comparison of Holistic Method and Feature-based Method

For holistic methods, the main advantage is that they solve recognition from the perspectives of global features, rather than concentrating on only limited regions or points of interest, so the whole information of an image is kept [27]. However, the advantage can also be a disadvantage. The global features of poses, illuminations, expressions et al. can be harmful for recognition. On the other hand, since they do not focus on local details, the computational cost is relatively low. From the point of methodology, holistic method recognizes human faces in a way of coarse searching, which focuses on overall features or rather general expression, as illustrated in Figure 2.22. To sum up, holistic methods make recognition decisions fast, accurate but sometimes may be bold.

For feature-based methods, the primary advantage is that they are relatively robust to distortions such as variations of expression, pose, illuminations, and so on, since the feature points are extracted from local details [27]. Similar with holistic methods, the advantage may go to the other side. Due to the fact that extracting local features is time-consuming,

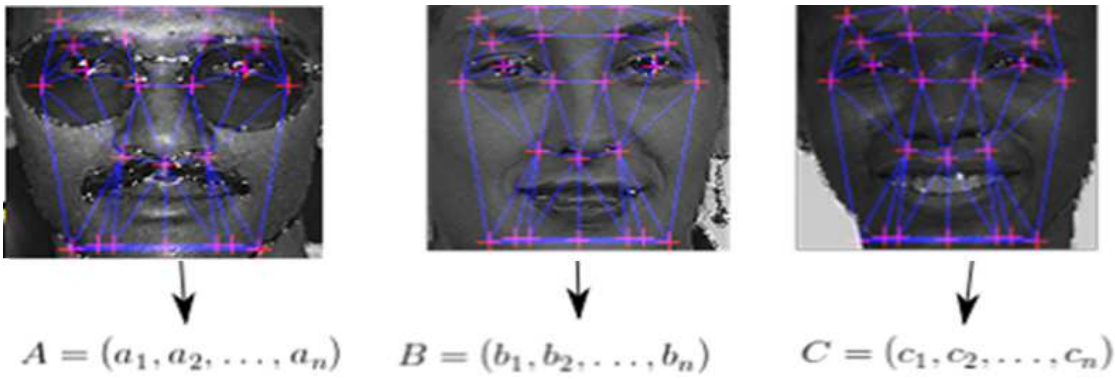

Figure 2.23: Facial details extracted by feature-based methods

feature-based methods execute slower than holistic methods, and sometimes the difference can be in several orders. From the perspective of methodology, feature-based methods recognizes human faces in a way of fine searching, which emphases facial details such nose, eyes, mouth, and so on, as demonstrated in Figure 2.23. Therefore, recognition decisions made by feature-based methods are more cautious but also slower.

Chapter 3

MULTI-STAGE MATCHING ALGORITHMS

In this chapter, a set of multi-stage matching algorithms are proposed to solve the recognition degradation problem. Three algorithms—n-ary elimination, divide and conquer, and two-stage hybrid—are presented respectively.

3.1 Recognition Degradation Problem

3.1.1 Two Circumstances

As mentioned in Section 1.3, face recognition has a recognition degradation problem under two circumstances: "single sample per person" and large gallery size. "Single sample per person" refers to a condition that one probe image corresponds to only one training image [5, 52]. For many reported face recognition algorithms, multiple training samples referencing to one probe image is a prerequisite so as that the different poses, lighting conditions, and other unexpected situations of testing images can be well dealt with. Otherwise, their recognition performances gradually degrade, and even fail to work effectively due to the extremely limited information provided by a single sample [48].

On the other hand, the large gallery size also brings difficulties to face recognition [34]. For holistic methods, say PCA, the recognition result is decided by the training image that has the shortest projection distance with the target probe image in the "eigenfaces" space. Since the relative projection of any one training image and the probe image is certain in the sub-faces space, larger gallery size means more possibilities for an unexpected training image to have a shorter distance than the corresponding training image has. If in that case, a false recognition happens. For feature-based methods, say NGM, the matching process is to match the feature points of the training images and the target probe image. As the number of training images increases, for one single feature point of the probe image, it is more possible to be matched to a wrong training image. If the corresponding training image fails to get enough matched feature points, a false recognition also occurs. It can be concluded that a large gallery size brings matching dispersion and degrades the recognition accuracy. In the next section, the matching dispersion of NGM due to a large gallery size is presented.

Table 3.1: Matching Dispersion of a probe image indexed 10

Gallery size=100		Gallery size=200		Gallery size=300	
Top 3 Index	Matched	Top 3 Index	Matched	Top 3 Index	Matched
10	53	10	45	51	39
51	45	51	42	10	37
89	23	89	18	89	15
Correct recognition		Correct recognition		False recognition	

3.1.2 Matching Dispersion of NGM

The matching process of NGM is to match feature points of training images and probe images. As the number of training images increases, for one single feature point of a probe image, it becomes more possible to be matched to a wrong training image. Figure 3.1 intuitively demonstrates a scenario that a probe image matches with two groups of training images. The first group has 3 training images while the second group has 3 other ones that make it 6. For the sake of demonstration, assuming that all the images have 20 feature points extracted. The feature points of the probe image are marked by red, and the feature points from the training images are also marked by red if they are matched, or they are shown with other colors. When matching with the first group, the corresponding training image has 10 feature points matched to the probe image and is correctly recognized as the result. However, when the gallery size goes to 6 in the second group, 3 new training images get 3 feature points matched that are previously matched to the corresponding training image in the first group. As a result, with 7 matched feature points, the corresponding training image fails to be recognized as the result, since another training image attains 8 matched feature points.

An experimental case that a false recognition happens due to gallery size increasing is illustrated in Table 3.1. A probe image indexed 10 has about 382 feature points, which are matched to three galleries with sizes of 100, 200, and 300 respectively. Top 3 indexes that have the most matched feature points are listed. As the gallery size increases, the numbers of matched feature points of the top 3 training images keep decreasing due to matching dispersion. When the gallery size goes to 300, a false recognition occurs when the corresponding training image indexed 10 fails to get the most matched feature points.

3.1.3 Accuracy Degradation

To intuitively demonstrate the performance degradation along with the gallery size, we tested the recognition accuracies of PCA, LDA, and NGM respectively, with sizes of gallery ranging from 100 to 1000 with an increment of 100. And each probe image references to

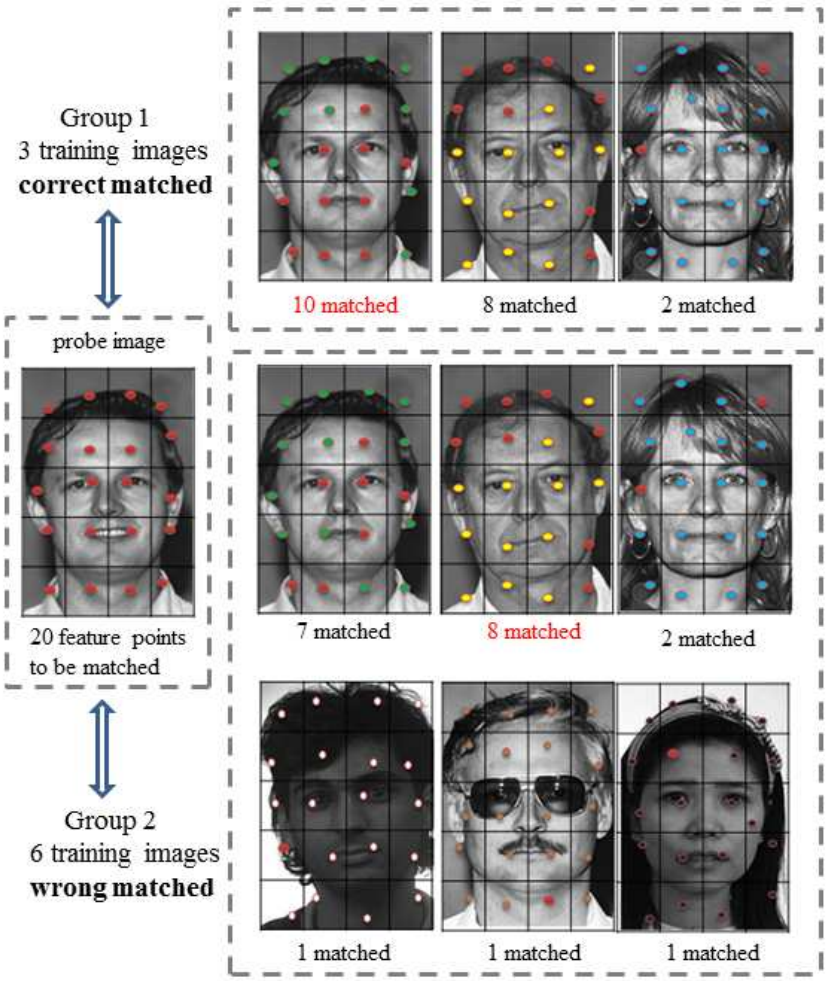

Figure 3.1: Demonstration of matching dispersion of NGM

only one training image. The recognition accuracies of PCA, LDA, and NGM along with the gallery sizes are plotted in Figure 3.2, from where obviously can be seen that the accuracies of the three methods continuously decline with the gallery size increasing.

3.2 The Core of Multi-stage Matching Strategy

The core of computer vision is to teach computers to view the world as human beings do. The matching process of finding the corresponding image from a bunch of training images can be viewed to deciding the championship in a sports tournament, which normally requires several rounds, such as preliminary contest, intermediary heat, semi-final, and final. Each training image is a competitor, and the one that matched as the recognition result is

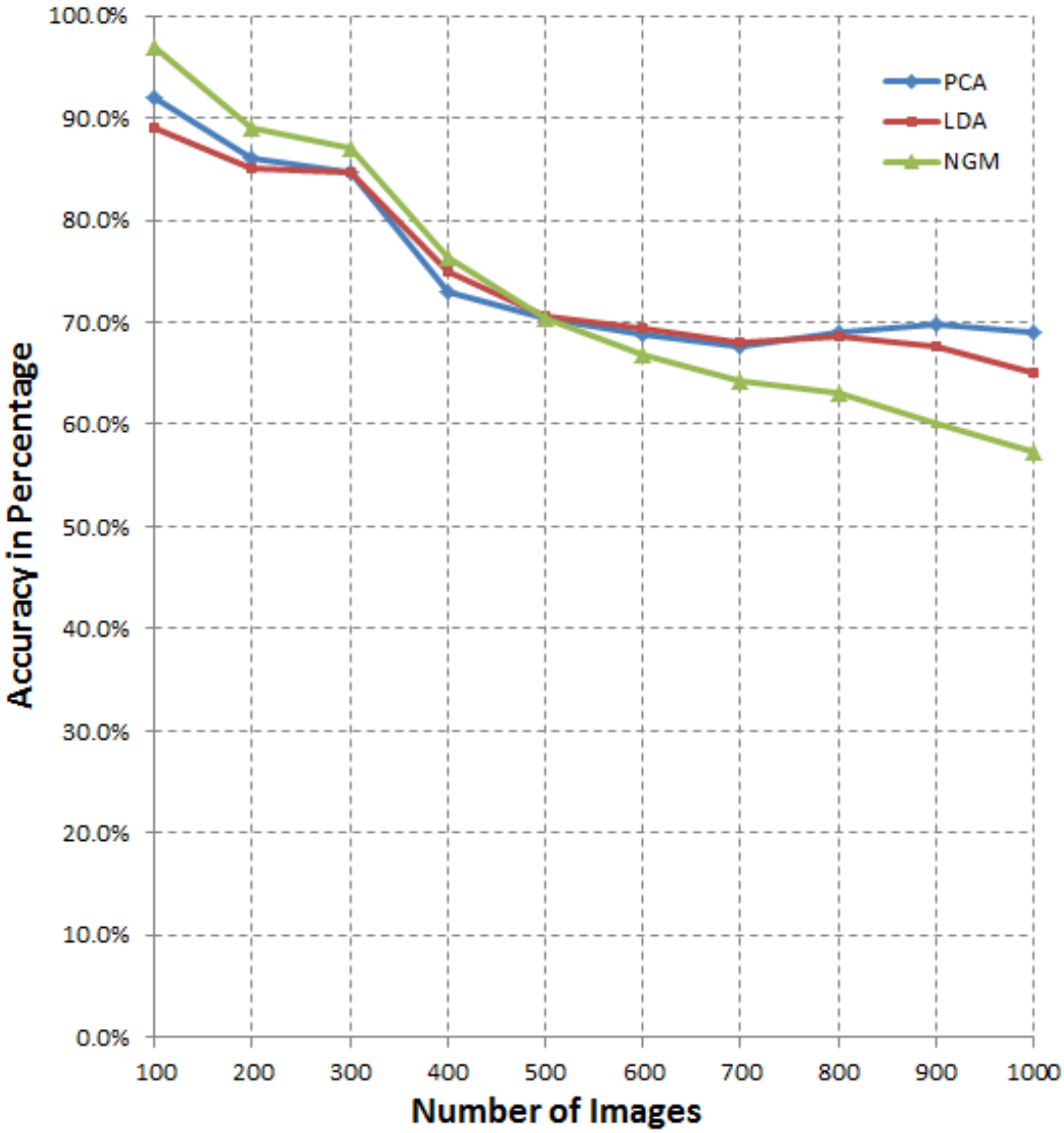

Figure 3.2: Recognition degradation of face recognition

the champion, which is required to pass through multiple steps. Each step of matching selects a small portion of the competitors that have the best similarities with the probe image as the new candidates for the next step, and eliminates the rest. Once the number of remaining candidates is small enough, the final round of matching executes and produces the recognition result. It has been found in practice that when the number of training images is no more than 15 [9], the matching dispersion of feature points becomes trivial and thus the multi-stage searching process should stop.

As illustrated in Figure 3.3, the multi-stage matching strategy has two properties as following:

Figure 3.3: The core of multi-stage matching strategy

1. Narrowing searching range: the recognition decision should be generated step-by-step. Each step picks some best candidates and removes others.

2. Making the recognition decision in a small range: the recognition decision is made when the number of remaining candidates is small enough.

3.3 N-ary Elimination Matching Algorithm

N-ary elimination multi-stage matching algorithm is introduced to the matching process of NGM presented in Section 2.3. It aims to solve the recognition degradation problem due to the large gallery sizes. The core part of this work has been accepted in [10].

3.3.1 Implementation

Recognizing a probe image from a bunch of training images can be considered as deciding the champion in a sports tournament, which has preliminary, intermediary, and final, and each stage of the tournament eliminates most of the candidates. When matching a probe image, the similarities of the training images are calculated and ranked in order, and the first $1/n$ of them are picked as the new candidates for the next step, while the others are eliminated. With n-ary convergence rate, the behavior of picking and eliminating repeats, until the number of remaining training images is not more than a certain small number. And the recognition result is generated from the remaining candidates. N-ary elimination accomplishes the multi-stage matching strategy from the perspective of global, since its candidate selection is based on the similarity ranking of all involved training candidates. In

our work, the certain small number is 15, an empirical number mentioned in Section 3.2. The procedure of recognizing a probe image from T training images by NGM with n-ary elimination matching algorithm is illustrated in Figure 3.4.

Under some circumstances, especially when running with a large gallery size, the original NGM makes a wrong recognition, but actually the corresponding training image has a significant similarity, even though not the greatest one. The cause is the matching dispersion of feature points that discussed in Section 3.1.2. N-ary elimination removes those disturbance from dissimilar training images, and provides more opportunities for the corresponding training image to stand out. Similarly in a sports tournament, the champion may not be the best one in preliminary game, but he or she defeats the other competitors in the final. On the other hand, if the corresponding training image has already obtained the greatest similarity in the first few rounds of matching, removing other weak candidates does no harm to the final result due to the fact that the smaller the gallery size, the easier the recognition.

3.3.2 A Typical Preliminary Result

Table 3.2 demonstrates a typical case that a corresponding training image does not rank the best in similarity in the first round of matching but finally is correctly recognized as the result with binary elimination ($n = 2$) matching algorithm. A probe image indexed 15 is to be recognized from 100 training images. With binary elimination, there are four rounds of matching with 100, 50, 25, and 13 candidates respectively. From Table 3.2, we can see that the corresponding training image indexed 15 does not rank the first position in the first and second round of matching, but as the number of candidates reduces in the next two rounds, it gradually stands out as the best similar candidate. It can be concluded that it is the decreasing number of candidates that has less matching dispersion and produces a better accuracy.

Table 3.2: A typical case of n-ary elimination ($n = 2$) matching algorithm

Matching Round	Candidates Amount	Probe Index	Top 10 Similarity Ranking Indexes
First	100	15	13 21 15 34 25 55 29 91 37 35
Second	50	15	21 15 13 7 34 25 37 2 55 67
Third	25	15	15 21 13 34 7 67 29 2 4 25
Final	13	15	15 21 37 4 7 34 2 13 41 25

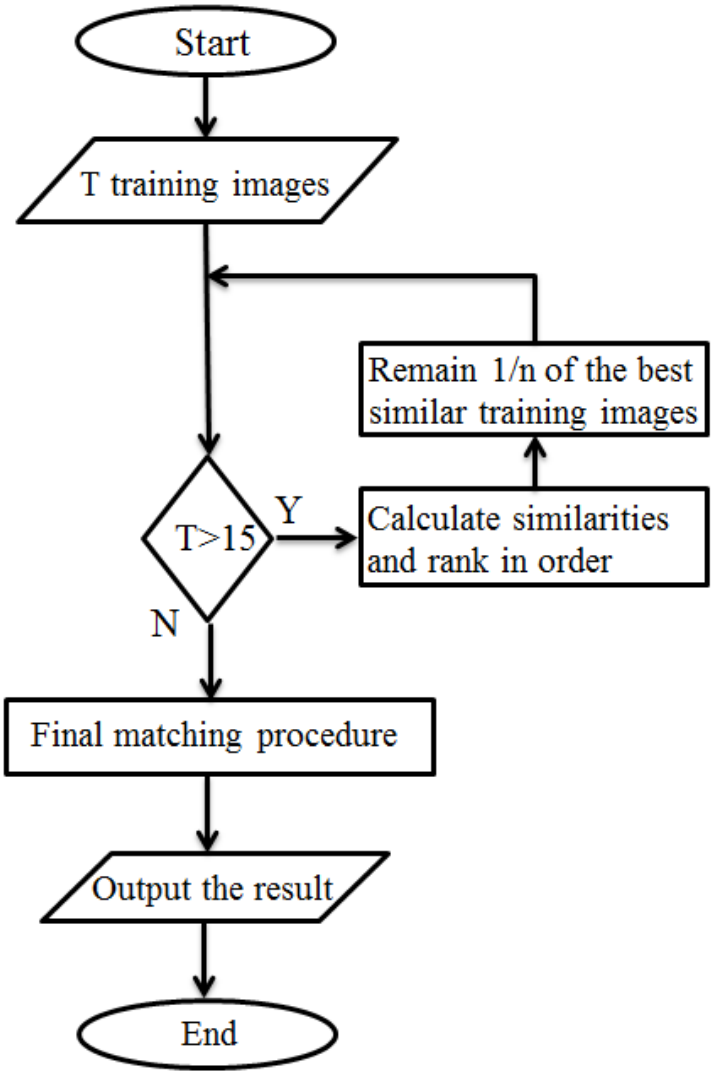

Figure 3.4: NGM with n-ary elimination matching algorithm

3.4 Divide and Conquer Matching Algorithm

In this section, divide and conquer matching algorithm is proposed for NGM presented in Section 2.3. In computer algorithm, divide and conquer breaks a big problem into several sub-problems of the same type, solves them individually and combines them until they are simple enough to be solved directly. The core part of this work has been accepted in [10].

3.4.1 Implementation

Different with n-ary elimination that solves from global, divide and conquer algorithm breaks a big problem into several sub-problems of the same type, solves them individually and combines them until they are simple enough to be solved directly. When matching a probe

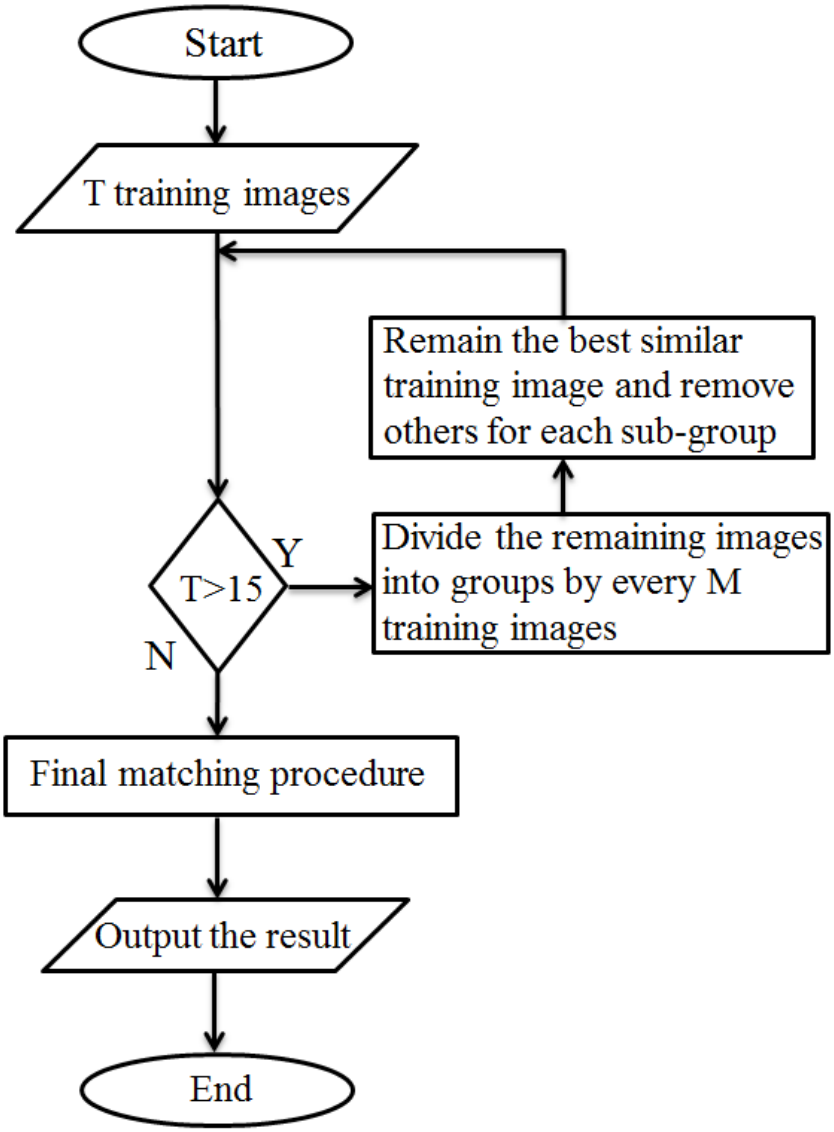

Figure 3.5: NGM with divide and conquer matching algorithm

image, training images are divided by a grouping number M. The matching process executes in each group and selects the best similar training image of each group as a candidate for the next step. The divide and conquer behavior repeats with the remaining candidates for the next steps, until the number of the remaining candidates is not greater than a certain small number to generate the final recognition result. In our work, the certain small number is 15, the same with that in Section 3.3. The procedure of recognizing a probe image from T training images by NGM with divide and conquer matching algorithm is illustrated in Figure 3.5.

Figure 3.6 demonstrates a case of recognizing a probe image indexed 1 from 100 training

images indexed from 1 to 100 with the grouping number equal to 2 ($M = 2$). With binary convergence, the number of candidates decreases half by half until it goes to 13, which is smaller than the setting certain number 15. Hence, the final matching executes after 3 stages of preliminary matching. The candidate selection of divide and conquer is based on the similarity comparison of each sub-group, so it accomplishes the multi-stage matching strategy from the perspective of local.

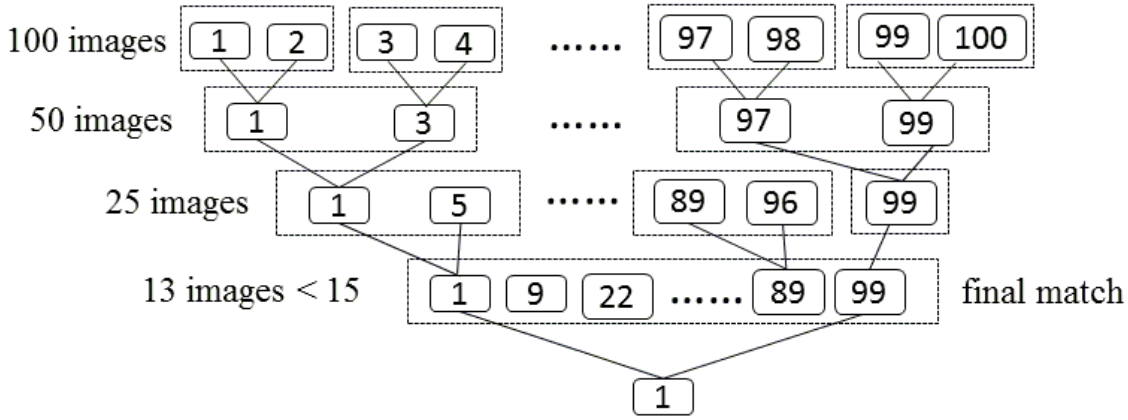

Figure 3.6: The demonstration of matching with divide and conquer ($M = 2$)

3.4.2 A Typical Preliminary Result

Table 3.3 illustrates a case that the corresponding training image fails to be recognized in the first round of matching but becomes the final recognition result in the next steps. A probe image indexed 45 is to be recognized from a gallery with 100 training images. With the original NGM matching, the recognition result is the training image indexed 57, which is wrong. With divide and conquer matching algorithm by setting grouping number 5 (i.e. $M = 5$), there are three rounds of matching. The number of remaining candidates are 100, 20, and 4 respectively. The training image indexed 45 succeeds to keep the qualification of a candidate and stands out as the recognition result out of the 4 candidates in the final round of matching. It can be also deduced that it is the smaller amount of training candidates that reduces the matching dispersion and helps a correct recognition.

3.5 Two-stage Hybrid Matching Algorithm

In this section, two-stage hybrid matching algorithm is presented to solve the recognition degradation problem, and this part of work has been published in [9]. Two-stage hybrid refers to hybridizing a holistic method and a feature-based method. By analyzing the

Table 3.3: Gallery size=100, grouping number M=5

Round	Candidates	Probe	Selected Candidates	Result
First	100	45	2 10 14 19 22 29 31 37 45 48 52 57 61 67 71 6 82 87 93 98	Wrong
Second	20	45	2 45 57 93	
Final	4	45	45	Correct

degradation features of the two types of methods, it combines the superiorities of the both yet avoids the inferiorities of either. Two implementations, PCA with NGM and LDA with NGM, are served to verify its validity.

3.5.1 Two Notable Observations and Analysis of Accuracy Degradation

In Section 2.4, the behavior mode of holistic methods and feature-based methods has been compared. Now, let's watch the recognition performance of the two types of methods along with different gallery sizes again in Figure 3.7, which has been illustrated in Section 3.1.

As the same as a holistic method, PCA and LDA share a similar behavior of recognition performance along with the size of gallery, which is obviously different with that of NGM. Two observations in Figure 3.7 are worth of mentioning. First, when the gallery size is less than 500, NGM is superior to PCA and LDA in recognition performance. Second, It is also watched that NGM has a shaper decline tendency, which makes it inferior to PCA and LDA with a gallery size larger than 500. From their accuracy curves, NGM behaviors more like a linear function, while PCA and LDA act approximately as an inverse proportion function.

Rather than a rigorous mathematical analysis, we prefer an intuitive but logical explanation as follow. For NGM, a correct recognition requires a corresponding training image match enough feature points with the probe image. However, the amount of feature points of an image is relatively certain and limited. As the gallery size increases linearly, the distribution of matched feature points also diverse in a linear trend, and a false recognition may happen due to matching dispersion that presented in Section 3.1.2. On the other hand, for PCA and LDA, more training images in the gallery indicates more eigenvectors will be generated, but the number of employed eigenvectors still stay relatively certain. From another perspective, assuming the relative projection positions of a pair of probe image and training image are certain in the “eigenfaces" space, those extra irrelevant training images are more possible to project somewhere far from the probe image, other than close to it.

Based on the observations and analysis above, two statements that can be claimed:

1. When the gallery size is small, feature-based method may be a better option due to its concentration on details and robustness.

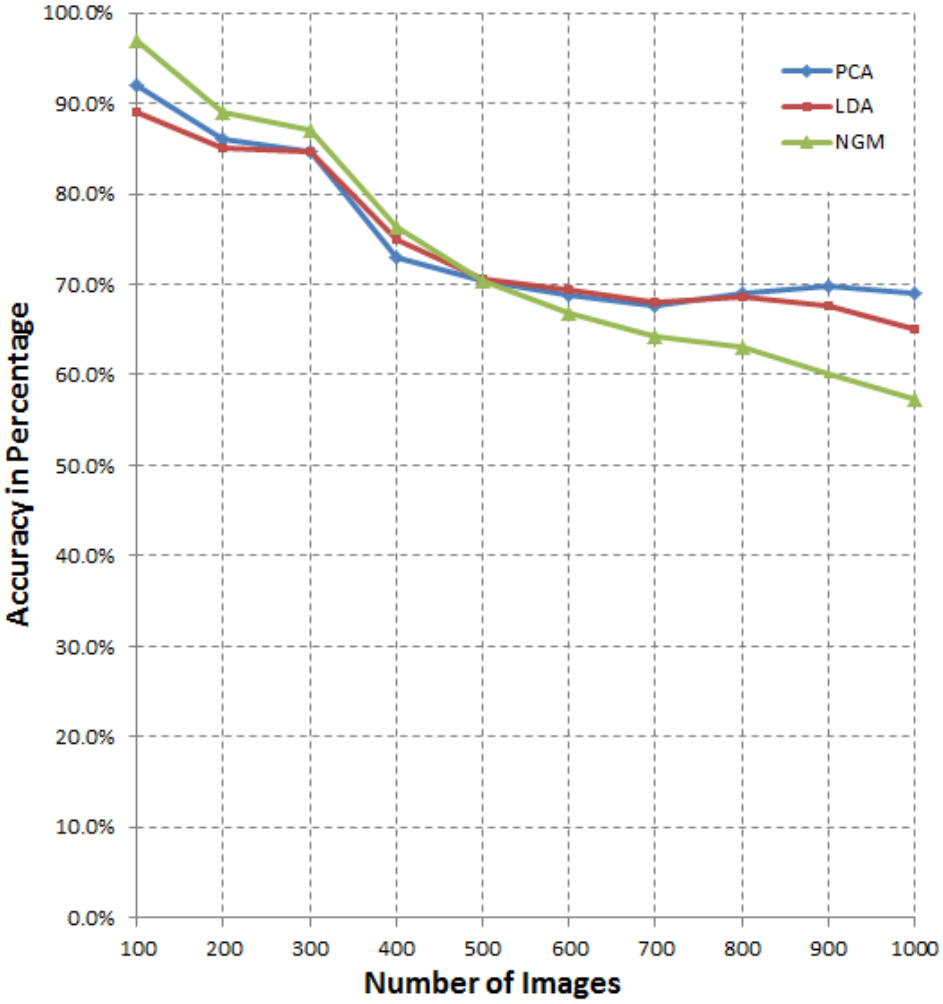

Figure 3.7: Recognition degradation comparison of two types of methods

2. When the gallery size becomes large, holistic method can be the first choice due to its slower degradation.

3.5.2 Implementation

Based on the statement above, the two-stage hybrid matching algorithm that combines the superiorities of holistic method and feature-based method yet avoids the inferiorities of either is proposed to deal with the recognition degradation problem. Rather than finding the recognition result directly in traditional methods, two-stage hybrid produces the recognition result in a two-stage procedure. When matching a probe image, a holistic method is firstly utilized to preprocess the gallery and choose a small amount of candidates that are most similar to the probe image. Second, a feature-based method is used to produce the final result from the selected candidates. The two-stage hybrid defers to a principle that holistic

methods preprocess the whole gallery and feature-based methods make the final decision. In our work, two implementations, PCA with NGM and LDA with NGM, are presented in Figure 3.8.

The strategy of the two-stage hybrid matching algorithm can be compared to a scenario that a detective cracks a criminal case. To find out the real criminal, he needs to range a small amount of potential suspects from all possible people for first. And then, each suspects should be interrogated detailedly one by one until the real criminal is revealed. The preprocess done by a holistic method corresponds to ranging suspects, while the final recognition by a feature-based method resembles interrogating suspects. In the preprocessing stage, it is crucial to decide an appropriate amount of candidates, since it should be large enough for a holistic method to cover the corresponding training image, as well as small enough for a feature-based method to work efficiently. In this paper, we set the number to 15, an empirical value that balances the holistic preprocessing and feature-based final procedure [9].

3.5.3 Some Rectified Examples by Two-stage Hybrid Matching Algorithm

Table 3.4 demonstrates some typical cases that the probe images are not identified correctly by either NGM or PCA individually, but are finally recognized by the two-stage hybrid method among 1000 training images. For instance, the probe image 49 is matched to 579 by PCA and 120 by NGM. From the perspective of traditional methods, both the recognition results are wrong. But the training image 49 is successfully selected as one of the candidates that preprocessed by the holistic method PCA. With one more opportunity, the training image 49 is matched finally by the feature-based method NGM. It is the same argument for the other two examples.

Table 3.4: A rectified examples by the two-stage hybrid matching algorithm

Probe Index	Original NGM	Candidates by PCA	Final by NGM
49	120	579 49 362 408 292 167 808 731 509 868 711 269 620 657 588	49
61	46	313 617 679 61 73 406 333 279 74 278 378 869 305 129 345	61
63	70	466 334 278 586 299 5 871 598 296 63 294 578 330 896 875	63

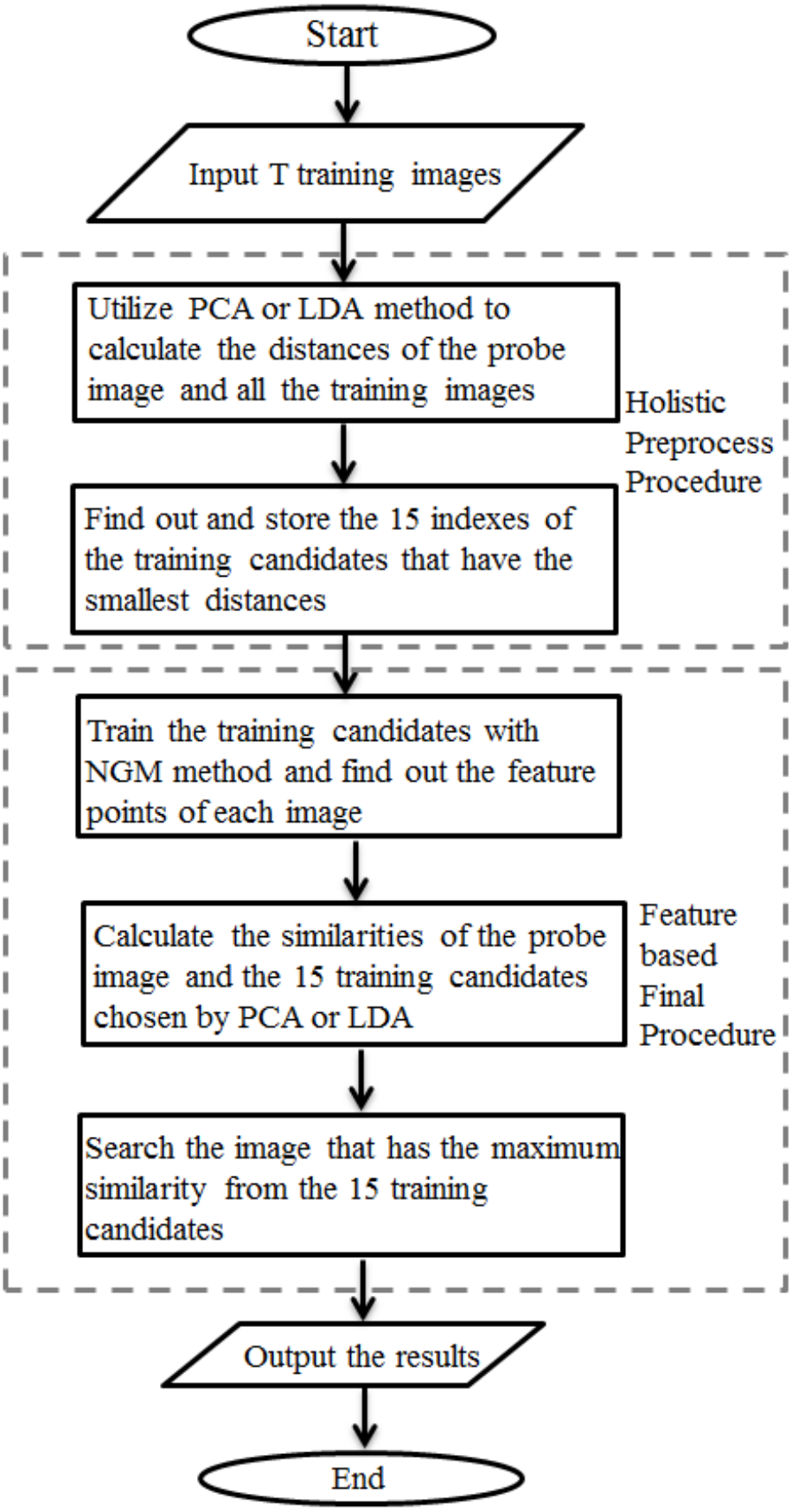

Figure 3.8: The two-stage hybrid matching algorithm work flow

3.6 Summary of the Three Multi-stage Matching Algorithms

From the angle of execution procedure, the three multi-stage matching algorithms share the common property that decides the recognition result step-by-step. Form the perspective of computer research, they accomplish the core of computer vision that views the world like human beings. N-ary elimination imitates the process of deciding the champion in sports tournament games. Divide and conquer simulates a promotion system. Two-stage hybrid copies the way of criminal investigation.

On the other hand, they also have different emphasis in methodology. N-ary elimination solves recognition from global, while divide and conquer searches the recognition result from local. Finally, two-stage hybrid utilizes the complementary advantages of a holistic method and a feature-based method.

Chapter 4

EXPERIMENTS

In this chapter, the performances of three multi-stage matching algorithms—n-ary elimination, divide and conquer, and two-stage hybrid—are comprehensively demonstrated. The primary goal of multi-stage matching algorithms is to improve recognition accuracy, meanwhile the computational costs should not greatly increase. Therefore, the evaluation items are mainly concentrated on the recognition accuracy and the computational costs.

4.1 Experiment Configurations

All the tested methods, including the original methods and the methods with multi-stage matching algorithms, are executed on a platform with the same configurations, including hardware, software, and image database.

4.1.1 Hardware and Software

For hardware, all kinds of the implemented codes run on an HP desktop computer with an Intel Core i7-2600 Sandy Bridge 3.4 GHz, 8 GB of memory.

For software, the implementations of PCA and NGM, with and without multi-stage matching algorithms, are developed by Microsoft Visual Studio 2012 C++ language, while LDA with and without multi-stage matching algorithms is implemented by Matlab 2012. The operating system is 64-bit Windows 7.

4.1.2 Database and Image Normalization

The mainstream database for face recognition includes FERET, AR, ATT, Yale, MIT, NIST, etc. However, it is only FERET that is able to provide a large amount of individual faces for the study of degradation problem caused by increasing gallery sizes. Thus, FERET [42] that can be tested up to 1000 individuals is selected as the only database for experiments. To create a large size of gallery, 1000 individuals are chosen randomly with different race, gender, age, expression, illumination, etc. Each individual corresponds to an fa image as a training image and an fb image as a probe image. Here, fa and fb means front facial images. In other words, there are 1000 training images and 1000 probe images one-to-one

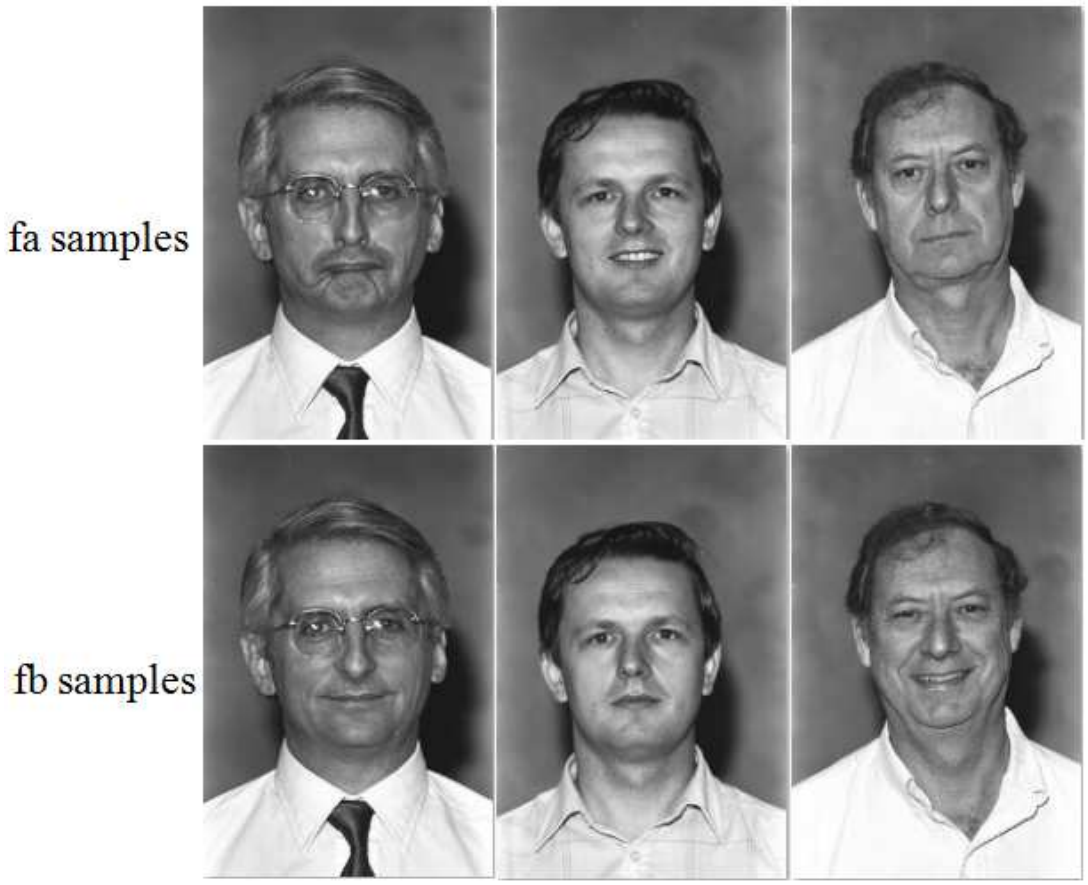

Figure 4.1: Original fa and fb sample images

corresponding each other. Figure 4.1 illustrates the original pairs of fa and fb sample images, which refer to the same faces but different taken times, expressions, illumination, etc.

Before the execution of the experiments, the images should be normalized for first by selecting the facial part and removing the other irrelevant information. For instance, without normalization that restricts a human face in a relatively certain area, the average face of PCA would be messed up. On the other hand, removing irrelevant part of image reduces computation cost and disturbance for recognition. The normalization process that implemented with Matlab code is listed as follows:

1. Locate the two positions of eyes and line with them.
2. Line the perpendicular bisector.
3. Extend the two lines horizontally and vertically to certain scale to cover head and the neck.

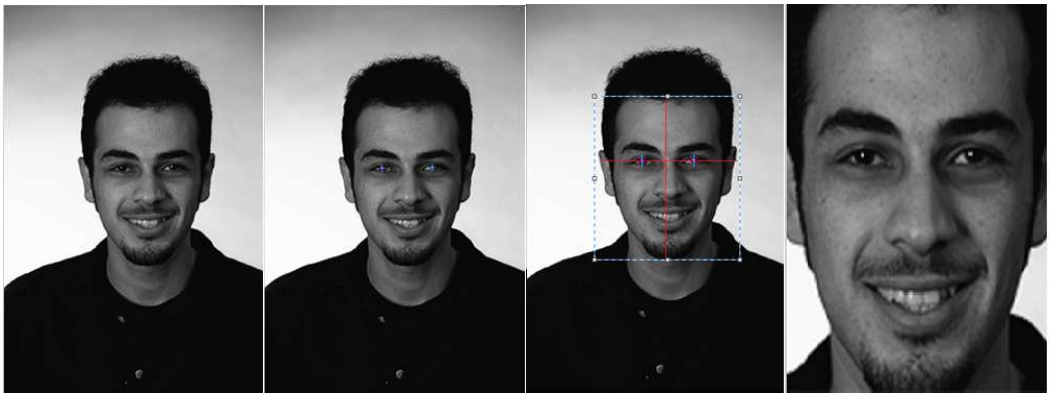

Figure 4.2: Normalization process

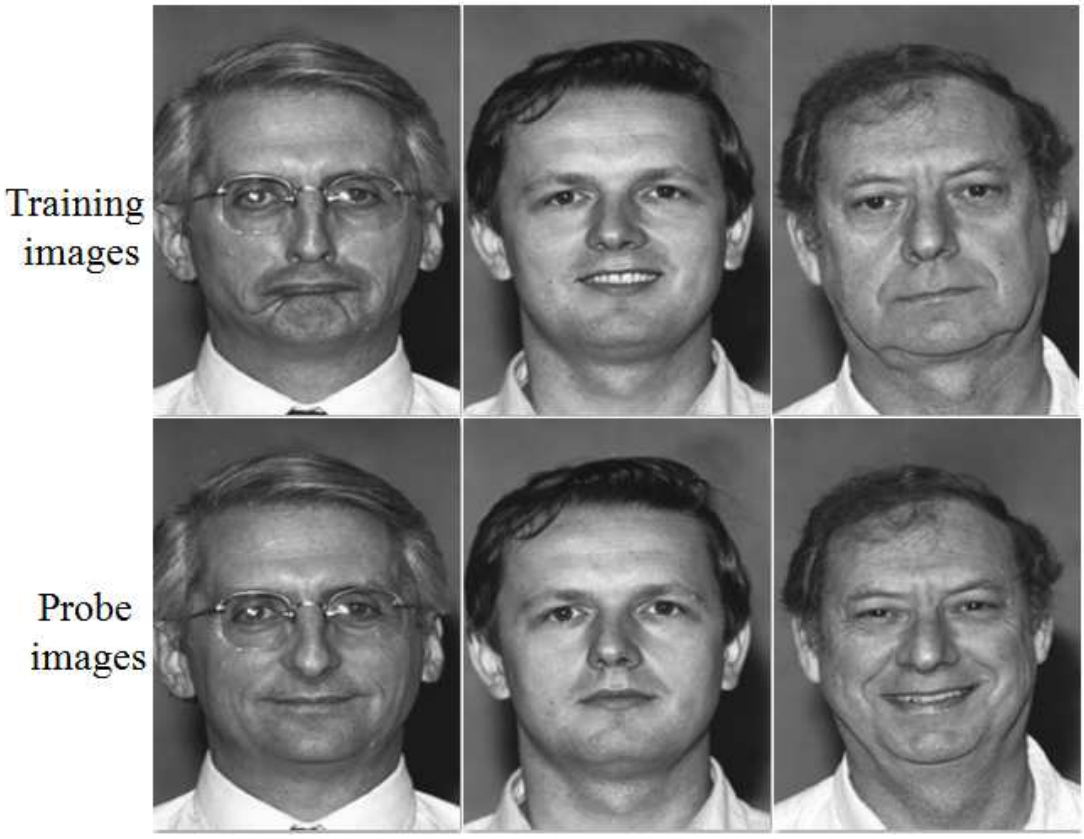

Figure 4.3: Sample normalized training and probing images

Figure 4.2 demonstrates the process of a raw image to be normalized step-by-step. For the size, All images are normalized to 311×232 pixels. The raw images showed in Figure 4.1 are normalized as illustrated in Figure 4.3.

4.2 Experimental Results

The original methods and the methods with multi-stage matching algorithms, n-ary elimination, and divide and conquer, and two-stage hybrid, are tested for recognition accuracy and computational costs with galleries of varying sizes ranging from 100 to 1000, with an increment of 100. The experimental results show that multi-stage matching algorithms provide a remarkably improved recognition performance, meanwhile they do not bring significantly extra computational times.

4.2.1 N-ary Elimination

The experimental results next indicate that the n-ary elimination matching algorithm has an advantage of higher recognition accuracy. Moreover, the improved accuracy also increases steadily with the gallery size increasing. Additionally, it dose not bring significantly extra computational costs.

Recognition Accuracy of N-ary Elimination

By setting n with 2, 4, 8, 16, 32, and 64, the recognition accuracies of the original NGM and NGM with n-ary elimination are tested on galleries of varying sizes ranging from 100 to 1000 with an increment of 100. All the recognition accuracies with different settings are collected in Table 4.1 and plotted in Figure 4.4. Also, the differences between the original NGM and that with n-ary elimination are calculated in Table 4.2.

From the two tables and one figure, three facts can be concluded. First, all the settings of n-ary elimination have an approximate performance, as the performance curves of n-ary elimination in Figure 4.4 almost stick together. Second, the NGM with n-ary elimination outperformances the original NGM with all gallery sizes, as the bunch of lines that represents NGM with n-ary elimination are always above the blue line that shows the original NGM in Figure 4.4. Finally, the improved level also increases steadily with the gallery size, as the gap between the bunch lines of n-ary elimination and the blue line that stands for the increment enlarges with the number of images in Figure 4.4. For instance, with binary elimination ($n = 2$), when the gallery size reaches to 1000 images, the recognition accuracy of NGM with n-ary elimination still maintains at a high level of 77.0%, compare with 57.3% of the original NGM, with a improvement of 19.7%. In other words, for 1000 cases of recognition, there are 197 cases that are rectified from false recognition by the binary elimination matching algorithm.

Table 4.1: Recognition accuracy of n-ary elimination

Gallery	Original	n=2	n=4	n=8	n=16	n=32	n=64
100	97.0%	98.0%	97.0%	97.0%	97.0%	97.0%	97.0%
200	89.0%	92.0%	92.0%	94.0%	93.0%	92.5%	92.0%
300	87.0%	92.7%	91.7%	92.7%	92.0%	92.0%	91.0%
400	76.3%	85.0%	86.3%	86.5%	84.3%	85.0%	83.5%
500	70.4%	83.4%	83.8%	83.6%	80.8%	81.6%	79.6%
600	66.8%	81.5%	81.5%	82.3%	81.0%	79.7%	79.2%
700	64.3%	80.3%	79.9%	79.9%	78.4%	76.9%	77.0%
800	63.1%	77.9%	79.1%	78.4%	78.9%	76.8%	75.5%
900	60.1%	76.8%	76.8%	77.4%	77.6%	75.4%	74.6%
1000	57.3%	77.0%	77.0%	76.8%	75.9%	73.8%	72.7%

Table 4.2: Recognition accuracy differences of n-ary elimination

Gallery	n=2	n=4	n=8	n=16	n=32	n=64
100	1.0%	0.0%	0.0%	0.0%	0.0%	0.0%
200	3.0%	3.0%	5.0%	4.0%	3.5%	3.0%
300	5.7%	4.7%	5.7%	5.0%	5.0%	4.0%
400	8.8%	10.1%	10.3%	8.1%	8.8%	7.3%
500	13.0%	13.4%	13.2%	10.4%	11.2%	9.2%
600	14.7%	14.7%	15.5%	14.2%	12.9%	12.4%
700	16.0%	15.6%	15.6%	14.1%	12.6%	12.7%
800	14.8%	16.0%	15.3%	15.8%	13.7%	12.4%
900	16.7%	16.7%	17.3%	17.5%	15.3%	14.5%
1000	19.7%	19.5%	19.5%	18.6%	16.5%	15.4%

Computational Costs of N-ary Elimination

Since n-ary elimination matching algorithm produces multiple rounds of matching process, it is of great significance to observe its extra computational costs. In our work, binary elimination ($n = 2$) that has the slowest convergent rate is chosen as the testing item. The computational costs of original NGM and NGM with binary elimination in seconds are recorded in Table 4.3 and plotted in Figure 4.5. It can be watched that the binary elimination dose not bring considerably extra computational costs, as the two time lines are close with each other. In Table 4.3, the greatest extra computational costs due to binary elimination is not more than 23% of that for traditional NGM.

The little extra computational costs of binary elimination can be explained in two factors. First, the extra computational costs are caused by extra matching processes, yet the costs for matching process is a minor part, compare with that of training process. So the two methods have the same major part for training process. Second, the amount of multiple matching

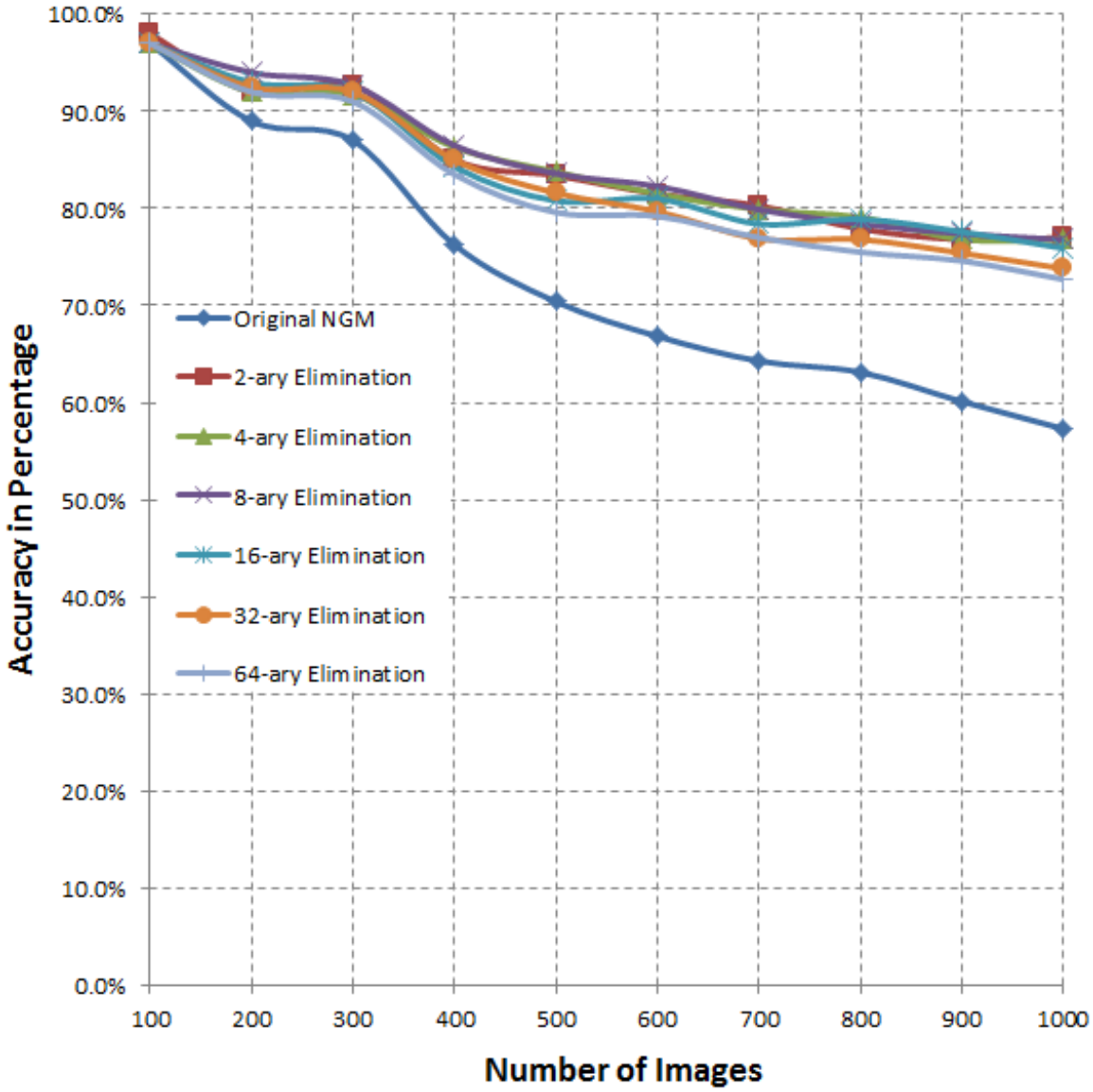

Figure 4.4: NGM accuracy comparison with n-ary elimination

process is not linear but logarithmic relation with the gallery size, which means that the number of extra rounds of matching increases slowly. So other than the linear growth of the original NGM, the computational costs of NGM with binary elimination matching algorithm are sectionally linear. For instance, from 100 to 500 images, 600 to 700 images, and 800 to 1000 images are three individual line segments.

4.2.2 Divide and Conquer Matching Algorithm

The experimental results next shows that the divide and conquer matching algorithm brings a higher recognition accuracy. Moreover, similar with n-ary elimination, the increment

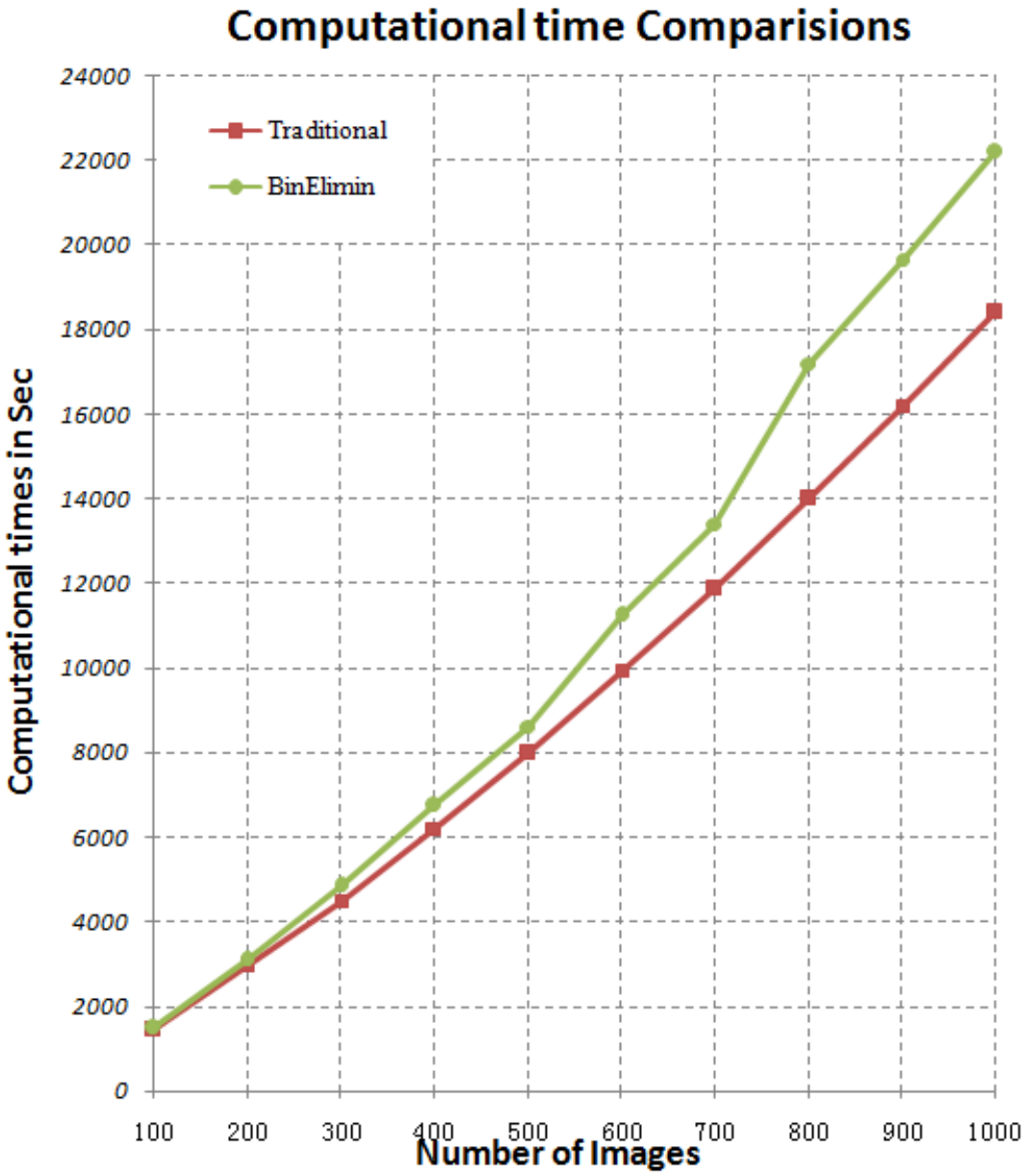

Figure 4.5: Computation times (in seconds) comparison with binary elimination

also has a trend of increasing with the gallery size. Meanwhile, unlike n-ary elimination, it can be theoretically proved that the extra computational costs due to multiple rounds of matching is insignificant.

Recognition Accuracy of Divide and Conquer

By setting the grouping number $M = 2, 4, 8, 16, 32$, and 64, the recognition accuracies of NGM with divide and conquer matching algorithm are tested on galleries of varying sizes

Table 4.3: Computation costs (in seconds) comparison with binary elimination

Gallery Size	Original Costs	Binary Elimination	Extra costs Percentage
100	1492	1477	3.38%
200	2950	3088	4.69%
300	4506	4870	8.07%
400	6206	6737	8.56%
500	7985	8624	8.00%
600	9949	11266	13.24%
700	11866	13408	13.00%
800	14013	17199	22.74%
900	16210	19672	21.36%
1000	18418	22197	20.51%

ranging from 100 to 1000 with an increment of 100. All the recognition accuracies with different grouping numbers are collected in Table 4.4 and plotted in Figure 4.6. Also, the differences between the original NGM and that with divide and conquer are calculated in Table 4.5.

From the two tables and one figure, the conclusions come with the same with n-ary elimination. First, the different settings of divide and conquer have a similar performance, as the accuracy curves of them in Figure 4.6 are close with each other. Second, it can be obviously seen that the performances of NGM with divide and conquer matching algorithm are superior to that of the original NGM, despite some exceptions with small sizes of gallery, as the bunch of lines that represent NGM with divide and conquer are always above the blue line that shows the original NGM in Figure 4.6. Finally, the improved level also increases steadily with the gallery size, as the gap between the bunch lines of divide and conquer and the blue line that stands for the increment enlarges with the number of images in Figure 4.6. For instance, with divide and conquer $M = 32$, when the gallery size reaches to 1000 images, the recognition accuracy of NGM with divide and conquer still maintains at a high level of 77.4%, compare with 57.3% of the original NGM, with a improvement of 20.1%. In other words, for 1000 cases of recognition, there are 201 cases that are rectified from false recognition by the divide and conquer matching algorithm.

Computational Costs of Divide and Conquer

Similar with n-ary elimination matching algorithm, divide and conquer matching algorithm also produces multiple rounds of matching that bring extra computational costs. Other than calculating the extra running times as reported in Section 4.2.1, it can be theoretically proved that the extra computational costs due to divide and conquer is negligible.

Table 4.4: Recognition accuracy of divide and conquer

Gallery	Original	M=2	M=4	M=8	M=16	M=32	M=64
100	97.0%	97.5%	97.0%	95.0%	99.0%	95.0%	93.0%
200	89.0%	89.5%	93.0%	91.0%	94.0%	90.0%	90.5%
300	87.0%	88.7%	90.1%	91.3%	91.3%	90.7%	89.0%
400	76.3%	82.0%	82.0%	84.3%	84.0%	85.0%	80.8%
500	70.4%	76.0%	80.0%	80.6%	81.0%	83.6%	79.2%
600	66.8%	75.8%	78.7%	80.0%	79.0%	82.0%	77.2%
700	64.3%	75.7%	78.1%	78.1%	78.3%	80.9%	76.3%
800	63.1%	75.5%	78.4%	78.6%	78.6%	80.1%	76.4%
900	60.1%	75.4%	77.3%	78.3%	78.0%	80.0%	75.7%
1000	57.3%	71.1%	73.0%	73.5%	75.6%	77.4%	74.3%

Table 4.5: Recognition difference of divide and conquer

Gallery	M=2	M=4	M=8	M=16	M=32	M=64
100	0.5%	0.0%	-2.0%	2.0%	-2.0%	-4.0%
200	0.5%	4.0%	2.0%	5.0%	1.0%	1.5%
300	1.7%	3.1%	4.3%	4.3%	3.7%	2.0%
400	5.8%	5.8%	8.0%	7.8%	8.8%	4.6%
500	5.6%	9.6%	10.2%	10.6%	13.2%	8.8%
600	9.0%	11.9%	13.2%	12.2%	15.2%	10.4%
700	11.4%	13.8%	13.9%	14.0%	16.6%	12.0%
800	12.4%	15.3%	15.5%	15.5%	17.0%	13.3%
900	15.3%	17.2%	18.2%	17.9%	19.9%	15.6%
1000	13.8%	15.7%	16.2%	18.3%	20.1%	17.0%

Generally, a smaller grouping number brings more rounds of matching, since the number of extra rounds are equal to the logarithm of the gallery size to base the grouping number. Assuming the computational costs of matching a pair of images is $O(1)$, the time complexity for matching N probe images and training images is $O(N)$. Choosing $M = 2$ that makes the slowest convergent rate, the total matching costs would be

$$O(N) + O(\frac{N}{2}) + O(\frac{N}{4}) + ... + O(2) + O(1) = O(2N),$$

so the time complexity for matching with divide and conquer is still $O(N)$. Actually, choosing $M = 2$ is the same case with binary elimination matching algorithm. Moreover, the computational costs of matching process in NGM is a small part, compare with that of training process. The extra computational costs of other grouping numbers would be even less than that of $M = 2$, since they have less rounds of matching.

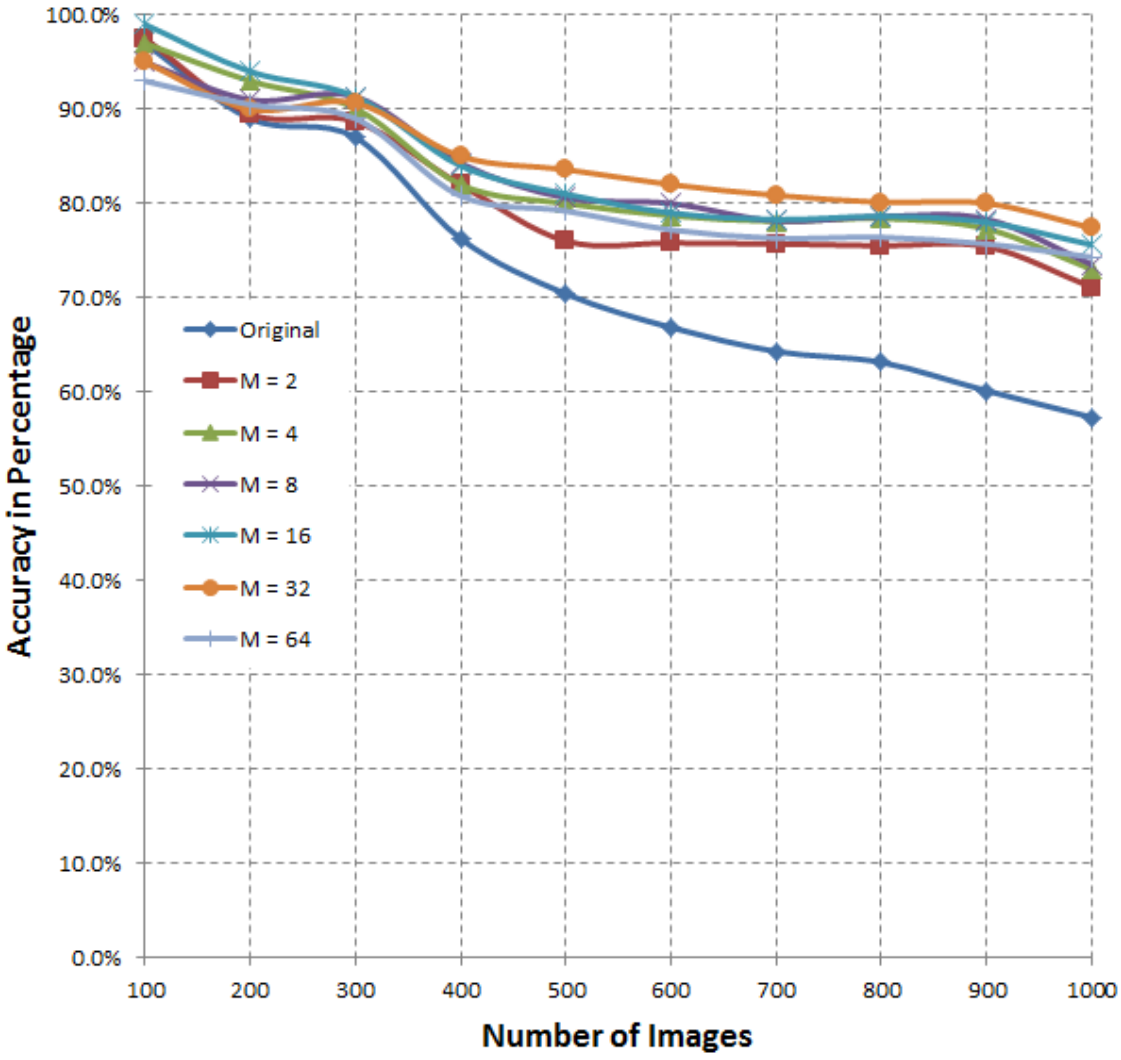

Figure 4.6: Recognition accuracy with divide and conquer

4.2.3 Two-stage Hybrid Matching Algorithm

The experimental result indicates that the two-stage hybrid matching algorithm possesses a higher recognition accuracy than any single method of PCA, LDA or NGM. Moreover, the improved accuracy also increases with the gallery size. Finally, the two-stage hybrid even provides less computational costs than the original NGM.

Recognition Accuracy of Two-stage Hybrid

By hybridizing PCA with NGM and LDA with NGM, the recognition accuracies of PCA, LDA, NGM, PCA with NGM and LDA with NGM are tested on galleries of varying sizes ranging from 100 to 1000 with an increment of 100. All the recognition accuracies with

different gallery sizes are collected in Table 4.6 and plotted in Figure 4.7. Also, the the improvements of accuracy are calculated in Table 4.7 and Table 4.8.

From Figure 4.7, it can be obviously seen that the performance of the hybrid methods is superior to either PCA, LDA or NGM with all different sizes of galleries. Moreover, the improvement degree also has a rising trend with the gallery size, based on Table 4.7 and Table 4.8. For instance, in Table 4.6, when the gallery size reaches 1000, the accuracy of the hybridizing LDA and NGM still maintains at a high level of 81.5%. Compared with 65.0% of LDA and 57.3% of NGM, the hybrid method achieves 16.5% improvement over LDA and 24.2% improvement over NGM. In other words, the hybrid method rectifies 165 and 242 false recognition cases for LDA and NGM respectively.

Table 4.6: Recognition accuracy of two-stage hybrid

Gallery	PCA	LDA	NGM	PCA+NGM	LDA+NGM
100	92.0%	89.0%	97.0%	98.0%	97.0%
200	86.0%	85.0%	89.0%	93.5%	94.0%
300	84.7%	84.7%	87.0%	93.3%	94.3%
400	73.0%	75.0%	76.3%	87.0%	89.5%
500	70.4%	70.6%	70.4%	83.0%	85.6%
600	68.8%	69.3%	66.8%	82.5%	83.3%
700	67.6%	68.0%	64.3%	81.3%	82.9%
800	69.0%	68.6%	63.1%	84.1%	82.0%
900	69.8%	67.7%	60.1%	84.4%	81.6%
1000	69.0%	65.0%	57.3%	79.6%	81.5%

Table 4.7: Improvement of hybridizing PCA and NGM

Gallery	Over PCA	Over NGM
100	6.0%	1.0%
200	7.5%	4.5%
300	8.7%	6.3%
400	14.0%	10.8%
500	12.6%	12.6%
600	13.7%	15.7%
700	13.7%	17.0%
800	15.1%	21.0%
900	14.7%	24.3%
1000	10.6%	22.3%

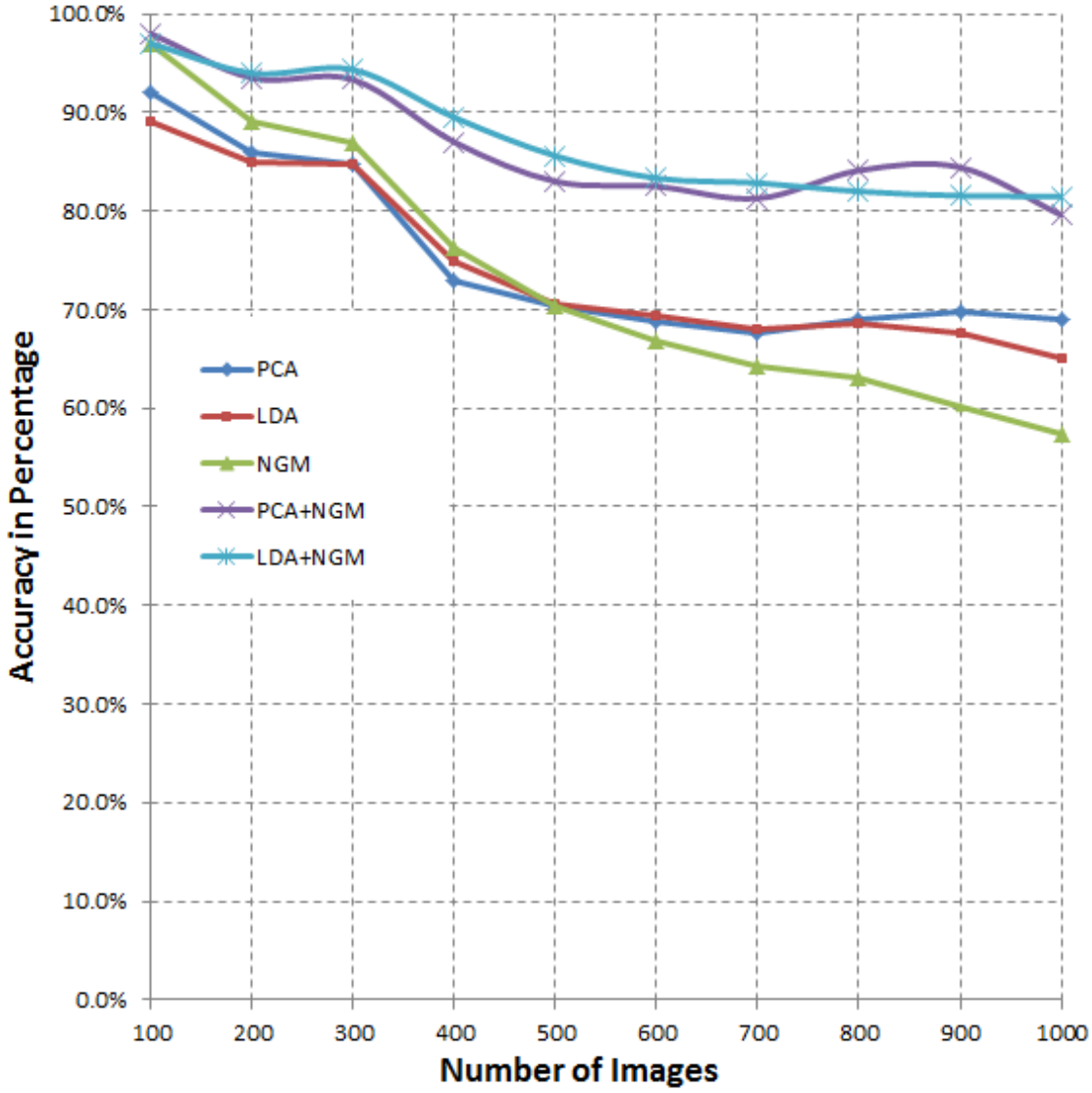

Figure 4.7: Recognition performance of two-stage hybrid

Computational Costs of Two-stage Hybrid

Since PCA has a similar computational costs with that of LDA, only the costs of hybridizing PCA and NGM is tested in this paper. All the computational costs in seconds of PCA, NGM, and the two-stage hybrid with different gallery sizes are collected in Table 4.9 and plotted in Figure 4.8. Though the two-stage hybrid executes a holistic method and a feature-based method, but the experimental result of hybridizing PCA and NGM shows that its computational costs actually is even less than that of NGM.

From Table 4.9 and Figure 4.8, three observations are worthy of mentioning. First, the computational costs of PCA can be negligible, compare with that of NGM and the hybrid

Table 4.8: Improvement of hybridizing LDA and NGM

Gallery	Over LDA	Over NGM
100	8.0%	0.0%
200	9.0%	5.0%
300	9.7%	7.3%
400	14.5%	13.3%
500	15.0%	15.2%
600	14.0%	16.5%
700	14.9%	18.6%
800	13.4%	18.9%
900	13.9%	21.4%
1000	16.5%	24.2%

method. Second, the computational costs of the hybrid method is even slightly less than that of NGM, except the gallery size is 100. It can be explained by the fact that the hybrid method requires only 15 candidates to match in the NGM part, while all the gallery images are required to participate in the original NGM. Finally, since the computational costs of the NGM part is stable due to the certain number of candidates, the larger the gallery is, the more computational costs can be saved by the hybrid method. It also explains why the gap between the hybrid method and the original NGM increases with the gallery size, as illustrated in Figure 4.8.

Table 4.9: Computation costs (in seconds) of hybridizing PCA with NGM

Gallery size	PCA	NGM	Hybrid
100	2.4	1718.4	1792.5
200	10.6	3541.7	3334.8
300	22.0	5417.6	5260.3
400	38.1	7638.2	6683.6
500	60.0	9685.7	8302.6
600	85.0	11822.4	10751.8
700	114.2	14388.8	11714.4
800	152.2	16440.2	14151.0
900	189.3	19107.1	15074.2
1000	232.9	21922.6	17117.9

4.3 Summary for Experiments

Through the detailed reports of experiments on multi-stage matching algorithms, three points can be summarized as next:

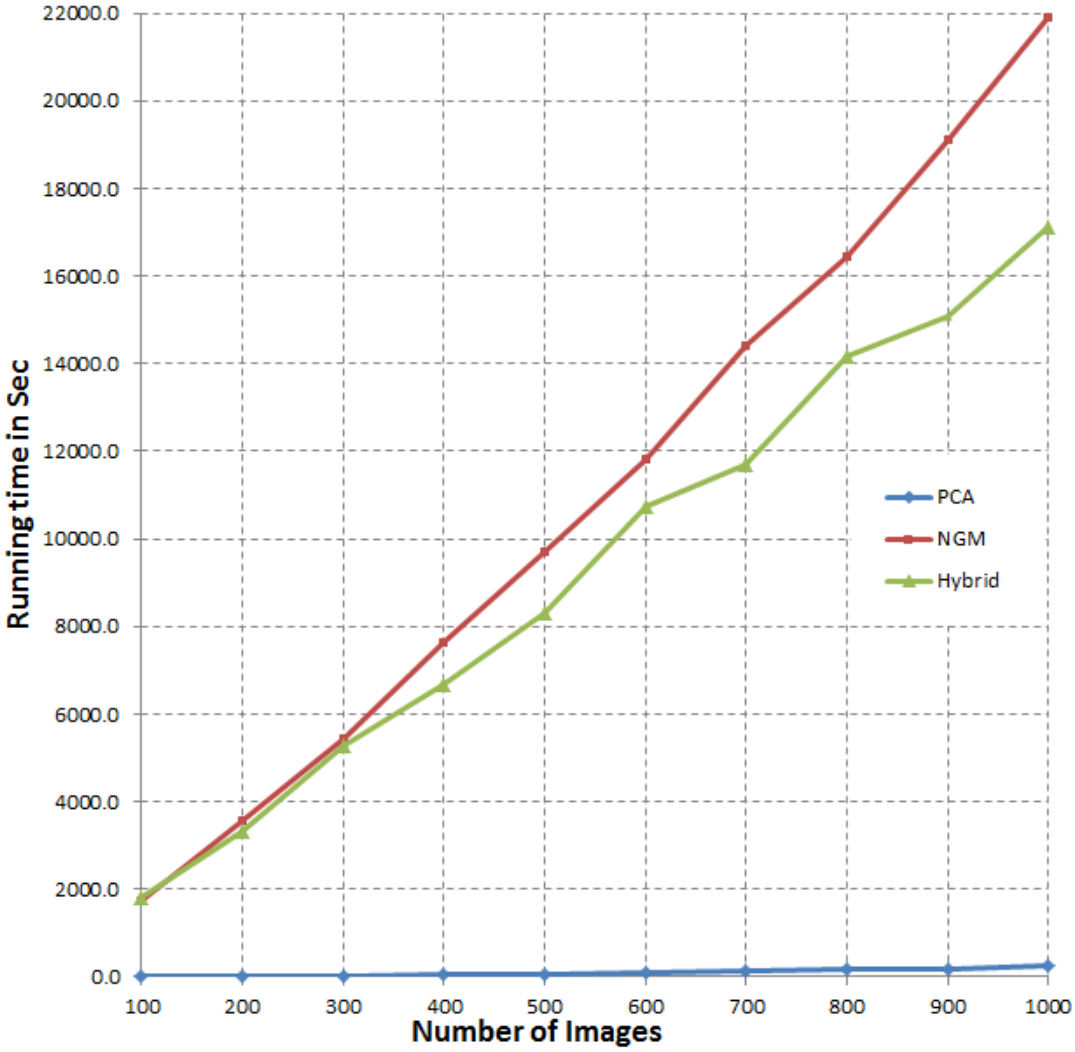

Figure 4.8: Computational costs of hybridizing PCA with NGM

1. The multi-stage matching algorithms provide a higher accuracy than that of the original methods.

2. With a larger gallery size, the multi-stage matching algorithms provide a larger improvement of accuracy.

3. The multi-stage matching algorithms have an insignificant impact on the computational costs.

In conclusion, the recognition degradation problem of face recognition is effectively solved by the multi-stage matching algorithms without raising the computational costs.

Chapter 5

CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions

The primary goal of the research in this paper is to achieve a higher recognition accuracy while reduce or maintain the computational costs of face recognition. One parallel algorithm that works in a coarse way was proposed to simultaneously reduce the running time and memory usage of a feature-based method NGM. To deal with the recognition degradation problem due to "single sample per person" and large gallery sizes, a set of multi-stage matching algorithms that determine the recognition result step-by-step were proposed.

To simultaneously reduce the running times and memory usages of feature-based methods, a coarse parallel method that equally divides the training images and probe images into the multiple processors was implemented with a typical feature-based method NGM. First, each processor finishes its own training workload and stores the extracted feature information respectively. And then, each processor simultaneously carries out the matching process for their own probe images by communicating feature information with each other. Finally, one processor collects the recognition result from the other processors. Due to the well-balanced workload, the speedup increases with the number of processors and thus the efficiency was maintained excellently, even if a large number of processors were participated. Furthermore, memory usage on each processor also reduced considerably as the number of processors increased. In sum, the parallel algorithm has an advantage of less running times and less memory usage for one processor simultaneously.

To solve the recognition degradation problem under "single sample per person" and large gallery sizes, the first multi-stage matching algorithm, n-ary elimination that searches the recognition result step-by-step from the perspective of global was introduced to the matching process of NGM. Each step of matching selects $1/n$ the training images that have the greatest similarities as the candidates for the next step and removes the others. The behavior of picking and eliminating repeats until the amount of the remaining training images is small enough to produce the final recognition result. By setting different n for elimination, the recognition performance of NGM with n-ary elimination matching algorithm was tested on different sizes of gallery. The experimental result showed that the NGM with n-ary elimination matching algorithm outperforms the original NGM in recognition accuracy, and

the improved accuracy also increases with the gallery sizes. Additionally, the multiple steps of matching have little affects to computational costs.

Divide and conquer matching algorithm, another type of multi-stage matching algorithm introduced to NGM, solves the recognition step-by-step from the local perspective. For each probe image, other than matching with the whole gallery, the training images are divided into groups with a grouping number M, and the best similar training image of each sub-group is selected as a candidate for the next step of matching. The behavior of dividing and conquering repeats with the remaining candidates until the amount of them is small enough to produce the final recognition result. By setting different grouping numbers, the recognition performance of NGM with divide and conquer matching algorithm was tested on different sizes of gallery. The experimental result showed that the NGM with divide and conquer gains a remarkably improved recognition accuracy. Moreover, the improvement level also increases with the gallery sizes. Similar with n-ary elimination, divide and conquer matching algorithm dose not bring considerably extra computational cost either.

The other multi-stage matching algorithm, two-stage hybrid matching algorithm hybridizes a holistic method and a feature-based method by combining the superiorities of the both types of methods yet avoiding the inferiorities of either. First, it utilizes a holistic method to pick up a small amount of candidates that are best similar with the probe image. Second, a feature-based method was used to find out the recognition result from the remaining candidates. Two implementations, hybridizing PCA with NGM and LDA with NGM, were tested on different sizes of gallery. The experimental result showed that the recognition performance of the hybrid methods was superior to any single method of PCA, LDA or NGM. And it also showed that the improvement level also increases with the gallery sizes. Additionally, the two-stage hybrid matching algorithm provides even less computational costs than that of NGM.

To summary, the work of our research considerably boosts the performance of face recognition. The parallel algorithm brings less running time and memory usage, while the multi-stage matching algorithms effectively solve the recognition degradation problem of face recognition.

5.2 Future Works

For n-ary elimination and divide and conquer matching algorithms, other than NGM, we believe that they are also promising to other types of face recognition methods. It would be of great significance to introduce them to other types of methods in the future. For two-stage hybrid matching algorithm, if the amount of candidates chosen by the holistic method can be

flexible to the gallery size, the pixel size or other parameters, we believe that the recognition performance can be further optimized.

BIBLIOGRAPHY

[1] B. Frey A. Colmenarez and T. Huang. Detection and tracking of faces and facial features. In *IEEE International Conference Image Processing*, volume 2, Kobe, Japan, October 1999.

[2] F. Jiang J. Ayyad N. Penard A. J. OToole, P. J. Phillips and H. Abdi. Face recognition algorithms surpass humans matching faces across changes in illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1642 –1646, September 2007.

[3] B. Klare A. K. Jain and U. Park. Face matching and retrieval in forensics applications. *IEEE Multimedia*, 19:20–28, January 2012.

[4] Ioannis Pitas Athanasios Nikolaidis. Facial feature extraction and pose determination. *Pattern Recognition*, 22:1090–1104, November 2000.

[5] David Beymer and Tomaso Poggio. Face recognition from one example view. In *Proceedings Fifth International Conference on Computer Vision*, Cambridge, MA, June 20-23 1995.

[6] W. W. Bledsoe. *The model of method in facial recognition*. Tech. rep. PRI:15, Panoramic research Inc., Palo Alto, CA, 1964.

[7] I. S. Bruner and R. Tagiure. *The perception of people*, volume 2. Addison-Wesley, Reading, MA, 1954.

[8] Xianming Chen, Chaoyang Zhang, Fan Dong, and Zhaoxian Zhou. Parallelization of elastic bunch graph matching (ebgm) algorithm for fast face recognition. In *2013 IEEE China Summit International Conference on Signal and Information Processing (ChinaSIP)*, Beijing, China, July 2013.

[9] Xianming Chen, Chaoyang Zhang, and Zhaoxian Zhou. Improve recognition performance by hybridizing principal component analysis (pca) and elastic bunch graph matching (ebgm). In *2014 IEEE Symposium Series on Computational Intelligence*, Orlando, FL, December 2014.

[10] Xianming Chen, Wenyin Zhang, Chaoyang Zhang, and Zhaoxian Zhou. Improve non-graph matching feature-based face recognition performance by using a multi-stage matching strategy. In *2015 11th International Symposium on Visual Computing (accepted)*, Las Vegas, Nevada, December 2015.

[11] C. Darwin. *The expression of the Emotions in Man and Animals*. John Murray, London, U.K, 1972.

[12] K. Etemad and R. Chellappa. Discriminant analysis for recognition of human face images. *Journal of the Optical Society of America A*, 14:1724–1733, August 1997.

[13] Kamran Etemad and Rama Chellappa. Face recognition using discriminant eigenvectors. In *IEEE International Conference on, Acoustics, Speech, and Signal Processing*, Atlanta, GA, May 7-10 1996.

[14] O. Carmona F. Smeraldi and J. BigÃijn. Saccadic search with gabor features applied to eye detection and real-time head tracking. *Image and Vision Computing*, 18:323–329, March 2000.

[15] R. A. Fisher. The use of multiple measures in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[16] Yun Fu, Umar Mohammed, James H. Elder, and Peng Li. Probabilistic models for inference about identity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:144–157, January 2012.

[17] P. J. Flynn G. Aggarwal, S. Biswas and K. W. Bowyer. A sparse representation approachto face matching across plastic surgery. In *2012 IEEE Workshop on Applications of Computer Vision (WACV)*, Breckenridge, USA, January 2012.

[18] D. Gabor. Theory of communication. *Journal of the Institution of Electrical Engineers*, 93:429–457, 1946.

[19] Quanxue Gao, Lei Zhang, and David Zhang. Face recognition using flda with single training image per person. *Applied Mathematics and Computation*, 205:726–734, November 2008.

[20] P.J. Phillips B. Draper Y.M. Lui D.S. Bolme G.H. Givens, J.R. Beveridge. Introduction to face recognition and evaluation of algorithm performance. *Computational Statistics and Data Analysis*, 67, November 2013.

[21] Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar. *Introduction to Parallel Computing (Second Edition)*. Pearson Education Limited, Edinburgh Gate, England, 2003.

[22] S. Baluja H.A. Rowley and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:23–38, January 1998.

[23] E. Hjelm and B.K. Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83:236–274, September 2001.

[24] http://www.i cube.co.za/Products.htm.

[25] A. Garg I. Cohen and T. Huang. Emotion recognition from facial expressions using multilevel hmm. *Neural Information Processing Systems*, 2000.

[26] J. Paik J. Heo, B. Abidi and M. A. Abidi. Face recognition: Evaluation report for faceit identification and surveillance. In *Proc. Of SPIE 6th International Conference on QCAV03*, volume 5132, Gatlinburg, TN, USA, May 2003.

[27] Rabia Jafri and Hamid R. Arabnia. A survey of face recognition techniques. *Journal of Information Processing System*, 5:41–68, June 2009.

[28] T. Kanade. *Computer recognition of human faces*. Birkhauser, Basel, Switzerland, and Stuttgart, Germany., 1973.

[29] B. Kepenekci. Face recognition using gabor wavelet transform. Master's thesis, The Middle East Technical University, Sep., 2001.

[30] M. Kirby and L.Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:103–108, January 1990.

[31] J. Klontz L. Best-Rowden, B. Klare and A. K. Jain. Video-to-video face matching: Establishing a baseline for unconstrained face recognition. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, Arlington, USA, September 2013.

[32] Juwei Lu, K.N. Plataniotis, and A.N. Venetsanopoulos. Face recognition using lda-based algorithms. *IEEE Transactions on Neural Networks*, 14:195–200, January 2003.

[33] Juwei Lu, K.N. Plataniotis, and A.N. Venetsanopoulos. Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition. *Pattern Recognition Letter*, 26:181–191, February 2005.

[34] Juwei Lu and Kostantinos N. Plataniotis. Boosting face recognition on a large-scale database. In *Proceedings of the 2002 International Conference on Image Processing*, volume 2, Rochester, New York, September 2002.

[35] S Mitra and T Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37:311 – 324, May 2007.

[36] E Murphy-Chutorian and M.M Trivedi. Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:607 – 626, April 2008.

[37] P. Belhumeur N. Kumar, A. Berg and S. K. Nayar. Attribute and simile classifiers for face verification. In *2009 IEEE 12th International Conference on Computer Vision*, Kyoto, Japan, September 2009.

[38] R. McCabe P. J. Phillips and R. Chellappa. Biometric image processing and recognition. In *In Proceedings, European Signal Processing Conference*, volume 1, 1998.

[39] Bo-Gun Park and Sang-Uk Lee. Face recognition using face-arg matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1982–1988, December 2005.

[40] JoÃčo P. Hespanha Peter N. Belhumeur and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:711–720, July 1997.

[41] P.C Petrantonakis and L.J Hadjileontiadis. Emotion recognition from eeg using higher order crossings. *IEEE Transactions on Information Technology in Biomedicine*, 14:186 – 197, October 2009.

[42] P.Jonathon Phillips, Harry Wechslerb, Jeffery Huang, and Patrick J. Raussa. The feret database and evaluation procedure for face recognition algorithms. *Image and Vision Computing*, 16:295–306, 1998.

[43] W.T. ; O'Toole A.J. ; Flynn P.J. ; Bowyer K.W. ; Schott C.L. ; Sharpe M Phillips, P.J; Scruggs. Frvt 2006 and ice 2006 large-scale experimental results. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:831 – 846, March 2009.

[44] S.A. Rizvi P.J. Phillips, H. Moon and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1090–1104, October 2000.

[45] M. Abdel-Mottaleb R.-L. Hsu and A. Jain. Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:696–706, May 2002.

[46] M.F. Hansen W.A.P. Smith V. Argyriou M. Petrou M.L. Smith L.N. Smith S. Zafeiriou, G.A. Atkinson. Face recognition and verification using photometric stereo: the photoface database and a comprehensive evaluation. *IEEE Transactions on Information Forensics and Security*, 8, January 2013.

[47] Linlin Shen and Li Bai. A review on gabor wavelets for face recognition. *Pattern Analysis and Applications*, 9:273–292, October 2006.

[48] Xiaoyang Tan, Songcan Chen, Zhihua Zhou, and Fuyan Zhang. Face recognition from a single image per person: A survey. *Pattern Recognition*, 39:1725–1745, September 2006.

[49] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.

[50] Matthew Turk and Alex Pentland. Face recognition using eigenfaces. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Maui, HI, June 3-6 1991.

[51] Laurenz Wiskott, Jean-Marc Fellous, Norbert Kruger, and Christoph von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:775–779, July 1997.

[52] Jianxin Wu and Zhihua Zhou. Face recognition with one training image per person. *Pattern Recognition Letters*, 23:1711–1719, December 2002.

[53] Chaoyang Zhang, Zhaoxian Zhou, Hua Sun, and Fan Dong. Comparison of three face recognition algorithms. In *2012 International Conference on Systems and Informatics*, Yantai, China, May 2012.

[54] Haiyang Zhang and Huadong Ma. Grid-based parallel elastic graph matching face recognition method. In *Proceeding of Advanced Web and Network Technologies, and Applications*, Harbin, China, January 16-18 2006.

[55] Lei Zhang and Samaras D. Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:351 – 363, March 2006.

[56] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing surveys*, 35:399–458, December 2003.

[57] Wenji Zhao, Arvindh Krishnaswamy, Rama Chellappa, Daniel L. Swets, and John Weng. *Discriminant Analysis of Principle Component for Face Recognition*, pages 73–85. Springer-Verlag, Berlin, 1998.