A Scheme of Human Face Recognition

in

Complex Environments

WEI CUI

A thesis submitted to Auckland University of Technology

in partial fulfilment of the requirements for the degree of

Master of Computer and Information Science (MCIS)

2014

School of Computer and Mathematical Sciences

Abstract

Face recognition is one of the most important biometrics in computer vision and it has been broadly employed in the area such as surveillance, information security, identification, and law enforcement. Over the last few decades a considerable number of studies have been conducted in face recognition such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), Linear Discriminant Analysis (LDA), and Elastic Bunch Graph Matching (EBGM), etc. What seems to be insufficient is the research in accuracy of face recognition, it could be affected by the factors such as luminance changes, pose changes, making up, complex backgrounds, head rotation, aging issues, and emotions, etc. This thesis will limit the discussions and concentrate on accuracy problem of face recognition in complex environments.

The complex environments are considered as a place with a large number of people such as big office, internet cafés, airport, train and bus station, and casino etc. In these environments, the target human faces for recognizing usually mingle with moving objects. However, the face recognition in complex environments also can be described as the face recognition for several people who might be interested. Therefore, in this thesis, we target only the person (referred as the "Target User") who is located closest to the camera and stationary.

In this thesis, a new scheme is proposed to recognize human faces in such complex environments. The proposed scheme can be split into three phases. The first is Moving Object Removal (MOR). The moving object could be a pedestrian, a vehicle or other moving object. The second phase is face detection which is a technology to locate a human face in a set of images or a video. The Open Source Computer Vision (OpenCV) locates human face features such as those of eyes, ears, mouth, and nose utilizing Viola-Jones algorithm. The final stage is face recognition. If a face is detected, it will be decomposed into PCA components, and then compared to other decomposed images in a face dataset.

The objective of this thesis is to propose a new scheme for human face recognition in complex environments so as to improve recognition precision and reduce false alarms. The scheme can be applied to prevent computer users against sitting too long in front of a screen.

Keywords: face recognition, face detection, PCA, LDA, OpenCV, digital image processing

Table of Contents

List of Figures

List of Tables

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature:　　　　　　　　　　　　　　　　　　　Date: 4 July 2014

Acknowledgements

Chapter 1 Introduction

1.1 Background and Motivation

Nowadays, the requirement for reliable personal identification in access control has resulted in an increased attention in biometrics (Kocjan & Saeed, 2012; Choi et al., 2001; Jain et al., 2004; Li & Lu, 1999); it is a pattern recognition technology which operates by obtaining biometrics data from a target object (Gandhe et al., 2007), correspondingly the closest matched object will be selected from the dataset. The biometrics data can be used for machine learning and training, they are collected from the objects such as deoxyribonucleic acid (DNA), ear, gait, fingerprints, palm print, voice, iris, retina scan, signature dynamics and face (Lawrence et al., 1997; Blue et al., 1994; Burton, 1987; Qi & Hunt, 1994) for authentication and identification purpose.

Most of those biometrics data have to be collected by using special hardware such as fingerprint scanner, palm print scanner, retina scanner, DNA analyser, and the target objects have to touch with the required hardware in the stage of data collection. The face recognition does not require target object to be touched with any hardware, so as the entire face recognition is completed without awareness of the target people. Face recognition has been widely used in the area such as intelligent surveillance, information security, human identification, and law enforcement (Kondo & Yan, 1999; Chellappa et al., 1995; Vinaya & Ashwini, 2009; Pentland, 2000). Recently, face recognition has become one of the most important biometrics in computer vision (Turk & Pentland, 1991). In general, face recognition is separated into two categories: online face recognition and offline face recognition (Lawrence et al., 1997).

Online face recognition is operated by live camera, namely, face recognition will be processed synchronously while the camera is recoding. The real-time based face recognition is not only used to identify a person under live security monitoring, object tracking, civil border checking, city safety control but also is used to authenticate a person such as grant the access while deny another (for instance, access authorization to a secured building, computer, car or a gambling machine) (Pentland et al., 1993).

The offline face recognition means that face recognition will be conducted using recorded data. For example, during the investigation, a police officer could input a video

or an image into a face recognition system to identify suspects. The face recognition system will return a list of most suspicious ones.

A prototype having online face recognition and offline face recognition is developed for experimental purpose in this thesis. The prototype consists of two main kinds of activities, the first one is to complete face detection using the OpenCV cascade classifier and the second is recognize a face using Principal Component Analysis (PCA) algorithm (Jonsson et al., 2000).

There is a large number of face recognition algorithms having been widely studied such as PCA, LDA, ICA and EBGM (Sirovich & Kirby, 1987; Turk & Pentland, 1991; Martinez & Kak, 2001; Lu et al., 2003; Delac et al., 2005). A great deal of literatures show that automatic face analytics such as face detection, recognition and facial expression recognition are useful in recent security and forensic research (Ahonen et al., 2006; Li & Jain, 2005; Pentland et al., 1994; Wong et al., 2001). The face recognition algorithms have been proudly studied in decades (Kondo & Yan, 1999; Lee et al., 1996; Bruce & Young, 1986; Valentin et al., 1994), and a group of algorithms have been already implemented in face recognition for various purposes.

The practical applications today for human face recognition have been diffusely applied to Facebook website, cameras (Smile detection function and face detection function) amounted to a laptops and mobile phone (face unlock function), and border control system, etc. It is more noteworthy that according to the well-known Federal Bureau of Investigation (FBI) website, the FBI agency has extended its existing Integrated Automated Fingerprint Identification System (IAFIS) which criminal and civil fingerprint database was implemented to include face recognition technology (Phillips et al., 1997). The expended system is named as Next Generation Identification (NGI) and will be fully operated in summer 2014. The NGI includes FBI's massive biometrics databases such as fingerprint database, DNA database, face database, and palm print database etc. The FBI agency also plans to collect 52 Million human face images in their NGI face database in 2015. Large scale of image collection for face recognition includes both non-criminal and criminal people's photos.

Recently, the FBI face recognition system has been applied to the area of combating passport fraud, supporting law enforcement, identifying missing children and minimizing benefit/identity fraud (Phillips et al., 1997). Also, authentications of driver's licenses and photo ID were used by other authorized government (Chellappa et al.,

1995). In addition, most of international airports have implemented the face recognition system for border check, such as the "SmartGate" system which is used to match the passport holder's face with their passport photo in New Zealand, Australia and Singapore etc. If a passport holder could not be identified by the system, the system will then send an alarm to the nearest custom officers.

Moreover, a face recognition system has been introduced to casino which could help security to identify the problem gamblers. For instance, when a problem gambler walks in and sits in front of a pokie machine, a camera mounted in the pokie machine is used to capture the gambler's face, the captured image will be compared to the trained face images in the existing human face database so as to identify if this is a problem gambler. The pokie machine will lock out the gamblers if the face recognition system identified them.

However, as one of the digital image processing and object recognition problems, face recognition still suffers from problems such as luminance changes (Choi et al., 2001), pose changes, mark-up (facial data points change in Figure 1.1), complex background, head rotation, aging issues, and expressions etc (Ahonen et al., 2006; Goldstei et al., 1971; Grudin, 2000; Lu et al., 2003; Maio and Maltoni, 2000). Most of these problems are still under investigation.

In Figure 1.1, the facial data points include the 35 points indicated the different parts of a human face such as eyebrows, right mid eyebrows, left mid eyebrows, nose bridge, nose tip, right lip upper bend, left lip upper bend etc., respectively.
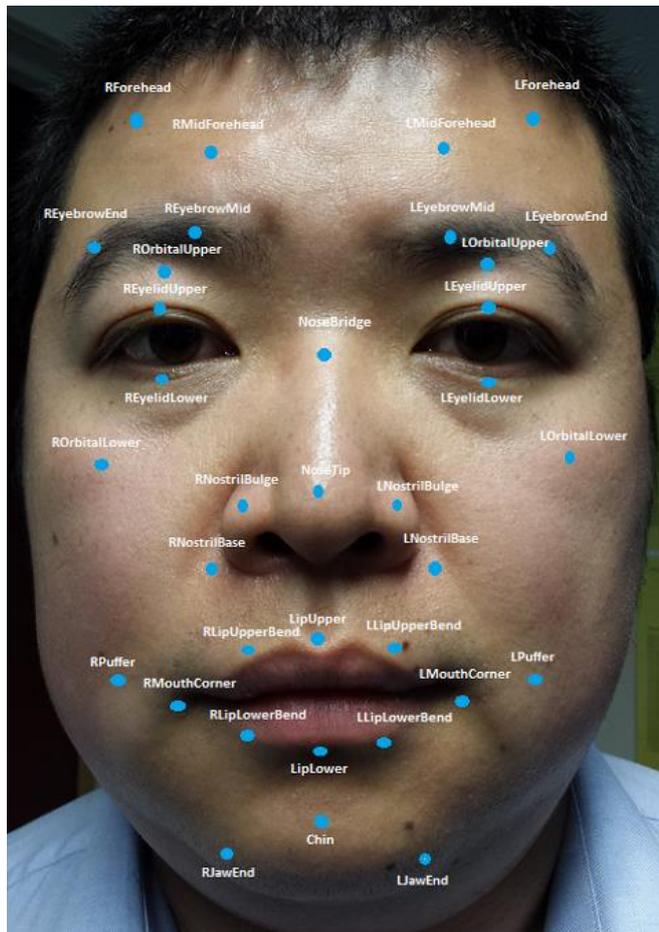
Figure 1.1 Facial data points

Therefore, in border control, a passport holder is usually identified by the officer using the information of eyes, nose and mouth provided by the passport photo. Sometimes, earmark also has to be taken into consideration as the complementary information since human ears have been changed seldomly which are not affected by any plastic surgeries. As shown in Figure 1.2, the OpenCV face detection identifies a person's face and ears using the information marked by red rectangles.
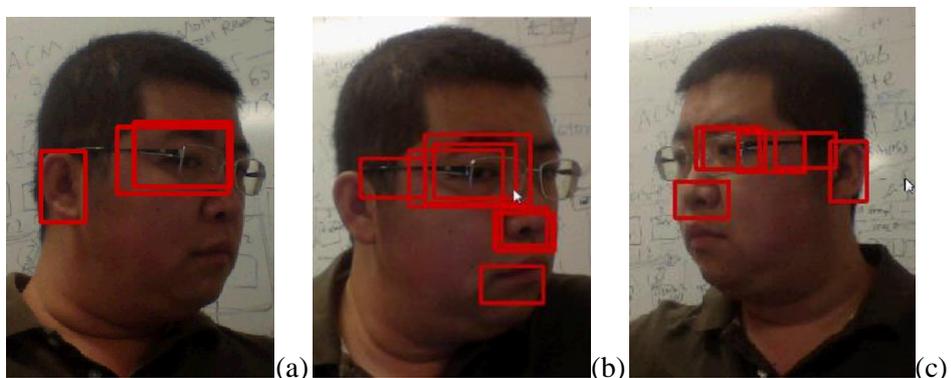
Figure 1.2 Human face recognition with the assistances of eyes, nose, mouth and ears.

In this thesis, a new scheme for face recognition in complex environments is proposed. In order to correctly count elapsed time and reduce false alarms in complex environments, the frame difference has been applied to detect motions and remove the moving objects from the background of motion pictures so as to increase face recognition precision.

1.2 Objectives of This Thesis

As mentioned in Section 1.1, the accuracy of human face recognition has always been the major concern for most of the security systems, identification systems and border check systems. In the past few decades a considerable number of studies have been made on the face recognition algorithms such as PCA, LDA, ICA, and EBGM. The understanding of those algorithms has been awarded, the face recognition approaches such as feature-based approaches and appearance-based approaches have been investigated as well.

A new scheme of human face recognition in complex environments has been developed in this thesis, to increase the precision and recall of face recognition is the main mission of this thesis. To achieve this goal, both theory and practice of face recognition have been studied. Theoretically, we have studied the algorithms of face detection and recognition; meanwhile we also developed a face detection and recognition prototype with MOR in practice.

1.3 Structure of This Thesis

The structure of this thesis is shown in Figure 1.3 below. This thesis consists of three main folds: understanding and planning, designing, implementation, testing and evaluation.

In Chapter 2, the literature review of face detection, face recognition and MOR are studied. The chapter covers detail understanding of various face recognition algorithms and approaches. The understanding of the OpenCV cascade classifier, Moving Object Detection (MOD) and MOR are also mentioned in this chapter.

In Chapter 3, the research methodology of this thesis is explained. The main hypotheses are tested as well as the main issues and potential solutions to the problems are explained. The experimental layout and design, dataset and implementation are depicted finally.

In Chapter 4, the experimental results and main outcomes are demonstrated. The chapter introduces the experimental test bed and analysis of experimental results. The outcomes of the experiments are detailed with the supports of tables and figures.

In Chapter 5, the analysis and discussions are presented based on the experimented outcomes and results. The main research hypotheses are tested and analysed with the final experiment results. The limitations of this research will be discussed as well. The conclusion and future work will then be presented in Chapter 6.
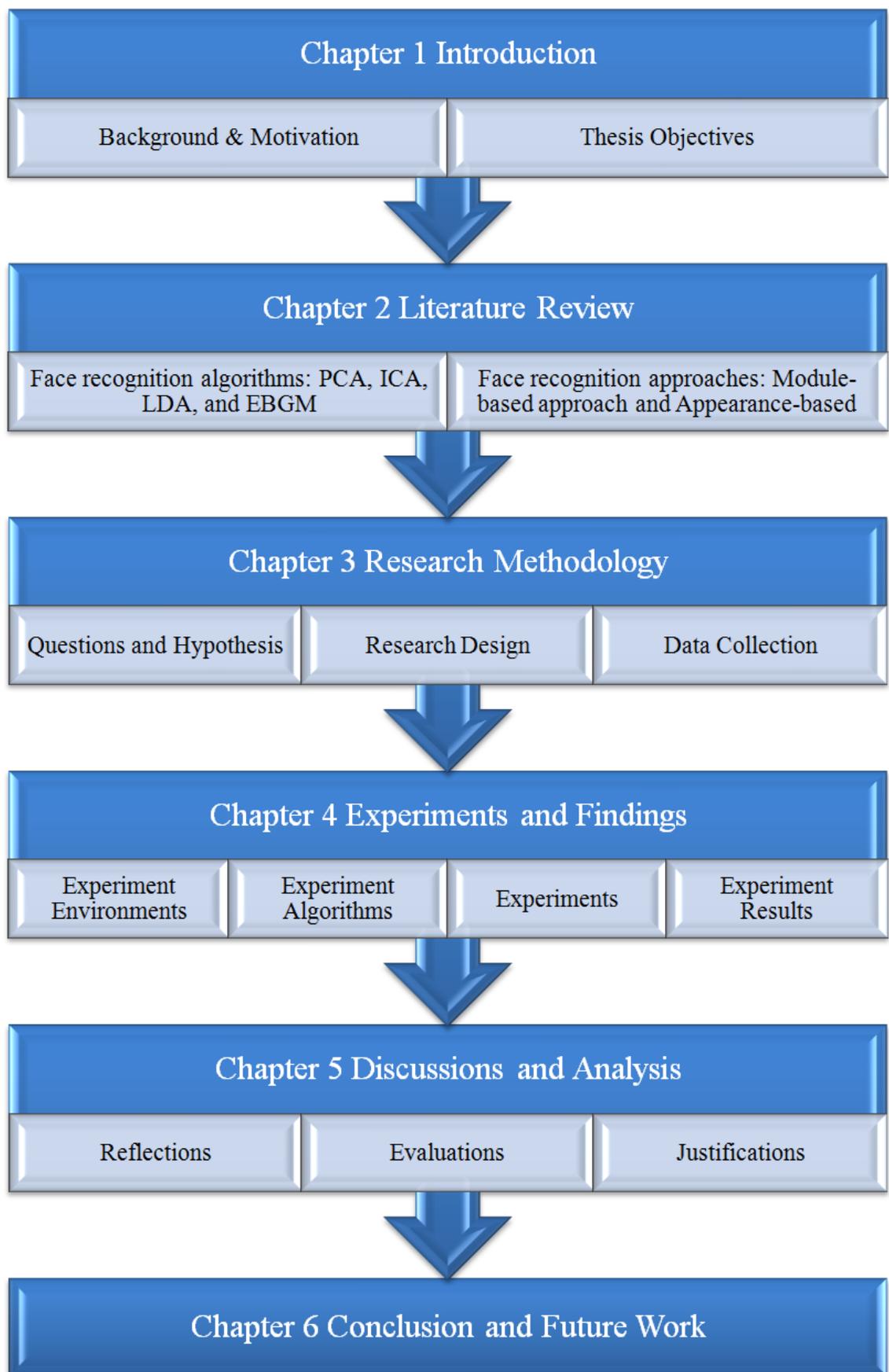
Figure 1.3 Structure of this thesis

Chapter 2 Literature Review

To achieve our objectives, three important research areas are expected to be studied. The first area is face detection, the second is face recognition and the third is Moving Object Removal (MOR). In this thesis, face detection was conducted by using OpenCV cascade classifier while face recognition was developed by using Principal Component Analysis (PCA) algorithm, moreover the MOR was implemented by using frame differencing.

In this chapter, a detailed understanding of OpenCV cascaded classifier will be discussed, followed by analysis of the most popular face recognition algorithms, for instance, PCA, ICA, LDA, and EBGM, as well as two important face recognition approaches, namely, appearance-based approach and module-based approach. The MOR sample frames will be presented in the last section.

2.1 Introduction

The main objectives of this chapter are to explain and evaluate the research studies that are related to face detection, face recognition and MOR. By addressing the recent studies and research backgrounds, this chapter presents an important understanding of literature review. This research aimed at finding a solution which improves precision of human face recognition in complex environments and the suggested solutions will be given.

In this chapter, literature review achieves an overall understanding of face detection, face recognition and MOR. In Section 2.2 a detailed understanding of the face detection will be presented, Section 2.3 as overall understanding of face recognition will be described, and it will be followed by detailed review of the most popular face recognition algorithms and approaches in Section 2.4 and 2.5. The evaluations and comparisons of the studied face recognition algorithms will be presented in Section 2.6 and the MOR is analysed in Section 2.7.

2.2 Face Detection

Face detection is a compulsory stage in face recognition. In this thesis, the OpenCV cascade classifier is used for face detection, it is one of the most popular classifications, it was proposed by Paul Viola and Michael J. Jones in 2001 known as Viola-Jones object detection framework (Viola & Jones, 2001).

The cascade classifier means that the subsequent classifier consists of several simpler stages (Castrillon et al., 2011). As shown in Figure 2.1 these stages are applied to a region of interest until the input samples are rejected or accepted.
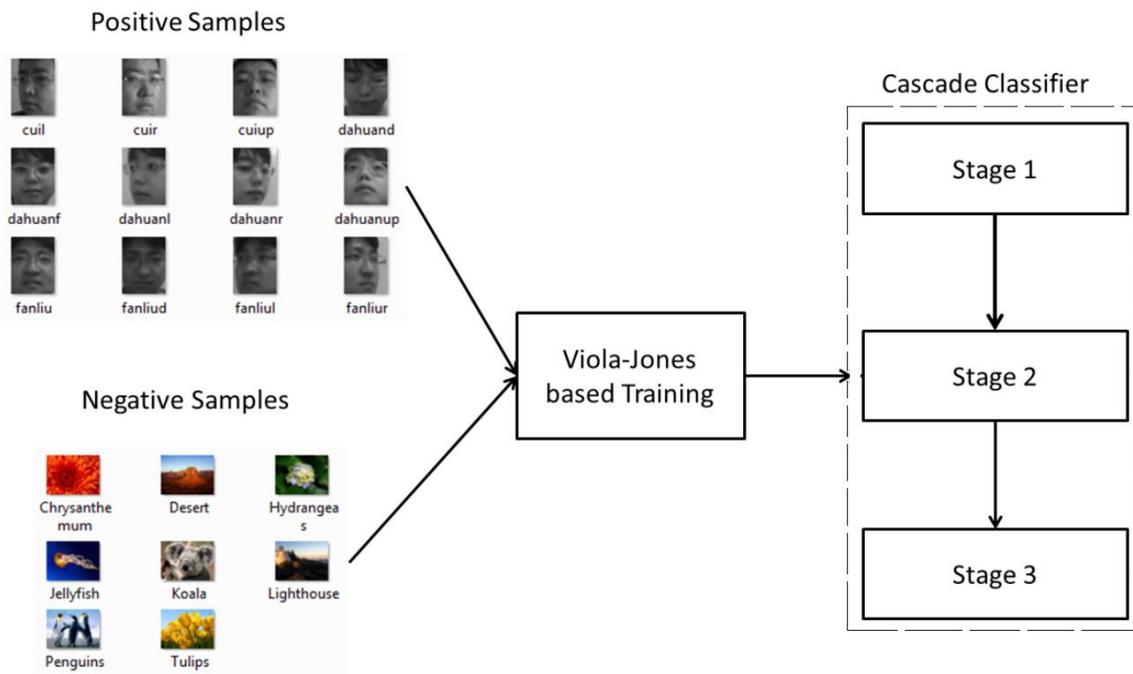
Figure 2.1 Training procedure for a Viola-Jones based cascade classifier

Haar-like features as shown in Figure 2.2 are adjacent rectangular regions that applied to a specific location in an image, and summed up the pixel intensities in each region and calculated the difference between these sums (Viola & Jones, 2001).
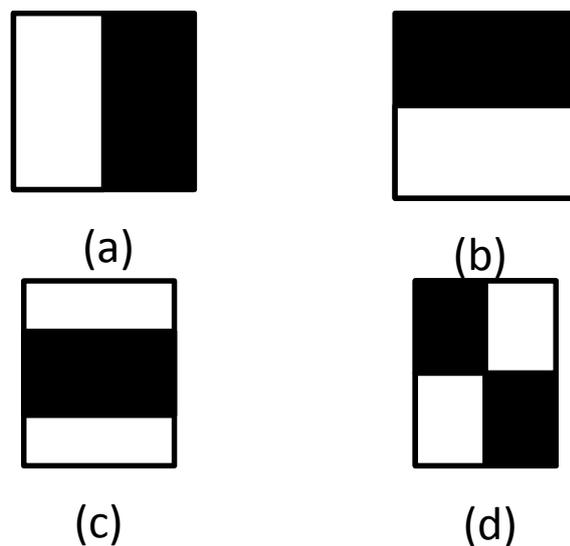
Figure 2.2 Examples of Haar-like features

As shown in Figure 2.2, (a) and (b) are represented as a two-rectangle feature, (c) is used to show a three-rectangle feature and (d) is described as a four-rectangle feature. The sum of the pixels in white rectangles is subtracted from the sum of the pixels in the black rectangles. For instance, the calculation result of two-rectangle feature such as (a) and (b) is the difference between the sum of the pixels within white rectangular area and black rectangular area. Both white rectangular area and the black rectangular area have the same size. The three-rectangle feature calculates the sum of the image pixels in white rectangle area subtracted from the sum of the image pixels in a black rectangle area. The four-rectangle feature calculates the difference between both white rectangle area and both black rectangle area (diagonal pair) (Viola & Jones, 2001; Papageorgiou et al., 1998).

However, the Haar feature detection framework is with the help of rectangular area, the computing speed is relatively slow. To increase the computing speed, an intermediate representation for the image is used, namely, Integral Image (Viola & Jones, 2001).

The Integral Image at point (*x, y*) contains the sum of intensities to the above this point (*x, y*) and to the left this point (*x, y*) shown in equation (2.1).

$$iImage(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{2.1}$$

where $iImage(x, y)$ is the integrated image point (*x, y*), and $i(x', y')$ is the original image.
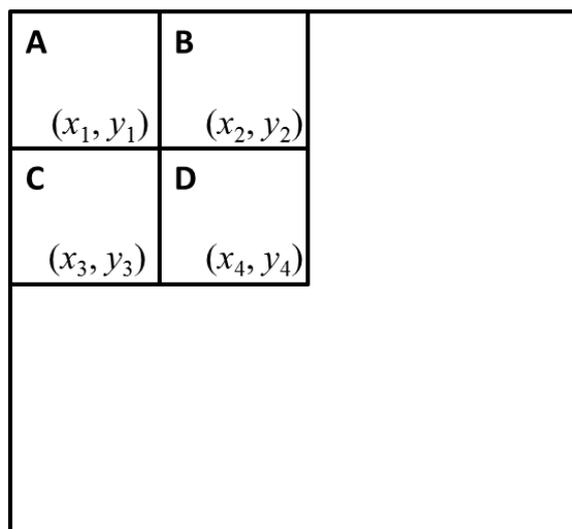
Figure 2.3 The calculation of the pixels in integral image

In Figure 2.3, the value of the integral image at point (x_1, y_1) is the total of the pixels in rectangle A. The value at pixel (x_2, y_2) is A + B. The value at pixel (x_3, y_3) is A + C. The value at pixel (x_4, y_4) is A + B + C + D.

By using the integral image as image representation, the rectangular features can be computed rapidly. It follows that any two-rectangle feature can be computed in six array references such as arrPoint1 (x_0, y_0), arrPoint2 (x_1, y_0), arrPoint3 (x_0, y_1), arrPoint4 (x_1, y_1), arrPoint5 (x_2, y_0), and arrPoint6 (x_2, y_2). The same logic could be applied to three-rectangle feature which can be computed in eight array preference, and any four-rectangle feature can be calculated in nine array preference etc.

2.3 Face Recognition

In the past few decades a considerable number of studies have been made on automatic face recognition algorithms such as PCA (Turk & Pentland, 1991), ICA (Bartlett, 200), LDA (Zhao et al., 1998), Evolutionary Pursuit (EP) (Liu & Wechsler, 2000), EBGM (Wiskott et al., 1997), Kernel Methods (Yang, 2002; Lu et al., 2003), Trace Transform (Kadyrov & Petrou, 2001; Srisuk et al., 2003), Active Appearance Model (AAM) (Cootes et al., 2002), 3D Morphable Model (Blanz & Vetter, 2003), 3D Face Recognition (Bronstein et al., 2004), Bayesian Framework (Liu & Wechsler, 1998), Support Vector Machine (SVM) (Guo et al., 2000), Hidden Markov Models (HMM) (Nefian & Hayes, 1999), and Boosting & Ensemble Solutions (Lu et al., 2006).

Over the past few decades a considerable number of researches have been made on face recognition, they focus on the comparisons of face recognition accuracy (Yang et al., 2004; Draper et al., 2003; Martinez & Kak, 2001; Hanmandlu et al., 2013; Delac et al., 2005; Belhumeur et al., 1997). However, some other researchers addressed the disagreements of the comparing results. There are three main factors caused the different outcomes. The first factor is experiment environments, the second is the face databases such as FERET Database, Yale face database and AT&T face database etc., and the third factor is the experiment methods. For instance, Belhumeur et al., present the comparison result for both PCA and LDA, claim that the PCA algorithm has more error than the LDA algorithm, also the PCA algorithm is more sensitive with the lighting than that of LDA algorithm. But the PCA algorithm is insensitive to small or gradual changes on the face (Belhumeur et al., 1997; Slavkovic and Jevtic, 2012) Draper et al., concluded the comparison for PCA and ICA indicated that the ICA-I performed better than PCA-L2, but the PCA-L2 performed better than the ICA-II and

the PCA-L1 (Draper et al., 2003). However, the researchers Yang et al. presented a different view; that the 2D PCA performs better than that of PCA and ICA. Moreover the PCA and ICA have the same performance result (Yang et al., 2004). Martinez and Kak stated that the PCA algorithm outperforms the LDA algorithm (Martinez & Kak, 2001). Hiremath and Mayakar presented that in a constrained environment, the PCA algorithm is faster and simpler compare with LDA algorithm (Hiremath & Mayakar, 2009; Slavkovic and Jevtic, 2012). As a result, it is impossible to conclude the best algorithm for face recognition. The selection of the algorithm completely depends on the purpose.

The PCA was applied by Kirby and Sirovich to face images that is an optimal compression scheme which minimizes the mean squared error between the original images and their reconstructions for any given level of compression (Sirovich and Kirby, 1987; Turk and Pentland, 1991; Draper et al., 2003). The LDA is a statistical approach for classifying samples of unknown classes based on training samples with known classes. The ICA minimizes both second-order and higher-order dependencies in the input data and attempts to find the basis along which the data are statistically independent (Delac et al., 2005). The EBGM consists of a group of nodes that located at fiducial points and edges between the nodes as well as connections between nodes of different poses are defined (Wiskott et al., 1997). The SVM is learning algorithm that analyses data and recognize patterns, and it is used for classification. It was developed by V. Vapnik (Osuna et al., 1997). The embedded Hidden Markov Model (HMM) is using a 2D Discrete Cosine Transform (DCT) feature vector as the reflection vector to recognize the faces detected (Chen et al., 2010). Most current face recognition techniques refer to the appearance-based recognition work since late 1980's (Draper et al., 2003).

2.4 Facial Recognition Algorithms

2.4.1 Principal Component Analysis (PCA)

Recently, one of the best algorithms on frontal face recognition is PCA. The PCA is a statistical dimensionality-reduction approach where face images are stated as a subset of their eigenvectors and eigenvalues. In other words, it turns out the best possible linear least-squares decomposition of a training set (Sirovich and Kirby, 1987; Turk and Pentland, 1991; Graham and Allision, 1998; Martinez, 2000; Moghaddam & Pentland, 1997; Gottumukkal & Asari, 2004; Moon & Phillips, 2001).

In addition, the PCA is an optimal compression scheme that minimizes the mean squared error between the original images and their reconstructions for any given level of compression. Turk and Pentland used PCA to compute a set of subspace basis vectors also called eigenfaces shown in Figure 2.4 for a face database (Turk & Pentland, 1991). The projected face images (mug shot) from the face dataset into the compressed subspace is known as eigenspace. The subspace crossed by the principal directions are identical to the cluster centroid-based subspace specified by the between-class scatter matrix. To recognize a face, the input test image will be decomposed, then the input face will be used to match face images in the face dataset by projecting them onto the basis vectors and then finding the nearest compressed face image in the subspace (Draper et al., 2003).

In eigenfaces method, jet colormap is presented in Figure 2.4. There are three important information we can read from the jet colormap based eigenfaces. The first information is face features (eyes, nose, and mouth in eigenfaces 8). The second information is greyscale values. And the last information is illumination in the images (Left lighting in eigenface 2).

eigenface 1   eigenface 2   eigenface 3   eigenface 4   eigenface 5

eigenface 6   eigenface 7   eigenface 8   eigenface 9   eigenface 10
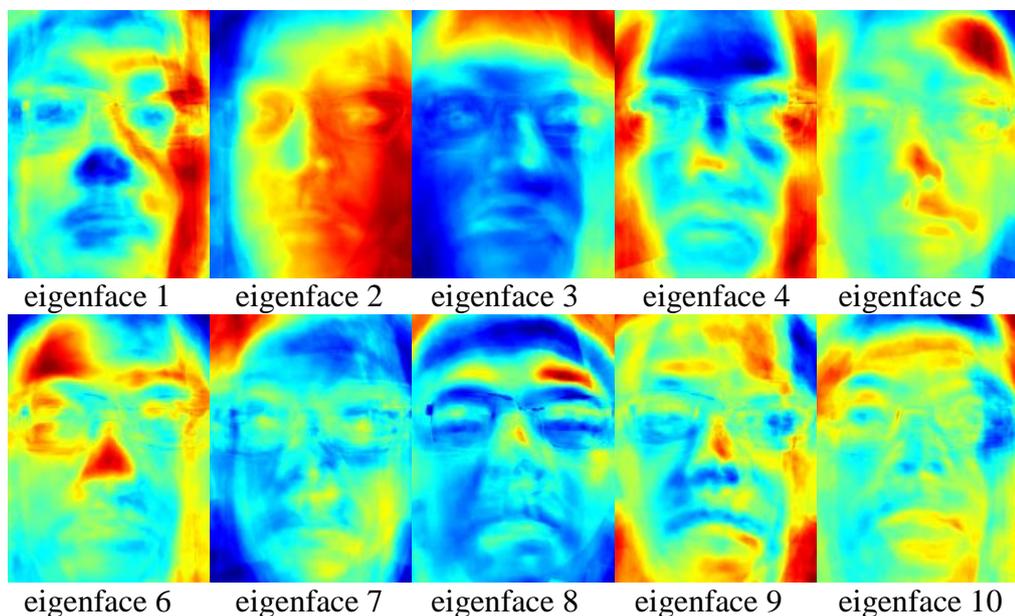
Figure 2.4 Eigenfaces

The eigenface is not visible to human, the face reconstruction is a way to reconstruct human face from eigenfaces to a visual one. The determination of how many eigenvectors are required to process the face reconstruction depends on face dataset. As

shown in Figure 2.5, the most important face features are already recovered with 50 eigenvectors. The eigenfaces could be recovered completely with 110 eigenvectors.

10 eigenvectors 20 eigenvectors 25 eigenvectors 30 eigenvectors 40 eigenvectors

50 eigenvectors 55 eigenvectors 60 eigenvectors 70 eigenvectors 80 eigenvectors

85 eigenvectors 90 eigenvectors 100 eigenvectors 110 eigenvectors
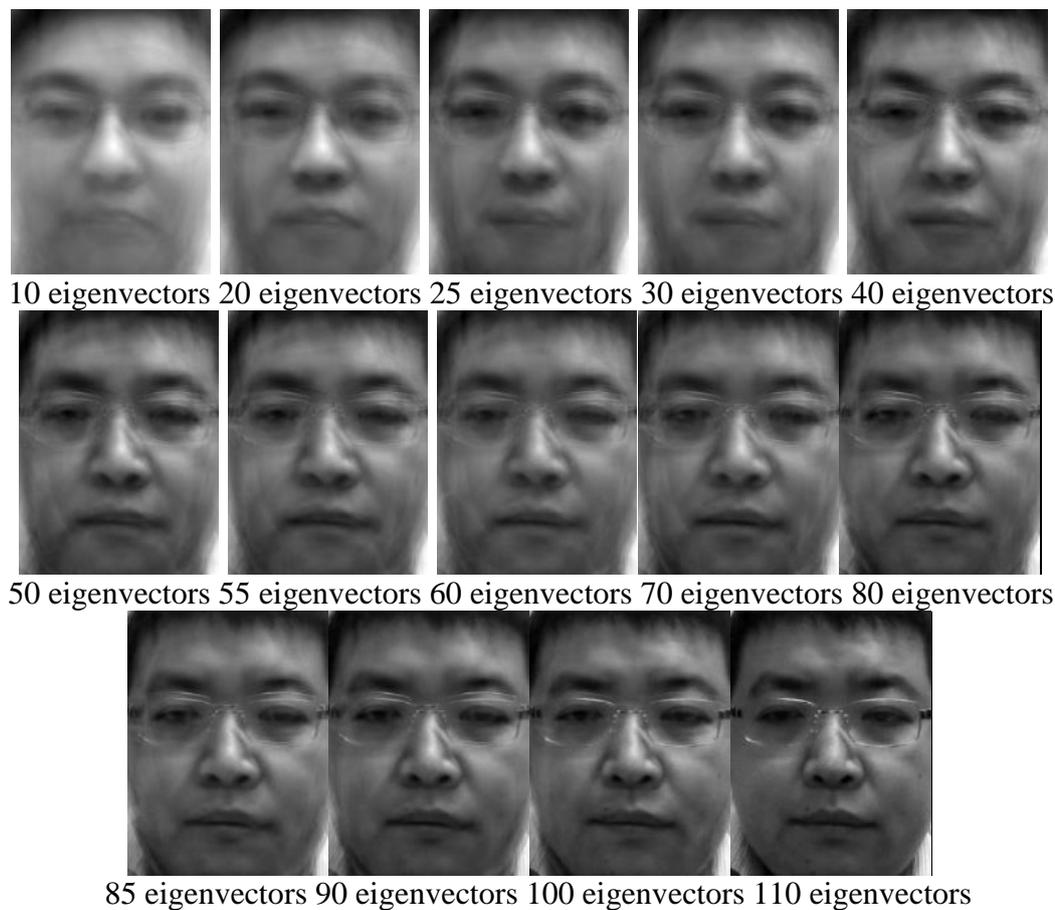
Figure 2.5 Eigenface reconstruction with various numbers of eigenvectors

The average face as shown in Figure 2.6 is the average of training images, it is used to get the mean-subtracted images. For example, an image set X has N images x_i ($i = 1, ..., n$) such as $X = \{x_1, x_2, x_3, ... x_N\}$, $\bar{X}$ is average face of the training images.

Figure 2.6 The PCA based average face

The relationship between eigenvectors and eigenvalues in a given matrix A is shown in equation (2.2),

$$AX = \lambda X \tag{2.2}$$

where λ is the eigenvalue and X is the eigenvector of the matrix A.

By using equation (2.2), we can generate equations (2.3) and (2.4),

$$\boldsymbol{Ax - \lambda x = 0} \tag{2.3}$$

$$(A - \lambda I)x = 0 \tag{2.4}$$

where A is the $n \times n$ matrix, I is the $n \times n$ identity matrix.

Covariance is to measure two or more matrixes which are same or not. The PCA finds the optimal linear least-squares representation shown in equation (2.5),

$$cov(X, Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)} \tag{2.5}$$

where X and Y are the vectors, $\bar{X}$ and $\bar{Y}$ are their means respectively.

A $n \times n$ image x_i is represented by,

$$X = \{x_1, x_2, x_3, \dots, x_n\} \tag{2.6}$$

where X is a collection of image set.

To find patterns by using PCA for a set of images, an image matrix (*iMatrix*) could be created by putting all the image vectors (*iVector*) together. Because each image vector is n^2-dimension, so it has n^2 *iVectors* which is shown in (2.7):

$$iMatrix = \begin{pmatrix} iVector\ 1 \\ iVector\ 2 \\ iVector\ 3 \\ iVector\ 4 \\ . \\ . \\ . \\ iVector\ n \end{pmatrix} \tag{2.7}$$

where *iVector* is the image vector and *iMatrix* is the image vector matrix which consists of n eigenvectors.

In 1991, Matthew and Alex implemented a formula as shown in equation (2.8) for the image training set of face images $x_1 \dots x_n$ (Turk & Pentland, 1991),

$$Cov = \frac{1}{n}\sum_{i=1}^{n}\Phi_i\Phi_i^T \tag{2.8}$$

where Φ_i is eigenvector represented by $(X_i - \bar{X})$ and $(Y_i - \bar{Y})$ from equation (2.5), n is the number of images in the training set.

(2.9) and (2.10) are used to calculate the image covariance matrix,

$$S_T = \sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T \tag{2.9}$$

$$G_T = \frac{1}{n}\sum_{j=1}^{n}(A_j - \bar{A})^T(A_j - \bar{A}) \tag{2.10}$$

where S_T and G_T are the covariance matrices, μ and $\bar{A}$ are the mean of the image training set x_i and A_j.

Taken the equation (2.8) into consideration, we produce the equation (2.11),

$$Cov = AA^T \tag{2.11}$$

where A is the matrix $A = [\Phi_1, \Phi_2, \Phi_3, \cdots, \Phi_n]$.

To calculate the eigenvectors and eigenvalues, we need to use a computationally feasible method to find these eigenvectors which as shown in (2.12). Assume that the eigenvectors v_i of AA^T, therefore,

$$AA^Tv_i = \mu_i v_i \tag{2.12}$$

where v_i is eigenvectors and μ_i is eigenvalues.

Following these analysis, we construct the $n \times n$ matrix $MA = AA^T$, where $MA_{nn} = \Phi_n\Phi_n^T$, and find the n eigenvectors V_i of matrix MA. Those vectors determine linear combinations of the MA training images to the eigenfaces E_i as shown in equation (2.13),

$$E_I = \sum_{k=1}^{n}v_{ik}\Phi_k \tag{2.13}$$

where eigenfaces E_i and $i = 1, 2, 3, \cdots, n$

Once a new face image I is projected into eigenface, and the weight can calculate in equation (2.14)

$$W_k = U_k^T(I - \psi) \tag{2.14}$$

17

where $k = 1, 2, 3, \dots n$. The weight from a projection vector is shown in equation (2.15),

$$\Omega^{\mathrm{T}} = \ [W_1, W_2, W_3, \cdots, W_n] \qquad (2.15)$$

Equation (2.15) describes the contribution of each eigenface in representing the input face image, treating the eigenfaces as a basis set for face images. The projection vector is then used in a standard pattern recognition algorithm to identify which of a predefined face classes, if any, best describes the face. The face class Ω_k can be calculated by averaging the results of the eigenface representation over a small number of face images of each individual. Classification is performed by comparing the projection vectors of the training face images with the projection vector of the input face image. This comparison is based on to find the closest square distance (Euclidean Distance) between the PCA subspace classes from face database and the PCA subspace class from the input face image as shown in equation (2.16). The Euclidean Distance classification is a decision rule which is based on the Euclidean distance between the sample projections class Ω and the projection of the k-th class mean Ω_k,

$$\varepsilon_k = \ \|(\Omega - \Omega_{\mathrm{k}})\| < \Delta \qquad (2.16)$$

where ε_k represents Euclidean Distance, Ω_k is a vector described the Kth faces class. If the Euclidean Distance is less than Δ (Threshold Value), then the new face is classified to class k.

The eigenvector can be calculated efficiently using Singular-Value Decomposition (SVD) and they are statistically determined by covariance matrix. The third module is to identify the face in a normalized image (Moon & Phillips, 2001). However, the PCA could only capture the invariance with the limitation of the information provided in the training data.

In Figure 2.7, the overall workflow of how face recognition works via using PCA algorithm. There are two pathways to be considered. The first pathway starts from input image, which means this image will be used for recognition purpose. The second pathway starts from face database, where all the source faces are stored and trained. The recognition process is to find out which image in the face database matching with the input image. The decision making of matching or not matching will be based on the Euclidean Distance between the testing class and the training class.
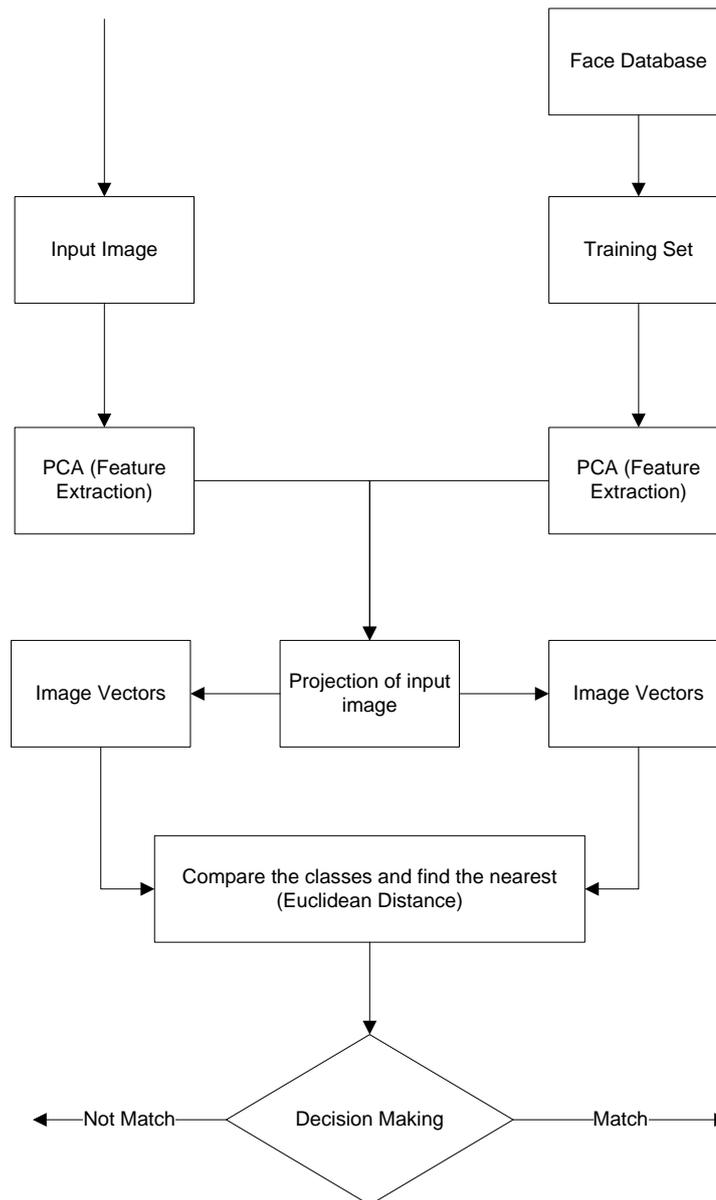
Figure 2.7 The PCA workflow

2.4.2 Independent Component Analysis (ICA)

Independent Component Analysis (ICA) is a PCA related method (Yang et al., 2004). ICA has the following potential advantages over PCA: (1) ICA provides a better probabilistic model of the data that identifies where the data focus in n-dimensional. (2) It uniquely identifies the mixing matrix. (3) It may rebuild the data better than PCA in the presence of noise by finding an unnecessary orthogonal basis. (4) It not only focus on the covariance matrix but also sensitive to high order (third order) statistics in data (Bartlett et al., 2002). Bartlett and Draper proposed using ICA for face recognition and found that ICA can perform remarkable result if the Euclidean distance is used. Liu and Wechsler also state that the ICA has the power to solve the problem of unknown source

separation. However, the computational time in ICA is longer than PCA (Liu & Wechsler 1999; Comon 1994; Hyvarinen & Oja 1997; Karhunen et al., 1997).

2.4.3 Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) which is known as Fisher Discriminant Analysis (FDA) as shown in Figure 2.8 (Martinez & Kak, 2001) was proposed to find the most discriminating features in the face classes (Hanmandlu et al., 2013). It is considered as one of the most successful algorithms in Face Recognition (Lu et al., 2003; Yu and Yang, 2001).

Because the Fisherfaces method does not capture the illumination as the eigenfaces does, therefore, the Fisherfaces have no chance to learn the illumination. As a result, if the Fisherfaces method was trained from an input image which was taken in a good lighting condition, then the method will find the wrong components with the recognized faces in bad lighting condition.
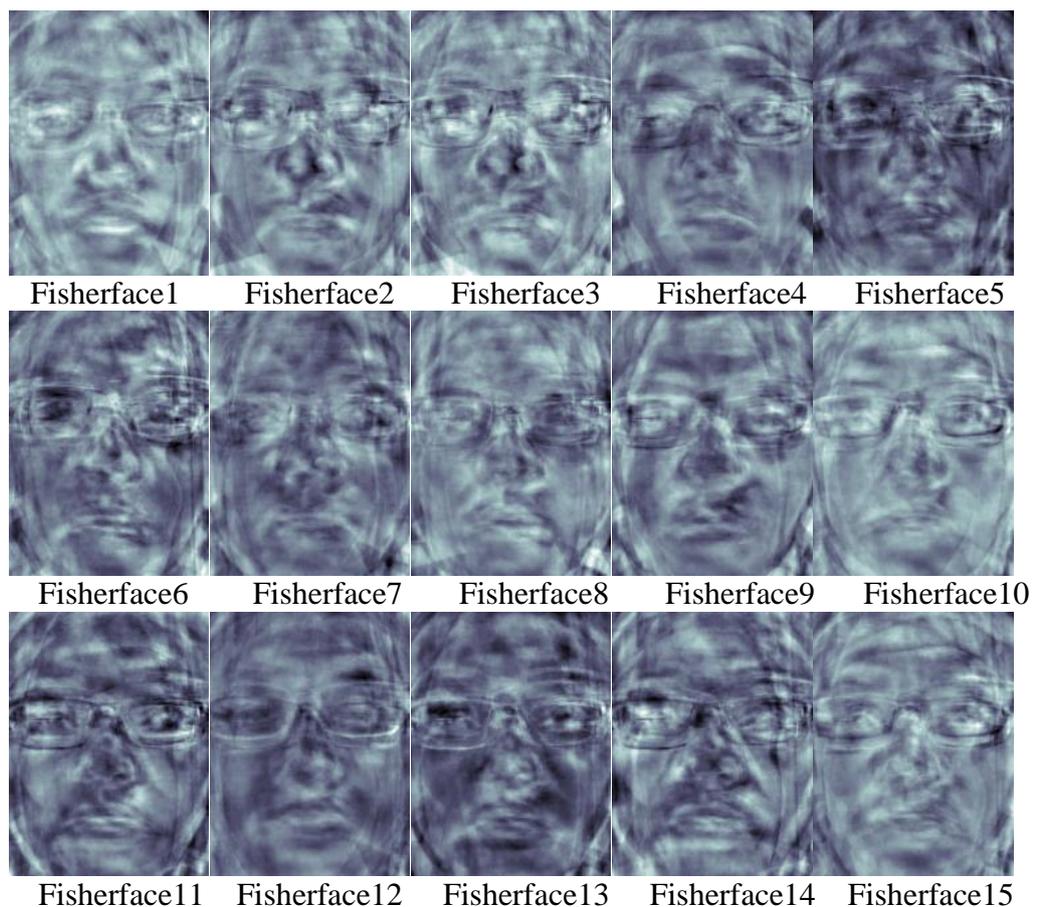
Figure 2.8 Fisherfaces

The Fisherfaces also allow face reconstruction by using Fisherface method which is shown in Figure 2.9. However, they are not good as the eigenface reconstruction does. There is no expectation of a nice reconstruction to original face.

Reconstruction from Fisherfaces 1	Reconstruction from Fisherfaces 2	Reconstruction from Fisherfaces 3	Reconstruction from Fisherfaces 4	Reconstruction from Fisherfaces 5
Reconstruction from Fisherfaces 6	Reconstruction from Fisherfaces 7	Reconstruction from Fisherfaces 8	Reconstruction from Fisherfaces 9	Reconstruction from Fisherfaces 10
Reconstruction from Fisherfaces 11	Reconstruction from Fisherfaces 12	Reconstruction from Fisherfaces 13	Reconstruction from Fisherfaces 14	Reconstruction from Fisherfaces 15

Figure 2.9 Face reconstruction from fisherfaces

The average face shown in Figure 2.10 calculated from training set.

Figure 2.10 The LDA based average face

There are two methods defined in LDA. Between-class scatter matrix (S_B) represents the scatter of features around the overall mean for all face classes shown in equation (2.17),

$$S_B = \sum_{j=1}^{c} N_j (\mu_j - \mu)(\mu_j - \mu)^T \qquad (2.17)$$

where N_j is the number of training samples in class j, c is the number of distinct classes, μ_j is the mean vector of samples belonging to class j.

Within-class scatter matrix (S_W) represents the scatter of features around the mean of each face class shown in equation (2.18),

$$S_W = \sum_{i=1}^{c} \sum_{j=1}^{N_i} (x_j^i - \mu_i)(x_j^i - \mu_i)^T \qquad (2.18)$$

where x_j^i is the Jth sample of class i, μ_i is the mean of class i, c is the number of classes and N_i is the number of samples in class i.

2.4.4 Elastic Bunch Graph Matching (EBGM)

Elastic Bunch Graph Matching (EBGM) describes faces using Gabor filter responses in certain facial landmarks and a graph described the spatial relations of those landmarks (Ahonen et al., 2006) shown in Figure 2.12

Figure 2.11 Elastic Bunch Graph Matching (EBGM)

A bunch graph consists of two characteristics. Its qualitative structure as a graph (a set of nodes plus edges) as well as the project of corresponding labels (jets and distances) for one initial image is manually marked, whereas the bulk of the bunch graph is extracted semi-automatically from sample images by matching the developing bunch graph to them, the most matched image will be considered as the matched face (Wiskott et al., 1999).

EBGM algorithm consists of five important stages: Normalization, Landmark Localization, Face Graph Creation, Similarity Measurement and Decision Making, shown in Figure 2.12.

Figure 2.12 Flowchart of the EBGM

2.5 Face Recognition Approaches

2.5.1 Appearance-Based Approach

Most current face recognition techniques are only referred to the appearance-based recognition work since late 1980's (Draper et al., 2003). The appearance based approach uses the holistic texture features and it is applied to either whole-faces or specific regions in a face image (Delac, Grgic & Grgic, 2005; Zhao, 2003). For example the algorithms such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), Linear Discriminant Analysis (LDA), Support Vector Machines

(SVM), Neural Networks and Statistical Methods are working well for classifying frontal face in face recognition (Heisele et al., 2001; Rowley et al., 1998).

However, those algorithms are highly sensitive to translation and rotation of the target face. Therefore, before classifying the face, an alignment stage can be added to avoid this defect. The face alignment will be used to computing correspondences between two images. The correspondence determined by the prominent points in the face such as eyes and corners of the mouth is shown in Figure 1.1.

2.5.2 Module-Based Approach

There are enormous face recognition techniques referring to module-based recognition. The Module-Based approach uses geometric facial features such as mouth, eyes, nose, brows, cheeks etc) and geometric relationships between them (Brunelli & Poggio, 1993).

In fact, the idea of module-based approach is to avoid the defect of pose changed by allowing a flexible geometrical relation between the face data in the classification process such as EBGM. The Module-Based Approach includes the algorithms such as EBGM, 3D Morphable Model, 3D Face Recognition and Dynamic Link Architecture (DLA) (Moghaddam et al., 2003).

2.5.3 Advantages and Disadvantages

There are few advantages and disadvantages of using appearance-based approach and module based approach.

In appearance-based approach, the information of the images will not be destroyed because the concentration only focused on limited regions or pixel points. The appearance-based approach also has faster processing speed. However, it does not perform effectively once the target face changes the pose, in the lighting of changing environment.

In module-based approach, it is designed to recognize a target face with pose change and lighting change. Moreover, the algorithms in module-based approach are also designed to distinguish twins. However, the obvious disadvantage of using module-based approach is the difficulty of allocating the facial feature automatically. The processing speed is slow and it is not suitable for real-time face recognition purpose.

2.6 Evaluations and Comparisons of Algorithms

There are two types of face recognition algorithms: Procedural Algorithms and Training Algorithms. The Procedural Algorithms includes DLA, and EBGM. The Training Algorithms include PCA, LDA, ICA, SVM, and Neural Networks. The Procedural Algorithms most likely require operator involved to manually allocate the facial points. The Training Algorithms are machine learning process. The identification stage is relying on the comparison of the testing classes and the training classes.

In this research, the PCA algorithm has been selected for the experiment with the below reasons:

1) PCA is more efficient comparing to LDA, ICA and SVM. However, this is closely related to the testing environments. There is no significant standard to distinguish which the algorithm is completely better than another.
2) PCA uses less computational time comparing to ICA and LDA when Euclidean Distance is used.
3) PCA supports multiple input images where LDA only supports single input image (Kim et al., 2007).

Although face recognition algorithms have been studied for a long time, there is little agreement for the performance and accuracy.

The comparison of the definition, pros and cons of the PCA algorithm and the LDA algorithm are presented in Table 2.1.

Table 2.1 Comparison of PCA and LDA

	PCA	LDA
Definition	Principal Component Analysis (PCA) is referred as Eigenface which is face recognition used to reduce the dimension of the face image data by means of density.	Linear Discriminant Analysis (LDA) is referred as Fisherface which is a statistical approach for classifying samples of unknown classes based on training samples with known classes. The result comparison between input image class and trained samples class will be used for decision making.
Pros	Fast process speed and less	Lower error rate.

	amount of memory required. Small face image size.	
Cons	It is highly sensitive to lighting, head rotation. Because the eigenfaces also record the lighting information as discussed in Section 2.3.1. The PCA face images contain limited information.	Heavy storage. Highly sensitive to self-shadowing, specularite (finsherface) and noise.

2.7 Moving Object Removal (MOR)

The Moving Objects Removal (MOR) is a way to remove the moving objects from the input video by using frame differencing technique. The frame differencing is a technique which checks the difference between two video frames (Prabhakar et al., 2012). The frame differencing algorithm (Jain and Nagel, 1979; Haritaoglu et al., 2000) is used to separate both static objects and dynamic objects. The static objects are considered as not moving and the dynamic object is considered as moving. Two important subjects needed to be studied in this research are moving objects detection and moving object removal.

2.7.1 Moving Object Detection (MOD)

Moving Objects Detection (MOD) in vision is a task to find the moving objects in a video frame sequence. As the movement of an object in a video frame sequence caused by pixels changing, therefore, the easiest way to extract the moving objects from a video frame sequence is to use frame difference algorithm.

The frame difference could be calculated as shown in equation (2.19). Once the objects moving in the video frames, the changes of those pixels will be presented by frames difference, is shown in Figure 2.15.

$$Diff_{(f1,f2)} = |F_{(2)} - F_{(1)}| \qquad (2.19)$$

The video frames shown in Figure 2.13 consist of three people, two people did not move and one person is moving pass behind the target user.

Figure 2.13 Sample frames of the original video footage

After applying the frame subtraction, the output video frames sequence is divided into two areas as shown in Figure 2.14. The first area is the black colour area. The black area is represented by the static objects which means there is no changes between the selected two frames and the result pixel subtraction will be 0. The second area is white colour area. An object is moving in the video frames, the result of the frame subtraction will be either greater than 0 or less than 0. In addition, the pixel changing direction also shows the possibility of the moving direction as shown in Figure 2.14.
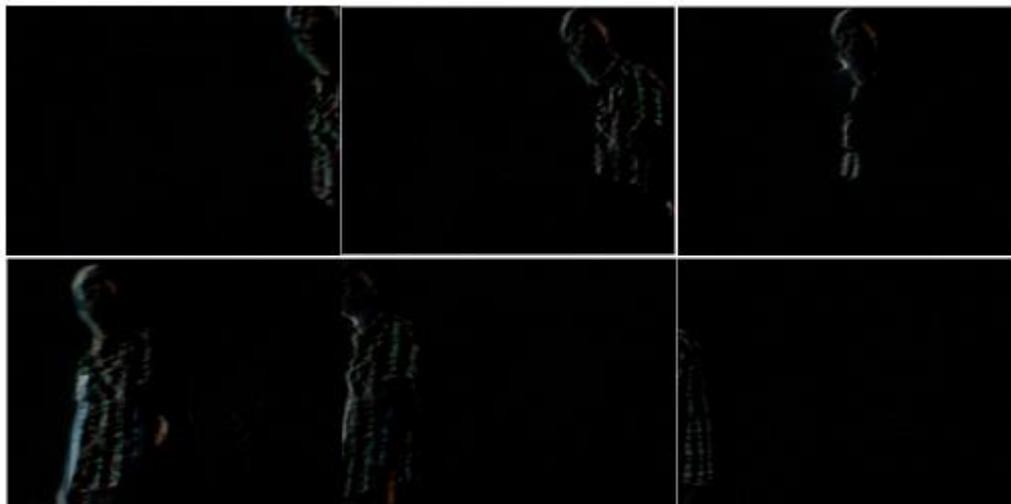
Figure 2.14 Sample video frames of the MOR

2.7.2 Moving Object Removal (MOR)

In this thesis, detecting a moving object is not enough. After detection stage, we need to apply moving object removal. In other words, we have to removal the moving object and update the background. To achieve this purpose, the simple frame subtraction will be not enough. Instead, we have to compare each of the frame pixels between two frames to indicate and locate the pixel changes, the output of the MOR is shown in Figure 2.15.

Figure 2.15 Sample video frames after the MOR

Chapter 3 Research Methodology

The research methodology explains key research questions developed from literature review in the field of face recognition. It is a key factor to lead the success of this research. In this thesis, the research has been consistently implemented and followed. Seven steps are required to be conducted in this research. The first step is to formulate a research problem which is presented in Section 3.3, the second step is to conceptualise a research design which is described in Section 3.4.1, the third step is to construct and instrument for data collection, which is illustrated in Section 3.4.2, the forth step is to identify any limitations of the research which is given in Section 3.5, the fifth step is to predict the research output and state out research expectation in Section 3.6, the sixth step is to apply research methods and gain the experiment result in Section 4.3 and 4.4 and the seventh step is to discuss the result to analyse and evaluate whether or not the research solution can solve the research problems or issues. The accuracy of face recognition is found to be one of concerns in computer vision. As technology advances, more and more face recognition algorithms are discovered and experimented. However, there is no research undertaken with regard to moving object removal (MOR) in face recognition system. To the best of my knowledge, this is the first time face recognition in complex environments is considered with the solution of MOR.

3.1 Introduction

Research methodology is a way to systematically solve the key research problems deployed from literature review in the field of face recognition. The challenges and current achievements of face recognition in complex background are presented in related studies section. The accuracy of face recognition is found to be the main concern in face recognition especially in a complex environment. There are many face recognition algorithms that experimented and presented in digital image processing. There are considerable similar researches of face detection in complex background. However, there is no research with regard to apply MOR on face recognition system so as to increase face recognition precision and recall rate. To the best of my knowledge, this is the first time face recognition in complex environments taken into consideration.

In this chapter, the research problems and hypothesises are discussed in details with relevance to the research gap identified from the literature review in Chapter 2. The main purpose of this chapter is to identify the research problems and explain research design, data collection, and experiment test bed.

There are five sections included in this chapter which cover the details of the research problems, hypotheses and relevant research designs. In Section 3.2, it covers the related studies of the research background. In Section 3.3, it identifies the research questions and hypotheses. In Section 3.4, a proper research design is presented and relevant data requirements are presented. In Section 3.5, the research limitations are clarified and the expected outcomes are presented in Section 3.6.

3.2 Related Work

In this section, the concept of face recognition is introduced and the previous research is presented.

Face recognition is a way to automatically identify and verify a person from a digital image or a frame of a digital video. Many image processing algorithms were proposed and experimented for face recognition. Some popular algorithms such as PCA, LDA and ICA for automatic face feature extraction and real-time face recognition are from appearance-based approach. The face recognition algorithms such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), 2-Dimensional Principal Component Analysis, and Independent Component Analysis (ICA) have been widely studied in the past decades. The algorithms from module-based approach such as

Elastic Bunch Graphic Matching (EBGM), 3-D Morphable Model and 3-D face recognition have solved the issues of the post changing in face recognition. However, the processing speed of those algorithms is one of the challenges when applied to real-time face recognition.

The researches of the face detection in complex backgrounds have achieved very remarkable results, the typical algorithms with long impact includes adaptive boosting (Adaboost) proposed by Freund and Schapire in 1996 (Chen et al., 2010). It is a machine learning algorithm which provides an effective learning algorithm and strong bounds on generalization performance in object detection (Viola & Jones, 2001; Manjunath et al., 1992; Wang & Yuan, 2001; Georghiades et al., 2001; Huynh-Thu et al., 2002; Pai et al., 2006; Li & Wang, 2013). Some researchers have experimented the face detection in complex background such as Lee et al., 1996 used motion and color information to locate human face automatically (Lee et al., 1996), Lin, 2007 and Kovac et al., 2003 presented the face detection using skin color clustering such as YCbCr (Lin, 2007; Kovac et al., 2003).

In general, face recognition in complex backgrounds is studied in four steps: 1) Locate face based on the feature and template (such as eyes, nose and mouth); 2) Use curve fitting to segment the face in image (face shape based); 3) Frontal face normalization; 4) Recognize faces with eigenface method. However, the accuracy of face recognition still yet takes into consideration since the faces to be detected and recognized are not suitable for recognition (Wu & Yang, 2004).

Yuan et al. proposed a method to detect frontal face in a complex environment by using image segmentation and face detection. The face detection consists of Valley-like detector and face verification with positive-negative attractor (Yuan et al., 2002). The Valley-like detector distinguishes the differences between surrounding pixels and core pixels in a mosaic image. The positive-negative is used to identify the positive attractor such as eyes and mouth, the negative attractor such as the surround pixel. However, all of the previous studies focused on the face detection and recognition without filtering the related face or not related face.

3.3 The Research Questions and Hypothesis

As mentioned in Chapter 1, the purpose of this research thesis is to propose a scheme of human face recognition in complex environments and the main objective of this thesis is to evaluate the current issues and prove the proposed scheme can increase the face recognition accuracy in complex environments such as office, reception, station and casino etc. In those environments, we set the camera always points in the fixed direction and the only changes are the moving objects such as the walkers and moving vehicles.

According to the purpose of the research, there are three points needed to be taken in consideration. The first point is to understand face detection and face recognition. The second point is to understand how to simplify the complex environments. The third point is the measurement. How we measure the result indicates that to what extent the proposed scheme can increase the accuracy of face recognition.

Face recognition is a type of biometrics that can recognize a human face by analysing and comparing the face patterns, which is used to automatically identifying or verifying a person from a digital image or a video frame. There are many algorithms in face recognition such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Independence Component Analysis (ICA) and Elastic Graph Bunch Matching (EGBM). The details of each algorithm were discussed in Chapter 2.

The precision, recall and F-measure will be calculated based on the experimental results. The experimentation is divided into two parts, the first one is to apply the face recognition without moving object removal and the second one is to apply the face recognition with moving object removal. In those experiment environments, precision, recall and F-measure comparison will be used to evaluate the proposed solution.

This research thesis attempts to identify a successful solution for the following research question:

"Whether the face recognition accuracy in complex environments can be increased by applying moving object removal technology?"

This research attempts to find the solutions for the following questions developed from the above main question:

1) *What are the face recognition algorithms and which algorithm is the best one for our research?*

2) What is the moving object removal (MOR) and how could it help to improve the face recognition accuracy in complex environments?

3.4 Research Design and Data Requirements

3.4.1 Research Design

The research design is explicitly same as this thesis structure presented in Figure 1.1. The main objective of this thesis is to find a new way to increase the precision of face recognition in complex environments.

In this research thesis, it consists of 5 main parts: motivation and objectives, literature review, research methodology, experiments, results, discussion and conclusion. There are two experimental methods were implemented in this research is shown in Figure 3.1. The first experiment process includes: face detection and face recognition. The second experiment process includes: moving object removal, face detection and face recognition.

The first experimental method consists of two stages. The first stage is face detection. In this thesis, the OpenCV cascade classification was implemented for face detection. The related study of face detection using cascade classifier was presented in Section 2.1. The second stage is face recognition and the PCA algorithm was used in face recognition as suggested in Section 2.6.

The second experimental method consists of three stages. The first stage is MOR. In this stage, the frame difference was conducted and the detail study was presented in Section 2.7 Chapter 2. The second stage is face detection. In this thesis, the OpenCV cascade classification was implemented for face detection. The related study of face detection using cascade classifier was presented in Section 2.1. The third stage is face recognition and the PCA algorithm was used in face recognition system as suggested in Section 2.6.
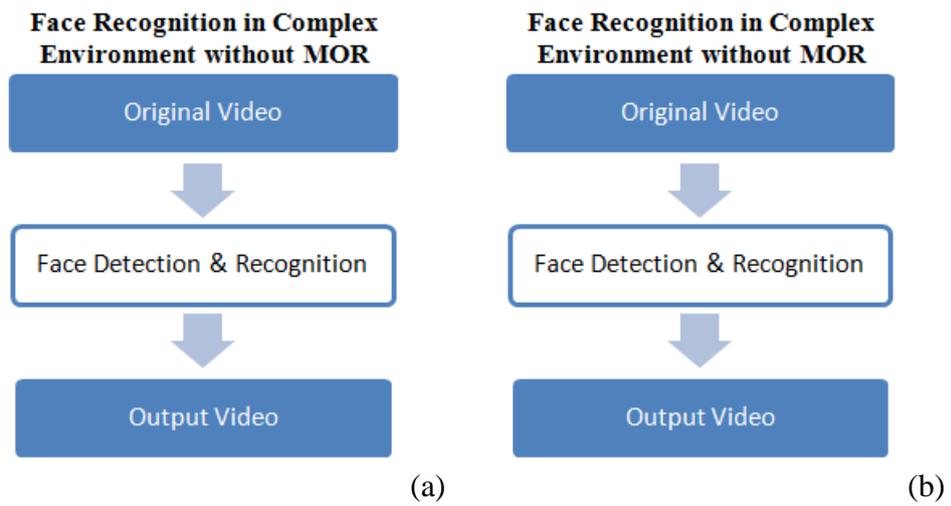
Figure 3.1 Two research workflows

As shown in Figure 3.1, the face recognition in complex environments with and without MOR are the two critical experiments. There are three important phases, needed to be taken in consideration which is shown in Figure 3.2. The first phase is MOR, the second phase is face detection and the last phase is face recognition.
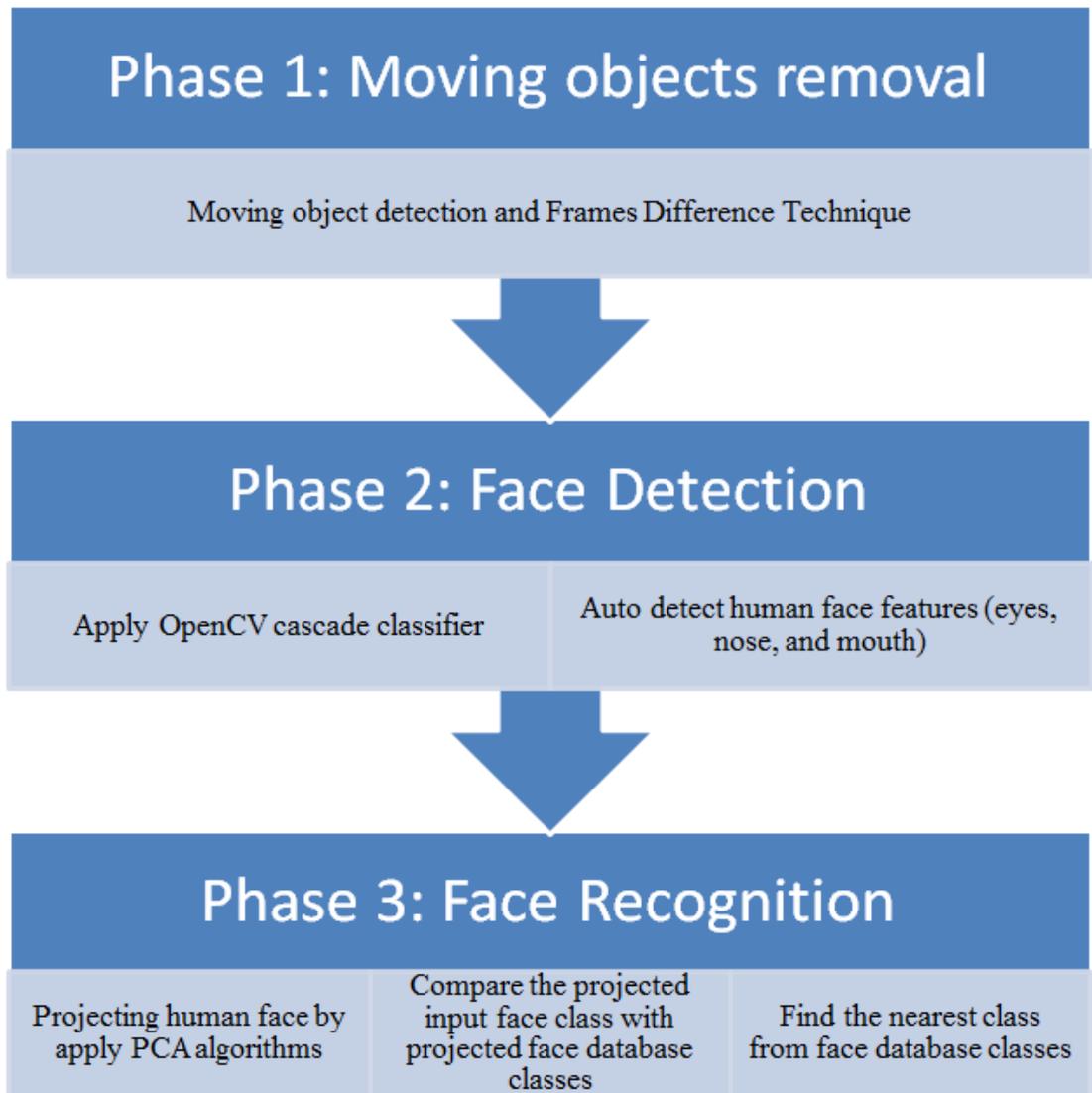
Figure 3.2 Three main phases included in the proposed scheme

Phase 1: Moving Object Removal

In MOR phase, we have to perform two tasks, the first one is the moving object detection and the second one is the moving object removal. The detailed study was presented in Section 2.7.

Moreover, the dynamic objects could also be divided into two types. The first type is represented as low speed motion object and the second type is represented as high speed motion object. As shows in Figure 3.3, the high speed motion objects are considered as background moving objects.

Figure 3.3 An image with high speed object and low speed object

After MOR phase, we will apply face detection in face recognition phases.

Phase 2: Face Detection

In face detection, the face detection face features must be extracted. The face features are defined as face objects such as eyes, nose, and mouth. To automatically detect those features, the OpenCV Cascade Classifiers has been used in this thesis. The OpenCV Cascade Classifiers was proposed by Paul Viola and Michael Jones in 2001 (Viola & Jones, 2001). It is also known as Viola-Jones object detection framework. The OpenCV Cascade Classifier is not only able to detect whole face and its features, but also human body shown in table 3.1. The second stage is decision making, it detects whether those detected features can group as a human face or not. The frontal face classifier was used to evaluate whether a frontal face detected or not. As a result, even if some features were detected, it is not considered as a face.

Table 3.1 OpenCV cascade classifiers

Face Features	Harr Feature Based Cascade Classifier	Size	Description
Eyes	haarcascade_eye.xml	20×20	Frontal eye detector.
Eyes with glasses	haarcascade_eye_tree_eyeglasses.xml	20×20	Tree-based frontal eye detector with better handling of eyeglasses.
Frontal face	haarcascade_frontalface_alt.xml	20×20	Stump-based gentle adaboost frontal face detector.

Frontal face	haarcascade_frontalface_alt_tree.xml	20×20	Stump-based gentle adaboost frontal face detector. This detector uses tree of stage classifiers instead of a cascade.
Frontal face	haarcascade_frontalface_alt2.xml	20×20	Tree-based gentle adaboost frontal face detector.
Frontal face	haarcascade_frontalface_default.xml	24×24	Stump-based discrete adaboost frontal face detector.
Full Body	haarcascade_fullbody.xml		"Haar"-based Detectors For Pedestrian Detection.
Left eye	haarcascade_lefteye_2splits.xml	20×20	Tree-based left eye detector. The detector is trained by 6665 positive samples from FERET, VALID and BioID face databases.
Lower body	haarcascade_lowerbody.xml	19×23	Lowerbody detector.
Eye pair	haarcascade_mcs_eyepair_big.xml	45×11	Eye pair detector computed with 7000 positive samples.
Eye pair	haarcascade_mcs_eyepair_small.xml	22×5	Eye pair detector computed with 7000 positive samples.
Left ear	haarcascade_mcs_leftear.xml	12×20	Left ear (in the image) detector computed with 5000 positive and 15000 negative samples.
Left eye	haarcascade_mcs_lefteye.xml	18×12	Left eye (in the image) detector computed with 7000 positive samples.
Mouth	haarcascade_mcs_mouth.xml	25×15	Mouth detector computed with 7000 positive samples.
Nose	haarcascade_mcs_nose.xml	18×15	Nose detector computed with 7000 positive samples.

Right ear	haarcascade_mcs_rightear.xml	12×20	Right ear (in the image) detector computed with 5000 positive and 15000 negative samples.
Right eye	haarcascade_mcs_righteye.xml	18×12	Right eye (in the image) detector computed with 7000 positive samples.
Head and shoulders	haarcascade_mcs_upperbody.xml	22×20	Head and shoulders detector.
Profile face	haarcascade_profileface.xml	20×20	Profile face detector.
Right eye	haarcascade_righteye_2splits.xml	20×20	Tree-based right eye detector. The detector is trained by 6665 positive samples from FERET, VALID and BioID face databases.
Smile	haarcascade_smile.xml		Smile detector.
Upper body	haarcascade_upperbody.xml	22×18	Upperbody detector.

Phase 3: Face Recognition

In this phase, face recognition is considered in three tasks: feature extraction, projection, and comparison. In face recognition, two sets of class are required. The first class is input face images, also called training class. The second class is the face database images, called test classes.

In feature extraction, the PCA algorithm will be used to extract the most relevant information from an input image and represent that information in a lower dimension space. For instance, a 200×200 dimensional image could generate 40,000 eigenvectors. However, not all of those eigenvectors are related to the face features. The key function of PCA is to reduce the high dimensional to lower dimensional, the purpose is to focus on the most of the useful information from an image and remove the information not useful.

In projection, the PCA will precisely decompose the face structure into orthogonal components as known as eigenface. Each of projected face will be represented as a weighted eigenvector of the eigenfaces.

In comparison, the projected input face image is used to compare with the projected trained face images. The nearest distance between the input image feature vector and trained image feature vectors will be used to measure the matching face.

3.4.2 System Architecture

In order to understand the working flow of the proposed scheme, the flowcharts are one of the important and indispensable components.

In this thesis, we have presented three important flowcharts. The first flowchart is shown in Figure 3.4 as the overall system workflow. The second flowchart as shown in Figure 3.5 is the workflow of face detection and recognition and the third flowchart as shown in Figure 3.6 is for the moving object removal.

In Figure 3.4, the proposed scheme consists of three kinds of input, corresponding processing components and one output. Face recognition which we developed accepts three types of input such as video clips, images and frames of live camera. After the input resource is provided, we will then process the environment by removing all the moving objects and then the processed video will be sent to face detection and

recognition unit. The outputs of this proposed scheme also support video clip output and image output.

In Figure 3.5, the detail of MOR is presented. In this research, we read each pixel in the frameset to identify the changes. Once the changes are found, then the background will be updated.

In Figure 3.6, the detailed workflow of face detection and face recognition is presented. In this research, we used the OpenCV cascade classifier to perform face detection and PCA algorithm with Euclidean Distance classification for face recognition purpose.
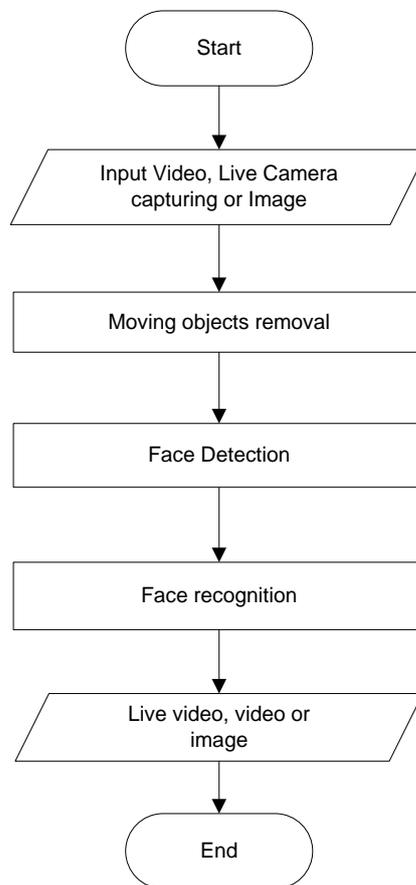
Figure 3.4 Flowchart of the proposed scheme
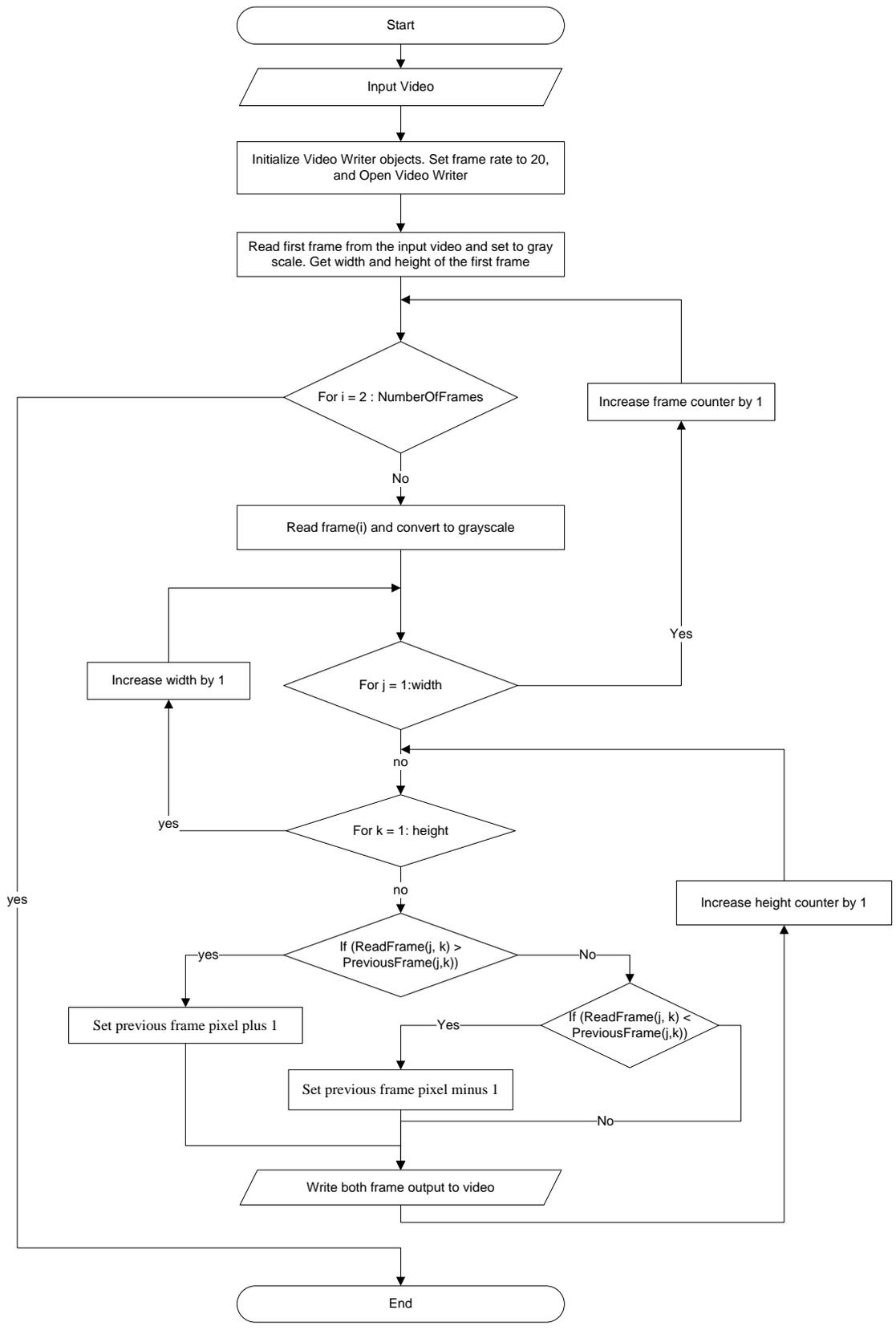
Figure 3.5 Flowchart of MOR
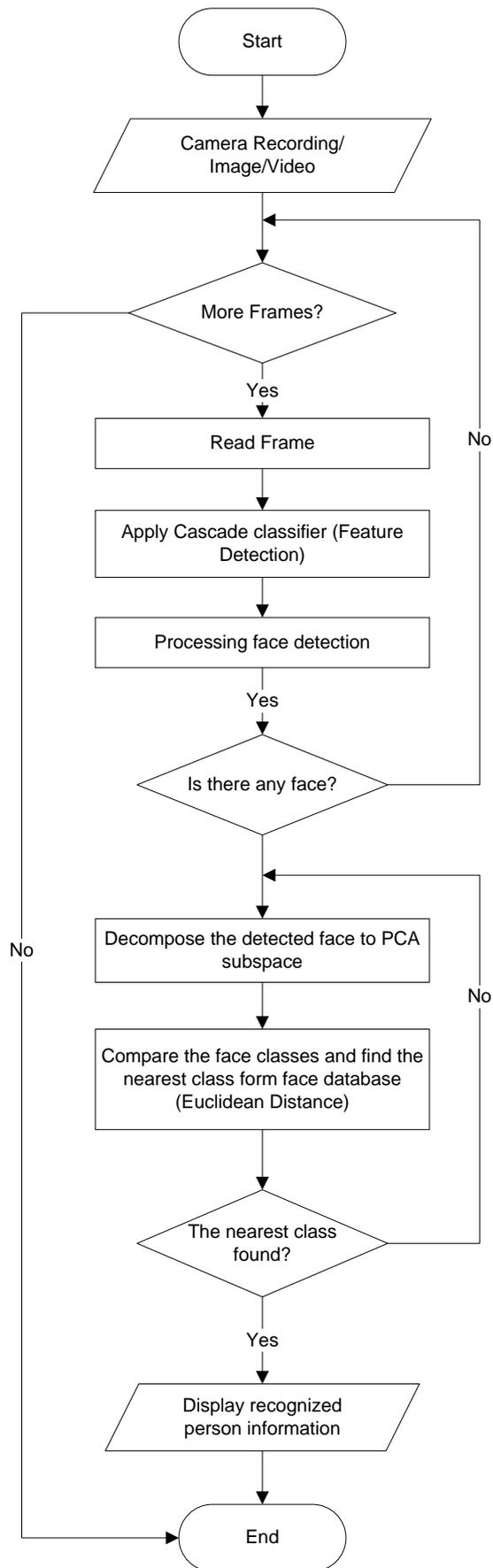
Figure 3.6 Flowchart of face recognition

3.4.3 Data Requirements

In this thesis, the majority data required are input video clips shown in Figure 3.7, and human face images are presented in Figure 3.8 (face dataset).

Figure 3.7 shows the video frames of the chosen samples. Those video clips cover a variety of different environments such as office, reception, and street, etc.

Video 1 face recognition in office environment

Video 2 face recognition in reception environment

Video 3 face recognition in reception environment
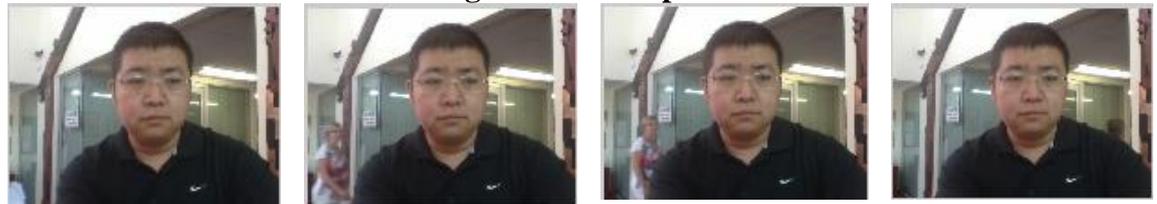
Video 4 face recognition in street environment

Video 5 face recognition in street environment

Figure 3.7 Face recognition in different environments

Figure 3.8 shows the face images dataset. It consists of 100 shots from 20 people and 5 shots per person.

	Front	Up	Down	Left	Right
Yan					
Bobby					
Cui					
Guang Yu					
JianYu					
Sumeet					

Figure 3.8 Face dataset

3.5 Limitations of the Research

In this research thesis, there are three major limitations: camera resolution, computer processing speed, and testing environments.

The first limitation is the camera resolution. Our entire research was conducted using the laptop build-in camera, the resolution of the video is low, and it may affect the precision of our recognition.

The second limitation is the processing speed. As this research involves heavy image processing and video processing, the memory and CPU processing speed is become as one of the factors affecting the process speed.

The third limitation is the testing environments. It is infeasible to perform the research experimentation in all kinds of environments.

3.6 Expected Outcomes

The main expected outcomes are the anticipated results developed in Section 3.3. From the proposed experiments, the analysis of face detection, face recognition and moving object removal is intended to be carried out. Thus, from literature review, the expected outcome is from understanding of the face detection, face recogniti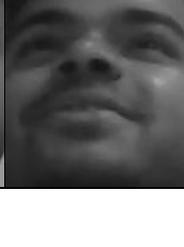on algorithms and moving objects detection and removal that can lead to evaluate the proposed scheme by applying those research methodologies.

In the next section, two main categories of experiments are grouped. The first category is to perform face recognition without MOR and the second category is to apply face recognition to the videos with moving object removal.

In this thesis, 5 videos were used for our experiments. Each of the exemplar video is presented by a set of frames to demonstrate the differences between before and after the proposed scheme is applied.

Chapter 4 Research Findings

4.1 Approach

In this thesis, we propose a scheme of human face recognition in a complex environment with the camera always pointing in the fixed direction. In such a environment, the only change is the moving objects such as pedestrians and moving vehicles etc.

In this section, two critical contributions are presented. The first contribution is face detection and recognition, the OpenCV cascaded classifier was used as detector and recognizer. The second contribution is moving object removal (MOR). We use frame difference to detect moving objects and remove them from background. According to our assumption, the background of our target environment will be changed frequently. However, the background changes will not affect our purpose since the scheme is used to a fixed camera with moving objects in that environment. We use face recognition on those post-processed videos, namely after irrelevant moving object removal.

4.2 Experimental Test Bed

The experimental environments are based on a standalone PC with a build-in camera. The CPU is an Intel I7 with 2.40 GHz and 6G memory. The graphic card is AMD Radeon HD 7670M with 4G graphics memory. The face detection and recognition prototype is developed in Visual Studio 2012 using VC++. The MOR is implemented based on Matlab R2011b.

The result of MOR, face detection and recognition are conducted under this environment. We present 5 video footages in this thesis. A face dataset was created by taking photographs of 20 people in different environments with 5 mug shots per person shown in Figure 3.8.

In this section, we demonstrate how our proposed scheme works. There are five video footages including three from an indoor environment and two from an outdoor environment. As shown in Figure 4.2, if the background people are not removed, two faces will be captured by the camera. However, after applying MOR, the face detection and recognition environment becomes simpler and easier. The same issue affects the

indoor video footages shown in Figure 4.3 and Figure 4.4, two faces were detected and recognized while a person was walking behind the target user. The outdoor video 4 in Figure 4.5 and video 5 in Figure 4.6 show that vehicles passing behind the current user can also generate a false result of face recognition.

4.3 Experiments

4.3.1 Experimental Algorithms

In this thesis, the proposed algorithms workflow which is shown in Figure 4.1
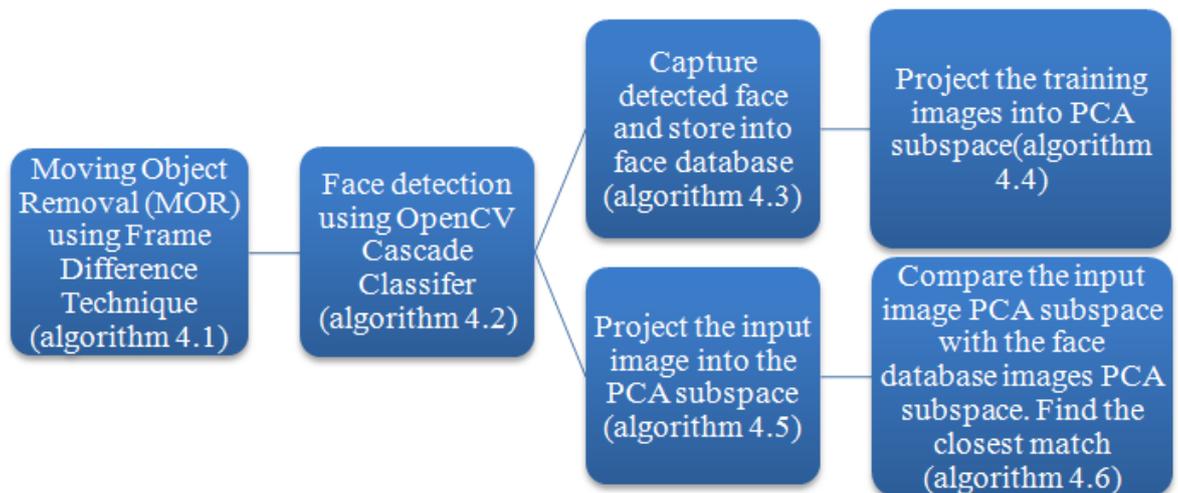
Figure 4.1 Workflow of the proposed algorithms

In Algorithm 4.1, the flow of moving object removal is presented. The input is a video with moving objects, W_i and H_j are the width the height of the video frame. The first frame of the input video will be set as background. The background frame will be used to compare with the rest of the frames from the input video so as to detect the moving objects and update the background by reading each pixel in the frames. The purpose of this operation is to make sure all background changes could be updated correspondingly. After this operation, the moving objects are able to be removed as expected. The output of this algorithm is a video footage without moving objects.

Algorithms 4.1: Moving object removal
Input: Video with moving objects
Output: Video without moving object
Read video file using Matlab mmreader method Set VideoWriter objects

Set W = width and H = height of the first frame

Read first frame from the input video file

For all frames f = 2, …, *MaxFrames*

   Set f = *frame2*

  For all width W_i (i=1, …, *MaxFrameWidth*)

      For all height H_j (j = 1, …, *MaxFrameHeight*)

         If the (i, j) greater than first frame (i, j) then

            Set first frame (i, j) to increase by 1

         Else the (i, j) greater than the first frame (i, j) then

            Set first frame (i, j) minus by 1

         End if

      End For

   End For

Write both original frames and processed frames to output video

End For

In Algorithm 4.2, face detection is the compulsory requirement before face recognition (Sung & Poggio, 1998). The OpenCV cascade classifier is used in this algorithm. It has included the frontal face cascade classifier. The output of this algorithm is to draw a rectangle on each detected faces based on OpenCV cascade classifiers.

Algorithms 4.2: Face detection

Input: Real-time captured frame

Output: Detected face with rectangle

Load the OpenCV classifier cascades

Declare face_cascade as haarcascade_frontalface_alt_tree

if system not terminated

{

   Declare capture as cvCaptureFromCamera

   Declare frame as cvQueryFrame(capture)

   Loaded detected face to memory based on frontal face cascade classifier.

   Set the detect area as 40×40

   Declare display window DISPLAY_WINDOW

   Declare pFaceRectSeq

  if face components detected

  {

Get face component from memory

Load pFaceRectSeq to CvRect class

Create CvPoints based on the CvRect

cvRectangle(frame, point1, point2, CV_RGB (204,0,0), 3, 4, 0) Draw a rectangular outline around each detection

cvShowImage(DISPLAY_WINDOW, frame)

   }

}

else clean up and release resources from memory

{

   cvReleaseImage → frame

   research resource from each cascade

}

In Algorithm 4.3, all the detected faces should be captured and normalized. The processed image will be stored. There are two import parameters: original image resolution of input image and resolution of output images. The processing is basically to convert the input image into grey image. Many OpenCV classes/structures/methods are used in this algorithm such as class *Mat*, struct *Size*, class *Rect*, and class *cvtColor*. The *Mat* class is *n*-dimensional dense array class. The *Size* struct is for specifying the $W \times H$ of an image or rectangle. The *Rect* class is for 2D rectangles with the parameters of *x*, *y*, *width* and *height*. The (*x*, *y*) dimension of the *Rect* class represents the coordinates of the top-left corner, and the width and height represent rectangle. The *cvtColor*(const *oclMat&src*, *oclMat&dst*, int *Code*) method is a colour converter which converts image from one colour space to another. It consists of three parameters, *src* (source image), *dst* (destination image) and *code* (colour code such as RGB2GRAY).

Algorithms 4.3: Processing Images

Input: Captured face images

Output: Processed face images

Declare cv: Mat resized

Declare cv: Size originalSize as input image size cv::size originalSize = image.size

Calculate ratio as cardinals (goalSize.height/originalSize.height)

Calculate newImageSize (originalSize.width × ratio, originalSize.height × width)

If (newImageSize.Height not equal to goalSize.Height

Then set newImageSize.Height equal to goalSize.Height

Resize image to newImageSize cv::resize(image, resized, newImageSize)

If the width of the resized image not equal to the width of the goal size

Find the different dif between the width of the goalSize and the width of the newSize

If the different dif less than 0

Store coordinates of a rectangle cv::Rect cRect(absolute(dif)/2, 0, goalSize width, new size height)

Resize image to new rectangle size

Else if the different dif great than 0

width needs to be widened, (goalSize n-dimensional array shape and array type) cv::Mat wid ((Size)goalSize, (Type)resized.type)

Store coordinates of a rectangle cv::Rect widRect((X)diff/2, (Y)0. (Width)goalSize width, (Height)goalSize height)

Get the rectangle at the centre that matches the size of resized cv::Mat widCenter=widened(widRect);

Copy resized matrix to widCenter matrix Resized.copyTo(widCenter);

End If

End If

Cv::resize(image, resized, goalSize);

Cv:Mat greyFrame;

Convert frame colour to grey colour cv:cvtColor(resized, greyFrame, CV_BGR2GRAY);

Return Grey Frame

In Algorithm 4.4, all of the processed images need to be trained before face recognition, as well as called training set. In this algorithm, the key part is to apply PCA algorithm to compute eigenspace. A method *cvNormalize* has been applied to normalized type of L1, which is the sum of all the absolute values. *cvEigenDescomposite* is applied to calculate the PCA space.

Algorithms 4.4: Training images

Input: Face dataset

Output: Normalized face images

Set number of eigenvalues numEigenValues = numTrainedImages – 1;

Declare face image size cvSize faceImageSize;

Get the width and height from the images and assign to cvSize structure.

faceImageSize.width equal to images[0] $\rightarrow$ width; faceImageSize.height equal to images[0] $\rightarrow$ height;

Allocate the eigenVector images eigenvector equal to (IplImage**) cvAlloc(sizeif(IplImage*) multiply numEigenValues);

For each numEigenValue

   Create eigenVector array. eigenVectors[i] = cvCreateImage(faceImgSize, IPL_DEPTH_32F, 1);

   Allocate eigenvalue array. Eigenvalues = cvCreateMat(1, numberEigenValues, CV_32FC1);

   Allocate average image. averageImage = cvCreateImage(faceImgSize, IPL_DEPTH_32F, 1);

   Set the PCA termination criterion. cvTermCriteria calcLimit = cvTermCriteria (CV_TERMCRIT_ITER, numEigenValues, 1);

   Compute average image, eigenvalue, and eigenvectors. cvCalEigenObjects (numTrainedImages, (void*) images, (void*) eigenVectors, CV_EIGOBJ_NO_CALLBACK, 0, 0, &calcLimit, averageImage, eigenvalues $\rightarrow$ data.fl);

   Normalize image. cvNormalize(eigenValues, eigenValues, 1, 0, CV_L1, 0);

Project the training images to PCA subspace projectedTrainingImage = cvCreateMat(numTrainedImages, numEigenValues, CV_32FC1);

Int offset = projectedTrainImage -> step/sizeof(float);

Read all images from face dataset

   Do decomposition cvEigenDecomposite(arrImages[i], numEigenValues, numeigenVectors, 0, 0, averageImage, projectedTrainImage -> data.fl + i × offset);

End loop

Return true (Training completed)

In Algorithm 4.5, all of the captured human face images should be normalized. A method *cvNormalize* has been applied to normalize L1, which is the sum of all the absolute values. *cvEigenDescomposite* is applied to calculate the PCA subspace. We compare the input face image PCA subspaces with that of training set images. The nearest neighbour will be selected as the recognized target object.

Algorithms 4.5: Face recognition
Input: Test human face image and a human face dataset
Output: Recognized image from the dataset

Detect the human face
 	
If face detected
 	
   TrainingImage::processImage(cv::Mat& image) Training stored face images (Refer to Algorithm 4.4).
 	
     Normalize images using *cvNormalize*.
 	
     Convert input face into Iplimage type
 	
     Apply *cvEigenDecomposite* to reflect the collected images into PCA space.
 	
     Compare PCA vector with that of the training set
 	
     If the distance between training set class and sample class is less than threshold
 	
        Display the name label
 	
     End if
 	
   End if

In Algorithm 4.6 the calculation of the Euclidean Distance between input image PCA feature vectors and trained that of images. The data pointer for float matrix (*data.fl*) array implemented to hold all classes of the training images from training set. While compared the square distance with the minimum square distance, the nearest square distance will be selected and the represented image will be as output.

Algorithms 4.6: Find the nearest neighbour
Input: Projected Test Image
Output: Nearest image from face dataset

Set the minimum distance to the max double value double minDistanceSquare = std::numeric_limits<double>::max();
 	
Declare an index of nearest image in face dataset int nearestImage = 0;
 	
Read all training images
 	
   Declare distance square. double distanceSquare = 0;
 	
   For each eigenvalue in eigenvalue matrix
 	
     Calculate the distance. float distance = the projected input face image eigenvalue subtract the projected training image -> data.fl[current trained image multiply numEigenvalues plus eigenvalue];
 	
     Square the distance and add it to the total distanceSquare equal to (distance power of 2) plus distanceSquare

End For
End Reading
    If distanceSquare less than minDistanceSquare
        Last distance square is the nearest neighbour Set lastDistanceSquare = distanceSquare;
        nearestIamge equal to current compare image class
    End If
End Read
Return nearestImage

4.3.2 Experiment Output

In this section, five video footages are presented from Figure 4.2 to Figure 4.6. Each of video footages is presented by four samples: original video, video with moving object removal (MOR), output video after face recognition and MOR video after face recognition. As shown in Figure 4.2 to Figure 4.6, the proposed scheme provides significant improvement with MOR in face recognition. The unexpected moving objects behind the target object no longer affect the face recognition.

In Figure 4.2, the face recognition was conducted in an office environment.

(a) Original video

(b) After moving object removal

(c) Applied face recognition on original video

(d) Proposed output

Figure 4.2 Face recognition in an office environment.

In Figure 4.3 and 4.4, face recognition was conducted in a reception environment.

(a) Original video

(b) After moving object removal

(c) Applied face recognition on original video

(d) Proposed output

Figure 4.3 Face recognition in a reception environment with an insignificant moving object close to the camera

(a) Original video

(b) After moving object removal

(c) Applied face recognition on original video

(d) Proposed output

Figure 4.4 Face recognition in a reception environment with an insignificant moving object far from the camera
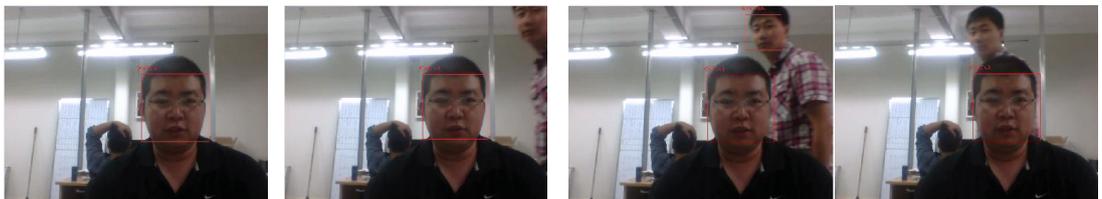
In Figure 4.5 and 4.6, face recognition was conducted in an outdoor environment.

(a) Original video

(b) After moving object removal

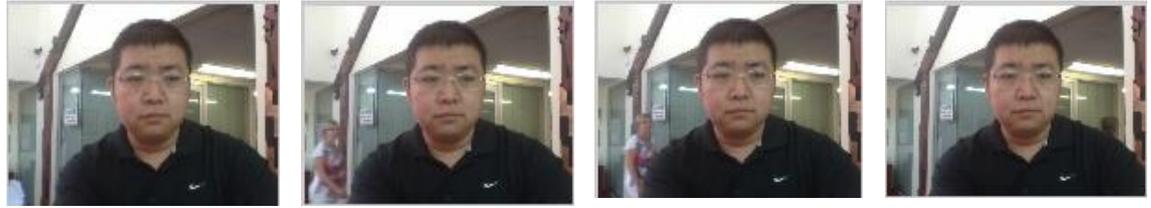(c) Applied face recognition on original video

(d) Proposed output

Figure 4.5 Face recognition in an outdoor environment

(a) Original video

(b) After motion object removal

(c) Applied face recognition on original video
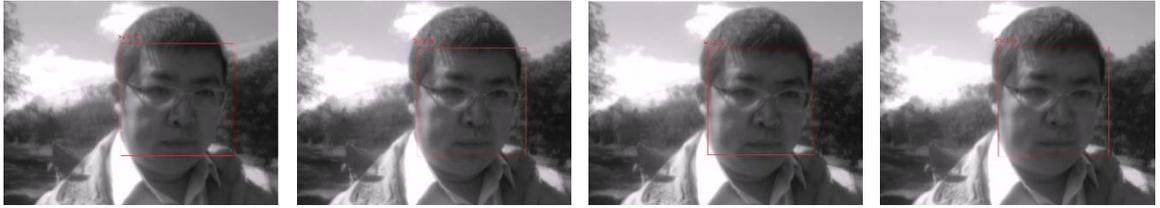
(d) Proposed output
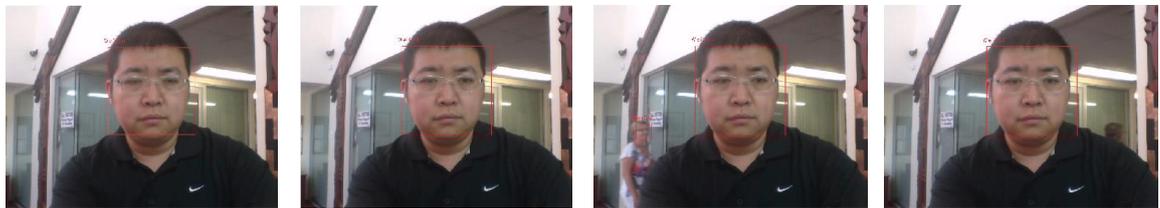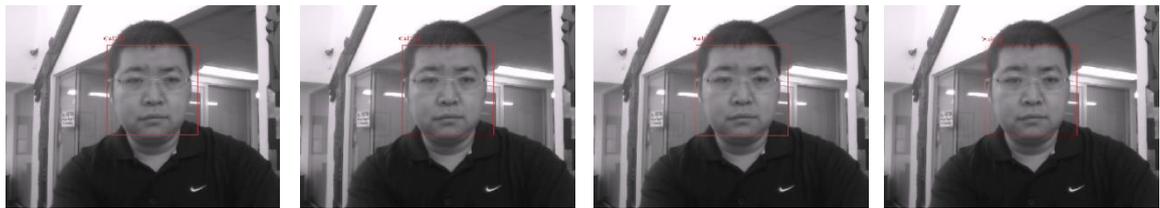
Figure 4.6 Face recognition in an outdoor environment with a truck near the camera.

4.4 Experiment Results

In this paper, the following three equations (4.1), (4.2) and (4.3) are used to calculate the precision, recall and F-measure. The precision, which is also known as positive predictive value, represents rate of the retrieved data that are relevant. In this thesis, the goal aimed at finding the number of times that the target face was retrieved. The recall is also known as sensitivity which reflects as the number of relevant data retrieved. In this thesis, it is the measurement of how many times the target face was recognized. The F-Measure or (F-Score) represents the average of the precision and recall, the best is 1.0 and the worst is 0.0.

$$\text{Precision} = \frac{A}{A + B} \times 100\% \tag{4.1}$$

$$\text{Recall} = \frac{A}{A+C} \times 100\% \tag{4.2}$$

$$\text{F-Measure} = 2 \times \frac{Precision * Recall}{Precision + Recall} \tag{4.3}$$

where A is the number of frames with the current user face recognized, B is the total frames with more than one face recognized, and C is the total number of frames with no faces recognized.

We calculate the precision and recall from our experiment based on the data shown in Table 4.1.

Table 4.1 Numbers of frames with face recognition before and after the MOR

Videos	Before/After	No. of frames with one face	No. of frames with two or more faces	No. of frames without faces
Video 1 (362 frames)	Before	205	98	59
	After	**264**	**31**	**67**
Video 2 (196 frames)	Before	106	39	51
	After	**128**	**11**	**57**
Video 3 (589 frames)	Before	403	70	116
	After	**444**	**24**	**121**
Video 4 (656 frames)	Before	428	103	125
	After	**461**	**64**	**131**
Video 5 (620 frames)	Before	383	97	140
	After	**413**	**58**	**149**

By applying the equations (4.1), (4.2) and (4.3), the results of precision, recall and F-Measure have been calculated and presented in Table 4.2 below. In Figure 4.7, the results of precision before/after MOR have been significantly improved.

Table 4.2 Precision, Recall and F-Measure results

Videos	Before/After	**Precision**	**Recall**	**F-Measure**
Video 1	Before	0.68	0.78	0.73
	After	**0.89**	**0.80**	**0.84**
Video 2	Before	0.73	0.68	0.70
	After	**0.92**	**0.69**	**0.79**
Video 3	Before	0.85	0.78	0.81
	After	**0.95**	**0.79**	**0.86**
Video 4	before	0.81	0.77	0.79
	After	**0.88**	**0.78**	**0.83**
Video 5	before	0.80	0.73	0.76
	after	**0.88**	**0.73**	**0.80**
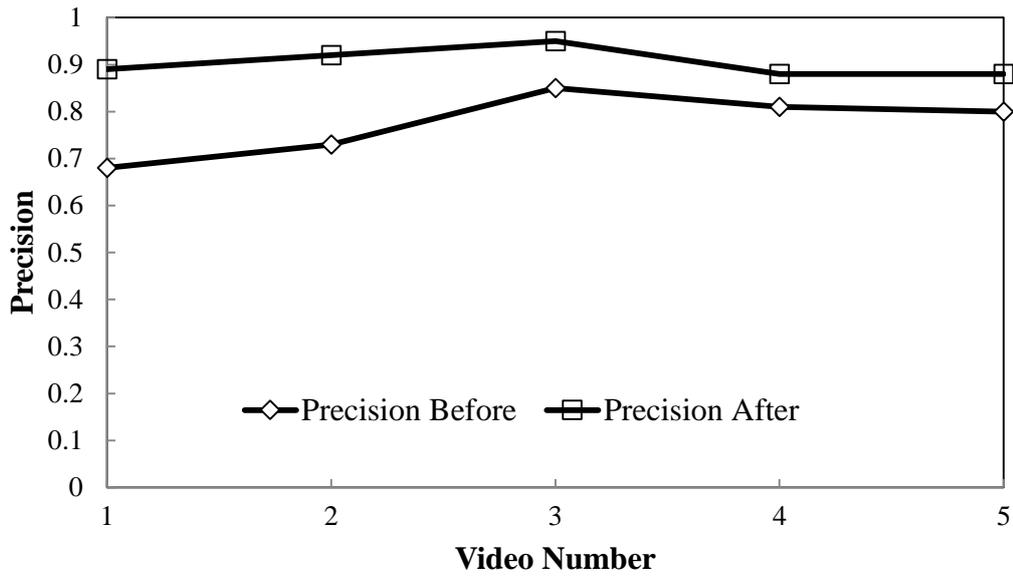
Figure 4.7 Comparison of precisions before and after MOR

In Figure 4.8, the results of Recall in before/after MOR. The results show that recall differences of face recognition before/after MOR are not obvious. In the experiment, because of the MOR, the false alarms of face detection were reduced. As a result some of videos such as video 1, video 2, video 3 and video 4 show a little increase in recall.

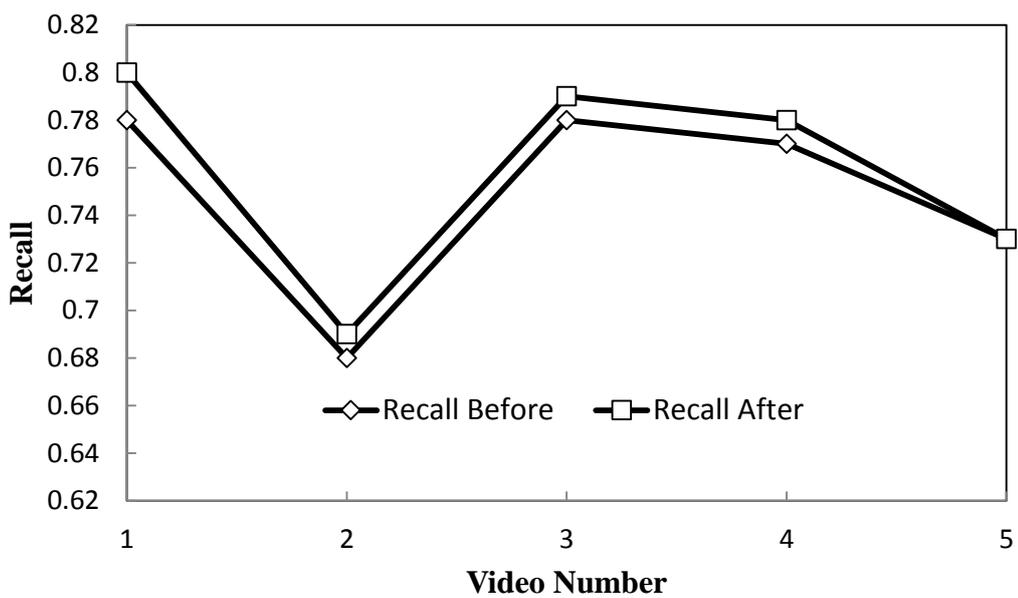

Figure 4.8 Comparison of recalls before and after MOR

In Figure 4.9, the calculation results of F-measure before and after moving object removal have significant improvement. The F-measure is explained as an average of the precision and recall, where the F-measure value is between 0.0 to 1.0 and the best is 1.0 and the worst is 0.0.
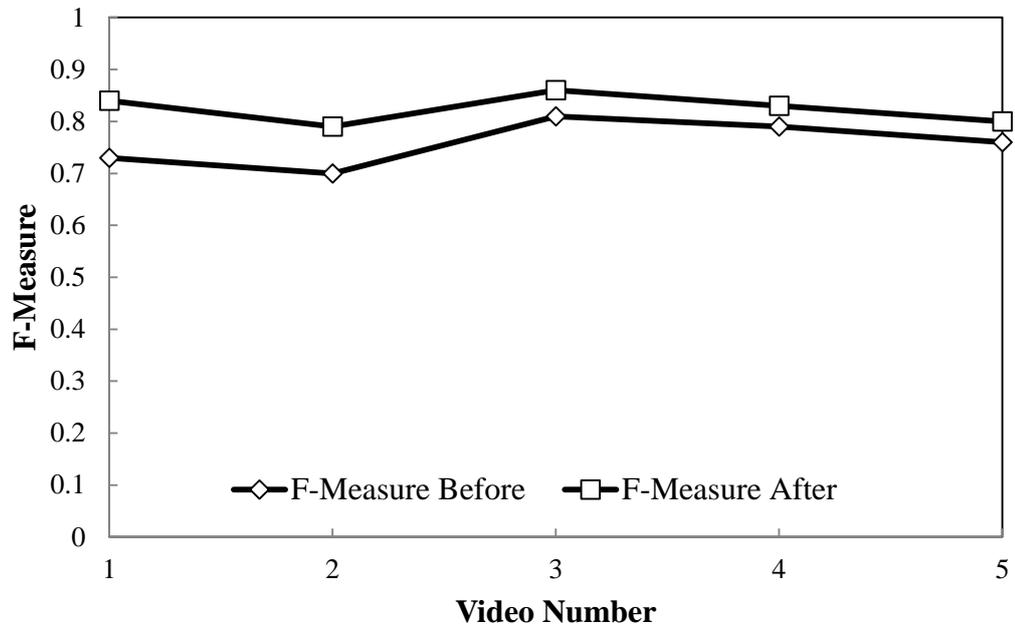

Figure 4.9 Comparison of F-measures before and after MOR

Chapter 5 Discussions and Analysis

5.1 Introduction

In Chapter 4, the experimental details and research findings were presented. The experiment is divided into two main parts. Both parts were conducted and analysed in details to find the evidence that the proposed scheme is the best solution to the research questions identified in chapter 3.

   In Chapter 5, the discussion of the main research findings to evaluate the importance of the outcomes will be presented. The reflection of the overall research will be presented in Section 5.2. The developed research questions in Section 3.3 are analysed in Section 5.3 and the justification is presented in Section 5.4.

5.2 Reflection

5.2.1 Work Description

Face recognition algorithms have been studies and evaluated. A research question has been formulated based on the literature review and related work. A prototype of moving object removal is developed. The moving object removal was conducted using the frame difference and pixel comparison. Moving Object Removal (MOR) result is evaluated and presented. A prototype of face recognition system is developed. OpenCV cascade classifier was used in face detection stage and the Principal Component Analysis (PCA) with Euclidean Distance classification was used for face recognition purpose. The Euclidean Distance classification was implemented for decision making. The experiments have been conducted to assess the face recognition precision and recall, and the experiment was conducted with two conditions: with and without MOR. The experimental results indicated that the proposed scheme is a suitable solution for our research question.

5.2.2 Evaluations

Since face detection is a critical part of face recognition, it will directly affect the recognition accuracy. Therefore, correctly detecting a human face in complex environments is considered as an effective way to improve the recognition accuracy of face recognition. Although face detection in a complex background using skin-color model and eigenface method (Monwar et al., 2006) or face region based detection

(Tanaka and Farah, 1993) has made great progress, the recognition accuracy could not be greatly improved. In this research, we detect and recognize a human face using a fixed camera pointing in a fixed direction, so only the possible changes of background with moving objects could affect our recognition rate. Therefore, we combine the moving object removal with face detection and recognition techniques together to increase the accuracy of our face recognition.

5.3 Research Questions

The research questions were developed based on the literature review in Chapter 2 and the similar studies in Chapter 3. In this section, the research solution is described in Chapter 4.

The derivation of the main research question postulates the key objective for different phases of experiments. The main research question is described as:

"Whether the face recognition accuracy in complex environments can be increased by applying moving object removal technology?"

In order to find the solution, an experimental test bed was designed and two prototypes were developed. This research attempts to find the solutions for the below questions:

Secondary question 1: **"What are the face recognition algorithms and which algorithm is the best one for our research purpose?"**

The face recognition algorithms are described in the literature review in the Chapter 2. It consists of two face recognition approaches: Appearance-based approach and Model-based approach. The appearance-based approach performs recognition using some spatial and/or temporal features extracted from the image sequences.

In this research, as mentioned in Chapter 2, the Principal Component Analysis (PCA) algorithm has been selected for the experiment with the below reasons:

- PCA is more efficient
- PCA is supporting multiple input image
- PCA uses less computational time comparing to ICA and LDA when Euclidean Distance is used

Secondary question 2: **"What is the moving object removal and how could it help to improve the face recognition accuracy in complex environments?"**

The Moving Object Removal (MOR) is a way to remove the moving objects from the input video by using frame differencing technique. If the pixels are changed, it means something changing in the frames (Prabhakar et al., 2012). The pedestrians and vehicles are the main components to constitute a complex environment.

In this research, as mentioned in Chapter 2, to simply the complex environment, we have to implement a solution to remove moving objects such as pedestrians and vehicles from the video clips.

5.4 Justifications

To justify the proposed solution for face recognition in complex environments, two prototypes have been developed in the Visual Studio 2012 and the MATLAB programming environments to demonstrate the applicability of the proposed scheme. The result of the precision from a direct comparison shows the significant improvement in face recognition throughout the entire input video footages.

Chapter 6 Conclusion

In this paper, a scheme for increasing accuracy of face recognition in a complex environment using Moving Object Removal (MOR) has been symmetrically presented, and the results were well demonstrated. The core of our scheme was to eliminate those moving objects from the complicated background so as to improve the recognition accuracy. The advantage of this scheme is that the face recognition rate has been significantly promoted, as supported by our results.

6.1 Conclusion

Face recognition is one of the most important biometrics in computer vision. It has been widely harnessed in the fields such as surveillance, information security, person identification, and law enforcement. Implementing a highly accurate face recognition system with ability to operate under a variety of environments is a tremendous challenge.

One of the paramount factors affects the face recognition accuracy is the interference from the complex environments, especially when human faces appeared in front of a camera or in a myriad of video footages. This is the primitive reason or motivation of research of this thesis. The proposed scheme as a solution is for increasing precision of face recognition in complex environments.

In this thesis, face recognition in complex environments is taken into consideration. An empirical approach for automated face recognition is formulated and a prototype is developed. The overall research consists of five steps. The first step is to clarify research motivation and background. The second is to manifest what other researchers have done in the past since it is critical to create a portfolio of how to conduct the research for this thesis. The third step is methodological analysis, in this step, an appropriate research methodology is systematically proposed. The fourth step is the experiments, it encompasses two main folds, the first is moving object detection and removal using the Matlab platform; the second is face detection and recognition using OpenCV and PCA algorithm. The last step is the results analysis. In this phase we compare the face recognition precision and recall before/after the complex environment was tackled. The comparisons will be utilized as an evidence to support that the accuracy of face recognition with MOR is superior to that without MOR.

6.2 Future Work

As the limitations of this thesis hereinbefore, the research could be extended by experiments with a myriad of different cameras such as Dynamic Vision Sensor (DVS). Moreover, various lighting conditions could be taken into consideration. The estimation of the moving direction and identification of the high speed moving objects will be also taken into consideration for MOR in future.

References

Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(12), 2037-2041. doi:Doi 10.1109/Tpami.2006.244

Abdel-Mottaleb, M., & Elgammal, A. (1999). Face detection in complex environments for color images, Image Processing 1999. Kobe: 1999 International Conference. pp. 622-626.

Bartlett, M.S., Movellan, J.R., & Sejnowski, T.J. (2002). Face recognition by independent component analysis. *IEEE Transactions on Neural Networks, 13*(6), 1450-1464. doi:Doi 10.1109/Tnn.2002.804287

Belhumeur, P.N., Hespanha, J.P., & Kriegman, D.J. (1997). Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 711-720. doi:Doi 10.1109/34.598228

Blue, J.L., Candela, G.T., Grother, P.J., Chellappa, R., & Wilson, C.L. (1994). Evaluation of Pattern Classifiers for Fingerprint and OCR Applications. *Pattern Recognition, 27*(4), 485-501. doi:Doi 10.1016/0031-3203(94)90031-0

Bruce, V., & Young, A. (1986). Understanding Face Recognition. *British Journal of Psychology, 77*, 305-327.

Blanz, V., & Vetter, T. (2003). Face recognition based on fitting a 3D morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*(9), 1063-1074. doi:Doi 10.1109/Tpami.2003.1227983

Bronstein, A., Bronstein, M., Kimmel, R., & Spira, A. (2003). 3D face recognition without facial surface reconstruction.

Brunelli, R., & Poggio, T. (1993). Face Recognition - Features versus Templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 15*(10), 1042-1052. doi:Doi 10.1109/34.254061

Burton, D.K. (1987). Text-Dependent Speaker Verification Using Vector Quantization Source-Coding. *IEEE Transactions on Acoustics Speech and Signal Processing, 35*(2), 133-143. doi:Doi 10.1109/Tassp.1987.1165110

Castrillon, M., Deniz, O., Hernandez, D., & Lorenzo, J. (2011). A comparison of face and facial feature detectors based on the Viola-Jones general object detection framework. *Machine Vision and Applications, 22*(3), 481-494. doi:DOI 10.1007/s00138-010-0250-7

Chellappa, R., Wilson, C.L., & Sirohey, S. (1995). Human and Machine Recognition of Faces - A Survey. *83*(5), 705-740. doi:Doi 10.1109/5.381842

Choi, J., Lee, S., Lee, C., & Yi, J. (2001). PrimeEye: A real-time face detection and recognition system robust to illumination changes. *Audio- and Video-Based Biometrics Person Authentication, 2091*, 360-365.

Comon, P. (1994). Independent component analysis, a new concept? *Signal Processing, Vol 36.* 287-314.

Chen, S., Zhang, T., Zhang, C., & Cheng, Y. (2010). A real-time face detection and recognition system for a mobile robot in a complex background, Artificial Life and Robotics, 439-443.

Cootes, T.F., Wheeler, G.V., Walker, K.N., & Taylor, C.J. (2002). View-based active appearance models. *Image and Vision Computing, 20*(9-10), 657-664. doi:Pii S0262-8856(02)00055-0 Doi 10.1016/S0262-8856(02)00055-0

Daugman, J.G. (1988). Complete discrete 2D Gabor transform by neural networks for image analysis and compression. *IEEE Transactions Acoustics, speech, and signal processing, vol. 36,* 1169-1179

Delac, K., Grgic, M., & Grgic, S. (2005). Independent comparative study of PCA, ICA, and LDA on the FERET data set. *International Journal of Imaging Systems and Technology, 15*(5), 252-260. doi:Doi 10.1002/Ima.20059

Delac, K., Grgic, M., & Grgic, S. (2005). Generalization abilities of appearance-based subspace face recognition algorithms. *the 12th International Workshop on Systems, Signals & Image Processing*, 271-274.

Delac, K., Grgic, M., & Grgic, S. (2005). Statistics in face recognition: Analyzing probability distributions of PCA, ICA and LDA performance results. *ISPA 2005: 4th International Symposium on Image and Signal Processing and Analysis*, 289-294. doi:Doi 10.1109/Ispa.2005.195425

Draper, B.A., Baek, K., Bartlett, M.S., & Beveridge, J. R. (2003). Recognizing Faces with PCA and ICA, *Computer Vision and Image Understanding (Special Issue on Face Recognition), Vols 91 (1-2),* 115-137.

Etemad, K., & Chellappa, R. (1997). Discriminant analysis for recognition of human face images, *Journal of the Optical Society of America A Optics Image Science and Vision, 14*(8), 1724-1733. doi:Doi 10.1364/Josaa.14.001724

Gandhe, S.T., Talele, K.T., & Keskar, A.G. (2007). Intelligent face recognition techniques: a comparative study. *International Journal Graph Vision and Image Processing (GVIP), 7*(2), 53-60.

Georghiades, A.S., Belhumeur, P.N., & Kriegman, D.J. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(6), 643-660. doi:Doi 10.1109/34.927464

Goldstei, A.j., Harmon, L.D., & Lesk, A.B. (1971). Identification of Human Faces. *Proceedings of the IEEE, 59*(5), 748-760. doi:Doi 10.1109/Proc.1971.8254

Gottumukkal, R., & Asari, V.K. (2004). An improved face recognition technique based on modular PCA approach. *Pattern Recognition Letters, 25*(4), 429-436. doi:DOI 10.1016/j.patrec.2003.11.005

Graham, D.B., & Allinson, N.M. (1998). Characterizing virtual Eigen signatures for general purpose face recognition. *Face Recognition: From Theory to Applications, NATO ASI Series F, Computer and Systems Sciences, vol. 163,* 446-456

Grudin, M.A. (2000). On internal representations in face recognition systems. *Pattern Recognition, 33*(7), 1161-1177. doi:Doi 10.1016/S0031-3203(99)00104-1

Guo, G., Li, S.Z., & Chan, K.L. (2000). Face recognition by support vector machines. *Automatic Face and Gesture Recognition, the Fourth IEEE International Conference,* 196-201. doi:DOI 10.1109/AFGR.2000.840634

Hanmandlu, M., Gupta, D., & Vasikarla, S. (2013). Face recognition using Elastic bunch graph matching. *IEEE Applied Imagery Pattern Recognition Workshop: Sensing for Control and Augmentation*, 1-7. doi:Doi 10.1109/AIPR.2013.6749338

Haritaoglu, I., Harwood, D., & Davis, L. S. (2000). W-4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(8), 809-830. doi:Doi 10.1109/34.868683

Heisele, B., Ho, P., & Poggio, T. (2001). Face recognition with support vector machines: Global versus component-based approach. *Eighth IEEE International Conference on Computer Vision, Vol II,* 688-694.

Hiremath, V., & Mayakar, A. (2009). Face recognition using Eigenface approach. In *IDT workshop on interesting results in computer science and engineering, Sweden*.

Huynh-Thu, Q., Meguro, M., & Kaneko, M. (2002). Skin-Color Extraction in Images with Complex Background and Varying Illumination, *the Sixth IEEE Workshop on Applications of Computer Visio*, 280-285.

Hyvarinen, A., & Oja, E. (1997). A fast fixed-point algorithm for independent component analysis. *Neural Computation, Vol. 9.* 1483-1492

Jain, A. K., Ross, A., & Prabhakar, S. (2004). An introduction to biometrics recognition. *IEEE Transactions on Circuits and Systems for Video Technology, 14*(1), 4-20. doi:Doi 10.1109/Tcsvt.2003.818349

Jian, R. & Nagel, H. (1979). On the analysis of accumulative difference prictures from image sequences of real world scenes. *IEEE transactions on pattern analysis and machine intelligence, 1(2),* 206-214

Jonsson, K., Matas, J., Kittler, J. & Li, Y.P. (2000). Learning support vectors for face verification and recognition. *Automatic Face and Gesture Recognition, the Fourth IEEE International Conference,* 208-213. doi:DOI 10.1109/AFGR.2000.8406346

Kadyrov, A., & Petrou, M. (2001). The trace transform and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(8), 811-828. doi:Doi 10.1109/34.946986

Karhunen, J., Oja, E., Wang, L., Vigario, R., & Joutsensalo, J. (1997). A class of neural networks for independent component analysis. *IEEE Transactions On neural networks, 8(3)*. 486-504

Kim, T.K., Kittler, J., & Cipolla, R. (2007). Discriminative learning and recognition of image set classes using canonical correlations. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(6), 1005-1018. doi:Doi 10.1109/Tpami.2007.1037

Kondo, T., & Yan, H. (1999). Automatic human face detection and recognition under non-uniform illumination. *Pattern Recognition, 32*(10), 1707-1718. doi:Doi 10.1016/S0031-3203(98)00176-9

Kovac, J., Peer, P., & Solina, F. (2003). Human skin colour clustering for face detection. *IEEE Region 8 Eurocon 2003, Vol B,* 144-148.

Kshirsagar, V.P., Baviskar, M.R., Gaikwad, M.E. (2011). Face Recognition Using Eigenfaces, *IEEE Computer Research and Development (ICCRD)*, 302-306.

Lawrence, S., Giles, C.L., Tsoi, A.C., & Back, A.D. (1997). Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks, 8*(1), 98-113. doi:Doi 10.1109/72.554195

Lee, C.H., Kim, J.S., & Park, K.H. (1996). Automatic human face location in a complex background using motion and color information. *Pattern Recognition, 29*(11), 1877-1889. doi:Doi 10.1016/0031-3203(96)00036-2

Li, M., & Wang, B. (2013). Face Detection under Complex Background, *International Journal of Applied Mathematics and Statistics, 49(19),* 457-462

Li, S.Z., & Jain, A.K. (2005). Handbook of Face Recognition, *eds. Springer*

Li, S.Z., & Lu, J.W. (1999). Face recognition using the nearest feature line method. *IEEE Transactions on Neural Networks, 10*(2), 439-443. doi:Doi 10.1109/72.750575

Lienhart, R., Kuranov, A., & Pisarevsky, V. (2003). Empirical analysis of detection cascades of boosted classifiers for rapid object detection. *Pattern Recognition, 2781*, 297-304.

Lin, C. (2007). Face detection in complicated backgrounds illumination conditions by using YCbCr color space and different and neural network. *Pattern Recognition Letters, 28*(16), 2190-2200. doi:DOI 10.1016/j.patrec.2007.07.003

Liu, C.J., & Wechsler, H. (1998). A unified Bayesian framework for face recognition. *1998 International Conference on Image Processing, Vol. 1*, 151-155.

Liu, C., & Wechsler, H. (1999). Comparative Assessment of Independent Component Analysis (ICA) for face recognition, *International Conference on Audio- and Video-based Biometrics Person Authentication, AVBPA'99*, 211-216

Liu, C.J., & Wechsler, H. (2000). Evolutionary pursuit and its application to face recognition.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(6), 570-582. doi:Doi 10.1109/34.862196

Lu, J.W., Plataniotis, K.N., & Venetsanopoulos, A.N. (2003). Boosting linear discriminant analysis for face recognition. 2003 International Conference on Image Processing, Vol 1, 657-660.

Lu, J.W., Plataniotis, K.N., & Venetsanopoulos, A.N. (2003). Face recognition using LDA-based algorithms. IEEE Transactions on Neural Networks, 14(1), 195-200. doi:Doi 10.1109/Tnn.2002.806647

Lu, J.W., Plataniotis, K.N., & Venetsanopoulos, A.N. (2003). Regularized discriminant analysis for the small sample size problem in face recognition. Pattern Recognition Letters, 24(16), 3079-3087. doi:Doi 10.1016/S0167-8655(03)00167-3

Lu, J.W., Plataniotis, K.N., & Venetsanopoulos, A.N. (2003). Face recognition using kernel direct discriminant analysis algorithms. IEEE Transactions on Neural Networks, 14(1), 117-126. doi:Doi 10.1109/Tnn.2002.806629

Lu, J.W., Plataniotis, K.N., Venetsanopoulos, A.N., & Li, S.Z. (2006). Ensemble-based discriminant learning with boosting for face recognition. IEEE Transactions on Neural Networks, 17(1), 166-178. doi:Doi 10.1109/Tnn.2005.860853

Maio, D., & Maltoni, D. (2000). Real-time face location on gray-scale static images. Pattern Recognition, 33(9), 1525-1539. doi:Doi 10.1016/S0031-3203(99)00130-2

Manjunath, B., Chellappa, R., & Malsburg, C.V. (1992). A feature based approach to face recognition, Computer Vision and Pattern Recognition. 373-377

Martinez, A.M., & Kak, A.C. (2001). PCA versus LDA. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(2), 228-233. doi:Doi 10.1109/34.908974

Martinez, A.M. (2000). Recognition of partially occluded and/or imprecisely localized faces using a probabilistic approach. Computer Vision and Pattern Recognition, vol. 1, 712-717

Moghaddam, B., Lee, J., Pfister, H., & Machiraju, R. (2003). Model-based 3D face capture with shape-from-silhouettes. IEEE International Workshop on Analysis and Modeling of Face and Gestures, 20-27.

Moghaddam, B., & Pentland, A. (1997). Probabilistic visual learning for object representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), 696-710. doi:Doi 10.1109/34.598227

Monwar, M.M., Paul, P.P., Islam, M.W., & Rezaei, S. (2006). A real-time face recognition approach from video sequence using skin color model and Eigenface method. 2006 Canadian Conference on Electrical and Computer Engineering, Vols 1-5, 567-571.

Moon, H., & Phillips, P. J. (2001). Computational and performance aspects of PCA-based face-recognition algorithms. Perception, 30(3), 303-321. doi:Doi 10.1068/P2896

Nefian, A.V., & Hayes, M.H. (1999). An embedded HMM-based approach for face detection and recognition. *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, Vols I-Vi*, 3553-3556.

Osuna, E., Freund, R., & Girosi, F. (1997). Training support vector machines: an application to face detection. *1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 130-136. doi:Doi 10.1109/Cvpr.1997.609310

Pai, Y.T., Ruan, S.J., Shie, M.C., & Liu, Y.C. (2006). A Simple and Accurate Color Face Detection Algorithm in Complex Background, *IEEE International Conference on Multimedia and Expo (ICME),* 1545-1548.

Papageorgiou, C.P., Oren, M., & Poggio, T. (1998). A general framework for object detection. *Sixth International Conference on Computer Vision*, 555-562. doi:Doi 10.1109/Iccv.1998.710772

Pentland, A. (2000). Looking at people: Sensing for ubiquitous and wearable computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(1), 107-119. doi:Doi 10.1109/34.824823

Pentland, A., Moghaddam, B., & Starner, T. (1994). View-Based and Modular Eigenspaces for Face Recognition. *1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 84-91.

Pentland, A., Starner, T., Etcoff, N., Masoiu, A., Oliyide, O., & Turk, M. (1993). Experiments with Eigenfaces. *International Joint Conference Artificial Intelligence.*

Phillips, P. J., Moon, H., & Rizvi, S. (1997). The FERET evaluation methodology for face-recognition algorithms. *Computer Vision and Pattern Recognition 97,* 137-143

Prabhakar, N., Vaithiyanathan, V., Sharma, A.P., Singh, A., & Singhal, P. (2012). Object Tracking Using Frame Differencing and Template Matching. *Research Journal of Applied Sciences, Engineering and Technology, 4(24),* 5497-5501

Qi, Y.Y., & Hunt, B.R. (1994). Signature Verification Using Global and Grid Features. *Pattern Recognition, 27*(12), 1621-1629. doi:Doi 10.1016/0031-3203(94)90081-7

Rowley, H. A., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(1), 23-38. doi:Doi 10.1109/34.655647

Sirovich, L., & Kirby, M. (1987). Low-Dimensional Procedure for the Characterization of Human Faces. *Journal of the Optical Society of America a-Optics Image Science and Vision, 4*(3), 519-524. doi:Doi 10.1364/Josaa.4.000519

Slavković, M., Jevtić, D. (2012). Face Recognition Using Eigenface Approach. *Serbian Journal of Electrical Engineering,* 9(1), 212-130.

Srisuk, S., Petrou, M., Kurutach, W., & Kadyrov, A. (2003). Face authentication using the trace transform. *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol 1,* 305-312.

Sung K.K., & Poggio, T. (1998). Example-Based Learning for View-Based Human Face Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(1),* 39-51.

Tanaka, J.W., & Farah, M.J. (1993). Parts and Wholes in Face Recognition. *Quarterly Journal of Experimental Psychology Section a-Human Experimental Psychology, 46*(2), 225-245.

Turk, M., & Pentland, A. (1991). Eigenfaces for Recognition. *Journal of Cognitive Neuroscience, 3*(1), 71-86. doi:DOI 10.1162/jocn.1991.3.1.71

Turk, M.A., & Pentland, A.P. (1991). Face Recognition Using Eigenfaces. *1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 586-591.

Valentin, D., Abdi, H., Otoole, A.J., & Cottrell, G.W. (1994). Connectionist Models of Face Processing - A Survey. *Pattern Recognition, 27*(9), 1209-1230. doi:Doi 10.1016/0031-3203(94)90006-X

Vinaya H., Ashwini M. (2009). Face Recognition using Eigenface Approach. *IDT Workshop on Interesting Results in Computer Science and Engineering*

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol 1,* 511-518.

Viola, P., & Jones, M. (2001). Robust real-time face detection. *Eighth IEEE International Conference on Computer Vision, Vol II,* 747-747.

Wang, Y.J., & Yuan, B.Z. (2001). A novel approach for human face detection from color images under complex background. *Pattern Recognition, 34*(10), 1983-1992. doi:Doi 10.1016/S0031-3203(00)00119-9

Wiskott, L., Fellous, J.M., Kruger, N., & vonderMalsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 775-779. doi:Doi 10.1109/34.598235

Wiskott, L., Fellous, J.M., Kruger, N., & vonderMalsburg, C. (1997). Face recognition by elastic bunch graph matching. *International Conference on Image Processing, Vol I*, 129-132.

Wiskott, L., Fellous, J.M., Kruger, N., & vonderMalsburg, C. (1999). Face recognition by elastic bunch graph matching. *In Intelligent Biometrics Techniques in Fingerprint and Face Recognition,* 355-396.

Wong, K.W., Lam, K.M., & Siu, W.C. (2001). An efficient algorithm for human face detection and facial feature extraction under different conditions. *Pattern Recognition, 34*(10), 1993-2004. doi:Doi 10.1016/S0031-3203(00)00134-5

Wu, L.F., & Yang, X.K. (2004). Face Recognition in Complex Backgrounds, *Proc. SPIE 5403, Sensors, and Command, Control Communications, and Itelligence (C3I) Technologies for Homeland Security and Homeland Defense III, 325*

Yang, M.H. (2002). Kernel Eigenfaces vs. Kernel Fisherfaces: Face recognition using Kernel methods. *Fifth IEEE International Conference on Automatic Face and Gesture Recognition,* 215-220.

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

Yu, H. & Yang, J. (2001). A direct LDA algorithm for high dimensional data with application to face recognition. *Pattern Recognition,* vol. 34. 2067-2070

Yuan, Q., Gao, W., & Yao, H.X. (2002). Robust frontal face detection in complex environment. *16th International Conference on Pattern Recognition, Vol I,* 25-28.

Zhao, W., Chellappa, R., & Krishnaswamy, A. (1998). Discriminant analysis of principal components for face recognition. *Automatic Face and Gesture Recognition - Third IEEE International Conference*, 336-341.

Kocjan, P. & Saeed, K. (2012). Face recognition in unconstrained environments, Biometrics and Kansei Engineering, Springer.

Appendix

In this appendix, we provide the C++ source code of face detection and recognition. Visual studio 2012 was used for the implementation of our algorithms. The library required for running this program is OpenCV 2.4.5 platform.

```cpp
///////////////////////////////////////////////////////////////
//Author: Wei Cui                                        /////
//Implemented using VC++ OpenCV Library (http://OpenCV.org)   /////
///////////////////////////////////////////////////////////////
#include "stdafx.h"
#include <Windows.h>
#include <iphlpapi.h>
#pragma comment(lib, "iphlpapi.lib")
#include <stdio.h>
#include <boost/uuid/uuid.hpp>              // uuid class
#include <boost/uuid/uuid_generators.hpp> // generators
#include <boost/uuid/uuid_io.hpp>          // streaming operators etc.
#include <boost/lexical_cast.hpp>
#include <opencv2/core/core.hpp>
#include <opencv2/contrib/contrib.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/objdetect/objdetect.hpp>

#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <time.h>
#include <vector>

//Declare OpenCV cascade classifier

string PCA_haar =
"C:\\opencv\\data\\haarcascades\\haarcascade_frontalface_default.xml";

//Main method

int main(int argc, const char *argv[]) {
    HWND console = GetConsoleWindow();
    RECT r;
    GetWindowRect(console, &r); //stores the console's current dimensions
    MoveWindow(console, r.left, r.top, 320, 450, TRUE); // 320 width, 450 height
        cout<< "--------------------------------------------------" << endl;
        cout<< "Face recognition system @ AUT."   << endl;
        cout<< " Example:"                        << endl;
        cout<< "facerec { trainingSet.txt } {PCA, LDA}"    << endl;
        cout<< "-------------------------------------" << endl;
        cout<< endl;
        if (argc != 3){
                cout << "Not enough parameters" << endl;
                help();
                return -1;

        }
        const string trainingSet = argv[1];
        const string algorithms = argv[2];
        if(algorithms == "PCA"){
```

```
                facerec_eigenface(trainingSet);
        }
        if(algorithms == "LDA"){
                facerec_eigenface(trainingSet);
        }
        else if (algorithms == "EXIT"){
                exit(-1);
        }
    return 0;
}
```

The result is shown as the below figure.