School of Electrical Engineering and Computing
Department of Computing

Robust Face Recognition based on Color and Depth Information

Billy Y.L. Li

This thesis is presented for the Degree of
Doctor of Philosophy
at
Curtin University

April 2013

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made. This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Billy Li	Date

Abstract

One of the most important advantages of automatic human face recognition is its non-intrusiveness property. Face images can sometime be acquired without user's knowledge or explicit cooperation. However, face images acquired in an uncontrolled environment can appear with varying imaging conditions. Traditionally, researchers focus on tackling this problem using 2D gray-scale images due to the wide availability of 2D cameras and the low processing and storage cost of gray-scale data. Nevertheless, face recognition can not be performed reliably with 2D gray-scale data due to insufficient information and its high sensitivity to pose, expression and illumination variations. Recent rapid development in hardware makes acquisition and processing of color and 3D data feasible. This thesis aims to improve face recognition accuracy and robustness using color and 3D information.

In terms of color information usage, this thesis proposes several improvements over existing approaches. Firstly, the Block-wise Discriminant Color Space is proposed, which learns the discriminative color space based on local patches of a human face image instead of the holistic image, as human faces display different colors in different parts. Secondly, observing that most of the existing color spaces consist of at most three color components, while complementary information can be found in multiple color components across multiple color spaces and therefore the Multiple Color Fusion model is proposed to search and utilize multiple color components effectively. Lastly, two robust color face recognition algorithms are proposed. The Color Sparse Coding method can deal with face images with noise and occlusion. The Multi-linear Color Tensor Discriminant method harnesses multi-linear technique to handle non-linear data. Experiments show that all the proposed methods outperform their existing competitors.

In terms of 3D information utilization, this thesis investigates the feasibility of face recognition using Kinect. Unlike traditional 3D scanners which are too slow in speed and too expensive in cost for broad face recognition applications, Kinect trades data quality for high speed and low cost. An algorithm is proposed to show that Kinect data can be used for face recognition despite its noisy nature. In order to fully utilize Kinect data, a more sophisticated RGB-D face recognition algorithm is developed which harnesses the Color Sparse Coding framework and 3D information to perform accurate face recognition robustly even under simultaneous varying conditions of poses, illuminations, expressions and disguises.

Acknowledgments

I would like to express my sincere thanks to the following people whose help and contribution make this thesis possible.

First, I would like to thank my supervisors. My main supervisor Wanquan Liu, who has given me continuous guidance and encouragement, starting from my undergraduate study all the way through here. I would not have commenced my PhD without him. My co-supervisor Aneesh Krishna who has provided much useful advice and support to me. My co-supervisor Ajmal Mian, who has taught me all the important knowledge about 3D face recognition, giving me inspiring ideas and going through my papers word by word patiently. Although he just supervised me for only one year, I have learned much from him. His contribution greatly improves the quality of my works as well as this thesis.

I am grateful to Patrick Peursum for saving my computer before I smashed it. He has provided much help and advice to me including both technical and academic, everything from low level programming problems to thesis writing. I would also like to thanks Senjian An who has provided useful knowledge and comments to me.

Thanks to my parents Albert and Wendy, my sister Kate, and my partner Katrina, for all their care, support and love. They are one of the main motivations that get me this far.

I must also thank my fellow PhD students who always have interesting discussion with me which inspires me with ideas and different ways of thinking.

Finally, thanks to Curtin University for providing me with a scholarship to support my PhD study. I would also like to thanks our administrative officer, Mary Mulligan, for getting me through all complicated paper works, all the IT staff for their professional help. Special thanks to all staff and students who participated in our Kinect data collection project which directly contributes to part of this thesis.

Published Work

This thesis includes the following works that have been published over the course of my PhD study, and they are listed in chapter order:

- Billy Y.L. Li, Wanquan Liu, Senjian An, Aneesh Krishna and Tianwei Xu. (2012) Face recognition using various scales of discriminant color space transform. *Neurocomputing*. Volume 94, pages 68-76. (Chapter 3)

- Billy Y.L. Li, Senjian An, Wanquan Liu and Aneesh Krishna. (2011) The MCF Model: Utilizing Multiple Colors for Face Recognition. *International Conference on Image and Graphics*, pages 1029-1034. (Chapter 4)

- Billy Y.L. Li, Wanquan Liu, Senjian An and Aneesh Krishna. (Under review) Robust Face Recognition by utilizing Color Information and Sparse Representation. *Information Sciences*. (Chapter 5)

- Billy Y.L. Li, Wanquan Liu, Senjian An and Aneesh Krishna. (2012) Tensor Based Robust Color Face Recognition. *International Conference on Pattern Recognition*. (Chapter 6)

- Billy Y.L. Li, Ajmal S. Mian, Wanquan Liu and Aneesh Krishna. (Under review) Face Recognition based on Kinect. *IEEE Transactions on Systems, Man and Cybernetics Part B, Special Issue on Computer Vision for RGB-D Sensors: Kinect and Its Applications*. (Chapter 6)

- Billy Y.L. Li, Ajmal S. Mian, Wanquan Liu and Aneesh Krishna. (2013) Using Kinect for Face Recognition Under Varying Poses, Expressions, Illumination and Disguise. *IEEE Workshop on the Applications of Computer Vision*. (Chapter 7)

- Billy Y.L. Li, Ajmal S. Mian, Wanquan Liu and Aneesh Krishna. (Under review) Robust RGB-D Face Recognition using Kinect Sensor. *International Journal of Computer Vision*. (Chapter 7)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Biometrics are personal characteristics that can be used to uniquely identify a person. Although finger prints and iris scans are highly accurate, they both require user cooperation. On the other hand, face images can be captured even without the user's knowledge. This non-intrusiveness property is an apparent advantage for face recognition, which allows broad applications. Therefore, face recognition research should focus on non-intrusive situations where face images are acquired under uncontrolled environment. These images can be in varied conditions such as different head poses, illumination, facial expressions, disguises and noises. Since, these conditions are unknown beforehand and can appear simultaneously, they pose additional challenges and complexity to the conventional face recognition problem.

Traditionally, face recognition was performed using gray-scale images. One of the main reasons using gray-scale over color images is due to the low storage and processing requirements for gray-scale data. A gray-scale image stores only the light intensity at each pixel location, whereas a color image is usually modeled in RGB, storing intensity values of three channels R, G and B, which when in combinations, infinite number of colors can be generated. Since three channels are used to represent color images, handling them requires high storage and processing cost.

Color is an important cue for face recognition. *Colors* can be defined as the spectra of light. Visible light source always consists of infinite spectra, which can be separated into rainbow-like colors using a dispersive prism (i.e. the visible wavelength range (400-700nm) can be partitioned into any number of bands). Different objects display different colors because they have different reflection properties on different spectra. For example, human lips are reddish because it has higher reflection rate on the red spectrum, while pupils appear black because it dose not reflect any light. This reflection property is also different from face to face, therefore color information can help discriminate different people. In addition, redundant information across different spectrum also facilitate error correction and thus increases recognition robustness to noise. In fact, color face recognition has been shown to outperform gray-scale face recognition in many situations. With the rapid development of technology, larger capacity of storage media and higher processing power of

computers are available at a much lower cost than before. Handling large amount of color image data is feasible nowadays. The advantages of using color data actually outweigh its disadvantages. In short, color information should not be ignored.

Despite the usefulness of color information, many face recognition algorithms proposed recently are still designed and evaluated only on gray-scale images. How much these methods can be benefited from color information, has not been investigated. On the other hand, existing research on color face recognition mainly focus on deriving new color spaces, where a few shortcomings can be noted. Firstly, most color spaces are derived based on holistic face images, while human faces display different colors at different parts. Secondly, most color spaces consist of three components only, while complementary information can be revealed from more than three color components across multiple color spaces. Lastly, most color spaces are evaluated using basic features and classifiers. The evaluations usually involve face images with moderate variations only. The advantages of these proposed color spaces with more advanced recognition algorithms are unknown. Their performance under large variation, especially with occlusion and noise, is highly questionable.

Color information can certainly aid face recognition, however adding color information alone to existing face recognition algorithms may not be sufficient to solve all real world problems. Consider that human can perform face recognition remarkably well for face images under large variation, while the accuracy of most of the existing face recognition algorithms degrade dramatically when the input face image is taken at a different view point, or is taken after several years. It is possible that there are some other key types of information, such as structural or spatial information, missing in existing face recognition algorithms.

One of the most obvious information that has been missed out but start receiving research attention recently is 3D information. In fact, whether using gray scale or color data, face recognition can not be performed reliably with only two-dimensional (2D) images alone. A 2D image is a projection of the 3D scene on to the image plane of a camera which is a one way process i.e. the original 3D scene cannot be recovered from the projected image. While a 2D image contains the reflective properties of an object, a 3D image describes its shape. Unlike 2D images, absolute measurements can be performed on 3D data. There is no way to tell how far two objects are apart from each other on one single 2D image, because they can be captured in any distance from the camera. The absolute measurement computed using 3D data can be used, for example, to correct the facial pose or to generate infinite novel poses using computer graphic techniques. Furthermore, facial geometry is invariant to illumination whereas 2D images are a direct function of the lighting conditions (i.e. direction and spectrum). Although, the 3D imaging process can

be influenced by lighting, the 3D data itself is illumination invariant. Facial images under different illumination conditions can be generated using a 3D face model. In short, many limitations of 2D face recognition can be overcome by using 3D data.

Although much research focus has been put on 3D face recognition recently, most algorithms proposed so far do not effectively utilize or completely ignore the 2D (texture) image, which is usually captured along with the 3D data by most of the 3D scanners. Existing *multi-modal* (2D+3D) methods mostly use gray-scale images only, however color information is important and should not be ignored as discussed before. Moreover, existing 3D methods assume the availability of high resolution 3D data, however 3D scanners that are able to provide such data usually are expensive and slow in capturing speed which limits their applicability. On the other hand, the recently released Kinect sensor has received increasing research interest due to its low cost and high speed. However, whether the low resolution 3D data captured by Kinect is useful for face recognition or not has not been justified.

1.1 Aims and Approach

The main aim of this thesis is to achieve robust and accurate automatic face recognition under uncontrolled environment, for non-intrusive applications. To this end, an algorithm is expected to be developed that utilizes color and depth information for face recognition achieving high and robust performance. The recognition process may involve four major steps with several sub-objectives described as follows:

1. Face (2D and 3D) data acquisition.

2. Automatic face detection.

3. Extracting discriminative (2D and 3D) features or representations.

4. Deciding the person identity based on some classification rules.

The objective associated with step (1) is to justify the feasibility of using low resolution 3D data for robust face recognition. Existing technology allows convenient acquisition of high quality colored 2D face images and therefore the choice of 2D acquisition devices is not critical. However, existing 3D scanners that can capture high resolution 3D data, are usually slow in capture speed. This means that the users are required to present their faces

still during acquisition, which is not achievable in many non-intrusive applications. On the other hand, high speed 3D devices provide low quality 3D data which leads to bumpy and non-realistic 3D face models. The usefulness of such 3D data has to be justified.

Step (2) is not the research focus of this thesis. All face recognition algorithms proposed in this thesis assume the bounding box of the face on a 2D image, or the nose tip position on a 3D face model is previously detected and available. Face detection in 2D images and nose tip detection in 3D face modelled is a well explored area with many existing algorithm such as (Viola and Jones, 2004; Mian *et al.*, 2007).

The objective associated with step (3) and (4) is to better utilize 2D color and 3D information to improve face recognition performance. For 2D color images, investigation in better color spaces or color models, which are more discriminative for face recognition, is required. For example, such a color model can be derived using local patches of human face or from multiple existing color spaces. It is also important to design better classifiers, which are more robust to varying conditions, after the color features are extracted. For example, multi-linear and sparse coding techniques can be used to handle large variation and outliers, but they have to be re-formulated to work on color data.

For 3D data, designing of better preprocessing methods is required, such as pose correction, that can maximize the advantage of 3D data of having absolute measurement. Furthermore, a more discriminative representation can be developed to represent the 3D data. For example, the normal map image converted from range image is more discriminative. After that, the formulation of robust classifier to make use of the representation is needed. For example, some 2D classification methods such as sparse coding can be extended to work on 3D data. Lastly, the interaction and complement of 2D and 3D data to each other also require consideration. For example, the 3D data can be used to correct the pose of the corresponding 2D texture.

1.2 Contributions

In this thesis, all aforementioned objectives are achieved, resulting in several contributions to the field. We describe some of our major contributions from methodological, theoretical and experimental perspectives.

1.2.1 Novel Methods

All methods and algorithms proposed in this thesis are novel and significant. They either set the new state-of-the-art performance for face recognition, or provide an alternative solution to the problem with different advantages, which usually lead to new theoretical insights. The following methods are proposed in this thesis.

Pixel-level Discriminant Color Space (PLDCS) and Block-wise Discriminant Color Space (BWDCS) are two novel color spaces generalizing Yang and Liu (2008b)'s Discriminant Color Space (DCS) to work on local patches of face image. The proposed color spaces outperform the original DCS which is among the state-of-the-art color spaces for color face recognition.

Multiple Color Fusion (MCF) is a novel algorithm which searches for optimal color component combination and fuses them for face recognition. This algorithm outperforms existing color spaces and models, which sets the new state-of-the-art performance for color face recognition. The MCF algorithm consisting of several components is also very general, where each of the components can be replaced by more advanced techniques to further improve performance.

Color Sparse Coding (CSC) is a novel extension of the popular sparse coding method for face recognition. The Sparse Representation Classifier (SRC) (Wright *et al.*, 2009) and Correntropy Sparse Representation (CESR) (He *et al.*, 2010) are originally designed for gray-scale images only, but they are reformulated to work on color images in this thesis. The CSC method achieves superior performance especially in cases of random pixel corruptions and occlusions.

Multilinear Color Tensor Discriminant (MCTD) model is a novel integration of two state-of-the-art methods, i.e. the MPCA-PS (Rana *et al.*, 2009) and Tensor Discriminant Color Space (TDCS) (Wang *et al.*, 2011). MCTD defines a novel representation for color images, modifies and utilizes TDCS for feature extraction and reformulates a classifier based on MPCA-PS. MCTD complements the shortages of these two methods while retains their advantages. As a result, MCTD outperforms MPCA-PS and TDCS especially in the presence of large imaging variation.

Finer Feature Fusion (FFF) proposed in this thesis is a novel approach to perform face recognition using low quality RGB-D data (i.e. Red, Green, Blue and Depth). It utilizes Local Binary Pattern (LBP) (Ahonen *et al.*, 2004), Haar-like features (Viola and Jones,

2001) and Gabor features (Liu and Wechsler, 2002). A novel feature fusion strategy is proposed which removes redundant information and retains only the meaningful features for maximizing class separability. This method benefits from depth information even it is noisy. FFF is able to achieve high face recognition accuracy under challenging conditions.

Lastly, a novel multi-modal (2D+3D) face recognition algorithm is proposed which consists of several innovative components, i.e. Canonical Preprocessing (CP), Multi-channel Discriminant Transform (MDT) and Multi-channel Weighted Sparse Coding (MWSC). The most important part of CP algorithm is that it exploits facial symmetry to handle large pose variation. The MDT method generalizes DCS to derive a discriminant transform that works well on any multi-channel data. MWSC is a reformulation of the state-of-the-art Robust Sparse Coding (RSC) (Yang *et al.*, 2011) method to utilize color and depth data. The proposed multi-modal method outperforms existing state-of-the-art algorithms. It can handle simultaneous variations in pose, expression, illumination and disguise. Moreover, it dose not require parameters tuning for each case and it performs equally well on both low and high resolution 3D data.

1.2.2 Theoretical Insights

Theoretical insights are significant contributions as they inspire and facilitate future research. The following theoretical insights are provided in this thesis.

In terms of color space, we provide strong analytical evidences on the advantages of considering DCS locally compared to holistically. We show that the local optimal color space can be very different to the holistical one. The inter-component correlation can be decreased when deriving DCS locally, thus increasing its discriminative power. The findings suggest that subdivision of color image is more desirable when deriving color spaces. Furthermore, we have given a clear definition to distinguish color space from color model, which aids clarifying color face recognition methods. Moreover, we introduce the concept of Variance Face (VF) which can help visualizing and estimating the available information in a specific color component image. We also justify the advantages of using multiple color components over using three fixed components, which is the case in most of the traditional color spaces.

We show that many recently proposed gray-scale methods can be reformulated to take advantage of color images. However, the amount by which they benefit from color is different. The way how color is used can also affect the performance. Therefore, we suggest researchers to consider the case of color images when proposing new methods.

We show that although the performance of popular sparse coding technique is claimed to be feature invariant when feature dimension is high enough, it can be affected by color information. Two novel concepts are introduced namely the *correctness* and *discriminativeness (DIS)*. We propose a mathematical formula to measure DIS for a given sparse code in order to determine its discriminative power.

Lastly, we prove that even though Kinect provides only noisy depth data, it is still very useful for face recognition when utilized effectively. We also argue that existing 3D face recognition algorithms do not consider color information and only account for high resolution 3D data which is a clear disadvantage. With a properly designed algorithm, robust face recognition can be performed reliably with low cost.

1.2.3 Experimental Data and Protocols

Experimental data and protocols are important as they help evaluating proposed methods and provide unbiased comparison.

In this thesis, we design a few novel and repeatable experimental setups that cover many aspects of face recognition problems including seen and unseen identification, as well as unseen and partially-seen verification. These test protocols that resemble some useful real-world applications and have not been considered before.

Lastly, a new database namely CurtinFaces is constructed to complement existing databases. It is captured using the Kinect sensor, which provides only low resolution and noisy depth data. Images in CurtinFaces are captured with extreme variations in poses, expression, illumination and disguise. This is, to the best of our knowledge, the first publicly available Kinect face database that consists of images with such large variations.

1.3 Thesis Structure

The rest of this thesis is organized as follows.

In Chapter 2, some background knowledge is presented to aid understanding of this thesis. Some definitions about 2D, color and 3D face recognition are clarified. Some related 2D subspace methods are introduced. We also detail some recent 2D approaches based on tensor and sparse coding. Background knowledge about color face recognition is given at

last, which includes some existing color spaces and color face recognition methods.

Chapter 3 proposes the Pixel-level Discriminant Color Space (PLDCS) and Block-wise Discriminant Color Space (BWDCS) methods for color face recognition. We contrast the difference between our proposal and the original DCS method. Experiment is designed to justify the advantages of our algorithms. We also investigate on the advantages of applying DCS locally instead of holistically.

Chapter 4 presents the Multiple Color Fusion (MCF) algorithm, which utilizes multiple color components from across various linear and non-linear color spaces, for face recognition. We detail the advantages of MCF and evaluate it against other existing color face recognition methods. Some weaknesses of MCF are also identified. We also show that different color components carry different but complementary information.

Chapter 5 formulates the popular sparse coding framework to work on color images. We first discuss the challenges and advantages of our formulation. We also analyze different choices of formulations, detail and justify our approach. The concept of *Correctness* and *Discriminativeness* (DIS) is introduced to describe the discriminative power of a given sparse code. We propose a mathematical way to measure DIS. Several experiments are designed to validate our claims. Experimental results are analyzed at the end.

Chapter 6 investigates the feasibility of using low quality RGB-D (Red, Green, Blue and Depth) data from the Kinect sensor for face recognition under challenging conditions. We introduce a new 3D face database acquired using the Kinect sensor for our experiments and for the research community. Two algorithms are presented to utilize Kinect data for robust face recognition. The first algorithm namely the Multilinear Color Tensor Discriminant (MCTD), makes use of only 2D color face images. We show that this method outperforms several other state-of-the-art 2D methods. The second algorithm is proposed to utilize the noisy Kinect depth data with a novel Finer Feature Fusion (FFF) technique. We show that FFF has more robust performance and therefore justifies the usefulness of noisy Kinect 3D data.

Chapter 7 proposes a multi-modal face recognition algorithm that is robust to variations in pose, illumination, facial expressions and disguise. This algorithm consists of several components which are described in detail with justification and analysis. Experiments are carried out extensively on four challenging public databases in order to show the outstanding performance of our proposed method.

Finally, Chapter 8 concludes this thesis in the perspective of color spaces, recognition

methods and RGB-D face recognition. Some interesting future directions are also described.

Chapter 2

Background

In this chapter, some background knowledge is presented. Firstly, we introduce several notations about 2D, color and 3D face recognition. Secondly, a few 2D subspace methods are introduced. Thirdly, we detail some recent approaches based on tensor and sparse coding. Lastly, background information on color face recognition is given, which includes the introduction of some existing color spaces and methods. A summary is given at the end.

2.1 Notations and Terminologies

Data structures, frequently used in image processing, are mathematically defined as follows. Assume the resolution of an image is $r \times c$. It is usually organized as a d-dimensional column vector (where $d = r \times c$ is the total number of pixels), by stacking all its columns.

Two dimensional face data (or *texture*) image can be either gray-scale or colored. *Gray-scale image* is a vector $v \in \mathbb{R}^d$ of the light intensity at each pixel location. A *Color image* is usually modeled in RGB color space, which is a matrix $[R, G, B] \in \mathbb{R}^{d \times 3}$, where R, G and B are the Red, Green and Blue *color components* (or *channels*), each representing the light intensity of its spectrum that has been sensed or captured by the camera.

Three dimensional face data is usually acquired in the form of range image (2.5D image). A *range image* is a single view of the object, containing part of the 3D information sampled on a rectangular $r \times c$ image grid that is visible to the camera. Each sampled data point has its 3D coordinates denoted as $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$. Converting each of them to vector form, results in a matrix $[\mathbf{x}, \mathbf{y}, \mathbf{z}] \in \mathbb{R}^{d \times 3}$, which is termed as the *point-cloud*. Along with range image, most of the 3D acquisition devices also acquire the corresponding 2D texture image with color. *Multi-modal data* refers to 2D+3D data, which is a matrix $[\mathbf{x}, \mathbf{y}, \mathbf{z}, R, G, B] \in \mathbb{R}^{d \times 6}$ or $[\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{v}] \in \mathbb{R}^{d \times 4}$ if the texture is in gray-scale.

Through out this thesis, several terms that refer to different types of approaches are defined

as follows. *Gray-scale face recognition* refers to approaches that use 2D gray-scale image, whereas *color face recognition* refers to approaches that use 2D color image. The term *3D face recognition* refers to approaches that use 3D or range image data. *Multi-modal* approaches are 3D face recognition methods that additionally use 2D data.

2.2 Subspace Methods

Statistical approaches are proven to be effective for face recognition. In order to obtain reliable statistical results, dense samples are necessary. However, face recognition is usually a Small Sample Sized (SSS) problem. Most of the cameras nowadays offer imaging resolution over 4 megapixels (or 2240 $\times$ 1680), whereas in most face recognition applications only one or a few training samples per subject are available. These samples lie in such a high dimensional space that is too sparse to draw statistical significance. This is commonly referred to as the Curse of Dimensionality problem. Therefore, dimension reduction is usually necessary for statistical methods. Subspace methods project the data into a low dimensional subspace and the Nearest Neighbor (NN) classifier is applied subsequently to decide the identities of the test samples.

Existing subspace methods are mostly proposed for gray-scale images, and some basic notations can be defined. Given n gray-scale training face images with d pixels, they are commonly organized into a single matrix A, by arranging each image vectors v as a column of A:

$$A = [v_1, v_2, \ldots, v_n] \in \mathbb{R}^{d \times n}. \tag{2.1}$$

Assume that there are C classes each having M_i images with class label P_i (where $n = \sum_{i=1}^{C} M_i$). Let v_j (j=1,2,...,n) denote the j-th image vector in A. The class mean image $\overline{A}_i$ can be computed as:

$$\overline{A}_i = \frac{1}{M_i} \sum_{v_j \in P_i} v_j, \tag{2.2}$$

and the grand mean image $\overline{A}$ can be computed as:

$$\overline{A} = \frac{1}{n} \sum_{i=1}^{n} v_i. \tag{2.3}$$

2.2.1 Principal Component Analysis (PCA)

This section introduces the original PCA and the whitened PCA methods. The original PCA method is used in this thesis mainly for dimensional reduction by preserving certain percentage of energy, which will be explained here in detail. The whitened PCA is an improvement over the original PCA method for face recognition.

Eigenface is one of the earliest successful statistical method for face recognition proposed by Turk and Pentland (1991) based on PCA. Following the image data arrangement defined in Eq. 2.1, the covariance matrix can be computed as follows:

$$\Sigma = (A - \overline{A})(A - \overline{A})^T, \tag{2.4}$$

where $\overline{A}$ is the mean image vector in Eq. 2.3. The PCA projection is an orthonormal eigenvector matrix $W = [w_1, w_2, \ldots, w_r]$, containing eigenvectors of Σ arranged in descending order of their associated eigenvalues $\Omega_r = [\lambda_1, \lambda_2, \ldots, \lambda_r]$, where r is the rank of Σ. Usually, the eigenvectors with small eigenvalues are related to noise and therefore are discarded to improve performance. Since the number of useful eigenvectors varies under different data conditions, one of the common rules is to keep s eigenvectors to preserve $g\%$ energy, i.e.

$$\underset{s}{argmax} \frac{\sum_{i=1}^{s} \lambda_i}{\sum \lambda} \quad s.t. \quad \frac{\sum_{i=1}^{s} \lambda_i}{\sum \lambda} \leq g. \tag{2.5}$$

Finally, the dimension of A can be reduced by projecting to the subspace W:

$$\widetilde{A}_{pca} = W^T A. \tag{2.6}$$

The matrix $\widetilde{A}_{pca}$, consisting of PCA features extracted from A, can be used directly for NN classification. The whitened PCA (wPCA) (Deng *et al.*, 2010a) increases the performance over PCA for face recognition under larger variations by balancing the eigenvectors. The physical meaning of the eigenvalues are the data variance along the corresponding eigenvector, thus the first few eigenvectors may dominate in magnitude due to large variations of illumination or noise. To stabilize the result, each principal direction is scaled to uniform the spread of the data:

$$\widetilde{A}_{wpca} = \Omega_r^{-1/2} \widetilde{A}_{pca}. \tag{2.7}$$

Applying NN classifier on $\widetilde{A}_{wpca}$ usually results in better performance than on $\widetilde{A}_{pca}$.

2.2.2 Linear Discriminant Analysis (LDA)

This section introduces the original LDA and regularized LDA methods. The LDA method is frequently used in this thesis for experiments. Several discriminative features and color

spaces are also derived based on LDA's idea. The regularized LDA is an improvement over the original LDA method for face recognition.

The Fisherface method is a successful subspace method for gray-scale face recognition proposed by Belhumeur *et al.* (1997) based on LDA. Unlike PCA which maximizes the data variance in the subspace, LDA maximizes inter-class separability. It is a supervised method which requires class labels for training. Following the image data arrangement defined in Eq. 2.1, the between-class matrix S_b and within-class scatter matrix S_w are defined as follows:

$$S_b = \sum_{i=1}^{C} (\overline{A}_i - \overline{A})(\overline{A}_i - \overline{A})^T, \tag{2.8}$$

$$S_w = \sum_{i=1}^{C} \sum_{v_j \in P_i} (v_j - \overline{A}_i)(v_j - \overline{A}_i)^T, \tag{2.9}$$

where $\overline{A}_i$ is the class mean in Eq. 2.2 and $\overline{A}$ is the grand mean in Eq. 2.3. S_b and S_w measure the between-class and within-class variance respectively and LDA finds a projection subspace W to maximize their ratio:

$$\underset{W}{argmax} \frac{tr(W^T S_b W)}{tr(W^T S_w W)}, \tag{2.10}$$

and the solution can be obtained by solving the equivalent generalized eigenvalue problem:

$$S_b W = \lambda S_w W. \tag{2.11}$$

Finally, the dimension of A can be reduced by projecting to the subspace W:

$$\widetilde{A}_{lda} = W^T A. \tag{2.12}$$

The matrix $\widetilde{A}_{lda}$ consisting of LDA features extracted from A can be used directly for NN classification. A problem encountered in LDA is related to the inverse of S_w in Eq. 2.11, which may not exist due to the fact that S_w is usually singular in most of the face recognition applications. To stabilize the result, the Fisherface algorithm applies PCA on A first for dimension reduction, resulting in a full rank S_w matrix in the PCA subspace, where LDA can then be applied afterward. One of the drawbacks of this approach is the loss of discriminative information during the PCA projection step. Instead of PCA projection, the regularized LDA method (Lu *et al.*, 2005) solves the original objective function directly with regularization:

$$\underset{W}{argmax} \frac{tr(W^T S_b W)}{tr(W^T (S_w + \lambda I) W)}, \tag{2.13}$$

where λ here is a user defined constant and I is the identity matrix. As a result, the regularized LDA outperforms Fisherface in terms of recognition accuracy.

2.2.3 Other Subspace methods

Although it is impossible to cover all existing subspace methods, this section briefly describes two other subspace methods that have been used in this thesis. These two methods, namely the supervised Locality Preserving Projection (sLPP) and Intrinsic Discriminant Analysis (IDA), can achieve good face recognition performance with their own advantages.

The Laplacianfaces method is proposed by He *et al.* (2005) as an unsupervised technique for face recognition, based on Locality Preserving Projection (LPP). Different from PCA and LDA which assume the face space to be Euclidean, LPP finds an embedding that preserves the local structure and obtains a subspace that best resembles the actual face manifold. When class labels are available, LPP projection can be computed with supervision to improve performance (Zheng *et al.*, 2006). The supervised LPP usually has higher face recognition accuracy than LDA if the underlying data structure is complex.

The Intrinsic Discrimnant Anlaysis (IDA) is a recent method proposed by Wang and Wu (2010). Unlike other subspace approaches, IDA is designed specifically for face recognition. It mathematically decomposes a face image into three components: facial commonness difference, individuality difference and intrapersonal difference. By maximizing the individuality difference while minimizing the intrapersonal difference, a subspace different to PCA, LDA and LPP is derived, which is more suitable for face recognition. As a result, IDA can achieve higher face recognition accuracy in comparison to LDA, LPP, etc.

2.2.4 The Nearest Neighbor Classification

All aforementioned subspace methods aim to find a projection subspace, and the Nearest Neighbor (NN) classifier is applied after projection. This section describes the NN classifier for sack of completeness. Two choices of the distance metric used for the nearest neighbor measurement are introduced.

Assume the face feature matrix $\widetilde{A}$ and the corresponding projection matrix W are obtained using one of the subspace methods. Let $q \in \mathbb{R}^d$ be a query (probe) face image vector. To decide the identity of q, it has to be projected onto the same subspace first, i.e.: $\widetilde{q} = W^T q$. Afterwards, q is classified to class P_i if its nearest neighbor belongs to P_i. Image A_j is said to be the nearest neighbor of q if q has the shortest distance to A_j, i.e.

$$\underset{j}{argmin} \;\; dist(\widetilde{A}_j \;,\; \widetilde{q}), \quad (j = 1, 2, \ldots, n). \tag{2.14}$$

The $dist(a, b)$ function is commonly evaluated using Euclidean distance, i.e. $\|a - b\|_2$. However, sometimes the cosine similarity is used instead, i.e. $-(a \cdot b)/(\|a\| \cdot \|b\|)$. The better choice of metric depends on the data, therefore one way to choose is via cross-validation.

2.3 Multi-linear Analysis

In this section, some multilinear based methods are introduced. As mentioned in Section 1.1, this thesis focuses on user non-intrusive face recognition applications and therefore face image probes may have large variations such as pose, illumination and expression. Subspace methods described in Section 2.2 are all formulated linearly, and therefore have limited ability to handle non-linear variations especially for poses. Although there are manifold methods in existing literature to deal with non-linear data, they have several disadvantages. Firstly, manifold methods usually require dense sampling which is hard to achieve when the user is not cooperated. Secondly, even with sufficient samples, the true manifold may be too complicated to be modeled accurately or reliably enough. Lastly, the computation cost is usually expensive for manifold methods. Multilinear analysis is an alternative approach to handle largely varied images with low computational cost. Multilinear methods assume that a face image is a multilinear function of various factors (i.e. person, lighting, pose, pixel etc). The main idea is to deal with each factor linearly, in which linear methods can be employed. For this purpose, all training samples are represented by a single data tensor, with different factors modeled as different modes. Before detailing the technique, some tensor properties and operations have to be defined first. Then the Multilinear PCA (MPCA) method as well as its improved version MPCA-PS are introduced for face recognition.

2.3.1 Tensor Operations

Tensor is an object describing relations between vectors. It can be viewed as a multi-dimensional array of numerical values, which extends the notion of scalar, vector and matrix. For example, a d-dimensional array can be referred to as a dth-order tensor when using tensor terminology. The *order* of a tensor is the number of indices required to label an entry. Each dimension of the tensor is denoted as a *mode*. Let $A \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_d}$ be a tensor of order d. The following tensor operations are defined:

Unfolding is a unary operation on a single tensor. If $d > 2$, A can be unfolded at kth

mode, resulting in a 2D matrix $A_{(k)} \in \mathbb{R}^{I_k \times (I_{k+1} \cdots I_d I_1 \cdots I_{k-1})}$ which puts the kth dimension data as the rows and concatenate all data in other modes as the columns.

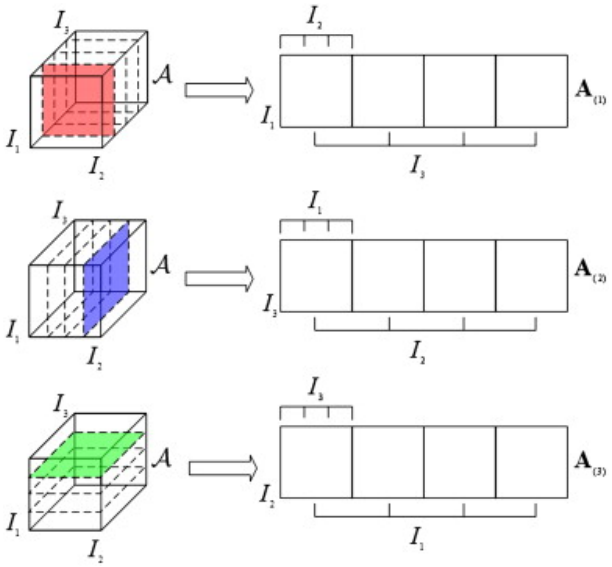


Figure 2.1: Illustration of tensor unfolding operation (Jia *et al.*, 2012).

Mode-k multiplication is an operation between a tensor and a matrix. A can be multiplied by the matrix $U \in \mathbb{R}^{I_k \times I_k}$ at kth mode denoted by $B = A \times_k U$. The tensor $B \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{k-1} \times I_k \times Ik+1 \times \cdots \times I_d}$ can be obtained by unfolding A at kth mode, to perform normal matrix multiplication $U \cdot A_{(k)}$ and finally fold the resulting matrix back to tensor by directly reversing the unfolding operation. In other words,

$$(A \times_k U)_{(k)} = U \cdot A_{(k)}. \tag{2.15}$$

Kronecker product is referred to as the tensor product of two matrices denoted by $\otimes$, which generalizes the outer product of two vectors. The Kronecker product of $U \in \mathbb{R}^{a \times b}$ and $V \in \mathbb{R}^{c \times d}$ results in a $ac \times bd$ block matrix:

$$U \otimes V = \begin{bmatrix} u_{11}V & \cdots & u_{1b}V \\ \vdots & \ddots & \vdots \\ u_{a1}V & \cdots & u_{ab}V \end{bmatrix}. \tag{2.16}$$

Higher Order SVD (HOSVD) is a generalization of Singular Value Decomposition (SVD) from matrix to tensor. Applying HOSVD on A yields the following decomposition:

$$A = C \times_1 U^1 \times_2 U^2 \times_3 \cdots \times_d U^d, \tag{2.17}$$

where $U^k = SVD(A_{(k)})$ is an orthonormal matrix containing the ordered eigenvectors for the kth mode. $C = A \times_1 U^{1^T} \times_2 U^{2^T} \times_3 \cdots \times_d U^{d^T}$ is the *core tensor* containing the projections of A in each mode-specific eigen-subspace.

Frobenius norm of a tensor is denoted by the $\|.\|_F$ operator and it is computed as:

$$\|A\|_F = \|A_{(k)}\|_F. \tag{2.18}$$

2.3.2 Multilinear PCA and MPCA-PS

Multilinear PCA (MPCA) is applied for gray-scale face recognition by Vasilescu and Terzopoulos (2002). Assuming that the training data contains samples of N_p people with N_l lighting conditions being captured under N_v viewpoints. All these training images are represented as a single fourth-order tensor T:

$$T \in \mathbb{R}^{N_p \times N_l \times N_v \times N_x}, \tag{2.19}$$

where

$$T(i_p, i_l, i_v) \in \mathbb{R}^{N_x} \tag{2.20}$$

indexes to an image vector of the i_p-th persion at i_l-th lighting condition and i_v-th viewpoint with N_x pixels. Applying HOSVD (as in Eq. 2.17) on T yields the following decomposition:

$$T = C \times_1 U^P \times_2 U^L \times_3 U^V \times_4 U^X, \tag{2.21}$$

where $U^P \in \mathbb{R}^{N_p \times N'_p}$, $U^L \in \mathbb{R}^{N_l \times N'_l}$, $U^V \in \mathbb{R}^{N_v \times N'_v}$ and $U^X \in \mathbb{R}^{N_x \times N'_x}$ are four orthonormal factor subspaces. N'_p, N'_l, N'_v and N'_x are the numbers of leading eigenvectors. C is the core tensor which controls the mutual interaction between the subspaces. Note that U^X is actually the traditional eigenface. Similarly, U^P, U^L and U^V represent the eigen-person, eigen-lighting and eigen-viewpoint respectively. Since each factor subspace is orthonormal, each of their rows contains coefficients that can be used to reconstruct a factor. The full training dataset can be reconstructed approximately by

$$T \approx C \times_1 U^P \times_2 U^L \times_3 U^V \times_4 U^X. \tag{2.22}$$

Based on the above MPCA framework for face image representation, there are several recognition approaches. Among them, MPCA-PS (Rana *et al.*, 2009) is one of the most promising. Assuming that a query image q is one of the people in training set T, MPCA-PS classifies a query q to person k who has the minimum value of the following optimization problem:

$$\min_{k,u_l,u_v} \|q - C \times_1 u_p^k \times_2 u_l \times_3 u_v \times_4 U_X\|_2, \tag{2.23}$$

where u_p^k is the kth row of U_p ($k = 1, \ldots, N_p$). u_l and u_v are two free variables used to reconstruct the lighting and viewpoint modes respectively. Rana *et al.* (2009) showed that there is a direct least square solution for solving u_l and u_v. High recognition performance is also achieved by MPCA-PS.

2.4 Sparse Coding

In this section, some sparse coding based methods for gray-scale face recognition are introduced. Unlike the problem of varied imaging conditions, a face image with disguise and noise carries unrelated pixels. The ability to identify these unrelated pixels is crucial to handle occlusion successfully. Methods that are based on sparse coding have been shown to perform exceptionally well for this problem. Specifically, the Sparse Representation Classifier (SRC) (Wright *et al.*, 2009) has achieved the state-of-the-art performance for corruption or occlusion problems, and has received much attention recently. Besides, the Correntropy-based Sparse Representation (CESR) (He *et al.*, 2010) and Robust Sparse Coding (RSC) (Yang *et al.*, 2011) methods have been proposed to improve over SRC. This section details these methods.

2.4.1 Sparse Representation Classifier

The Sparse Representation Classifier (SRC) is proposed by Wright *et al.* (2009), who cast the face recognition problem as a linear regression problem. It assumes all human face images lie in a linear space, faces of the same person lie in its local linear subspace and the linear spaces for different people are separable. Therefore if the training sample size is large enough to represent the whole face population (i.e., spanning the entire space for each person), any facial image can be represented as a linear combination of training images from the same class. Given a query image, though its membership is unknown, it can still be represented by a linear combination of all the training images. With intuition of the sparsest representation, the recovered combination coefficients are expected to be zero except those associated with the same class as the query, and this will reveal the membership of the query.

Let $A = [A_1, A_2, \ldots, A_N] \in \mathbb{R}^{d \times N}$ be N training samples, where d is the data dimension. Statistically, assume that the training samples span the whole human face space, any new

query face image y can be represented as a linear combination of all training samples:

$$y \approx Ax_0 \in \mathbb{R}^d \tag{2.24}$$

where $x_0 \in \mathbb{R}^N$ is the coefficient vector whose entries are expected to be zero except those associated with the same class as y and this can be found by maximizing its sparsity by solving the following equation:

$$x_0 = \min \ \|x\|_0 \quad \text{s.t.} \quad Ax = y \tag{2.25}$$

where $\|\cdot\|_0$ denotes the ℓ^0-norm, which counts the number of non-zero entries in a vector. In fact, solving Eq. 2.25 is NP-hard, however theoretical analysis (Sharon *et al.*, 2009) allows recovery of the correct x_0 via the following ℓ^1-norm minimization,

$$x_1 = \min \ \|x\|_1 \quad \text{s.t.} \quad Ax = y \tag{2.26}$$

as long as x_0 is sparse enough.

To increase robustness, Wright *et al.* (2009) further extended the model in Eq. 2.26 to deal with noise. Let $e \in \mathbb{R}^d$ be the error vector, the linear model in Eq. 2.24 becomes

$$y \approx Ax_0 + e \tag{2.27}$$

and the objective function in Eq. 2.26 in this case changes to:

$$\min_x \|x\|_1 \quad \text{s.t.} \quad \|Ax - y\|_2 \leq \varepsilon \tag{2.28}$$

Since the noise level ε is unknown beforehand, one possible alternative to solve this problem is to use the Lasso (Tibshirani, 1994) formulation by solving:

$$\min_{x,e} \|y - Ax + e\|_2^2 + \lambda \left(\|x\|_1 + \|e\|_1\right) \tag{2.29}$$

which is the same as:

$$\min_w \|y - Bw\|_2^2 + \lambda \left(\|w\|_1\right) \tag{2.30}$$

with

$$Ax + e = [A, I] \begin{bmatrix} x \\ e \end{bmatrix} = Bw \tag{2.31}$$

where λ is a given regularization parameter controlling the sparsity.

In recognition stage, the query y is classified to class i in which its associated coefficients would reproduce y with the smallest error, i.e.,

$$\min_i r_i(y) = \|y - A\delta_i(x_1)\|_2 \tag{2.32}$$

where $\delta_i : \mathbb{R}^N \to \mathbb{R}^N$ is the characteristic function that selects the coefficients associated with the ith class and $\delta_i(x) \in \mathbb{R}^N$ is a vector whose non-zero entries are the entries in x corresponding to class i.

2.4.2 Correntropy-based Sparse Representation

Recently, He *et al.* (2010) proposed a more robust improvement by integrating correntropy measure and non-negative constraint into the sparse model, namely the correntropy-based sparse representation (CESR). The correntropy firstly proposed by Liu *et al.* (2007) from concepts of correlation and entropy. It is a nonlinear similarity measure between two random variables P and Q. Let P and Q have finite number of samples $\{(P_j, Q_j)\}_{j=1}^{N}$, the sampling correntropy is estimated as:

$$\hat{V}_{N,\sigma}(P,Q) = \frac{1}{N}\sum_{j=1}^{N} k_\sigma(P_j - Q_j) \tag{2.33}$$

where $k_\sigma(\cdot)$ is a kernel function satisfying the Mercer condition (Vapnik, 1995). Instead of minimizing the reconstruction error in Eq. 2.29, CESR aims to find a non-negative sparse representation coefficient $x = (x_1, \ldots, x_N)^T \in \mathbb{R}^N$ to maximize the correntropy between the original query image $y = (y_1, \ldots, y_d)^T \in \mathbb{R}^d$ and the reconstructed query image $\hat{y} = Ax = (\sum_j A_{j1}\ x_j, \ldots, \sum_j A_{jd}\ x_j)^T \in \mathbb{R}^d$, for $j = 1, \ldots, N$. Thus the objective function for CESR becomes:

$$\max_x \sum_{k=1}^{d} g\left(y_k - \sum_{j=1}^{N} A_{jk}x_j\right) - \lambda\sum_{j=1}^{n} x_j \quad \text{s.t.} \quad x_j \geq 0 \tag{2.34}$$

where $g(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)$ is a Gaussian kernel function and σ^2 is the kernel size. As presented by He *et al.* (2010), Eq. 2.34 can be solved via the half-quadratic technique and expectation maximization method. They proposed an iterative solution, which also compute the kernel size σ^2 in each iteration.

For classification, instead of finding the minimum residual as in Eq. 2.32, y is classified to class i which its associating coefficients would best reproduce y in terms of the maximum correntropy (similarity):

$$\max_i\ S_i(y) = \sum_{k=1}^{d} g\left(y_k - \sum_{j=1}^{N} A_{jk}\hat{x}_j^i\right) \tag{2.35}$$

where $\hat{x}^i = \delta_i(x) \in \mathbb{R}^N$ is a new vector whose non-zero entries are from x that associate with class i as defined in Eq. 2.32.

Unlike the error modeled SRC in Eq. 2.30, optimizing both the error e and coefficient x at the same time, which is computationally intensive, CESR iteratively updates the weights for each pixels (locating the errors) and finds the sparse coefficients based on the pixel-weighted samples. Furthermore, the correntropy measure is claimed to be robust

against non-gaussian noise (Liu *et al.*, 2007). Therefore, CESR improves over SRC not only in computational efficiency but also in accuracy.

2.4.3 Robust Sparse Coding

The Robust Sparse Coding (RSC) method (Yang *et al.*, 2011) greatly improves the performance of SRC by introducing pixel weights with a different formulation to CESR. For the sake of simplicity, consider Eq. 2.29 without the error term:

$$\min_x \|y - Ax\|_2^2 + \lambda\|x\|_1. \tag{2.36}$$

Yang *et al.* (2011) pointed out that, the fidelity term $\|y - Ax\|_2^2$ implicitly assumes the data has gaussian distribution. However, the actual distribution may be far from this assumption. For example, if the data has Laplacian distribution, ℓ_1-norm will be more suitable (i.e. $\|y - Ax\|_1$). Since the distribution is unknown beforehand, a distribution function f_θ should be used in the fidelity term (i.e. $f_\theta(y - Ax)$) instead. Yang *et al.* (2011) showed that this distribution actually induces weights and therefore they reformulated Eq. 2.36 as the following weighted Lasso problem:

$$\min_x \|W(Ax - y)\|_2 + \lambda\|x\|_1, \tag{2.37}$$

where a robust W can be estimated by:

$$W = \frac{\exp(\mu\delta - \mu(e)^2)}{(1 + \exp(\mu\delta - \mu(e)^2))}. \tag{2.38}$$

Here in Eq. 2.38, $e = Ax - y$ is a vector of reconstruction residuals, μ and δ are user defined parameters controlling the rate of decrease and the location of demarcation point respectively. $W^{(1)}$ is initialized as the residual to the mean dictionary atom $e^{(1)} = \overline{A} - y$. Eq. 2.37 is then iteratively solved for x and W. The iteration stops at the t-th iterations when the change in W is smaller than ε, i.e:

$$\|W^{(t)} - W^{(t-1)}\|_2 / \|W^{(t-1)}\|_2 < \varepsilon \tag{2.39}$$

In recognition stage, a strategy similar to Eq. 2.32 is used. The query y is classified to class i if it satisfies:

$$\min_i r_i(y) = \|W(y - A\delta_i(x_1))\|_2 \tag{2.40}$$

where δ_i is the same characteristic function defined in Eq. 2.32.

2.5 Color Face Recognition

All methods described in Section 2.2, 2.3 and 2.4 are proposed for gray-scale images only. However, as discussed in Chapter 1, color information is very useful and should not be ignored. Most of the existing color face recognition methods focus on deriving new color spaces. Once the new color space is derived, traditional subspace methods are used to decide the identity. We briefly describe their recognition pattern as follows. Given a color image, it is usually modeled in a color space of three components (e.g. Red, Green and Blue in RGB color space) organized as a matrix $A_s = [c1, c2, c3] \in \mathbb{R}^{d\times 3}$. In order to apply a subspace method on A_s, it is converted into one column vector by stacking the three color components (image level fusion), i.e. $A_v = [c1\ c2\ c3]^T \in \mathbb{R}^{3d}$. After fusion, different subspace methods can be applied directly on A_v as in the case of gray-scale image. In this section, several color spaces and some approaches with slight variation to this pattern are introduced.

2.5.1 Conventional Color Spaces

The RGB color space is an additive color space. Any color can be produced by mixing the three elemental colors: Red, Green and Blue. Therefore, storing these three color components is sufficient to reproduce a color image. The RGB space is a fundamental color space as a RGB color image can be converted to another color space by either linear or non-linear transformation (which is referred to as *linear* and *non-linear color space* respectively hereafter).

Linear color spaces is obtained by linear matrix multiplication, i.e. $A_s = A_{rgb}T_s$, where $A_{rgb} = [R, G, B] \in \mathbb{R}^{d\times 3}$ is the RGB color image, and $T_s \in \mathbb{R}^{3\times 3}$ is the transformation matrix. The transformation matrix for RGB color space can be defined as the identity matrix:

$$T_{rgb} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.41}$$

The $I_1I_2I_3$ color space (Ohta, 1985) is obtained by de-correlating RGB using Karhunen-Loève transform. The transformation is defined as follows:

$$T_{I_1I_2I_3} = \begin{bmatrix} 1/3 & 1/2 & -1/2 \\ 1/3 & 0 & 1 \\ 1/3 & -1/2 & -1/2 \end{bmatrix} \tag{2.42}$$

The YIQ, YUV and YC_bC_r color spaces are designed to take advantage of human color-response characteristics, by separating the eye-sensitive colors and less sensitive ones in different components. They represent a colour in luminance (Y) and chrominance (IQ, UV or CbCr) components respectively. Therefore they allow better compression and are less sensitive to perceptible errors during image or video processing, transmission and display (Buchsbaum, 1975). Their transformation matrices are defined as follows:

$$T_{YIQ} = \begin{bmatrix} 0.2990 & 0.5957 & 0.2115 \\ 0.5870 & -0.2744 & -0.5226 \\ 0.1140 & -0.3213 & 0.3111 \end{bmatrix}, \tag{2.43}$$

$$T_{YUV} = \begin{bmatrix} 0.2990 & -0.1471 & 0.6148 \\ 0.5870 & -0.2888 & -0.5148 \\ 0.1140 & 0.4359 & -0.1000 \end{bmatrix}, \tag{2.44}$$

$$T_{YC_bC_r} = \begin{bmatrix} \frac{0.2126\times219}{255} & \frac{0.2126\times224}{1.8556\times255} & \frac{0.5\times224}{255} \\ \frac{0.7152\times219}{255} & \frac{0.7152\times224}{1.8556\times255} & -\frac{0.7152\times224}{1.5748\times255} \\ \frac{0.0722\times219}{255} & \frac{0.5\times224}{255} & -\frac{0.0722\times224}{1.5748\times255} \end{bmatrix} \tag{2.45}$$

(note that for the YC_bC_r color space, an offset of $[16, 128, 128]$ is required to be added to the three components after the transformation).

The XYZ color space is derived from a series of experiments in the study of the human perception by the International Commission on Illumination (CIE)(Weeks, 1996). The transformation matrix is:

$$T_{XYZ} = \begin{bmatrix} 0.607 & 0.299 & 0.000 \\ 0.174 & 0.587 & 0.066 \\ 0.201 & 0.114 & 1.117 \end{bmatrix} \tag{2.46}$$

Non-linear color space is obtained by applying some functions that involve non-linear operations. The $L^*a^*b^*$ color space defined by CIE (Weeks, 1996) as a device-independent color space which describes all colors that are visible to the human eyes. It is converted from the XYZ color space non-linearly as follows:

$$\begin{aligned} L^* &= 116f(Y/Y_n) - 16 \\ a^* &= 500[f(X/X_n) - f(Y/Y_n)] \\ b^* &= 200[f(Y/Y_n) - f(Z/Z_n)] \end{aligned} \tag{2.47}$$

where

$$f(t) = \begin{cases} t^{1/3}, & \text{if} \quad t > (\frac{6}{29})^3 \\ \frac{1}{3}(\frac{29}{6})^2 + \frac{4}{29}, & \text{otherwise} \end{cases}$$

and $[X_n, Y_n, Z_n]$ is the reference white point defined as $[0.950456, 1, 1.088754]$. Similar to $L^*a^*b^*$ color space, $L^*u^*v^*$ and $L^*c^*h^*$ are two other color spaces from the same family having similar transformations and purposes.

The HSL color space rearranges the geometry of RGB relating to human perception of color. The HSL components represent the *Hue*, *Saturation* and *Lightness* which are defined as follows:

$$H = 60^o \times \begin{cases} 0, & \text{if} \quad C = 0 \\ \frac{G-B}{C} \bmod 6, & \text{if} \quad M = R \\ \frac{B-R}{C} + 2, & \text{if} \quad M = G \\ \frac{R-G}{C} + 4, & \text{if} \quad M = B \end{cases}, \tag{2.48}$$

$$S = \begin{cases} 0, & \text{if} \quad C = 0 \\ \frac{C}{1-|2L-1|}, & \text{otherwise} \end{cases}, \tag{2.49}$$

$$L = \frac{1}{2}(M + m), \tag{2.50}$$

where R, G and B are the Red, Green and Blue components of the RGB image, $M = \max(R, G, B)$, $m = \min(R, G, B)$ and $C = M - m$.

2.5.2 Normalized Color Spaces

Color Space Normalization (CSN) is proposed by Yang *et al.* (2010a) to improve the discrimination power of weak linear color spaces. Although all color spaces introduced in former section can be used for face recognition, it has been shown that the performance of some linear color spaces such as RGB and XYZ are relatively weak due to high correlation between color components. Observed by Yang *et al.* (2010a) that all color spaces such as $I_1I_2I_3$ and YUV satisfying the so-called Double Zero Sum (DZS) property (Yang *et al.*, 2010b), have lower inter-component correlation, which leads to higher face recognition performance. A color space is said to satisfy the DZS property if two row sums of its transformation matrix equal to zero, while the remaining one dose not. Defining it precisely, given a transformation matrix

$$T = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix},$$

let the row-sum-vector $b = [b_1, b_2, b_3]^T$ contains the sum over each row of T, i.e., $b_i = \sum_{j=1}^{3} a_{ij}$. The transformed color space is DZS if only two numbers in b equal to zero.

There are several ways to convert a non-DZS color space to DZS. One way is by removing the mean from the second and third rows of their transformation matrix, i.e.,

$$\widetilde{T} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} - m_2 & a_{22} - m_2 & a_{23} - m_2 \\ a_{31} - m_3 & a_{32} - m_3 & a_{33} - m_3 \end{bmatrix},$$

where $m_2 = b_2/3$ and $m_3 = b_3/3$ are the second and third row means. The experiments carried out by (Yang *et al.*, 2010a) have shown that the normalized RGB and XYZ spaces (nRGB and nXYZ) perform significantly better than their original. Their transformation matrices after normalization are:

$$T_{nRGB} = \begin{bmatrix} 1 & -1/3 & -1/3 \\ 0 & 2/3 & -1/3 \\ 0 & -1/3 & 2/3 \end{bmatrix}, \tag{2.51}$$

$$T_{nXYZ} = \begin{bmatrix} 0.6070 & -0.0343 & -0.3940 \\ 0.1740 & 0.2537 & -0.3280 \\ 0.2000 & -0.2193 & 0.7220 \end{bmatrix}. \tag{2.52}$$

Notice that both matrices satisfy the DZS property.

2.5.3 Learned Statistical Color Space

The **Uncorrelated Color Space (UCS)** proposed by Liu (2008) derives a color space based on PCA, where the three color components are statistically uncorrelated. The transformation matrix is derived from the training samples, therefore there is no predetermined transformation matrix like other linear color spaces mentioned in the former section. Since the Eigen vectors found by PCA are orthogonal, the derived UCS has no inter-component correlation. In detail, let $A_i \in \mathbb{R}^{d \times 3}$, for $i = 1, \ldots, N$, be the set of N RGB training images. The *color space covariance matrix* is calculated as:

$$\Sigma_A = \frac{1}{N} \sum_{i=1}^{N} (A_i - \bar{A})^T (A_i - \bar{A}), \tag{2.53}$$

where $\bar{A}$ is the grand mean image calculated as:

$$\bar{A} = \frac{1}{N} \sum_{i=1}^{N} A_i. \tag{2.54}$$

The transformation matrix T_{ucs} consists of the three left singular vectors obtained by factorizing Σ_A:

$$\Sigma_A = T_{ucs}\ \Lambda\ \tilde{T}_{ucs}, \tag{2.55}$$

where Λ is the Eigen value matrix.

The **Discriminant Color Space (DCS)** proposed by Yang and Liu (2008b) is derived in a similar way, but is based on LDA, utilizing class labels. Transforming to DCS not only reduces inter-component correlation (Yang *et al.*, 2010b), but also forms a statistically optimal space for maximum discrimination. The DCS transformation matrix W is chosen to maximize the following objective function:

$$J(W) = \frac{tr(W^T L_b W)}{tr(W^T L_w W)}, \tag{2.56}$$

where L_b and L_w are the *color space between-class scatter matrix* and *color space within-class scatter matrix* respectively. Assuming that there are C classes each having N_j $(j = 1, \ldots, C)$ samples with class label P_j and $N = \sum N_j$. The color space scatter matrices are derived as:

$$L_b = \sum_{j=1}^{C} (\bar{A}_j - \bar{A})^T (\bar{A}_j - \bar{A}), \tag{2.57}$$

$$L_w = \sum_{j=1}^{C} \sum_{A_i \in P_j} (A_i - \bar{A}_j)^T (A_i - \bar{A}_j), \tag{2.58}$$

where $\bar{A}_j$ is the jth class mean computed as:

$$\bar{A}_j = \frac{1}{N_j} \sum_{A_i \in P_j} A_i. \tag{2.59}$$

The solution to Eq. 2.56 is the generalized eigenvectors corresponding to the largest eigenvalues λ satisfying the following equation:

$$L_b W = \lambda L_w W. \tag{2.60}$$

After obtaining the transformation, the original image is converted to DCS with three discriminant color components, i.e. $[D^1, D^2, D^3] = [R, G, B]W$. Before stacking the three components to form one vector for subsequent LDA application, these components are normalized to zero mean and unit standard deviation to avoid magnitude dominance in one over the others. Specifically, let $\mu_k = \sum D^k$ be the mean of all elements in D^k and σ_k be corresponding standard deviation, the normalization is done by:

$$D^k \leftarrow \frac{D^k - \mu_k \mathbf{1}_{N \times 1}}{\sigma_k}, , \tag{2.61}$$

where $\mathbf{1}_{N \times 1}$ is an N-dimensional column vector with all ones.

2.5.4 Color Image Discriminant Model

Addressing the drawback of the DCS method described in last section, the extended General Color Image Discriminant (eGCID) model (Yang and Liu, 2008a) is proposed. Observing from Eq. 2.56, the objective of DCS is to maximize the ratio between the color space scatter matrices L_b and L_w. These two matrices account for within and between class variance for the original color images before vectorization. However, after DCS transformation, LDA is applied on the vectorized color images where the within and between class variance is no longer accounted by L_b and L_w but by S_b and S_w as in Eq. 2.10. Therefore, DCS may not be optimal in terms of class discrimination for the subsequent LDA operation.

The eGCID algorithm seeks the set of optimal color transformation coefficients and the LDA projection together using an iterative approach. Given N RGB training samples $A_i \in \mathbb{R}^{d\times 3}$, (for $i = 1, \ldots, N$) from C classes, each class has N_j ($j = 1, \ldots, C$) samples with class label P_j and $N = \sum N_j$. Let $X \in \mathbb{R}^{3\times 3}$ be the color transformation matrix, such that A_i can be converted to D-space by $D_i = A_i X$. The between-class and within-class scatter matrices in D-space are defined as follows:

$$\begin{aligned} S_b(X) &= \sum_{j=1}^{C} (\bar{D}_j - \bar{D})(\bar{D}_j - \bar{D})^T \\ &= \sum_{j=1}^{C} (\bar{A}_i - \bar{A}) X X^T (\bar{A}_i - \bar{A})^T, \end{aligned} \tag{2.62}$$

$$\begin{aligned} S_w(X) &= \sum_{j=1}^{C} \sum_{D_i \in P_j} (D_i - \bar{D}_j)(D_i - \bar{D}_j)^T \\ &= \sum_{j=1}^{C} \sum_{A_i \in P_j} (A_i - \bar{A}_j) X X^T (A_i - \bar{A}_j)^T. \end{aligned} \tag{2.63}$$

Let $W \in \mathbb{R}^{d\times \widetilde{d}}$ be the LDA projection matrix (where $\widetilde{d} \ll d$ is the user-defined number of LDA eigenvectors), such that A_i can be projected to LDA subspace by $\widetilde{A}_i = W^T A_i$. The color space between-class and within-class scatter matrices in the LDA subspace are defined as follows:

$$L_b(W) = \sum_{j=1}^{C} (\bar{A}^j - \bar{A})^T W W^T (\bar{A}^j - \bar{A}), \tag{2.64}$$

$$L_w(W) = \sum_{j=1}^{C} \sum_{A_i \in P_j} (A_i - \bar{A}^j)^T W W^T (A_i - \bar{A}^j). \tag{2.65}$$

The objective of eGCID is to maximize the following function:

$$J(W,X) = \frac{tr(W^T S_b(X) W)}{tr(W^T S_w(X) W)}.. \tag{2.66}$$

Yang and Liu (2008a) have shown that Eq. 2.66 is equivalent to

$$J(W,X) = \frac{tr(X^T L_b(W) X)}{tr(X^T L_w(W) X)}.. \tag{2.67}$$

For these two equations, when one variable is fixed, they become the generalized eigenvalue problems as in Eq. 2.11 and Eq. 2.60. Therefore, an iterative solution can be derived by first initializing X with, for example random numbers, and solves for W. Then uses W to solve for X and iterates, until the change in $J(W,X)$ is less then ε.

In above algorithm, $X_1 = [x_{11}, x_{21}, x_{31}]^T$ is obtained as the eigenvector with the largest eigenvalue. It is the combination coefficients to obtain one discriminant color component $D^1 = x_{11}R + x_{21}G + X_{31}B$. As pointed out by Yang and Liu (2008a), one discriminant color component is not enough to retain all useful information in general. Therefore X_2 and X_3 are required and can be derived from the null-space of L_w. It is known that $X_i (i = 1, 2, 3)$ are required to be $L_w(W)$-orthogonal, i.e.:

$$X_i^T L_w(W) X_j = 0 \quad \forall i \neq j, \quad i, j = 1, 2, 3. \tag{2.68}$$

Let u_1 and u_2 be the remaining eigenvectors which is $L_w(W)$-orthogonal to X_1. In other words, u_1 and u_2 are the bases of the null space of L_w. Then X_2 can be expressed as a linear combination of u_1 and u_2, i.e.

$$X_2 = [u_1, u_2] \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = UY, \tag{2.69}$$

where y_1 and y_2 are two coefficients. Substituting $X_2 = UY$ into X in Eq. 2.67 results in the following equations:

$$\begin{aligned} J_2(W, X_2) &= \frac{tr(X_2^T L_b(W) X_2)}{tr(X_2^T L_w(W) X_2)} \\ &= \frac{tr(Y^T (U^T L_b(W) U) Y)}{tr(Y^T (U^T L_w(W) U) Y)} \\ &= \frac{tr(Y^T \widetilde{L}_b(W) Y)}{tr(Y^T \widetilde{L}_w(W) Y)}. \end{aligned} \tag{2.70}$$

Y can be obtained as the generalized eigenvector of $(\widetilde{L}_b(W), \widetilde{L}_w(W))$ with the largest eigenvalue and X_2 can be derived using Y. The the same iterative procedures used to find X_1, can be adopted to solve X_2. Similarly, the remaining color component must be $L_w(W)$-orthogonal complement of both X_1 and X_2, which is actually 1-D. Therefore, X_3 is unique when the length is fixed and the sign is neglected. Let Z be the smallest generalized eigenvector of $(\widetilde{L}_b(W), \widetilde{L}_w(W))$, then X_3 can be obtained by $X_3 = UZ$.

2.5.5 Tensor Discriminant Color Space

The eGCID method introduced above is based on vectorized image. However, image vectorization is a mechanical step that breaks the original image structure. Wang *et al.* (2011) proposed the Tensor Discriminant Color Space (TDCS) to extract the discriminant features, while preserves its matrix structure. Given N training RGB color images from C classes, each image can be naturally represented by a third-order tensor $A_i \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ (for $i = 1, \ldots, N$), where I_1, I_2 and $I_3 = 3$ are the number of rows, columns and color components of the images. Assumes that each class has N_j ($j = 1, \ldots, C$) samples with class label P_j and $N = \sum N_j$. TDCS finds two discriminant projection matrices $W_1 \in \mathbb{R}^{I_1 \times I_1'}$, $W_2 \in \mathbb{R}^{I_2 \times I_2'}$ and a color space transformation matrix $W_3 \in \mathbb{R}^{I_3 \times I_3'}$ (usually $I_1' < I_1$, $I_2' < I_2$ and $I_3' \leq I_3$), in order to extract the feature tensor $D_i \in \mathbb{R}^{I_1' \times I_2' \times I_3'}$:

$$D_i = A_i \times_1 W_1^T \times_2 W_2^T \times_3 W_3^T, \tag{2.71}$$

where $\times_k$ denotes the mode-k multiplication defined in Eq. 2.15. The grand mean and class mean image tensors can be computed as follows:

$$\bar{A} = \frac{1}{N} \sum_{i=1}^{N} A_i, \tag{2.72}$$

$$\bar{A}_j = \frac{1}{N_j} \sum_{A_i \in P_j} A_i. \tag{2.73}$$

The k-mode between-class scatter matrix $\Psi_b^{(k)}$ in the feature space can be computed as follows:

$$\begin{aligned}
\Psi_b^{(k)} &= \sum_{j=1}^{C} \|\bar{D}_j - \bar{D}\|^2_{(k)} \\
&= \sum_{j=1}^{C} \|(\bar{A}_j - \bar{A}) \times_1 W_1^T \times_2 W_2^T \times_3 W_3^T\|^2_{(k)} \\
&= \sum_{j=1}^{C} \|W_k^T (\bar{A}_{j(k)} - \bar{A}_{(k)}) \tilde{W}_k\| \\
&= \sum_{j=1}^{C} tr\Big[W_k^T (\bar{A}_{j(k)} - \bar{A}_{(k)}) \tilde{W}_k \tilde{W}_k^T (\bar{A}_{j(k)} - \bar{A}_{(k)})^T W_k\Big] \\
&= tr(W_k^T S_b^{(n)} W_k),
\end{aligned} \tag{2.74}$$

where $\tilde{W}_k = W_d \otimes \cdots \otimes W_{k+1} \otimes W_{k-1} \otimes \cdots \otimes W_1, k = 1, 2, \ldots, d$ and $d = 3$. The $\|.\|$ denotes the Frobenius norm of the tensor, $\cdot_{(k)}$ denotes k-mode unfolding and $\otimes$ denotes the Kronecker product, as defined in Section 2.3.1. Similarly, the k-mode within-class

scatter matrix $\Psi_w^{(k)}$ in the feature space can be defined as :

$$\begin{aligned}\Psi_w^{(k)} &= \sum_{j=1}^{C} \sum_{D_i \in P_j} \|D_i - \bar{D}_j\|^2_{(k)} \\ &= \sum_{j=1}^{C} \sum_{A_i \in P_j} tr\Big[W_k^T (A_{i(k)} - \bar{A}_{j(k)}) \tilde{W}_k \tilde{W}_k^T (A_{i(k)} - \bar{A}_{j(k)})^T W_k\Big] \\ &= tr(W_k^T S_w^{(n)} W_k).\end{aligned} \tag{2.75}$$

The objective of TDCS is to maximize the class separability in the feature space with the following criterion:

$$\max J(W_k) = \frac{\Psi_b^{(k)}}{\Psi_w^{(k)}} = \frac{tr(W_k^T S_b^{(k)} W_k)}{tr(W_k^T S_w^{(k)} W_k)}, (k = 1, 2, 3). \tag{2.76}$$

Since the above objective function consists of three variables, it can be optimized alternatively by solving its corresponding generalized eigenvalue decomposition problem while fixing any two of the three variables, until the change in $J(W_k)$ is less than ε.

2.5.6 Non Negative Matrix Factorization

Besides subspace and tensor methods, face recognition based on Non-negative Matrix Factorization (NMF) has also been investigated. This section briefly introduce some NMF based face recognition methods for sake of completeness.

The idea of NMF is to factorize a high dimensional matrix F into two low rank matrices H and W such that $F = W \times H$ and W and H contain no negative values. In the context of face recognition, F is usually the collection of training faces where each column is a image vector. The low rank factor matrix W can be interpreted as the dictionary where its columns are basis images that can be linearly combined to reconstruct the original face images. These combination coefficients (weighting) are stored in H which can be interpreted as the encoding matrix. Since faces are naturally discriminative, therefore, face images of the same person would have similar encoding while different people would have very different encoding vector. Utilizing this property, faces can be recognized by comparing its derived encoding vector with the database. Although there is no closed form solution for NMF, it can be approximated using a iterative solution (Lee and Seung, 1999).

There are a few variation of NMF algorithm proposed for face recognition. Li *et al.* (2001) proposed the Local NMF (LNMF), which aims to improve the locality of the learned

features by imposing constraints. Wang *et al.* (2005) proposed the PNMF and FNMF. They are derived with additional PCA and Fisher LDA criteria respectively and have shown improved recognition accuracy. For color face recognition, Rajapakse *et al.* (2004) proposed the color NMF algorithm. It performs NMF on each color channel separately and fuses the result in similarity score level for decision. They have shown that the performance can be improved significantly with color cover gray-scale images in their experiment.

2.6 Summary

This chapter has introduced some background knowledge related to this thesis. It begins with giving some basic definitions, followed by describing a few subspace methods (i.e. PCA, LDA, LPP and IDA) and the Nearest Neighbour (NN) classifier with different metrics such as Euclidean or Cosine distance. Then some definitions related to tensor are introduced in order to explain the multilinear PCA and MPCA-PS algorithm for face recognition. Some sparse coding methods including SRC, CESR and RSC are detailed next. We discuss that both multilinear and sparse coding techniques are more effective than linear subspace methods in term of dealing with images under large variations and outliers. Lastly, we introduce some conventional color spaces as well as some existing color face recognition methods such as DCS, CSN, CID and TDCS which are some of the current state of the art methods.

Chapter 3

Local Discriminant Color Space

Many researches on color face recognition have investigated on which color space is more suitable for face recognition (Yang *et al.*, 2010a,b). The most recent and promising one is the discriminate color space (DCS) (Yang and Liu, 2008b). As described in Section 2.5.3, DCS linearly combines R, G and B components of a color image using a set of optimal coefficients based on Fishers criterion. This space thus provides a theoretically optimal representation of a color image for recognition purpose, and it has shown outstanding performance with only one traditional classifier.

The shortcoming for DCS method is that it converts the entire image from RGB into DCS. This may not provide the best discriminant information for some parts of the color image since a color image consists of many different colors at various locations. For example, different lips may be more recognizable by red component, while different eyes may be more discriminable by blue component. Therefore, different blocks or pixels should be treated differently instead of as one image in DCS. Based on this motivation, this chapter proposes two novel color spaces namely the block-wise DCS (BWDCS), and the pixel-level DCS (PLDCS). They aim to find optimal discriminant color space for each block or each pixel in RGB images.

In this chapter, four subspace algorithms are considered, and each of them is integrated with the proposed color spaces. Improvement in their performances is noted on five different databases. The most recent preprocessing pipeline in (Tan and Triggs, 2010) is also integrated into the proposed system to form a complete color face recognition framework.

The rest of this chapter is organized as follows. The original DCS method is reviewed briefly in Section II. Section III addresses the pixel-level DCS and Section IV addresses the block-wise DCS. Experiments are provided in Section V. In Section VI PLDCS is compared against DCS and Section VII concludes the chapter.

3.1 Discriminant Color Space

The Discriminant Color Space (DCS) was originally proposed by Yang and Liu (2008b) for color face recognition. For detailed introduction on DCS, we shall refer the reader to Section 2.5.3. Nevertheless, some important DCS formulas are redefined in this section for sake of completeness and referencing purpose. Let A be a color image with a resolution of $H \times W$ consisting of three color components R, G and B. Assume they are represented as column vectors: $A = [R, G, B] \in \mathbb{R}^{P \times 3}$, where $P = H \times W$ is the number of pixels. The idea is to convert the image A in RGB space to an image D in a more discriminable space namely the DCS. Let the image D be:

$$D = [D^1, D^2, D^3] \in \mathbb{R}^{3 \times 3}. \tag{3.1}$$

The conversion is done by the following linear combination:

$$D^k = x_{1k}R + x_{2k}G + x_{3k}B = AX_k, \tag{3.2}$$

where $k = 1, 2, 3$ and $X_k = [x_{1k}, x_{2k}, x_{3k}]^T \in \mathbb{R}^{3 \times 1}$ is the combination coefficients that can be found via maximizing the following objective function:

$$J(X) = \frac{X^T L_b X}{X^T L w X}, \tag{3.3}$$

where $X = [X_1, X_2, X_3]^T \in \mathbb{R}^{3 \times 3}$, L_b and L_w are the between-class and within-class color-space scatter matrices. Assumes that we have N training images with C classes, T_i be the ith class label ($i = 1, 2, \cdots, C$) and N_i be the number of images in T_i. Then L_b and L_w are defined as:

$$L_b = \sum_{i=1}^{C} (\bar{A}_i - \bar{A})^T (\bar{A}_i - \bar{A}), \tag{3.4}$$

$$L_w = \sum_{i=1}^{C} \sum_{A_j \in T_i} (A_j - \bar{A}_i)^T (A_j - \bar{A}_i), \tag{3.5}$$

where $j = 1, 2, \cdots, N$. $\bar{A}$ and $\bar{A}_i$ are the global mean and class mean computed as:

$$\bar{A} = \frac{1}{N} \sum_{j=1}^{N} A_j, \tag{3.6}$$

$$\bar{A}_i = \frac{1}{N_i} \sum_{A_j \in T_i} A_j. \tag{3.7}$$

The optimal X that maximizes Eq. 3.3 is the generalized eigenvectors corresponding to the largest eigenvalues λ satisfying the equation:

$$L_b X = \lambda L_w X. \tag{3.8}$$

3.2 Pixel-level Discriminant Color Space

Unlike DCS which assigns the same coefficients for all pixels, we propose the Pixel-level Discriminant Color Space (PLDCS) which assigns different coefficients to different pixels, i.e.:

$$D_p^K = x_{1kp}R_p + x_{2kP}G_p + x_{3kp}B_p, \tag{3.9}$$

where $p = 1, 2, \cdots, P$. These coefficients are obtained by optimizing the objective function locally for each pixel. Then we convert each image A in RGB space to image E in PLDCS, which consists of P different DCS. Image E can be defined as:

$$E = D_p = [D_p^1, D_p^2, D_p^3]. \tag{3.10}$$

The algorithmic procedure to convert a color image from RGB to PLDCS is stated below:

Algorithm 3.1 Pixel-level Discriminant Color Space

Require: A, RGB training samples

1. Store each color image using color space rearrangement in which R, G and B are three column vectors, i.e. $A = [R, G, B] \in \mathbb{R}^{P\times 3}$.
2. Pixel extraction. From $p = 1$, extract the first pixel $A_p = [R_p, G_p, B_p] \in \mathbb{R}^{1\times 3}$.
3. Compute $\bar{A}_p$ and $\bar{A}_{ip}$ using (3.6) and (3.7). Then L_{bp} and L_{wp} using (3.4) and (3.5).
4. Obtain the local projection matrix X_p by solving (3.8).
5. Convert A_p to E_p using (3.9) and (3.10).
6. Repeat Step 2 for $p = p + 1$, until $p = P$.

return E

PLDCS is locally optimal for each pixel, while DCS is only holistically optimal, in terms of color face recognition. By allowing different projections at different locations, we expect the discriminative power of PLDCS will be stronger than DCS.

3.3 Block-wise Discriminant Color Space

Since PLDCS repeats calculations for each single pixel, it is expected to work slower. The main bottleneck is in the P times DCS calculations. PLDCS may also suffer from over-fitting problem, since single pixel does not have sufficient discriminative information to separate different identities. Instead of pixel calculation, we propose the Block-wise Discriminant Color Space (BWDCS) method which operates on blocks of an image to overcome the shortcomings stated.

For an image with resolution $H \times W$, we can divide the image into small blocks with size $h \times w$ ($h < H$ and $w < W$) and convert each block to DCS. For simplicity, we just reduce the block size to fit in the image edges, but other handling rules may also be applied, for example zero-padding or circulating pixels. Clearly, BWDCS is a generalization of PLDCS and DCS. If the block size is defined as $H \times W$, then it is same as the DCS method. Similarly, if the block size is defined as 1×1, then it is same as the PLDCS method. The algorithmic procedure to convert a color image from RGB to BWDCS is as follows:

Algorithm 3.2 Block-wise Discriminant Color Space

Require: A, RGB training samples

1. Store each color image using image matrix rearrangement in which R, G and B are three matrices, i.e. $A = [R, G, B] \in \mathbb{R}^{H \times W \times 3}$.
2. Extract h rows and w columns as a block (or sub-image), i.e. $\hat{A} = [\hat{R}, \hat{G}, \hat{B}] \in \mathbb{R}^{h \times w \times 3}$.
3. Rearrange $\hat{A}$ using color space rearrangement by converting $\hat{R}$, $\hat{G}$ and $\hat{B}$ to three augmented vectors $\hat{A} = [\hat{R}, \hat{G}, \hat{B}] \in \mathbb{R}^{(h*w) \times 3}$.
4. Compute $\overline{\hat{A}}$ and $\overline{\hat{A}}_i$ using (3.6) and (3.7). Then $\hat{L}_b$ and $\hat{L}_w$ using (3.4) and (3.5).
5. Obtain the local projection matrix $\hat{X}$ by solving (3.8).
6. Convert $\hat{A}$ to $\hat{D}$ using (3.1) and (3.2).
7. Repeat Step 2 for the next unconverted block, until every block is converted.

return $\hat{D}$

BWDCS has two main advantages. Firstly, it runs faster with the increase of block size. In terms of color space conversion, PLDCS has a complexity of $O(H \times W)$ while the complexity of BWDCS is $O(\frac{H \times W}{h \times w})$. Secondly, each calculation of BWDCS makes use of information from the adjacent pixels which are always correlated. Thus, the performance of BWDCS is expected to be more robust.

3.4 Experiments

In this section, we have evaluated our proposed color spaces by integrating them with four subspace methods respectively. We have also compared the performance against other color spaces as well as gray-scale (which is the origin of these four methods). This section presents details of the experiments in terms of performance statistic, datasets, integration methods and results.

3.4.1 Evaluation Protocol and Performance Analysis

Following the protocol in Biometrics (2006), we divide face recognition problems into two types: face identification problem and face verification problem. In our experiments, both problems are evaluated. The evaluation protocol used in this chapter is as follows. Each database is divided into three mutually exclusive sets: training set, gallery set and probe set. Training is done on the training set while testing is done on both gallery and probe sets. For identification problems, the probe images are matched to all of the gallery images to find the match. For verification problems, we consider accepting or rejecting the probe against the claimed gallery image. For each type of problem, the performance is calculated in a different manner. For identification problems, we report the rank-one identification rate on the Cumulative Match Characteristics (CMC) curve. For the verification problems, we report the face verification rate (FVR) at 0.001 false accept rate (FAR) on the Receiver operating characteristic (ROC) curve.

3.4.2 Databases and Experimental Setup

To ensure that our experimental results are unbiased, we extensively evaluate them on five color face databases.

- The Aberdeen database from Psychological Image Collection at Stirling (PICS). (Hancock, 2004)
- Georgia Tech face database (GT). (Nefian, 2007)
- AR face database (AR). (Martinez and Benavente, 1998)
- The Facial Recognition Technology (FERET). (J. Phillips, 2000)
- Face Recognition Grand Challenge ver. 2 exp. 4 (FRGC-204) (Phillips *et al.*, 2005)

Images in PICS (Hancock, 2004) are collected for research in Psychology, however, it is also suitable for the face recognition community given its variations in illumination, facial expression and pose. We select a subset of the database that consists of 29 people each having 9 images. Since this is a small database, we formulate the test as face identification problem: i.e., 4 images per person are randomly chosen for training as well as being the gallery images and the remaining 5 for probe images. To ensure that result is not depending on specific training/testing samples, the process is repeated 10 times and the average rank-one identification rate is reported with the standard deviation.

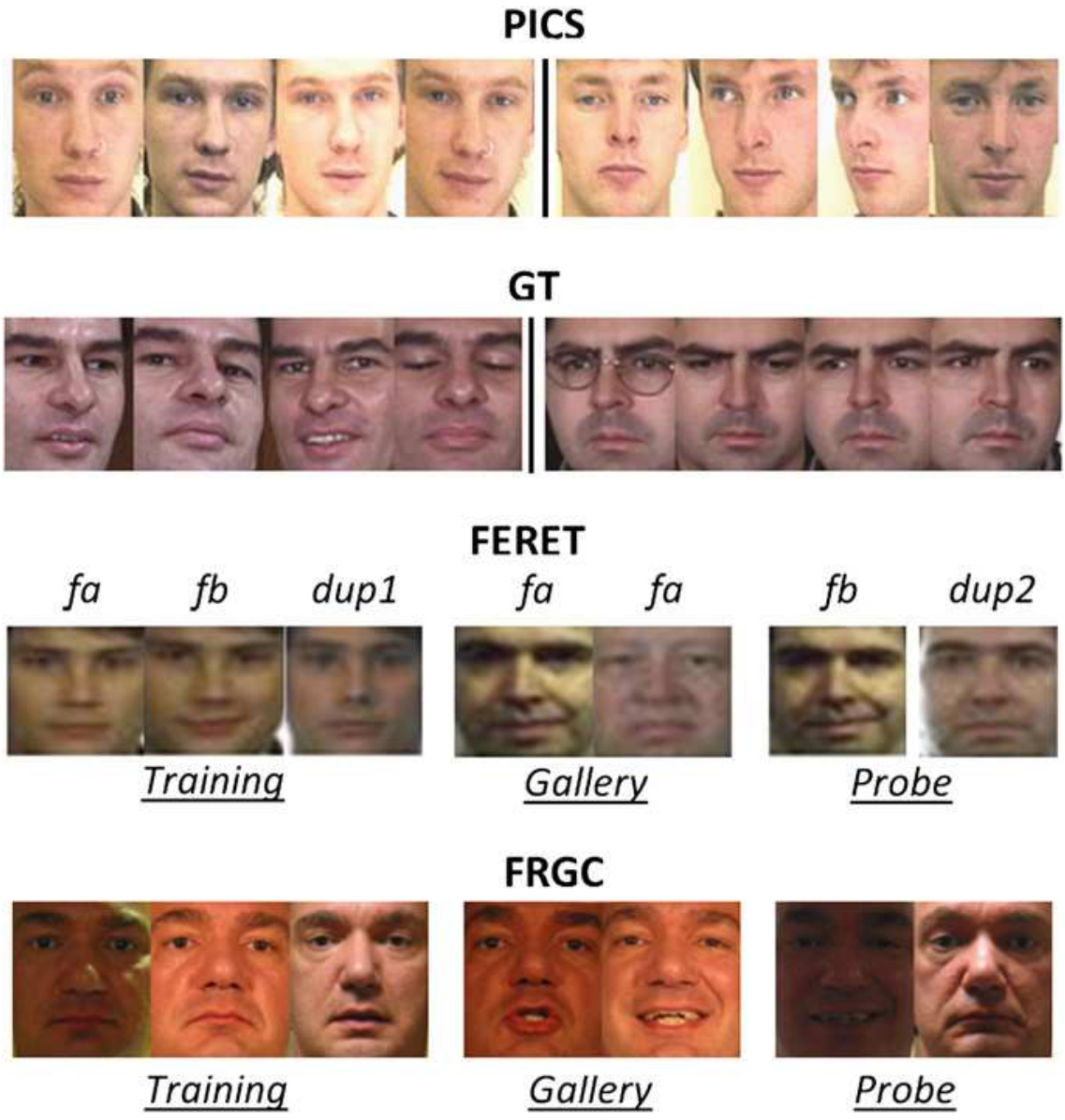


Figure 3.1: Example images from the four databases used in our experiments.

GT (Nefian, 2007) is a database often used for face recognition because of its variations in facial expression/details and pose. The database consists of 50 people each having 15 images. We test them as a face identification problem: i.e., using all images of 25 people randomly chosen for training and for the other 25 people, 8 images per person are randomly chosen as gallery and remaining images for probe, such that the subjects for testing are never been seen during training. By repeating 10 times, the average rank-one identification rate is reported with the standard deviation.

AR (Martinez and Benavente, 1998) is a publicly available face database, widely used for evaluating face identification problems. This database contains over 4000 images which are captured in two sessions with different facial expressions, illumination conditions and occlusions. We use a subset of all unoccluded images from the first 50 males and 50 females. As a result, a total of 1400 images from two sessions (14 images per subject) are involved in our experiment. We formulate a time-delayed face identification problem using images from session one for training and images from session two for testing. Note that the size of this database is much larger than PICS and GT. Therefore, this allows us to resemble a large-scale identification situation.

For FERET (J. Phillips, 2000), we are using the color version (though some images are still in gray-scale) which was released in year 2003. It was the de-facto standard dataset to evaluate face recognition systems at that time because of the strictly controlled parameters and support from US government. The database provides standard subsets namely fa, fb, dup1 and dup2. We select only the color images with eyes/lip coordinates provided, resulting in a total of 2419 images (808+806+593+212), from the four sets. We then construct the training set by selecting from fa the first 400 images/subjects that are not appearing in dup2 and then combining with the images of the corresponding subjects in fb and dup1 , resulting in 1013 images (400fa + 399fb + 214dup1). The rest of the 408 images from fa serve as gallery, the rest of 407 images from fb serve as one probe set and all 212 images from dup2 serve as another probe set. This design allows training on images that vary in facial expression and time delay up to 1.5 years, while testing on facial expression effect (fb) and aging effect of longer than 1.5 years (dup2). Subjects in training and testing sets are mutually exclusive. We report face verification rates on the two tests respectively.

FRGC version 2 (Phillips *et al.*, 2005) having more than 50000 records is constructed by the same organization as FERET, and is becoming the next major challenge for face recognition systems. We consider only the most challenging protocol, i.e. the experiment 4 ROC-III. This subset consists of controlled gallery images and uncontrolled probe images with pairs that is one semester different in collection time. This large-scaled dataset

exhibits large intrapersonal variations and some people look very similar due to strong illumination.

Some example images used in our experiments from each database are shown in Figure 3.1 as for ready reference. A summary is given in Table 3.1.

Table 3.1: Databases Summary

Name	Problem	Training	Gallery	Probe
PICS	Seen Identification	116	-	145
GT	Unseen Identification	375	200	175
AR	Large-scale Seen Identification	700	-	700
FERET	Unseen Verification	1013	408	407 (fb) 212 (dup2)
FRGC-204	Large-scale Partially-seen Verification	12776	16028	8014

3.4.3 System Pipeline

The face recognition system used in our experiment is detailed in this section. The major steps are image preprocessing, color space conversion, feature extraction and similarity score calculation as illustrated in Figure 3.2.

Preprocessing is done as following. Firstly, the face region is cropped out and resized to resolution of 32×32 after eyes and lip aligned to same position. We follow a tradition alignment procedure which first manually locates the two eyes and the lip on the image, then transits and scales the image such that these three identified points are aligned to the same pixel. Subspace methods usually have some tolerance to facial misalignment, however their performance may be greatly affected in case of significant misalignment. Nevertheless, the current state of art face detection algorithms are proven to be reasonably accurate and robust. Since the focus of this thesis is on the recognition stage rather than the detection stage, in this work we assume that the two eyes and lip have been detected on the face image.

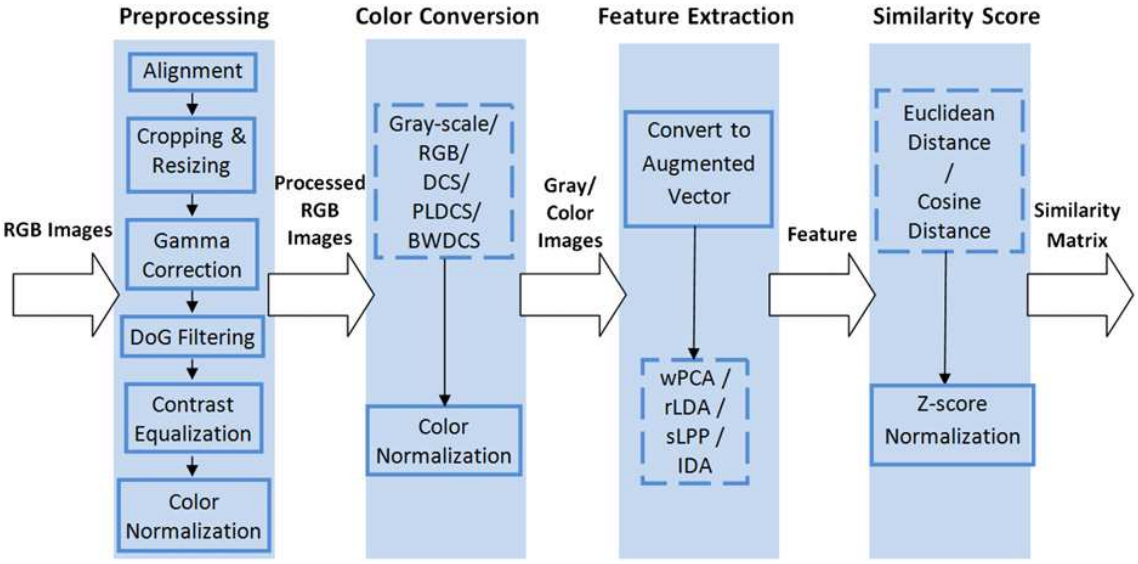


Figure 3.2: The face recognition system used in our experiment.

After facial alignment, we apply the preprocessing chain in (Tan and Triggs, 2010) on each of the color image components (R,G and B) separately, i.e. gamma correction, DoG filtering and contrast equalization to reduce illumination effect and enhance facial feature. The preprocessing parameters are tuned manually to the best for gray-scale images and recorded in Table 3.2. Note that the default parameters used by Tan and Triggs (2010) is not used here since they are optimized for descriptor methods rather than for subspace methods.

Color spaces that we have compared include the proposed PLDCS and BWDCS with DCS, RGB as well as gray-scale. The color normalization step is done similarly to DCS by Eq. 2.61, to avoid magnitude dominance in one color component over the others, which is explained in Section 2.5.3. Note that we apply the normalization before and after color space conversion as we find that it increases the performance.

Feature extraction methods that we have considered are:

- whitened Principal Component Analysis (wPCA) (Deng *et al.*, 2010a)
- regularized Linear Discriminant Analysis (rLDA) (Lu *et al.*, 2005)
- supervised Locality Preserving Projection (sLPP) (Zheng *et al.*, 2006)
- Intrinsic Discriminant Analysis (IDA) (Wang and Wu, 2010)

The main reason of choosing these methods is because we are extending the work of Yang

and Liu (2008b) where DCS was tested with appearance-based subspace method, thus we are testing on methods of the same type. Further, we want to test on methods that are stable and well-developed like PCA and LDA, but recent enough to compete with the state of the art. For each method, parameters are tuned manually to the best for gray-scale images and recorded in Table 3.2. Note that the parameter w of sLPP is a constant number added to the weight matrix. The reason is that when we construct the weighted graph for color space that allows negative values such as DCS, PLDCS and BWDCS, some weights become negative. However, constructing this graph using the original RGB image or Euclidean distance decreases the performance. We find that simply adding a constant number w to the weights increases the performance, thus we introduce this parameter. More detail of these subspace methods is given in Section 2.2. In order to apply these methods with color images, the basic image level fusion strategy is used which stack the color components to form an augmented color image vector.

Table 3.2: Method parameters.

Method	Parameter	PICS	GT	AR	FERET	FRGC-204
Gamma Correction	γ	0.9	0.9	1	0.2	0.5
DoG filitering	σ_0	0	0	0.4	0.4	0.8
	σ_1	0	0	0	-1	-3
Constrast equalization	α	0.1	0.1	0.1	0.1	0.1
	τ	1	2	10	3	-10
wPCA	Features	35	250	550	700	1000
rLDA	λ	0.5	0.5	5e-3	10	5
	Features	28	24	99	385	220
sLPP	Dist. Metric	cos	cos	cos	cos	cos
	PCA Dim.	35	250	550	700	1000
	w	0	0	0	1	100
	Features	18	35	200	335	220
IDA	λ	50	100	100	0.1	10
	Features	28	24	99	390	221
Feature Vector Similarity Metric		L2	L2	cos	cos	cos

Similarity score is calculated either by Euclidean distance or Cosine similarity depends on which measurement gives higher performance on gray-scale image. The score is then normalized using z-score (Jain *et al.*, 2005). Lastly, the similarity matrix is evaluated using the specific performance statistic described in Section 3.4.1.

3.4.4 Experimental Results

The experimental results are presented in Table 3.3. For PICS and GT, the average rank one identification rate r is reported along with the standard deviation s in the format of $r \pm s$. For AR, we report the rank one identification rate. For FERET and FRGC, the face verification rates (FVR) at 0.1% false accept rate (FAR) is reported. We have evaluated PLDCS and BWDCS with various square block size ($b \times b$), but in order to make it clearer, just the best ones are shown with the block size bracketed (b). Note that the color space for $b = 1$ is actually PLDCS. Results for various block sizes are shown separately in Table 3.4.

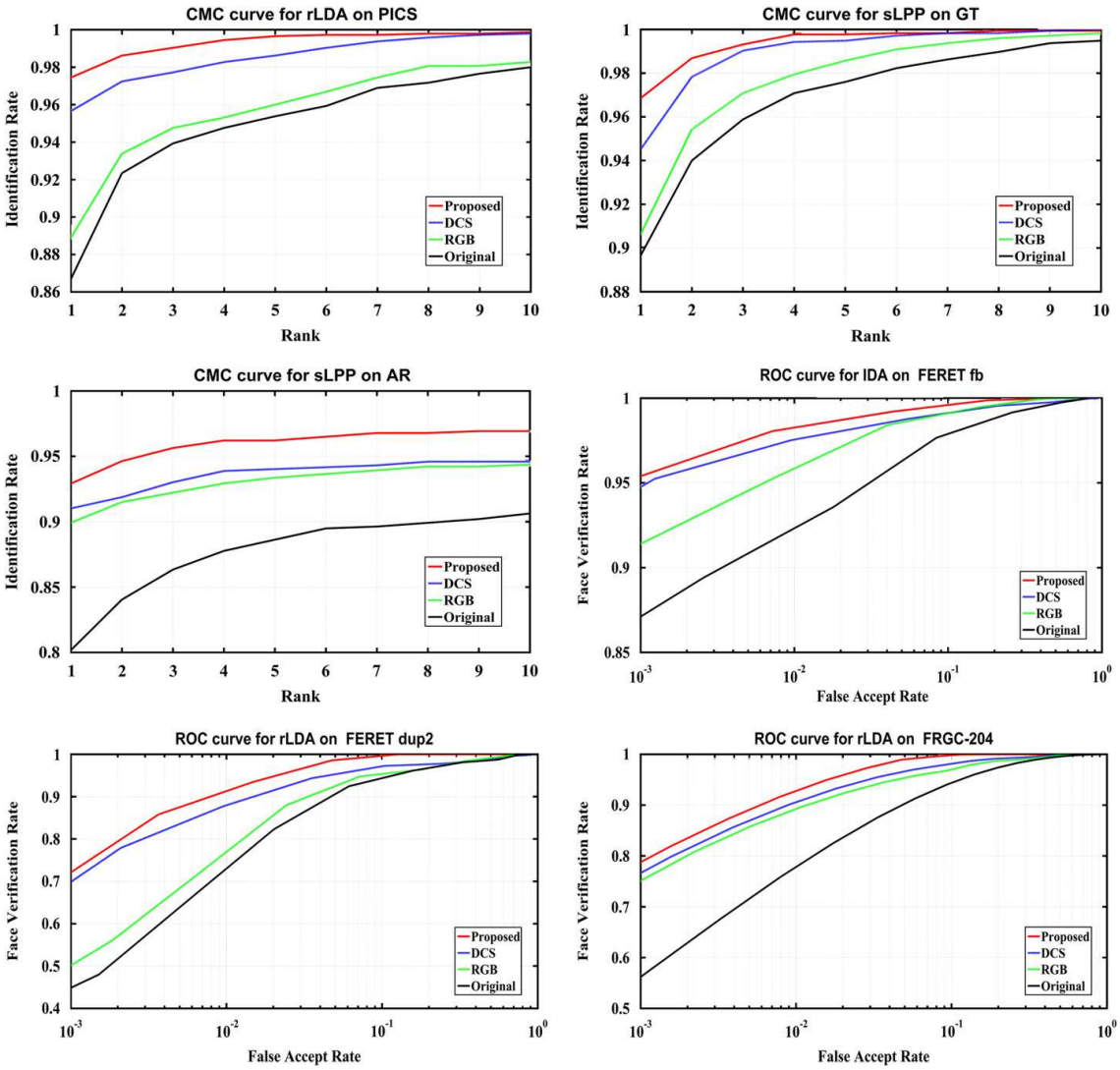


Figure 3.3: CMC/ROC curves for best performing method in each database.

All the methods considered were originally proposed and evaluated with the gray-scale

Table 3.3: Experiment results for gray-scale, RGB, DCS and the proposed BWDCS. For block size of BWDCS, only square block is considered. The optimal block size is in bracket.

	Original	*RGB*	*DCS*	*Proposed*
PICS database				
wPCA	71.8% ± 1.2	72.9% ± 1.3	83.4% ± 0.8	**90.8%** ± 0.6 (1)
rLDA	86.7% ± 0.7	88.9% ± 0.5	95.7% ± 0.4	**97.4%** ± 0.5 (1)
sLPP	77.7% ± 1.1	78.7% ± 1.2	86.9% ± 1.0	**93.8%** ± 1.0 (1)
IDA	86.3% ± 0.9	87.9% ± 0.5	95.4% ± 0.5	**97.0%** ± 0.5 (2)
GT database				
wPCA	84.4% ± 1.0	85.2% ± 1.0	90.1% ± 0.6	**95.0%** ± 0.5 (1)
rLDA	87.5% ± 1.0	90.6% ± 0.8	94.6% ± 0.5	**95.7%** ± 0.4 (1)
sLPP	89.7% ± 0.8	90.6% ± 0.7	94.5% ± 0.7	**96.9%** ± 0.6 (2)
IDA	91.6% ± 0.8	92.3% ± 0.7	95.0% ± 0.5	**96.1%** ± 0.3 (2)
AR database				
wPCA	50.6%	74.0%	74.5%	**77.0%** (6)
rLDA	80.5%	86.6%	87.6%	**90.3%** (2)
sLPP	80.3%	90.1%	91.6%	**93.0%** (1)
IDA	79.0%	80.4%	81.0%	**82.0%** (6)
FERET (fa-fb set)				
wPCA	85.1%	88.3%	90.1%	**93.1%** (4)
rLDA	90.5%	92.0%	93.9%	**94.7%** (4)
sLPP	85.5%	89.9%	93.7%	**95.3%** (2)
IDA	88.2%	91.5%	95.7%	**96.4%** (2)
FERET (fa-dup2 set)				
wPCA	35.2%	39.8%	59.6%	**66.2%** (2)
rLDA	42.9%	50.5%	71.5%	**73.7%** (2)
sLPP	37.1%	43.4%	59.8%	**66.8%** (8)
IDA	34.1%	40.7%	70.6%	**71.7%** (16)
FRGC v2 exp4				
wPCA	18.2%	20.2%	21.5%	**23.3%** (6)
rLDA	56.5%	75.7%	76.5%	**78.2%** (16)
sLPP	56.2%	73.0%	74.0%	**75.9%** (6)
IDA	37.9%	73.5%	74.4%	**75.5%** (16)

Table 3.4: Experiment results for BWDCS with various block size.

	1×1	2×2	4×4	6×6	8×8	16×16
PICS database						
wPCA	90.8%±0.6	90.6%±0.6	88.1%±1.0	86.9%±0.8	84.8%±0.8	84.8%±0.8
rLDA	97.4%±0.4	97.4%±0.4	97.0%±0.5	96.1%±0.4	96.4%±0.4	95.9%±0.4
sLPP	93.8%±0.9	92.7%±1.0	91.3%±0.8	89.7%±0.8	89.0%±0.8	89.6%±1.0
IDA	96.8%±0.5	97.0%±0.5	96.6%±0.5	95.8%±0.4	96.3%±0.4	95.8%±0.5
GT database						
wPCA	95.0%±0.5	95.0%±0.6	94.7%±0.6	93.4%±0.6	94.1%±0.7	91.4%±0.8
rLDA	95.7%±0.4	95.5%±0.4	95.2%±0.5	94.6%±0.5	95.6%±0.5	94.3%±0.6
sLPP	96.6%±0.6	96.9%±0.4	96.1%±0.4	95.5%±0.5	96.3%±0.4	95.3%±0.5
IDA	96.0%±0.3	96.1%±0.3	95.8%±0.4	95.1%±0.5	95.8%±0.5	94.9%±0.5
AR database						
wPCA	76.3%	76.3%	76.3%	77.0%	76.3%	75.7%
rLDA	90.1%	90.3%	90.1%	90.3%	90.1%	89.0%
sLPP	93.0%	92.4%	92.6%	92.3%	92.0%	92.7%
IDA	81.8%	81.4%	81.0%	82.0%	81.4%	80.3%
FERET (fa-fb set)						
wPCA	92.3%	93.0%	93.1%	92.0%	91.3%	92.5%
rLDA	94.4%	94.6%	94.7%	93.7%	94.1%	94.3%
sLPP	94.6%	95.3%	94.3%	94.9%	94.2%	93.7%
IDA	96.1%	96.4%	96.1%	95.5%	95.8%	95.8%
FERET (fa-dup2 set)						
wPCA	64.2%	66.2%	64.5%	58.5%	60.3%	59.7%
rLDA	72.9%	73.7%	72.5%	70.8%	73.5%	71.3%
sLPP	63.4%	62.2%	62.2%	58.9%	66.8%	60.4%
IDA	69.6%	67.8%	70.2%	66.7%	70.8%	71.7%
FRGC v2 exp4						
wPCA	22.5%	22.4%	22.5%	23.3%	22.2%	22.5%
rLDA	77.3%	77.7%	77.7%	77.7%	77.5%	78.2%
sLPP	75.7%	75.3%	75.6%	75.9%	75.2%	75.5%
IDA	74.3%	74.0%	74.4%	74.0%	74.6%	75.5%

images in their published paper. By integrating with RGB color space, the performance is increased by up to 36% (i.e. IDA on FRGC). The performance can be further increased by converting to DCS with a margin of up to 30% (i.e. IDA on FERET dup2). Our proposed color space further increases the performance in all cases, with about 7% improvement in the best case (i.e. wPCA and sLPP on both PICS and FERET dup2). While comparing with the original methods in gray-scale, our proposed color space boosts the performance by 17% on average. Moreover, the proposed color space is more stable since it has smaller standard deviation for the ten experiments repeated on PICS and GT.

Notice in Table 3.3 that, PLDCS sometimes performs better than BWDCS. This is because PLDCS optimizes the Fisher's criterion on each pixel where as only each block is optimized in BWDCS. Nevertheless, the fact that PLDCS operates on pixel reduces its robustness when there are large variations where a single pixel may not provide sufficient discriminative information. Therefore, one can see from the table that PLDCS performs the best on smaller datasets (i.e. PICS and GT) while larger blocks are usually required in more challenging datasets such as AR, FERET and FRGC to achieve the best performance.

From Table 3.4, one can see that the performance varies with different block sizes (about 2% on average). Similar to selecting parameters for other subspace methods, there is no scientific way to select the best block size in general situation. In real world application, these parameters are often determined by cross validation. However, the performance of even the worst block size is still comparable to DCS in our experiment.

The relative CMC/ROC curves for the best performing method in each database are presented in Figure 3.3. It is clear that for identification problem on PICS and GT, our proposed color space reaches the top rate at lower rank and outperforms DCS. For verification problems on FERET and FRGC, our proposed color space is on top of DCS over the entire curve. In all cases, our proposed color space is superior to gray-scale (in which these methods originally developed for).

3.5 An Investigation on Holistic and Local DCS

DCS makes use of holistic information while we generalize it to use different scales of local information. As a result, optimal projections can be obtained to suit each local area. In addition, the correlation between each color components can be further reduced which is shown to be an important factor for performance increment (Yang *et al.*, 2010b). Some color spaces are illustrated in Figure 3.4. For DCS, PLDCS and BWDCS, we scale the

pixel values of each color component to image domain [0,255]. One can see that PLDCS is visually very different to DCS. For example, the mouth area is forced to be in yellow color in DCS, while the local optimal color is actually near blue and green in PLDCS.

Notice in Figure 3.4 that the images for PLDCS and BWDCS is harder to recognize by human, but the computer recognition accuracy is increased with the transformation. Understanding of the human visual perception system would often inspire the development of computer vision algorithms as they are common in many aspect, however what is perceived as discriminative (i.e. the discriminant information or cue) by human visual system may not hold for machine vision. Human visual system is optimized for many more tasks such as navigation, obstacle avoidance, etc whereas the algorithms proposed in this thesis have been optimized to discriminate only human faces by transforming them to a more discriminant color space. This color space may not be so discriminant when it comes to other tasks.

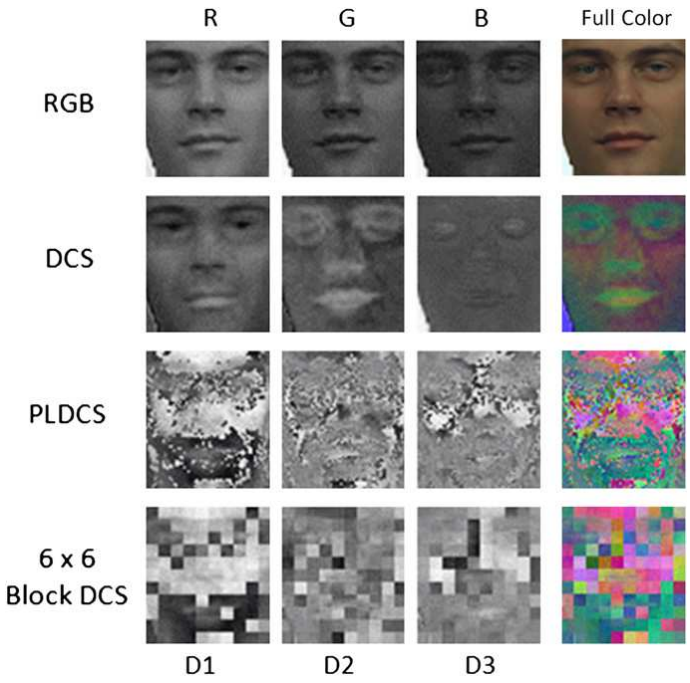


Figure 3.4: Illustration of R, G, B color component images and the three DCS component images generated by the proposed method.

The projection vector associated with the largest Eigen value on FERET training data is further investigated. Since, the image resolution is 32×32, there are 1024 PLDCS projection vectors. Angles between the DCS projection vector to each of these PLDCS vectors are calculated. The summary of these 1024 angle differences are shown using box plot in Figure 3.5 (left). All 1024 angle differences are then scaled from degree domain [0, 90] to image domain [0, 255] and shown as a gray-scale image in Figure 3.5 (right). It

is important to note that this figure is not a transform of a facial image. Black color (0) means minimum angle difference (0.40 degrees) while white color (255) means maximum angle difference (89.99 degrees). Each region on a face having similar color is also having similar projection vector, while the greatest difference is at around the lip, eyes and edges, thus making the image look like a face. DCS is based on Fisher Criterion which maximizes class separation. Nevertheless, according to our illustration, every pixel has its own optimal local projection vector and most of them (obtained by PLDCS) are different to the holistic projection vector obtained by DCS method, with a 41 degrees difference on average. The fact that DCS method uses one DCS projection for every pixel diminishes its performance.

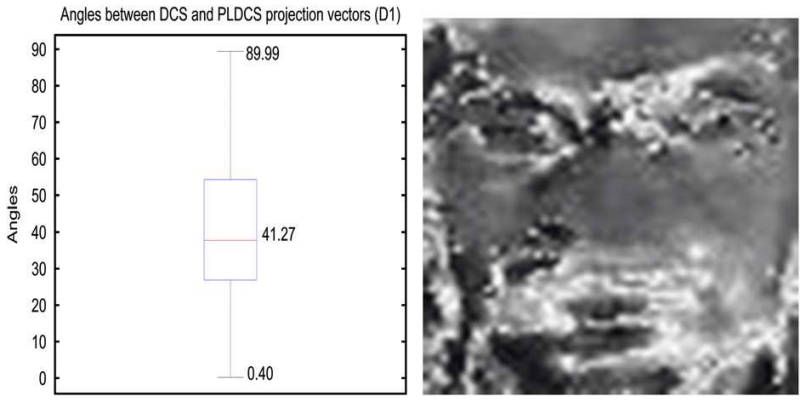


Figure 3.5: Angle differences between DCS and PLDCS projection vectors. Left: box plot for the differences. Right: gray-scale image obtained by scaling the differences to (0,255).

In a more theoretical perspective, Yang *et al.* (2010b) shows that color spaces that have Double Zero Sum (DZS) property is better for face recognition. As explained in Section 2.5.2, a color space is DZS if two rows of its transformation matrix have zero sums. DZS is important because color spaces that are DZS have lower correlation between their color components. For example, the three color components of RGB is highly correlated and therefore is not a good color space for recognition. DCS is a near DZS color space hence it has low inter-component correlation and high discrimination power. PLDCS has DZS property for each single pixels, which further reduces the inter-component correlations. Table 3.5 shows the average inter-component correlation for the training images in FRGC-204 and FERET. Given K color images $A_i = [A_i^1, A_i^2, A_i^3](i = 1, \cdots, K)$, where $A^1A^2A^3$ are three color components. The correlation matrix is calculated first from each individual image as:

$$\rho_i^{x,y} = \frac{E[(A_i^x - E(A_i^x)(A_i^y - E(A_i^y)^T]}{\sigma_i^x \sigma_i^y}, \quad (x, y = 1, 2, 3, x \neq y). \tag{3.11}$$

Since $\rho_i^{x,y} = \rho_i^{y,x}$ and the sign does not affect the correlation magnitude, the average

correlation R is calculated by averaging across all images and taking absolute value, i.e.

$$R = \frac{1}{K}\sum_{i=1}^{K}(\frac{|\rho_i^{1,2}| + |\rho_i^{1,3}| + |\rho_i^{2,2}|}{3}). \tag{3.12}$$

R is ranged from zero to one. Zero means no correlation at all while one means perfectly correlated. We can see from Table 3.5 that image in PLDCS has much lower correlation.

Table 3.5: Average Inter-component Correlation.

	FRGC-204	FERET
RGB	0.9283	0.9332
DCS	0.2992	0.3795
PLDCS	0.1669	0.0828

To conclude, different locations of a face have different optimal color space for classification. The improvement achieved by PLDCS or BWDCS is mostly due to the reason that it captures the optimal space for each pixel or block, and it further reduces correlation between color components.

3.6 Summary

We have made the following contributions in this chapter. We propose two new color spaces for face recognition. We incorporate some promising subspace algorithms that are originally proposed and evaluated on gray images with our new color spaces. An effective preprocessing pipeline proposed recently is integrated in our recognition framework to work along with color images. We designed repeatable experimental setups that covered many aspects of face recognition problems including seen and unseen identification as well as unseen and partially-seen verification. By improving DCS, two new color spaces are proposed for color face recognition namely PLDCS and BWDCS. Experimental results show that, for all subspace methods considered, the performances are improved significantly in the following order: gray-scale, RGB, DCS and our proposed PLDCS/BWDCS. The improvement is due to the fact that local DCS allow locally optimal color space which is important for face recognition. Locally derived DCS also has lower inter-component correlation which is one of the factor for the improved performance. Since many recently proposed methods are still being developed using gray-scale images, the state of the art can be advanced by using the color spaces proposed in this work.

A shortcoming can be observed on our proposed color spaces. Both PLDCS and BWDCS

are derived from RGB linearly and consist of three color components only. In fact, complementary information can be found on more than three color components across multiple linear and non-linear color spaces. This problem will be further investigated and a novel color model for face recognition is proposed in Chapter 4.

Chapter 4

Hybrid Color Model

Color has been proven to have the capability of increasing face recognition accuracy. As stated multiple times in this thesis, main research effort on color face recognition has been devoted to constructing a more discriminative color space. Starting from transformed color models to statistically learned color models, all have shown improving performance with different scales. In summary, the Color FisherFace (CFF) model (Thomas *et al.*, 2008) concatenates the R, G and B components to one vector in image level and then applies LDA on the augmented vector. Shih and Liu (2006) proposed a hybrid color space YQCr which outperforms RGB. Yang *et al.* (2010a) proposed a color space normalization technique which can enhance the discriminative power of color spaces. Their experiments show that the normalized RGB (nRGB) and normalized XYZ (nXYZ) have achieved significant performance improvement. Liu (2008) derived three statistical color spaces namely the Uncorrelated Color Space (UCS), Discriminant Color Space (DCS) and Independent Color Space (ICS). UCS aims to de-correlate RGB based on PCA. DCS aims to learn a discriminant space using the class label based on LDA. ICS aims to separate RGB into independent source images based on Independent Component Analysis (ICA). Yang and Liu (2008a) developed the extended General Color Image Discriminant (eGCID) model to iteratively optimize both the color transformation and feature extraction specific to LDA. Jing *et al.* (2010) proposed a color model to extract uncorrelated features from each of the RGB component images separately based on Holistic Orthogonal Analysis (HOA). HOA features are combined in feature level for face recognition. Some of these methods, which are more representative, are introduced in detail in Section 2.5.

We distinguish color models from color spaces based on whether the method poses restriction on recognition procedure or not. The CFF, eGCID and HOA are regarded as color models. The eGCID involves LDA feature extraction in constructing the color space, while the CFF and HOA restricts to LDA when deriving color features. Other methods are color spaces since their proposed algorithms only generate a new color space and it is up to the user to choose the recognition methods (e.g. what feature to use, when to fuse the colors, etc.), although the underlying recognition algorithm used in their experiments is still based on LDA.

As we can see, most of the state-of-the-art color face recognition methods use at most three color components transformed only linearly from RGB. However, using only three color components may not capture sufficient information to solve complex face recognition problems. In this chapter, we propose a novel color model namely Multiple Color Fusion (MCF) model to utilize more than three color components across both linear and non-linear color spaces. After reducing the number of available colors by correlation analysis, a greedy search based algorithm is developed to find the optimal color combination. Decision level fusion is designed to overcome the high dimensionality problem as well as to utilize most of the information from different colors. The effectiveness of MCF is evaluated on FRGC v2 experiment 4 (Phillips *et al.*, 2005) and AR database (Martinez and Benavente, 1998), for large-scale face verification and identification problems. Experimental results show that the proposed MCF model outperforms all the state-of-the-art color face recognition algorithms considered.

The rest of the chapter is organized as follows. Section 4.1 explains the MCF model and presents the corresponding algorithms. Experimental results and time complexity are reported next in Section 4.2. Section 4.3 demonstrates that different colors provide quite different information for face recognition, justifying the motivation of MCF model and a summary is given in Section 4.4.

4.1 Multiple Color Fusion model

The proposed method includes four steps. First, for highly correlated color components, we keep one and remove the others. Then, a simple greedy search method is applied to search for the optimal color combination set. Next, for each individual color, we extract features and compute the similarity scores. And finally, we fuse the similarity scores from all colors. In this section, we detail each step of the proposed algorithm with justification.

4.1.1 Stage 1: Correlation

To develop the MCF model, we use the following thirteen color spaces which involve 39 color components. Most of them are introduced in Section 2.5:

- RGB (fundamental color space)
- $I_1I_2I_3$ ((decorrelated RGB)
- YIQ, YUV and YCbCr (NTSC video transmission color spaces)

- XYZ, L*a*b*, L*u*v* and L*c*h* (CIE uniform color spaces)
- HSL (human perceptual color space)
- DCS (statistic learning color space)
- nRGB and nXYZ (normalized RGB color spaces)

For these 39 color components, some are identical, for example the Y components from YIQ, YUV and YCbCr. After removing duplicated color components, a color correlation matrix is computed using the training samples. For highly correlated colors (> 0.99), only one of them is kept and the others are removed.

4.1.2 Stage 2: Greedy Search

Assumes that $n(< 39)$ color component candidates are left for selection after first stage. We design a greedy search approach to find the best color combination, which is detailed as follows.

Our selection criterion is based on recognition rate. Let $A = [A_1, A_2, \cdots, A_N]$ be N training samples. First, we have to divide the training data into two subsets, i.e. the learning set $L = [L_1, L_2, \cdots, L_{N_l}]$ and validation set $V = [V_1, V_2, \cdots, V_{N_v}]$, where $A = L \cup V$ and $N = N_l + N_v$. The learning set is used for color-feature extraction, while the validation set is used for color-performance computation. This division can be done and repeated in a similar fashion to the cross validation technique for improving the generalization capability of the resulting chosen optimal color set.

Next, assumes that each sample image from the two subsets consists of d pixels and n color component candidates, which can be denoted by

$$\begin{aligned} L_i &= [l_i^1, l_i^2, \cdots, l_i^n] \in \mathbb{R}^{d \times n} \quad (i = 1, 2, \cdots, N_l), \\ V_j &= [v_j^1, v_j^2, \cdots, v_j^n] \in \mathbb{R}^{d \times n} \quad (j = 1, 2, \cdots, N_v), \end{aligned} \tag{4.1}$$

where $l_i^k \in \mathbb{R}^d$ and $v_j^k \in \mathbb{R}^d$ (for $k = 1, 2, \cdots, n$) denote the k-th color component candidate of the i-th image in learning set and the j-th image in the validation set, respectively. They are both organized as d-dimensional column vectors. In order to extract LDA feature, the projection matrices W^k for each color candidate is first learned from the learning set. It can be computed directly with respect to each color components of the learning samples, i.e.

$$W^k = LDA(L^k), \tag{4.2}$$

where $L^k = [l_1^k, l_2^k, \cdots, l_{N_l}^k] \in \mathbb{R}^{d \times N_l}$ is a matrix containing all learning samples with the k-th color component and $LDA : \mathbb{R}^{d \times N_l} \to \mathbb{R}^{d \times \widetilde{d}}$ is the function that computes the

FisherFace projection matrix (see Eq. 2.10). In a similar fashion to L^k, we can define $V^k \in \mathbb{R}^{d \times N_v}$ as a matrix containing all validation samples with the k-th color component. Then. the following projections can be performed

$$\begin{aligned} \widetilde{L^k} &= W_k^T L^k, \\ \widetilde{V^k} &= W_k^T V^k. \end{aligned} \tag{4.3}$$

In order to compute the recognition rate for each color candidate, we first compute their decision scores, then the Nearest Neighbor (NN) classifier is used to classify each sample in validation set to the learning set, color by color. As described in Section 2.2.4, the decision score can be computed by different distance metrics such as Euclidean or Cosine distance. Let this distance function be $dist : \mathbb{R}^{\widetilde{d}} \times \mathbb{R}^{\widetilde{d}} \to \mathbb{R}$, then the decision score matrix $D^k \in \mathbb{R}^{N_l \times N_v}$ for the k-th color candidate can be computed as

$$D^k_{(i,j)} = dist(l^k_i, v^k_j), \tag{4.4}$$

where $D^k_{(i,j)}$ indexes to (i, j) entry of the matrix D^k. An array of correctness denoted by $r^k \in \mathbb{R}^{N_v}$ can be computed as

$$r^k_j = \begin{cases} 1, & \text{if} \quad \text{label}(l^k_m) = \text{label}(v^k_j) \\ 0, & \text{otherwise} \end{cases}, \tag{4.5}$$

where $m = \underset{i}{argmin}(D^k_{(i,j)})$ is the index of the nearest neighbor of v^k_j. The performance for the k-th color candidate denoted by p^k_1 is then computed as

$$p^k_1 = \frac{\sum_j^{N_v}(r^k_j)}{N_v} \tag{4.6}$$

Let k_1 be the index of the first chosen color candidate. It is set as follows: $k_1 = \underset{k}{argmax}\, p^k$. The index of second color k_2 is chosen if it results in the highest performance after fusing with the first color in decision score level based on sum rule (Ross and Jain, 2003), i.e. $D^{k_1}_{(i,j)} + D^{k_2}_{(i,j)}$, and so on. The index of c-th color k_c is chosen if the fused decision score matrix, i.e.

$$D^{k_1}_{(i,j)} + D^{k_2}_{(i,j)} + \cdots + D^{k_c}_{(i,j)} \tag{4.7}$$

results in the highest performance. Note that the z-score normalization (Jain *et al.*, 2005) is applied before fusion to balance the score distribution. This normalization minus the mean and divides the standard deviation of the decision scores for each validation sample. It is omitted from the formulas above for sake of simplicity. Finally, the selection procedures stop at the $(C + 1)$-th iteration when the performance decreases, i.e. $p^{C+1} < p^C$. As a result, C colors are chosen to be the optimal color set. Since the same color is not allowed to be fused twice, the computation complexity is $n(n + 1)/2$.

4.1.3 Remarks

The MCF model described above is used in our experiment in this work, however it can be applied for more general cases. For example, the original color space candidates can be chosen according to domain knowledge and the threshold for defining high correlation can be set differently. Also, the feature extractor, similarity distance metric, score normalization technique, classifiers and performance statistic can be changed to preferred choices. In real-world applications, these parameters are usually tuned by performing k-fold cross validation.

Here we give some justification for each step of our algorithm. We use correlation as a selection criterion at the first step because highly correlated color components encode very similar information and noise. Fusing two highly correlated color components has limited gain on benefit whereas noise will be emphasized. The low cost correlation based selection criterion we used at first stage not only eliminates this problem, but also speed up the subsequent process.

For the combination search in step two, it is a NP-hard problem naturally. An exhaustive search for all possible combinations has $O(2^n)$ complexity. Therefore a greedy search approach is desired for simplicity and reliability. Although global optimal solution is not guaranteed, sub-optimal solution can always be derived, which is usually sufficient to improve recognition performance. More importantly, as the same color is not allowed to be fused twice, the complexity of our proposed searching algorithm is only $O(n(n+1)/2)$.

In terms of information fusion, the approach we adopted is at decision level. This approach not only overcomes the high dimensionality problem, but also utilizes most of the information from different colors.

4.1.4 Algorithms

Algorithm 4.1 summarizes the proposed MCF model. Algorithm 4.2, which is also used in our experiment, presents a possible application to face verification problem. Algorithm 4.2 can be modified easily for identification problem, by replacing step 9 to 13 by the NN classifier and return the identity of the matched image.

Algorithm 4.1 MCF - Training

Require: A, RGB training samples

Ensure: -

$\{T_{index}\}$, the chosen color transformations

$\{F_i\}$, the feature extractor for each chosen colors

$T() \leftarrow$ Construct multiple color space transformer.

$A \leftarrow T(A)$: Convert A to multiple color spaces.

$\{train\ ,\ test\} \leftarrow$ Split $\{A\}$ into two halves.

for all individual color i in $\{train\}$ **do**

  $F_i() \leftarrow$ Construct the feature extractor.

  $B_i \leftarrow F_i(test)$: Extract features.

end for

$index \leftarrow \{\}$

$d \leftarrow 1$

repeat

  $chosen = \{\ B_{index}\ \}$

  $remain = \{B\} - \{B_{index}\}$

  for all individual color i in $remain$ **do**

    Fuses $chosen$ and $remain_i$ in decision score level

    Evaluate the performance of the fused scores

  end for

  Add to $index$ the best performing color i after fusion

  $d \leftarrow$ the performance difference after fuses color i

until (no more remaining color) OR $(d < 0)$

$A \leftarrow T_{index}(A)$: Transform A to the chosen colors

for all individual color i in $\{A\}$ **do**

  $F_i() \leftarrow$ Construct the feature extractor.

end for

return $\{T_{index}\}$ and $\{F_i\}$

Algorithm 4.2 MCF - Verification

Require: -

$\{F_i\}$, Feature extractor for each chosen colors

g, RGB a Gallery Images

x, a RGB Query Image

th, Acceptance threshold

$T()$, The chosen color transformers

Ensure: Accept or reject that g and x is the same person

1: $g \leftarrow T(g)$: Transform g to the chosen multiple colors
2: $x \leftarrow T(x)$: Transform x to the chosen multiple colors
3: **for all** individual color i **do**
4: $H_i \leftarrow F_i(G)$: Extract features.
5: $y_i \leftarrow F_i(x)$: Extract features.
6: $S_i \leftarrow$ Compute decision scores for the pair (H_i, y_i)
7: **end for**
8: Normalize and fuse all scores S_i
9: **if** The fused score $> th$ **then**
10: **return** ACCEPT
11: **else**
12: **return** REJECT
13: **end if**

4.2 Experiments

In our experiments, both face identification and verification problems are evaluated. For identification problems, the test images are matched to all of the training images to find the match. For verification problems, we consider accepting or rejecting the probe against the claimed gallery image. In this section, we evaluate the proposed MCF under both applications.

4.2.1 Face Verification

The face verification problem is evaluated using the Face Recognition Grand Challenge version 2 (FRGC2) (Phillips *et al.*, 2005). FRGC2 is a publicly available, large-scale database, widely used to test face verification algorithms. According to FRGC2 standard experiments setting, experiment 4 is the most challenging test containing 12776 training images, 16028 controlled target images and 8014 uncontrolled query images. Three Receiver Operating Characteristic (ROC) curves (ROC-I, ROC-II and ROC-III) are used by FRGC2 as the performance statistics. We considered the most difficult ROC-III which only considers target/query pairs that the query is taken in a year later than the target. All the face images are cropped and resized to 32×32 with eyes and mouths aligned to the same position using the coordinates provided by FRGC2. Some sample images are shown in Figure 4.1. The experiment setup and evaluation protocol follow exactly the FRGC2 standard to ensure fair and comparable experiment results. The face verification rate (FVR) at 0.001 false accept rate (FAR) on the Receiver operating characteristic (ROC) curve is reported.

Figure 4.1: Sample images from FRGC2.

The MCF model is applied on the database as described in Section 4.1. In stage one, it removes 20 of the 39 colors which are either identical or highly correlated to the remanning. The remaining 19 colors are RGB, YIQ, nG nB (the G and B competents in nRGB), nY (the Y component in nXYZ), $D_1D_2D_3$ (the three components in DCS), HS (from HSL), a*b* (from L*a*b*), v* (from L*u*v*) and c*h* (from L*c*h*). In stage two, 12 colors are finally chosen as the optimal color set in this order: Y, nY, a*, D_2, I, R, nG, nB, G, H, B and D_1. This 12-color MCF model is compared to 9 state-of-the-art 3-color spaces/models

as introduced previously: CFF , YQCr , nRGB , nXYZ , UCS , DCS, ICS, eGCID and HOA . The same feature extraction procedure for each space/model is implemented as in their published references. In particular, after converting to the specific color space, the dimension is reduced using PCA to 1000 and then 220 LDA features are extracted. Image level fusion is used for CFF, YQCr, RGB-NI, nXYZ, UCS, DCS, ICS and CID, feature level fusion is used for HOA and decision level fusion is used by the proposed MCF model. Before image level fusion, each color component image is normalized to zero mean and unit standard deviation in order to avoid the negative effect of magnitude dominance of one component over the others.

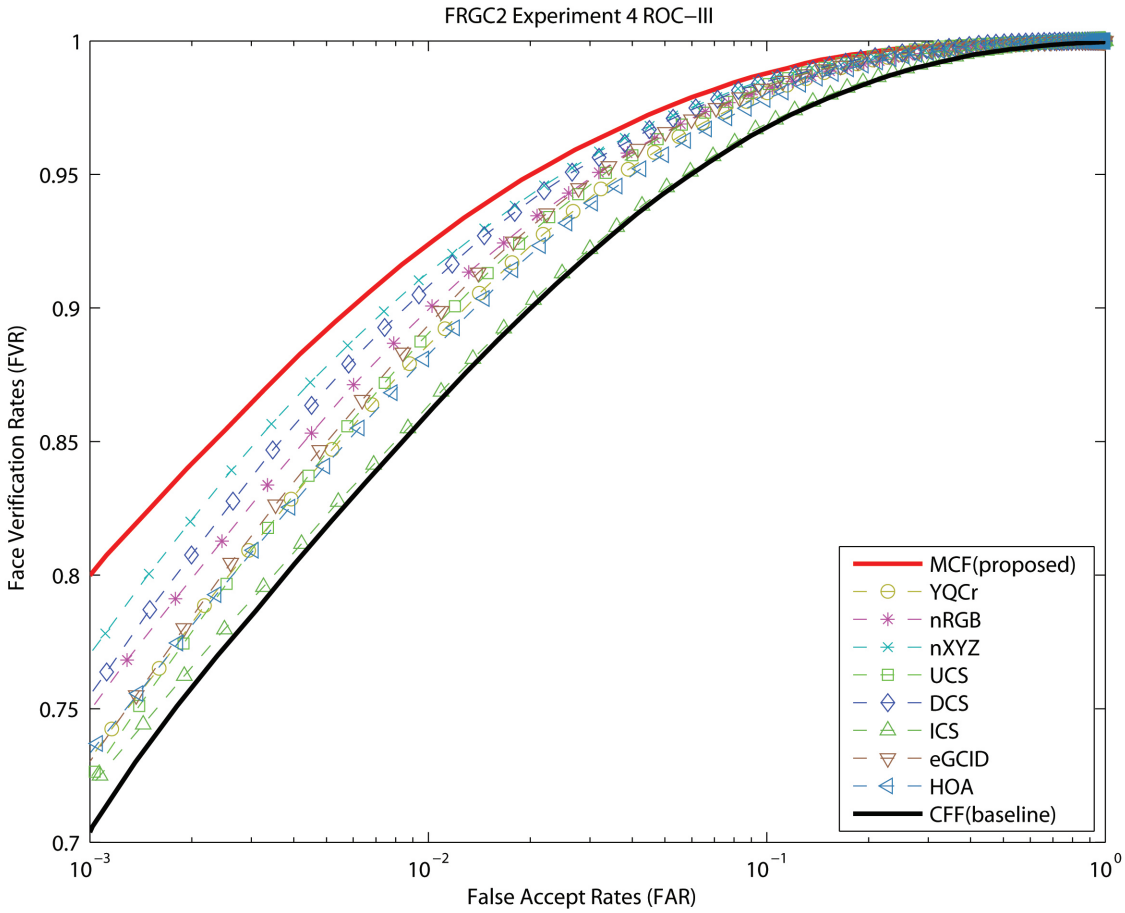


Figure 4.2: The ROC-III curve for each method compared on FRGC2 experiment 4.

Figure 4.2 shows the ROC-III curve for different methods compared on FRGC2 experiment 4. The CFF model applying LDA directly on the image-level-fused RGB image serves as a good benchmark baseline for color face recognition. It is clear to see that the proposed MCF model outperforms all other methods in comparison. Table 4.1 further reports the corresponding Face Verification Rates (FVR) at 0.1% False Accept Rate (FAR). The proposed model achieves 80%, outperforming the second best method nXYZ by 3%. In

Table 4.1: The FVR (@FAR=0.1%) for different methods compared on FRGC2 experiment 4 ROC-III.

Methods	**FVR (%) @ FAR = 0.1%**
MCF(proposed)	*80.01*
YQCr	73.01
nRGB	74.86
nXYZ	75.97
UCS	72.34
DCS	75.43
ICS	72.01
eGCID	73.01
HOA	73.29
CFF(baseline)	71.96

Table 4.2: The FVR (@FAR=0.1%) for 10 random 12-color combinations

12-Color Combination	**FVR (%) @ FAR = 0.1%**
MCF(proposed)	*80.01*
1	76.56
2	75.11
3	75.95
4	76.86
5	76.74
6	72.69
7	74.98
8	73.70
9	75.73
10	75.70

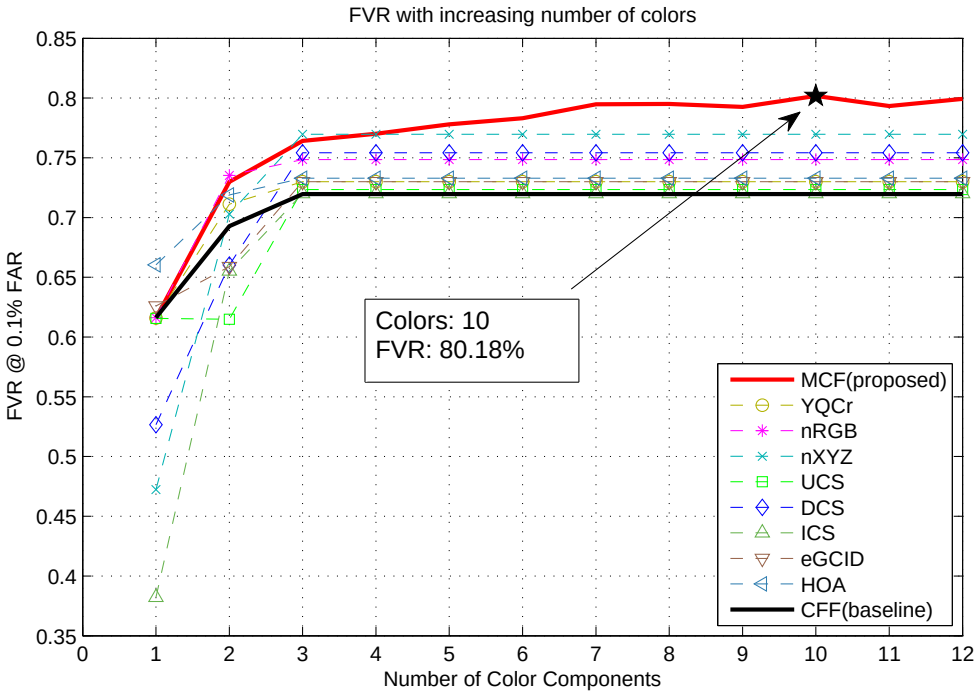


Figure 4.3: The FVR @ 0.1% FAR with increasing number of colors.

addition, MCF improves the CFF baseline method significantly by a margin of 8%.

To show the color selection effectiveness of MCF, we further compare the performance of the 12-color combination selected by MCF with 10 other randomly selected 12-color combinations in Table 4.2. The 10 randomly selected color combinations perform slightly better than all 3-color model/space reported in Table 4.1 on average, but with unstable performance. The best performing color combination is still the one selected by MCF.

Figure 4.3 shows the FVR with increasing number of colors. When using only 3 colors, nXYZ perform slightly better than MCF. This is due to the local maximum problem associated with the greedy search algorithm. However, the performance can still be improved with more than three colors. The maximum possible FVR is 80.18% when using 10 colors, but the optimal number found by MCF searching is 12 based on the training images. This is due to the generalization problem that any supervised algorithm suffers when training size is limited. This problem can be reduced by acquiring better training samples (represent the testing samples better) as well as by applying k-fold cross validation. Nonetheless, the performance different between 10 and 12 colors is negligible (just 0.17%).

Notice that the proposed MCF outperforms the Block-wise Discriminant Color Space

(BWDCS) proposed in Chapter 3. Since the experiment protocol we adopted in this section is the FRGC standard, therefore results are directly comparable to Table 3.3. The best performance for FRGC in Table 3.3 is 78.2% achieved by the BWDCS with a 16×16 block size, utilizing regularized LDA (rLDA) for feature extraction. Whereas the proposed MCF achieves 80.01% in Table 4.1 using the basic LDA algorithm for feature extraction. Higher performance can be expected when MCF is applied on image patches (i.e. block-wise MCF) with the use of advanced face recognition features. However, since there are large number of such feature proposed for face recognition, and some of them are more suitable in certain situations or proposed for tackling a specific challenge, therefore it is impossible to experiment on every combination. The main purpose of this chapter is to show that face recognition performance can be increased in general when effectively utilizing multiple color components across color spaces. We achieve this objective by comparing face recognition performance in this section with other holistic color face recognition methods that use LDA feature, and therefore color space become the only changing factor.

4.2.2 Face Identification

AR face database (AR) (Martinez and Benavente, 1998) is a publicly available face database. It is widely used for evaluating face identification problems. This database contains over 4000 images which is captured in two sessions with different facial expressions, illumination conditions and occlusions. We use a subset of all the un-occluded images from the first 50 males and 50 females. As a result, a total of 1400 images from two sessions (14 images per subject) are included in our experiment. All the face images are cropped and resized to 32×32 with eyes and mouths aligned to the same position manually. Some sample images are shown in Figure 4.4. Following Yang *et al.* (2010a) and Yang *et al.* (2010b), we formulate a time-delayed face identification problem using images from session one for training and images from session two for testing. The rank-one identification rate on the Cumulative Match Characteristics (CMC) curve is reported.

Figure 4.4: Sample images from AR.

All methods with same parameters setting as in previous section are evaluated on AR, except the dimension for PCA and LDA are set to 650 and 99 respectively. The best color

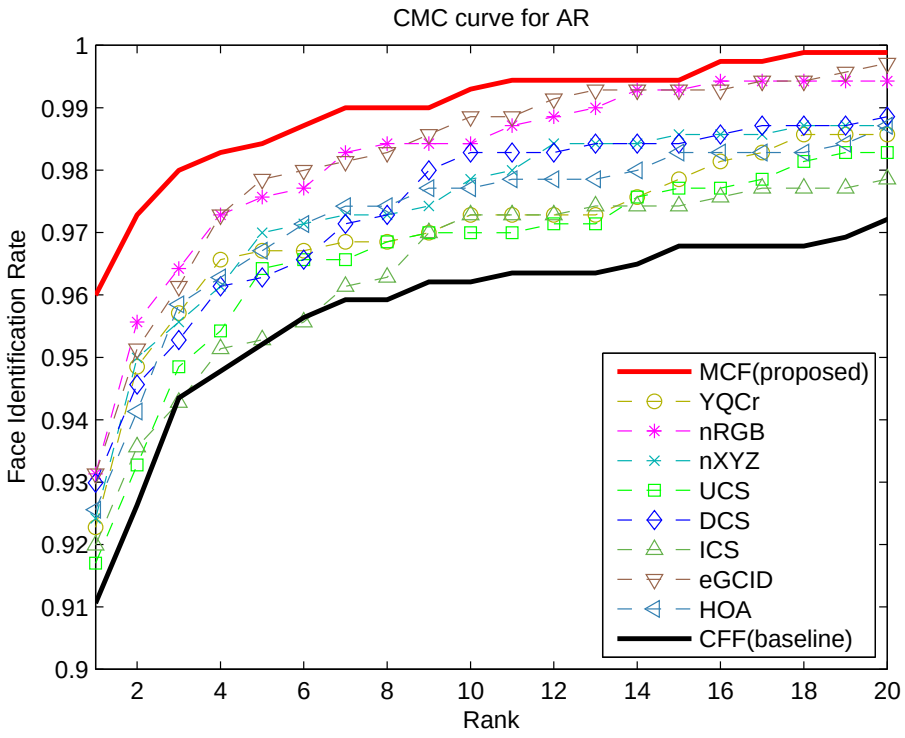


Figure 4.5: The CMC curve for each method evaluated on AR.

Table 4.3: The rank one identification rate for each method evaluated on AR.

Methods	**Identification Rate (%)**
MCF(proposed)	*96.00*
YQCr	92.27
nRGB	93.13
nXYZ	92.42
UCS	91.70
DCS	92.99
ICS	91.99
eGCID	93.13
HOA	92.56
CFF(baseline)	91.00

Table 4.4: Identification rate (%) for 10 random 8-color combinations

8-Color Combination	**Identification Rate (%)**
MCF(proposed)	*96.00*
1	94.56
2	93.71
3	93.71
4	93.28
5	92.99
6	93.42
7	92.13
8	89.13
9	93.28
10	94.99

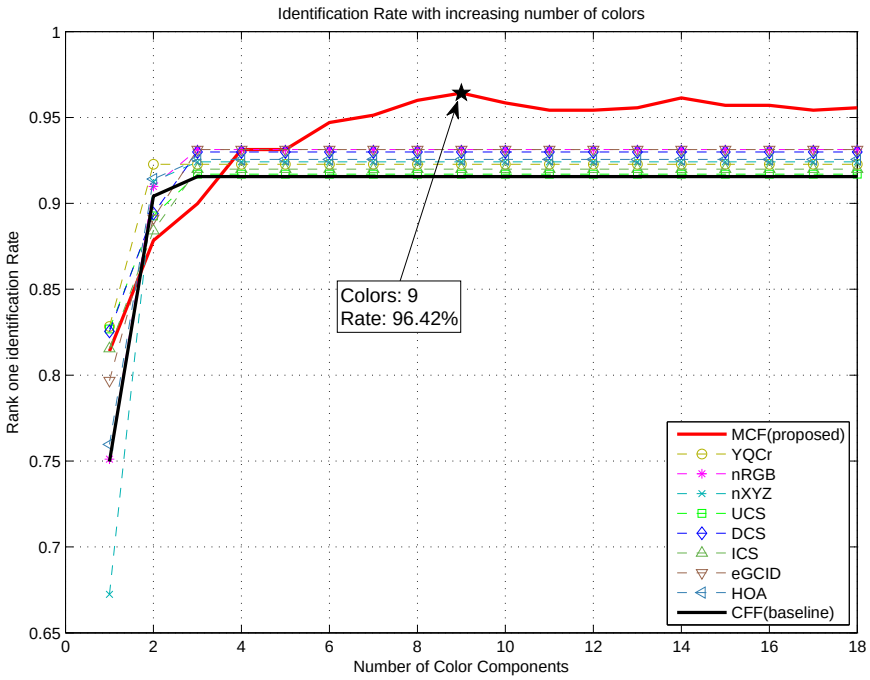


Figure 4.6: The Face Identification Rate with increasing number of colors.

combination found by MCF consists of 8 colors: B (from RGB), b* (from L*a*b*), nY (from nXYZ), D1D2 (from DCS), YI (from YIQ) and nB (from nRGB). The corresponding CMC curve is ploted in Figure 4.5 with the rank one identification rates reported in Table 4.3. Similar result trend is observed. MCF model achieves 96%, outperforming the second best method eGCID by nearly 3% and improves the CFF baseline method by 5%. Since performance are already over 90%, a further 3 to 5 % difference can be considered significant.

Next we evaluate color selection effectiveness of MCF as same as on FRGC in previous section. 10 randomly generated 8-color combinations are compared against MCF in Table 4.4. Although the highest performance (i.e. the 10^th combination) is nearly 95%, which is comparable to MCF, the performance can not be guaranteed due to its random generation. MCF does provide a systematic way to search for the optimal color combination.

From Figure 4.6, we can observe the same generalization problem for MCF. The optimal color should include 9 color componenets but MCF can only pick up 8. The performance difference is however negligible.

4.2.3 Time complexity

Although the proposed MCF algorithm is based on combinatorial search, time complexity is not a problem. According to Algorithm 4.1, most of the steps in training stage are linear except for the nested for-loop when searching for color combinations. For n colors, it can take up to $n(n+1)/2$ steps. The most computational intensive step is the re-computation of the decision scores in each iteration, however if there are enough memory to pre-compute all the decision scores for each color and keep them in memory, then this step can be carried out in real time.

As shown in Table 4.5, only around 6 seconds is required for training in AR. Due to the large scale of FRGC, the time needed becomes much longer, but is still acceptable. Further, training time is actually not so important as it is an off-line process and only required to be done once. Note that the training time in Table 4.5 is the total time spent for the overall training process involving 12776 training images for FRGC and 700 images for AR.

For testing, since the color transformation matrix and LDA projection matrix have been constructed, the color transformation and feature extraction for a single image is very quick. Since the testing time shown in Table 4.5 is the total time needed to answer more

than 32 millions gallery/probe pairs of query for FRGC and 700 queries for AR. Therefore, on average answering a single query can be completed in real time.

Table 4.5: Total training and testing time in seconds

Training	FRGC2	AR
Color Transformation	198.3	23.5
Color Selection	2071.1	6.3
TOTAL	2269.4	29.8
Testing	FRGC2	AR
Color Transformation	39.7	3.2
Evaluation	151.6	14.3
TOTAL	191.3	17.5

4.3 The usefulness of multiple color components

Face recognition is a complex pattern recognition problem, such that there is no single feature that can capture all necessary information to solve it well. Research shows that fusing different biometric models (e.g. finger print/ palmprint + face) (Yao *et al.*, 2007), different features, different scales of the same feature (Liu and Liu, 2010) or different LDA based methods (Zuo *et al.*, 2007) can also increase recognition performance significantly. Evidenced from these findings, we expect the same property holds for fusing different color models. Human faces display different colors at different locations. For example, different eyes may be easier to classify in a specific color space, while lips may be easier to discriminate in another color space. In fact, different color representations encode very different information from a human face and thus complementary features can be extracted from them. To illustrate this idea further, we introduce the variance face (VF). Given a random face variable X and its expected value μ, the population variance is computed as:

$$\sigma^2 = E[(X - \mu)^2]. \tag{4.8}$$

Given N face image samples each arranged as a d dimension column vector in A (i.e. $A \in \mathbb{R}^{d \times N}$ is a collection of these N samples). If N is sufficiently large, μ in Eq. 4.8 can be approximated with the sample mean $\bar{A} = \frac{1}{N} \sum A_i$ and VF can be computed from A as:

$$VF = \sum_{i}^{N} (A_i - \bar{A})^2 \tag{4.9}$$

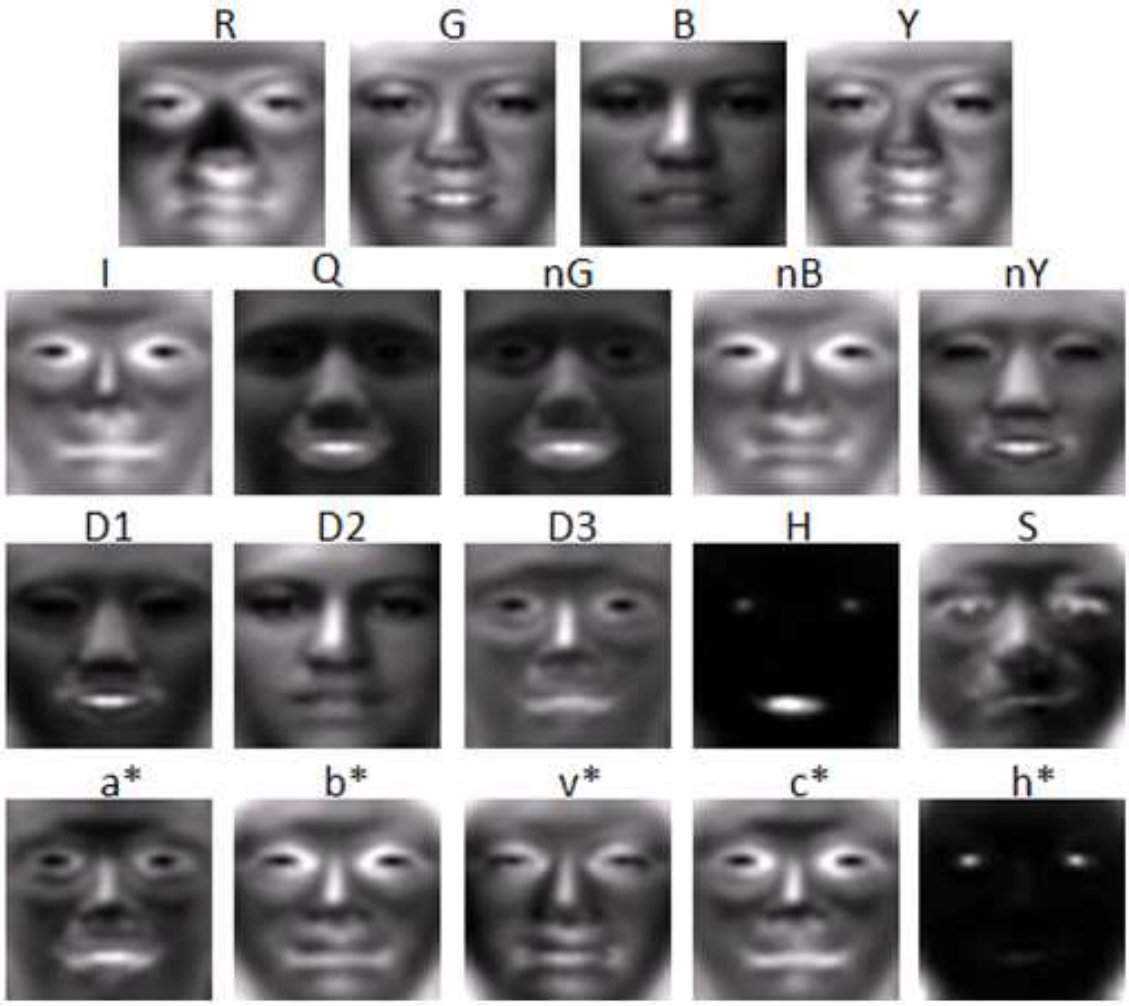


Figure 4.7: Variance face: variance of each pixel is computed and mapped to a 256-intensity image. White color represents higher variance while black color represents lower variance.

Figure 4.7 shows the VF computed from different colors using all the 12776 training samples in FRGC2 with Eq. 4.9. For visualization, the VF is scaled and displayed as an 256-intensity image. Therefore, black color means low variance while white color means high variance. Note that these VFs are not real face images but are synthetic face analogous to mean face image. The fact that some of these images look like a face demonstrates that the variance is distributed according to regions on a human face. These VFs show that different colors encode different complementary information. For example, the R component has higher variance in the area between nose and lips, the G component has higher variance for lips and the B component encodes higher variance for nose. Although R, G and B are highly correlated, their variance distribution is complementary which explains why fusing them can still offer performance improvement over single intensity image. It is also interesting to point out that most of the colors encode high variance for mouth area, which is caused by different facial expression. This finding may facilitate facial expression recognition. In addition, most of the colors encode low variance for eyeball (maybe due to low resolution) except for color H and h^* which encode very different information from other colors.

Since most of the color components used by the proposed MCF model can be linearly transformed from RGB, therefore the RGB model may already contain all necessary information. Nevertheless, RGB is not a good color model for face recognition. We believe that this is because most of the information is hidden in RGB space. Our experiment results show that, effective combination of different color components can increase recognition performance significantly. The purpose of using hybrid color model is not to produce new information but to reveal existing one in a better form for face recognition. In terms of information fusion between color components, although a simple voting algorithm may work, MCF utilizes the advantage of Linear Discriminant Analysis and fuses the colors in image level to deliver a robust yet low complexity solution. In this chapter, the major conclusion to be drawn is that effective utilization of multiple color models outperforms any single model. The way to better utilize these models may deserve future investigation.

4.4 Summary

In this chapter, the following contributions have been made. We proposed a novel color face recognition algorithm namely the MCF model, using more than three colors across both linear and non-linear color spaces. It starts with a number of color spaces and (after removing identical and highly correlated color candidates) a greedy search approach is adopted to find the optimal color combination. Decision level fusion with sum rule

is used to fuse the optimal color set at last. Since different color components have different variance distribution, they capture different information for face recognition. By greedy search, MCF finds the optimal color combination that encodes complementary information. The experiments on FRGC2 and AR show that MCF outperforms most of the existing state-of-the-art methods which based on three colors only. Further, the concept of Variance Face (VF) is introduced and we have shown that different colors capture different variance of a human face. These results and findings suggest that using only 3 color components are often not enough to encode all available information for a complex face recognition problem like FRGC.

Two shortcomings can be identified. Firstly, in terms of recognition, our proposed algorithm is based on LDA. As discussed previously in this thesis, LDA is linear subspace method that has limited effectiveness when dealing with large non-linear variation and noise. Nevertheless, a novel color face recognition method is proposed in Chapter 5 to address this problem. Secondly, in terms of color combination, the adopted greedy approach is only sub-optimal. An interesting future research direction will be to formulate the Multiple color selection problem as an optimization problem possibly with an elegant solution.

Chapter 5

Sparse Coding for Color Images

Face recognition is a very challenging problem under uncontrolled conditions, where system robustness is very important. Images of the same person can look very different due to uncontrolled effects such as illumination and pose. Furthermore, the facial images can even be corrupted or occluded. Among many existing robust face recognition solutions, the sparse representation based algorithms are the most promising. Specifically, the Sparse Representation Classifier (SRC) (Wright *et al.*, 2009) described in Section 2.4.1 has achieved the state-of-the-art performance for corrupted or occluded problems, and has received much attention recently. The Correntropy Sparse Representation (CESR) (He *et al.*, 2010) method described in Section 2.4.2 further improves the performance and robustness over SRC.

Despite sparse coding technique can effectively deal with face image under difficult conditions, most of the existing sparse coding methods, including SRC and CESR, are designed and tested on gray-scale images only. To the best of our knowledge, none of the existing state-of-the-art color face recognition algorithms harness sparse representation. For instance, an algorithm proposed by Yang *et al.* (2010a) first normalizes the RGB color image to reduce the inter-color-component correlation. Then it extracts Local Binary Pattern (LBP) and applies FLD (Belhumeur *et al.*, 1997) for face recognition. Another example is the method proposed by Deng *et al.* (2010b). This method first applies Gram-Schmidt (GS) orthogonalization procedure to reduce inter-color-competent correlation, then applies Regularized Couple Mappings (RCMs) for face recognition. Both methods mentioned above are examples of some solid framework for color face recognition, which unlike those described in Section 2.5 that simply aim to find a color space. Nevertheless, they have limited ability to deal with noisy or occluded face images compared to sparse coding methods.

In this work, we propose an explicit approach by utilizing color information when recovering the sparse representation. The overview of the framework is depicted in Figure 5.1. Unlike gray-scale images, each color image has three matrices. Let A_{rgb} be an $m \times n$ color image in RGB space, it is usually arranged as a matrix $A_{rgb} \in \mathbb{R}^{m \times n \times 3}$, while its gray-scale conversion is arranged as $A_{gray} \in \mathbb{R}^{m \times n}$. Therefore, all of the sparse representation based

algorithm working on A_{gray} can not work directly on A_{rgb}.

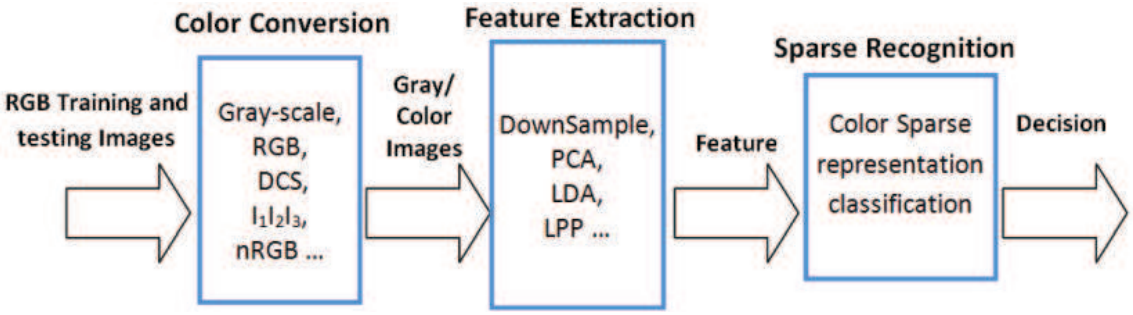


Figure 5.1: The Sparse Representation Framework for color face recognition.

The organization for rest of the chapter is as follows. Section 5.1 proposes the color sparse representation framework. Section 5.2 discusses the role of feature and color space as well as the novel *discriminativeness* measurement. All claims are then evaluated in Section 5.3 and we summarize the chapter in Section 5.4.

5.1 Sparse Coding Framework for Color Images

In this section, we propose a new color face recognition model based on sparse representation. The complete framework involves three steps: color space transformation, feature extraction and classification. Figure 5.1 depicts the framework. Face detection is not included in this framework since it is beyond the scope of this work.

5.1.1 Color Space Transformation

Color space transformation has been described in detail in Section 2.5. Given N RGB training images $A_{rgb} \in \mathbb{R}^{p \times 3 \times N}$ with pixels $p = m \times n$, it can be transformed to another space A_{space}.

Let $f : \mathbb{R}^{p \times 3 \times N} \rightarrow \mathbb{R}^{p \times r \times N}$ be the function that transforms A_{rgb} to A_{space}, where r is the number of color components in the new space:

$$A_{space} = f(A_{rgb}) \tag{5.1}$$

Except for gray-scale, A_{space} usually has 3 color components (i.e. $r = 3$) which encounters difficulty when using traditional feature extraction approaches.

5.1.2 Feature Extraction

Typical feature extraction approaches like PCA (Turk and Pentland, 1991), LDA (Belhumeur *et al.*, 1997) and Locality Preserving Projection (LPP) (He *et al.*, 2005) can be applied directly on gray-scale images arranged as $A_{gray} \in \mathbb{R}^{p \times N}$ where each column represents one face image. Let $g : \mathbb{R}^{p \times N} \to \mathbb{R}^{d \times N}$ be the feature extractor working on A_{gray}, where d is the dimension to be reduced to, we have

$$B_{gray} = g(A_{gray}) \tag{5.2}$$

For color images, one choice is to stack the three color components to one augmented vector (Thomas *et al.*, 2008) such that conventional approaches can be applied directly. This approach however decreases sparse recognition performance for two reasons. First, for some color space like DCS, when stacking the three components together, we need to normalize each component to zero mean and unit standard deviation in order to avoid magnitude dominance on one component over others (Yang and Liu, 2008b). Since variation is normalized, some important discriminant information is removed or weakened. Second, our experiment result shows that the performance of color images usually converges only at dimension that is much higher than $N - 1$. However, PCA dimension is bounded by $N - 1$ while LDA can have at most $C - 1$ dimension for C classes, which is not high enough.

To take full advantage of color images, we should extract features from each color component individually. Let $g_{color} : \mathbb{R}^{p \times r \times N} \to \mathbb{R}^{d \times r \times N}$ be the feature extraction working on color image with r color components [1] $A_{color} = [a_1\ a_2\ \ldots\ a_r]$. The color feature B_{color} can be extracted as:

$$B_{color} = g_{color}(A_{color}) = [g(a_1)\ \ldots\ g(a_r)] \tag{5.3}$$

5.1.3 Classification

The current Sparse Representation algorithm is designed for gray images. Let $h : \mathbb{R}^{d \times N} \times \mathbb{R}^d \to \mathbb{R}^C$ be a kind of sparse representation classifier as described in Section 2.4. However, instead of returning the identity of the query, h returns the residual to each class i. Let

[1] Although r is usually 1 (for grayscale) or 3 (for color image), we use the variable r here for the sake of generality. It is possible for r to be greater than 3 (e.g. when one combines multiple color components). One can also just use two color components with $r = 2$.

$B_{gray} \in \mathbb{R}^{d \times N}$ be N gray-scale training samples each with d features and $y \in \mathbb{R}^d$ be the query image. We obtain the distance to each class:

$$S = [S_1, S_2, \ldots, S_C] = h((B_{gray}, y)) \tag{5.4}$$

where S_i is the distance (residual) to class i ($i = 1, \ldots, C$). We can classify y to class i with $\min_i S_i$.

Next, we discuss how to apply h to color face images. After feature extraction as mentioned in last section, we have obtained training color features with r color components $B_{color} = [b_1 \ \ldots \ b_r] \in \mathbb{R}^{d \times r \times N}$ and query color features $y_{color} = [y_1 \ \ldots \ y_r] \in \mathbb{R}^{d \times r}$. One choice is to calculate the residual for each individual component and sum the distances up for a decision:

$$\hat{S} = h_{color}((B_{color}, y_{color})) = h((b_1, y_1)) + \ldots + h((b_r, y_r)) \tag{5.5}$$

This approach however is r times slower. High computational cost is one of the main drawbacks for sparse based approach, therefore this amount of computational time increment is undesirable. Furthermore, computing the sparse representation on each color individually does not make use of the complementary information among different color compoenents effectively.

In order to take full advantage of complementary information, we want to find the sparse representation directly for B_{color} and y_{color} by optimizing the following proposed model:

$$\begin{aligned} x = \min \ \|x\|_1 \quad & \text{s.t.} \\ y_{color} &= B_{color} \ x \\ &= x_1 B_1 + \ldots + x_N B_N \\ &= x_1 [b_{11}, \ldots, b_{1r}] + \ldots + x_N [b_{N1}, \ldots, b_{Nr}] \end{aligned} \tag{5.6}$$

where $B_j \in \mathbb{R}^{d \times r} (j = 1, \ldots N)$ is a particular color training sample with r color components. Unlike B_{gray} where each face is a column vector, here each face B_j in B_{color} is a matrix and we want to find a linear combination of these matrices. Therefore, the original sparse representation algorithm can not be applied directly. However, we notice that finding x in Eq. 5.6 is equivalent to finding x for the rearrangement of B_{color} and y_{color} by stacking the color component features into augmented column vectors which is called the *color image arrangement*. Let D and z be the rearranged training samples and the query, we have:

$$D = [\ [b_{11} \ \ldots \ b_{1r}]^T, \ldots, [b_{N1} \ \ldots \ b_{Nr}]^T\ ] \in \mathbb{R}^{rd \times N} \tag{5.7}$$

$$z = [y_1 \ \ldots \ y_r]^T \tag{5.8}$$

Then the coefficients x for the model in Eq. 5.6 can be solved by optimizing the convention sparse model:

$$x = \min \ \|x\|_1 \quad \text{s.t.} \quad z = Dx \tag{5.9}$$

Therefore, the distance metric h will work on D and z, which can be computed directly by:

$$S_{color} = h((D, z)) \tag{5.10}$$

Lastly, we classify the query to class i with $\min_i S^i_{color}$.

5.1.4 Algorithm

Algorithm 5.1 summarizes the complete sparse coding framework for color face images.

Algorithm 5.1 Color Sparse Representation Classifier

Require: -

A_{rgb}, RGB training samples
y_{rgb}, RGB query image
c, choice of color space
F, choice of feature extractor
s, choice of sparse algorithms

1. Color Space Conversion by Eq. 5.1:
   $A_c \leftarrow f_c(A_{rgb})$
   $y_c \leftarrow f_c(y_{cs})$
2. Feature Extraction by Eq. 5.3:
   $B_{c,F} \leftarrow g_{color,F}(A_c)$
   $y_{c,F} \leftarrow g_{color,F}(y_c)$
3. Color image arrangement using Eq. 5.7 and (5.8):
   Rearrange $B_{c,F}$ to D
   Rearrange $y_{c,F}$ to z
4. Normalize columns of D and z to unit ℓ^2-norm.
5. Compute the distance using Eq. 5.10:
   $S = h(D, z)$

return identity$(y) = \min_i S_i$

The computational time of algorihtm 5.1 is almost the same as the conventional algorithm working on gray images. The time increases slightly with increasing number of color components, which is usually 3, due to feature extraction on 3 times higher dimension data.

Unlike the computation of sparse representation, conventional feature extraction methods as well as downsampling are usually computed in real time. In this case, the feature extraction time increased is almost negligible. In fact, the most time consuming step is actually the sparse representation recovery. The proposed algorithm has exactly the same complexity as the conventional ones regardless of the training sample size. Its complexity is slightly higher than the conventional ones only when data dimension increases. Therefore, the proposed framework does not have noticeable time complexity increment.

5.2 Roles of Features and Color Spaces

In this section, we define *correctness* and *discriminativeness*, which are the two factors affecting performance of any sparse representation based classifiers. We argue that different choices of feature space and color space do not affect *correctness*, but will affect *discriminativeness* and hence the performance.

5.2.1 Correctness

Correctness measures how close is the representation found via ℓ^1-norm minimization to the one found via ℓ^0-norm minimization. The representation is said to be ideally correct if x_1 recovered by Eq. 2.26 is equivalent to x_0 in Eq. 2.25. There are established theories to guarantee the equivalence between ℓ^l and ℓ^0 minimization under mild conditions stated by Sharon *et al.* (2007). Loosely speaking, sparse representation based classifiers are guaranteed to be correct if the dimension of the data is high enough (Wright *et al.*, 2009). Therefore, different choices of features and color spaces with sufficiently high dimension do not affect correctness.

5.2.2 Discriminativeness

Discriminativeness describes how uniquely and precisely the sparse coefficient x reconstructs the query image y. The representation is said to be ideally discriminative if the coefficients associating with the same class as y would reconstruct y with no error while all other coefficients can not even approximate[2] y. Since x is found by enforcing sparsity,

[2]A reconstruction $\hat{y}$ does not approximate y if the reconstruction error is greater than or equal to the reconstruction error of a random reconstruction.

it tries to find a linear combination involving fewest possible training samples. In order to achieve minimum number of training samples involvement, only training samples belong to the same class as y should contribute to the representation and thus x becomes discriminative. Ideally, entries of x should be all zero except those associated with the same class as y. Obviously, sparse representation is not designed for classification purpose, it can be used for classification because it is usually discriminative. It relies heavily on whether x is close enough to the ideal case mentioned.

However, in practical context of robust face recognition, it is impossible to gather complete training set to span the whole space of each class (modeling every possible intra-person variation) without class overlap. Imagine to find the sparse representation for a query that lies in the overlapped space of two classes. The representation may involve linear combination of some number of training samples from both classes, hence increasing the chance for misclassification. Therefore, it is impossible to find an ideal x. Different features and color spaces disclose different information that may reduce the intra-class variation and increases the inter-class distance, hence affecting the *discriminativeness* and performance.

In this chapter, we propose a new measurement to describe discriminative power of x as the ratio of the reconstruction error using coefficients associated with the same class as y over the reconstruction error using all other coefficients. Specifically, let A be the training samples and y belongs to class J, we define $\delta_J(x)$ similar to Eq. 2.32 being a vector whose non-zero entries are entries in x corresponding to class J and similarly the complement $\tilde{\delta}_J(x)$ be a same vector as x with entries of class J set to zero. The within-class reconstruction error of y is $E_w = \|y - A\delta_J\|_2$ and the between-class reconstruction error is $E_b = \|y - A\tilde{\delta}_J\|_2$. The discriminativeness of x is defined as:

$$DIS(x) = \begin{cases} 0 & \text{if } E_b + E_w = 0 \\ E_b/(E_b + E_w) & \text{otherwise} \end{cases} \tag{5.11}$$

Since E_w and E_b is always non-negative, the range of DIS is always bounded between $[0, 1]$. If y can be reconstructed perfectly using $\delta_J(x)$ while $\tilde{\delta}_J(x)$ can not, then $E_w = 0$ and $E_b \neq 0$. We have $DIS = E_b/(E_b + 0) = 1$, which implies perfect discriminativeness. If y can be reconstructed perfectly using $\tilde{\delta}_J(x)$, then we have $DIS = 0/(0 + E_w) = 0$ which indicates nil discriminative power. Furthermore, if both E_w and E_b are not zero, DIS will be 0.5 when $E_w = E_b$, over 0.5 when $E_w < E_b$ and less than 0.5 when $E_b < E_w$.

5.3 Experimental Evaluations

In this section, we evaluate the proposed color sparse framework and validate the claims in previous sections using two publicly available face databases. For fair comparison, we follow exactly the same experiment protocols as reported in literatures (Wright *et al.*, 2009; He *et al.*, 2010; Yang *et al.*, 2010a; Jiang *et al.*, 2008; Li *et al.*, 2009). Specifically, we will compare sparse representation methods under different color and feature spaces. We show how our approach performs in comparison to the state-of-the-art performances. We also compare the performances of color with gray-scale in cases of random pixel corruption and occlusion.

5.3.1 Algorithm implementation and parameter selection

We use the source code provided by He *et al.* (2010) and use the same notations in our experiments:

- *SRC0*: Solves the standard SRC in Eq. 2.26 via an active set algorithm.
- *SRC2*: Uses the Lasso optimization algorithm to solve Eq. 2.29.
- *CESR*: Integrating the maximum correntropy criterion and non-negative constrain as in Eq. 2.34.

The parameters are estimated using five-fold cross-validation on each dimension and training set. Specifically, λ_{src2} for SRC2 is set to 0.001 and λ_{cesr} for CESR is set to 0.05.

5.3.2 Databases

To allow fair comparison to (Wright *et al.*, 2009), we use the same AR face database (AR) as in their experiment. However, the Extended Yale B database used in (Wright *et al.*, 2009) only consists of gray-scale images, it is not suitable for our experiment here. Therefore, we choose the Georgia Tech face database (GT) which is also widely used in face recognition community to evaluate color face recognition algorithms.

5.3.2.1 AR Face Database

AR (Martinez and Benavente, 1998) consists of over 4000 color frontal images for 126 individuals each having 26 images taken in two sessions. The uncontrolled variations mainly include facial expression, illumination and occlusion. All images used are cropped to 165 $\times$ 120 manually with eyes and mouth aligned to the same location. Illustrative images of one person are shown in Figure 5.2.

Figure 5.2: Sample images of one person from AR database

5.3.2.2 Georgia Tech Face Database

GT (Nefian, 2007) consists of 700 color images for 50 individuals each with 15 images. The uncontrolled variations mainly include facial expressions, illumination and poses. All images used are cropped to 146 $\times$ 120 manually with eyes and mouth aligned to the same location. Example images of one person are shown in Figure 5.3.

Figure 5.3: Sample images of one person from GT database

5.3.3 Downsampling

The downsampling method used through out our experiments is done with Matlab's "imresize" function, which uses the bi-cubic interpolation technique by default. The ratios between the height and width of the downsampled images are kept as close as to the original. For color images, as described in Section 5.1.2, downsampling is first applied on each individual color component image, the downsampled features are then combined. Note that we have to maintain the same final dimensions for color images to allow fair comparison with gray images. To this end, the downsampled dimension for each individual color component is lower than that for gray image. For example, if the final dimension is set to 100, a gray-scale image is downsampled directly to 100 pixels. While a RGB image is downsampled to 33 pixels such that after stacking the R, G and B components, the final dimension is 99 (which is the closest to 100 that we can achieve).

5.3.4 Various Color Spaces

We first evaluate the performance of SRC0 under uncontrolled condition but without corruption or occlusion. Although SRC2 is more robust when dealing with noise, SRC0 is usually enough in this situation with the advantage of faster computation. We show how different color spaces affect the performance with varying downsampled dimensions and how it performs in comparison with gray-scale on both AR and GT databases.

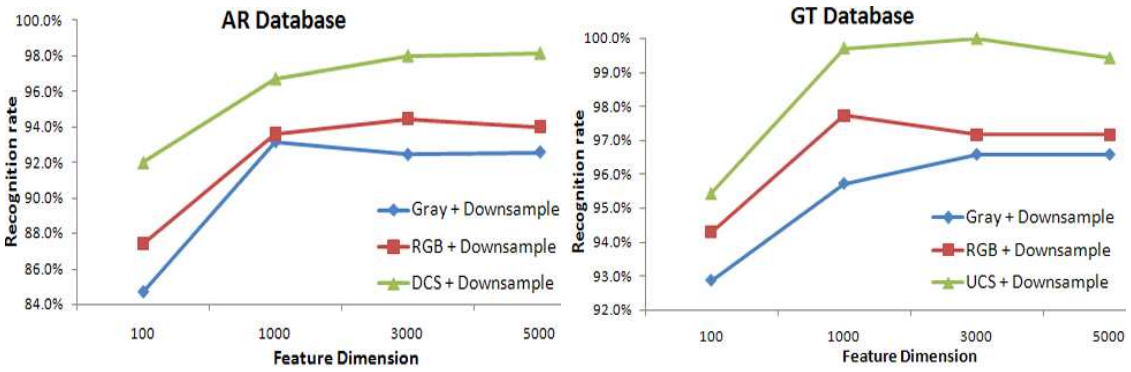
Our experiment protocol follows exactly the same as in (Wright *et al.*, 2009; Naseem *et al.*, 2010). For AR, only the subset of 50 males and 50 females without occlusion is used, resulting in 14 images per subject from two sessions. Consider the aging effect in real world application, the first 7 images per subject from session 1 are used for training while the rest from session 2 for testing. For GT, the first 8 images per subject are used for training and the last 7 per subject for testing.

The results are presented in Table 5.1 and Figure 5.4. Table 5.1 reports the face recognition rate for AR and GT with various color spaces, as well as the total time taken. Figure 5.4 plot the recognition rate for image in gray-scale, RGB space and the best performing color space in each database with different downsampled dimension.

Four findings can be highlighted here. First, color outperforms gray-scale consistently, which reveals that color can always improve the performance of SRC, no matter what color space is used. Second, different color spaces have different performances. This

Table 5.1: Recognition rates (%) and total time needed (second) with various color spaces.

Recognition Rate						
	Gray	RGB	DCS	UCS	$I_1I_2I_3$	nRGB
AR database	92.6	94.0	**98.1**	95.6	98.0	98.0
GT database	96.6	97.1	96.0	**100.0**	99.7	99.7
Total Time in Second						
	Gray	RGB	DCS	UCS	$I_1I_2I_3$	nRGB
AR database	217.0	222.3	231.5	239.0	227.7	225.1
GT database	22.4	23.9	24.8	24.4	24.2	22.7



(a) (b)

Figure 5.4: With various downsampled feature dimensions: (a) Comparisons of Gray-scale, RGB and DCS on AR. (b) Comparisons of Gray-scale, RGB and UCS on GT.

implies that the choice of color space is important for SRC. Although SRC is claimed to be invariant to the choice of feature (Wright *et al.*, 2009), its performance can be affected by color. By choosing the right color space, the performance for SRC can be improved by 5.5% in AR and 3.4% in GT. Third, color information is more useful than appearance information (resolution) for SRC. This can be best illustrated by the result when images are downsampled to 100 dimensions. In this case, as described in Section 5.3.3, gray scale images are downsampled to 100 pixels while color images are downsampled to 33 pixels (where each of the three color component images has 33 pixels) in order to maintain the dimension as close as to-100 (99 in this case) after combining the three color components. However, even color images contain three times less the appearance information, it still outperforms gray-scale. Lastly, the total time [3] needed for color images are slightly larger

[3]These time are obtained on a computer with an Intel Core Quard CPU @ 3GHz and 4GB of RAM. The sparse coding computation is done using the SPAMS package available at: http://spams-devel.gforge.inria.fr/, which is a c implementation optimized for sprase coding, with Matlab interface. All other steps are done in 64-bit Matlab without extra code optimization effort.

Table 5.2: Discriminativeness measures (DIS): mean ± variance, with various color space

AR database			
Dimension	Gray	RGB	DCS
100	0.49 ± 0.21	0.47 ± 0.21	$\mathbf{0.51 \pm 0.19}$
1000	0.51 ± 0.18	0.53 ± 0.18	$\mathbf{0.55 \pm 0.17}$
3000	0.50 ± 0.17	0.52 ± 0.17	$\mathbf{0.54 \pm 0.16}$
5000	0.49 ± 0.17	0.52 ± 0.17	$\mathbf{0.53 \pm 0.15}$
GT database			
Dimension	Gray	RGB	UCS
100	$0.60 \pm \mathbf{0.21}$	$\mathbf{0.61} \pm 0.22$	$\mathbf{0.61} \pm 0.23$
1000	0.60 ± 0.18	0.63 ± 0.18	$\mathbf{0.64 \pm 0.17}$
3000	0.58 ± 0.18	0.62 ± 0.18	$\mathbf{0.63 \pm 0.17}$
5000	0.58 ± 0.17	0.61 ± 0.18	$\mathbf{0.62 \pm 0.17}$

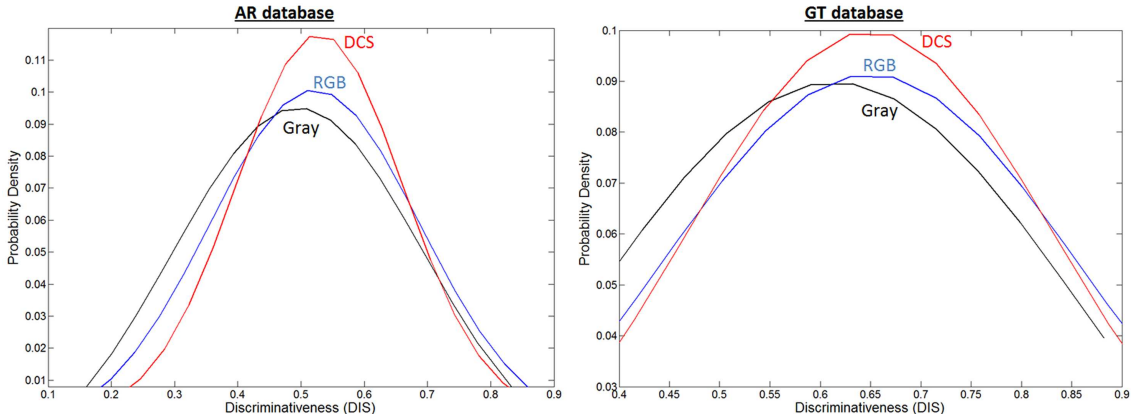


Figure 5.5: The distribution plot of DIS values with dimension 5000 for AR and dimension 3000 for GT.

than that for gray-scale images as expected. This complies with our time complexity analysis in Section 5.1.4.

To analyze how color affects the performance of SRC, the discriminativeness measure (DIS) of the sparse representation recovered in different color spaces and gray-scale are shown in Table 5.2. Their corresponding distribution is plotted in Figure 5.5. For each query, we compute the DIS using Eq. 5.11 and report the mean DIS and its standard deviation ($\mu_{DIS} \pm \sigma_{DIS}$). Since the recognition rates for both databases are over 90%, the DIS difference is not large. However, a small difference is significant since it is averaged over a few hundreds of query samples. One can see that the distribution of DIS with color images are always with higher mean values and smaller deviations.

One can find that the μ_{DIS} values computed for color spaces are always higher with smaller σ_{DIS}. This implies that the sparse representation recovered in color space is always more discriminative, hence classifying queries more precisely. This is the main factor why color can improve the performance of SRC over gray-scale.

5.3.5 Various Feature Extractors

We next evaluate the performance with different feature extractors. We use the same experimental set-up as in former section, however instead of only downsampled features, we also compared PCA (Turk and Pentland, 1991), LPP (He *et al.*, 2005) and LDA (Belhumeur *et al.*, 1997). These are some well known feature extractors which obtain features from the original data by projecting it to the feature space linearly. To obtain the projection matrix, the respective objective functions have to be solved and the solution to the three mentioned methods can actually be converted to Eigen value decomposition problems. After solving the Eigen problem, the first d Eigen vectors corresponding to the d largest Eigen values are chosen to form the projection matrix. Then the data can be projected to the d-dimension feature space. For color images, as described in Section 5.1.2, feature is extracted on each color components separately and then combined in feature level. Therefore, for a color space that has three color components, the final dimension becomes $3d$. Our evaluation is done on the dimension with the best performance for each method.

Since all subspace methods compared here are all based on ℓ^2-norm minimization, it is also interesting to investigate whether color can improve the performance of SRC under a random projection (randomFace). RandomFace is defined in (Wright *et al.*, 2009) as dimension reduction technique based on linear projection, where the projection matrix is sampled from a standard normal distribution. In case of color images, one random projection matrix is created for one color channel and it is used for all three channels for dimension reduction in order to make sure that the same number of random features are extracted from gray-scale and color images. Similarly, only the best performing dimension is reported.

Table 5.3 reports the results for both AR and GT. The first three columns compare the performance of gray, RGB and DCS images with different feature extractors using SRC0 as the classifier. The last two columns report the performance for the Nearest Neighbor (NN) classifier. PCA+NN, LDA+NN and LPP+NN are some well-known traditional approaches widely used for benchmarking comparison. Here we integrate them with the DCS color space and UCS color space using either image level fusion (I) or feature level

Table 5.3: Recognition rates (%) with various feature extractors.

AR database					
	SRC0			NN Classifier	
Feature(Dimension)	Gray	RGB	DCS	DCS(I)	DCS(F)
Downsample(5000)	92.6	94.0	**98.1**	-	-
RandomFace(3000)	93.0	95.0	**96.0**	-	-
PCA(699)	92.9	93.9	**97.6**	82.0	76.8
LPP(300)	86.7	90.4	**94.1**	90.1	53.0
LDA(99)	89.2	92.0	**96.0**	95.6	93.0
GT database					
	SRC0			NN Classifier	
Feature(Dimension)	Gray	RGB	UCS	UCS(I)	UCS(F)
Downsample(3000)	96.6	97.1	**100.0**	-	-
RandomFace(3000)	96.3	96.5	**96.6**	-	-
PCA(399)	96.3	96.3	**97.3**	79.4	82.3
LPP(200)	88.6	**93.1**	90.1	86.3	47.1
LDA(49)	94.8	95.4	**95.6**	95.5	93.7

fusion (F). For (I), after transformation to the new color space, the three color components are normalized to zero mean and unit standard deviation before stacking together to one column. This fusion scheme is also used by Yang and Liu (2008b) and Liu (2008). For (F), it is the one used in our proposed framework in Section 5.1. Here we just replace the classification step in Section 5.1.3 by NN classifier to yield a meaningful comparison to our approach.

Three findings are highlighted here. The first one is similar to former section, RGB performs better than gray-scale, while DCS/UCS performs the best among all the feature extractors, except for LPP in GT. Same pattern is observed even when using random features. This once again justifies the effectiveness of color when using SRC for face recognition. Second, in DCS/UCS, SRC is always better than NN. The proposed framework outperforms traditional approaches. Lastly, downsampling is the best among all feature extraction methods. This may be due to the fact that downsample method allow much higher feature dimensions for the performance of SRC to converge (5000 for AR and 3000 for GT in this case).

Table 5.4 presents the corresponding discriminative measures (DIS) of the sparse representations recovered in different feature spaces. Similar to Table 5.2, the DIS is computed using Eq. 5.11 for each query and the mean and standard deviation is reported

Table 5.4: Discriminativeness measures(DIS): mean ± variance, with various feature extractors.

AR database			
Feature(Dimension)	Gray	RGB	DCS
Downsample(5000)	0.49 ± 0.17	0.52 ± 0.17	$\mathbf{0.53 \pm 0.15}$
RandomFace(3000)	0.48 ± 0.16	0.49 ± 0.15	$\mathbf{0.49 \pm 0.14}$
PCA(699)	0.48 ± 0.17	0.50 ± 0.16	$\mathbf{0.50 \pm 0.15}$
LPP(300)	0.35 ± 0.12	0.40 ± 0.10	$\mathbf{0.41 \pm 0.08}$
LDA(99)	0.44 ± 0.15	0.46 ± 0.11	$\mathbf{0.48 \pm 0.07}$
GT database			
Feature(Dimension)	Gray	RGB	UCS
Downsample(3000)	0.58 ± 0.18	0.61 ± 0.18	$\mathbf{0.63 \pm 0.18}$
RandomFace(3000)	0.57 ± 0.17	0.59 ± 0.17	$\mathbf{0.60 \pm 0.17}$
PCA(399)	0.57 ± 0.18	0.59 ± 0.18	$\mathbf{0.60 \pm 0.18}$
LPP(200)	0.39 ± 0.13	$\mathbf{0.42 \pm 0.12}$	0.40 ± 0.12
LDA(49)	0.51 ± 0.15	0.52 ± 0.13	$\mathbf{0.53 \pm 0.13}$

($\mu_{DIS} \pm \sigma_{DIS}$). The proposed DIS measurement precisely describes the performance reported in Table 5.3. Methods that have higher recognition rate always have either higher μ_{DIS} or lower σ_{DIS}.

5.3.6 Comparisons to the State-of-the-Art Algorithms

In this section, we compare our approach with some state-of-the-art methods reported in the literature for face recognition under uncontrolled condition but without corruption or occlusion. To allow meaningful comparisons, we have selected methods that are evaluated under the same experimental protocol as ours.

For AR database, the Nonparametric Discriminant Analysis (NDA) method for face recognition (Li *et al.*, 2009) is published in a reputed journal (IEEE PAMI) in 2009. In their work, two extensions on NDA are proposed namely the Principal Nonparametric Feature Analysis (PNFA) and Null-space Nonparametric Feature Analysis (NNFA). Their experimental result shows that the fusion of PNFA and NNFA using voting rule achieved the state-of-the-art performance (91.9%), outperforming PCA, LDA, Bayesian method, Kernel LDA, LDE and Multi-class NDA.The Sparse Representation Classifier (SRC) (Wright *et al.*, 2009), which is described in detail in Section 2.4.1, is published in PAMI as well in

2009. The method has achieved 92% to 94.7% using various feature extractors. These two methods use only gray-scale images. For color face recognition, one of the state-of-the-art methods, as described at the beginning of the chapter, is nRGB+LBP+FLD (Yang *et al.*, 2010a) which is published in another reputed journal (Pattern Recognition) in 2010. The best performance they have achieved is 94.6% on AR database.

For GT database, one of the best methods is the Regularized Eigenfeature method (Jiang *et al.*, 2008) published in PAMI in 2008. They have compared to 8 other state-of-the-art methods but the proposed ERE_S^t has achieved the best performance (93.1%). The Linear Regressing Classifier (LRC) (Naseem *et al.*, 2010) is published in PAMI in 2010. They have achieved 92.57%. Unlike ERE_S^t, which uses carefully engineered features, LRC just uses the downsampled features. These two methods use only gray-scale images. For color face recognition, as described at the beginning of the chapter, the GS+RCMs (Deng *et al.*, 2010b) method is one of the state-of-the-art methods for color face recognition that has achieved 98.1% on GT database.

Our framework achieved 98.1% in AR and 100% in GT. It outperforms the second best method by 3.4% percent in AR and 1.9% in GT. This result suggests that by carefully selecting the suitable color space and feature dimension, the proposed framework can achieve the new state-of-the-art performance. Table 5.5 summaries the performance. Note that although we only show the best performance of the proposed framework, the choice of parameters does not affect much of its performance. As previously shown in Table 5.1 and Figure 5.4, the best performance achieved by the proposed framework in both AR and GT does not vary much when dimension is high enough (>1000).

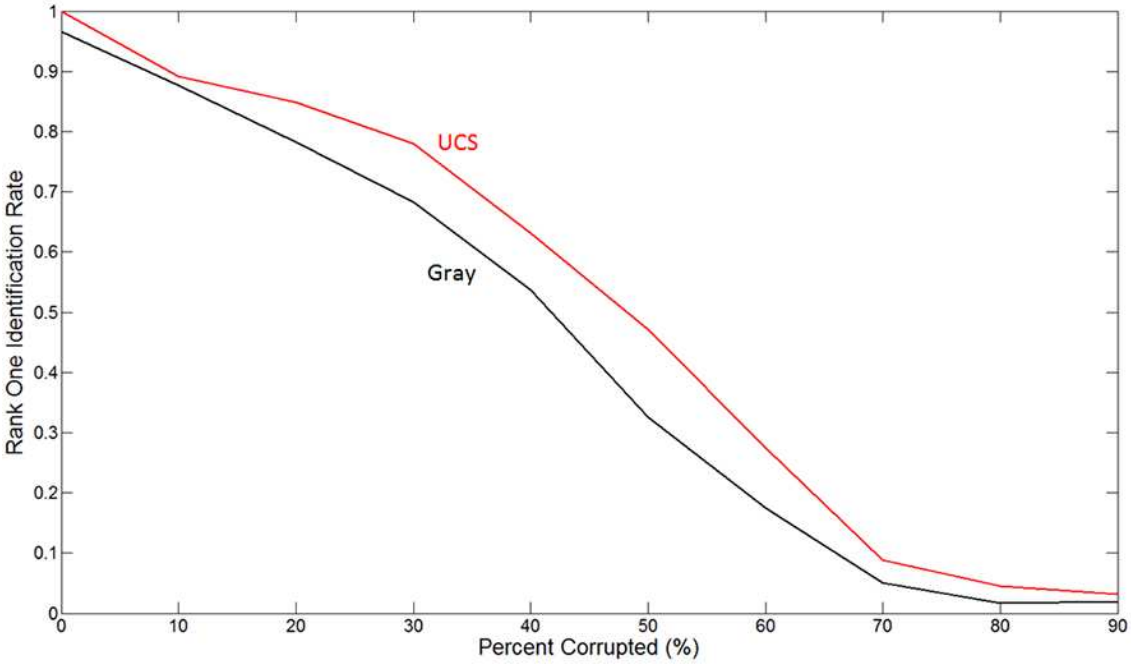
Table 5.5: State-of-the-art recognition rates on AR and GT.

AR database			
PNFA+NNFA+Voting	SRC	nRGB+LBP+FLD	*proposed*
91.9%	92-94.7%	94.6%	**98.1%**
GT database			
ERE_S^t	LRC	GS+RCMs	*proposed*
93.1%	92.6%	98.1%	**100.0%**

5.3.7 Evaluation on Random Pixel Corruption

In this section, we will evaluate the proposed framework under random pixel corruption. One of the main goals for robust face recognition is to tolerate error and noise. Real world

images can be corrupted in random possibly during the process of capture or transmission. We simulate this kind of error by randomly choosing entries of the query image matrix and replacing its intensity value with a random value in $[0, 255]$ (which is applied to all three color channels in case of color image). Each query has different corrupted locations which is not known to the algorithm in advance.



Percent corrupted	0%	10%	20%	30%	40%
Gray-Scale	96.6%	87.7%	78.3%	68.3%	53.7%
UCS	**100.0%**	**89.1%**	**84.8%**	**78.0%**	**63.1%**
Percent corrupted	50%	60%	70%	80%	90%
Gray-Scale	32.6%	17.4%	5.1%	1.7%	2.0%
UCS	**47.1%**	**27.4%**	**8.8%**	**4.5%**	**3.1%**

Figure 5.6: Recognition on GT under random pixel corruption.

We setup this problem on GT database and employ the SRC2 algorithm. Same experimental setting is used as in previous section. We do not repeat the experimental setting in (Wright *et al.*, 2009) because the Extended Yale B is a gray-scale database. In order to isolate the effect of corruption while testing precisely how good the proposed approach can tolerate corruption, we need a database that the performance is expected to decrease with increasing percentage of corruption. For dimension reduction, Both gray-scale and color image are downsampled (as described in Section 5.3.3) to 4000 (final) dimension.

Figure 5.6 reports the recognition rate for gray-scale and UCS up to 90% corruption. UCS completely outperforms gray-scale as expected. The maximum difference is 15.5% with

50% corruption. We conclude that the richer information in color space can help tolerating more noise and errors.

5.3.8 Evaluation on Occlusion

One of the most difficult challenges for robust face recognition is to identify an occluded person. Under occlusion, only part of the face is useable for recognition. In addition, the location of occlusion may vary and is not known in advance. Unlike random pixel corruption, the affected pixels in real world occlusion are usually continuous over a region. Therefore, this is actually the worst kind of error for robust face recognition.

We evaluate our framework using CESR on subset of the AR database. As claimed in (He *et al.*, 2010), CESR is more robust and outperforms SRC when recognizing occluded people. Further, DCS is the best performing color in AR database. Therefore, we integrate CESR with DCS in this experiment. In addition, we find that instead of initializing the weighting to all ones as in (He *et al.*, 2010), better performance can be achieved by initializing to the correntropy similarity between the query and mean face (i.e. $g(y - \overline{A})$).

Following the protocol in (Wright *et al.*, 2009), for the same 100 subjects used in previous section, we select per subject the first 4 images from session 1 and first 4 images from session 2 for training (except for a corrupted image w-027-14.raw). These eight images per subject is all un-occluded and frontal with varying facial expressions. Every image is downsampled to 4000 dimension. For the experiment on sunglasses occlusion, 2 images wearing sunglasses per subject are selected as queries. Similarly for the experiment on scarves occlusion, 2 images wearing scarf per subject are selected as queries. The total number of testing images for the two experiments are both 200 each. Examples of such images can be found in Figure 5.2. Table 5.6 compares the recognition rates between gray-scale and DCS as well as for various other methods designed specifically to tackle real world occlusion problem.

Table 5.7 lists results with different CESR weighting initializations. Three results should be highlighted. First, CESR performs better in DCS compared to gray-scale. Especially for scarf occlusion, the original CESR using gray-scale images only achieves 89%, however color improve the performance to 95%, which is significant. Second, the proposed approach outperforms all other state-of-the-art algorithms for occluded face recognition in the existing literature. Lastly, the proposed weighting initialization (i.e. $g(y - \overline{A})$) boosts the recognition rate, while the original all-ones, as well as 10 random initializations all converge to similar performance. Especially for scarves occlusion on Gray image, the

Table 5.6: Recognition rates on AR under real world occlusion. Unlike (He *et al.*, 2010), we set the initial weighting of CESR to the correntropy similarity between the query and mean face (i.e. $g(y - \overline{A})$).

Approach	Sunglasses	Scarves
DCS + CESR	**100.0%**	**95.0%**
Gray + CESR (He *et al.*, 2010)	97.0%	89.0%
Other reported performance		
SRC in (Wright *et al.*, 2009)	87.0%	59.5%
LRC in (Naseem *et al.*, 2010)	96.0%	26%
PCA + NN (Turk and Pentland, 1991)	70.0%	12.0%
ICA I + NN (Kim *et al.*, 2005)	53.5%	15.0%
LNMF + NN (Li *et al.*, 2001)	33.5%	24.0%

Table 5.7: Recognition rates for CESR with different initial weighting. "All-ones" denotes the original strategy used in (He *et al.*, 2010), initializing weighting to all ones. "Random" denotes 10 random initializations. "$g(y - \overline{A})$" denotes initializing the weighting using the correntropy between the query and the mean training sample face.

Sunglasses	All-ones (He *et al.*, 2010)	Random (min-max)	$g(y - \overline{A})$
DCS + CESR	99.5%	99.0 - 99.5%	**100.0**%
Gray + CESR	95.5%	95.0 - 95.5%	97.0%
Scarves	All-ones (He *et al.*, 2010)	Random (min-max)	$g(y - \overline{A})$
DCS + CESR	79.5%	80.0 - 81.0%	**95.0**%
Gray + CESR	45.0%	45.5 - 46.0%	89.0%

performance is increased from 45% to 89%. We believe this is due to that the all-one initialization is more likely leading to local optimal. Since the mean face is computed using some real human face images, thus the computed mean pixel values will be highly correlated to those on a face image. Using the mean face to compute the initial weighting mask can effectively mask out non-face shading caused by for example uneven illumination or occlusion. Therefore, noise or highly uncorrelated pixels will receive lower focus (or weighting) at the beginning which eventually increases the chance of converging to a global optimal weighting after several iterations.

We further analyze why DCS can significantly improves the performance of CESR on scarf occluded images. As discussed in Section 2.4.2, CESR can be interpreted as a pixel-weighted SRC, which iteratively update the pixel weighting and the sparse representation.

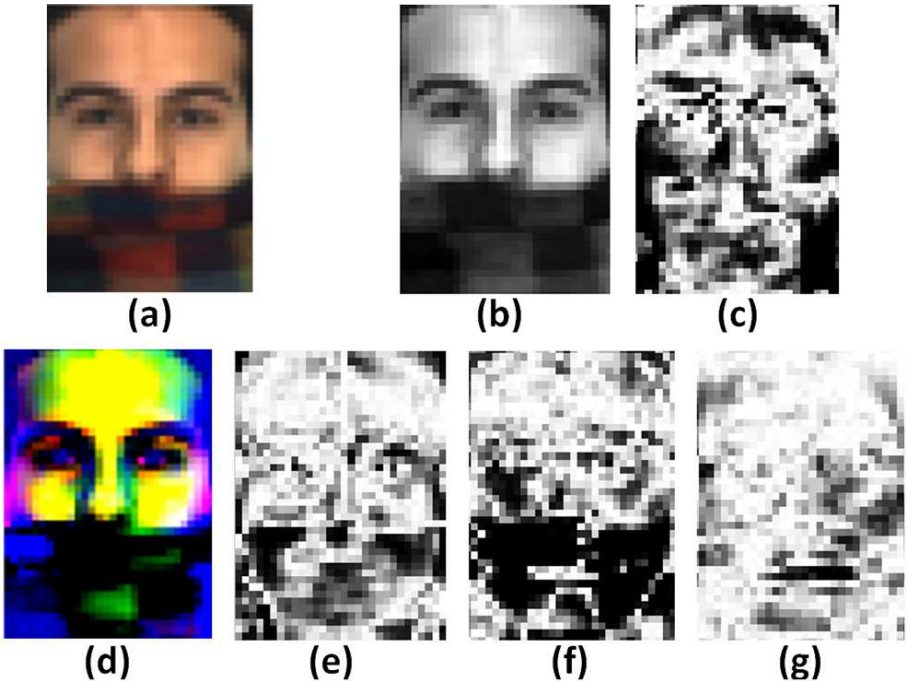


Figure 5.7: A query from AR database with scarf occlusion. The corresponding pixel weighting obtained by CESR is shown as an intensity image, which darker pixel means lower weighting. (a) The original RGB image. (b) The original image in gray-scale. (c) The weighting on gray-scale image. (d) The original image in DCS. (e-g) The weighting on the 1st, 2nd and 3rd DCS components respectively.

The way that CESR computes the weighting is based on the reconstruction residual of each pixel. The weighting maps for one query are shown in Figure 5.7 (c, e, f and g), where darker pixel represents lower weighting, which in other words, higher reconstruction residual. Notice on the gray-scale image (b) that the intensity values of the scarf is just a little darker but still very close to the skin intensity values. As this scarfed area is nearly 40% of the face region, finding a linear combination of the training samples to best reconstruct (b) will result to a darker face image like the scarf. This would cause larger reconstruction residual on the valid face pixels than on the scarf pixels. Therefore, the low-weighted pixels in (c) are clustered around the forehead and two side of the face, but not the scarfed area due to the fact that it is being resembled. Since some valid pixels are treated as error, the performance decreases by 33% when it compares to the original SRC in Table 5.6. On the other hand, one can see on image (d) that the color of the scarf and the face have significantly higher contrast in DCS space. Therefore, the low weighting are concentrated on the scarf pixels as they can not be reconstructed. This effect is more apparent in (f) where the weighting is computed in the 2nd DCS component. As a result, the performance can be increased significantly.

5.4 Summary

In this chapter, the following contributions have been made. We have shown that color can be used to improve performance of robust face recognition. The proposed color sparse representation framework outperforms both gray-scale methods and some other state-of-the-art algorithms. Supporting results are obtained with experiments on uncontrolled condition, different feature extractors, random pixels corruption as well as occlusions. We also introduce a concept about *discriminativeness* and contrast its difference to *correctness*. We argue that when the data dimension is high enough, the choice of features or color spaces do not affect correctness but discriminativeness. The sparse representation recovered in color space is very different from the one recovered with gray-scale. The main contribution of color is that its richer information can effectively increase discriminativeness of the sparse representation and help correct errors, which is critical for the success in robust face recognition. Therefore, the proposed color sparse framework provides a better solution to this problem.

There are two main shortcomings of the proposed method. Firstly, sparse coding is based on linear combination. Although it can effectively handel noisy data, it is not a reliable solution to the pose problem. Pose variation usually causes non-linear variation in data which can not be modeled linearly. Secondly, the proposed algorithm vectorizes the color image to a single vector in order to apply sparse coding methods directly. Vectorization is a mechanical step which may destroy important image structure. Color image should be represented naturally by a 3-rd order tensor. We will address these two problems and propose two novel face recognition algorithms in Chapter 6.

Chapter 6

Face Recognition using Kinect

As discussed in Chapter 1, face recognition can be done without user cooperation. However, in this context, the query image is usually in uncontrolled condition, posing great challenges for face recognition. These challenges can not be tackled reliably with only 2D images. In fact, it has been shown that face recognition based on RGB-D (Red, Green, Blue and Depth) information outperforms traditional 2D methods (Bowyer *et al.*, 2006). Existing techniques (Wang *et al.*, 2010; Queirolo *et al.*, 2010) are able to achieve over 99% accuracy on difficult experiments such as the Face Recognition Grand Challenge (FRGC) (Phillips *et al.*, 2005). However, all these methods are based on high resolution 3D scanners which are usually expensive, bulky and have slow acquisition time. Although, low cost 3D acquisition devices are available in the market, they usually generate very noisy and low resolution depth information. Whether such low quality data can be used to improve face recognition performance is an unknown question.

The recent release of Kinect sensor has received much attention because it can provide low cost 3D data with high speed. Along with 3D data, the corresponding 2D color texture data is also produced. In this chapter, we investigate whether Kinect is suitable for robust face recognition. Due to the lack of publicly available Kinect face database that consists of large amount of variations, we have constructed one namely CurtinFaces for our experiments and for the research community. Two algorithms are proposed and evaluated on this new dataset. The first algorithm namely the Multilinear Color Tensor Discriminant (MCTD) model is a novel color face recognition algorithm. It makes use of only 2D data. By utilizing tensor structure and multilinear analysis technique, it outperforms other state-of-the-art 2D methods when handling face images with large variations. The second algorithm namely Finer Feature Fusion (FFF) is a novel RGB-D face recognition algorithm. We show that, by utilizing the low quality 3D data Kinect provided, the performance of FFF is more robust. As a result, we can justify the usefulness of Kinect 3D data.

The rest of this chapter is organized as follows. Section 6.1 compares several commercial 3D acquisition devices with the Kinect sensor. Section 6.2 describes the challenge of using Kinect data. Section 6.3 details the specifications of the CurtinFaces database. Section 6.4 details the formulation and evaluation of the proposed MCTD method, whereas the

Finer Feature Fusion algorithm is proposed and evaluated in Section 6.5. A summary is presented in Section 6.6.

6.1 Commercial 3D Acquisition Devices

Figure 6.1: The Kinect Sensor (left) used in this work. The Minolta VIVID 910 3D scanner (right) used in FRGC.

Table 6.1: Various 3D data acquisition devices.

Device	Speed (sec)	Charge Time	Size (inch^3)	Price (USD)	Acc. (mm)
3dMD	0.002	10 sec	423.9	>$50k	<0.2
Minolta	2.5	no	1408	>$50k	∼0.1
Artec Eva	0.063	no	160.8	>$20k	∼0.5
BLITZ	0.9	no	n/a	>$14k	∼0.2
3D3 HDI R1	1.3	no	n/a	>$10k	>0.3
SwissRanger	0.02	no	17.53	>$5k	∼10
DAVID SLS-1	2.4	no	n/a	>$2k	∼0.5
Kinect	0.033	no	41.25	<$200	∼1.5-50

A comparison of 3D acquisition devices is summarized in Table 6.1 (see also (Boehnen and Flynn, 2005)). 3dMD is designed specifically for instant 3D face acquisition and is used for medical applications. Although, it can capture a single scan in 2 milliseconds, it requires

10 seconds charging time prior to every scan. The VIVID 910 3D scanner from Konica Minolta was used to acquire 3D data in the well-known Face Recognition Grand Challenge (FRGC) (Phillips *et al.*, 2005). The SwissRanger SR4000 is a time of flight 3D scanner. Although it has faster capture time compared to the Kinect, its cost is relatively higher and its accuracy is much lower than Kinect at medium range. Table 6.1 shows that high resolution 3D scanners are generally expensive, slow in capturing time and bulky in size. On the other hand, the Kinect sensor is low cost, has high acquisition speed, no recharge time and is compact in size. More precisely, its size is about $11\times1.5\times2.5$=41.25 cubic inch and it weighs 1400 grams. It is available off-the-shelf and costs less than $200 USD. The size, weight and cost of Kinect can be further reduced if its multi-array microphone and motorized tilt are excluded since they are not necessary for face recognition.

6.2 Challenges of Kinect Data

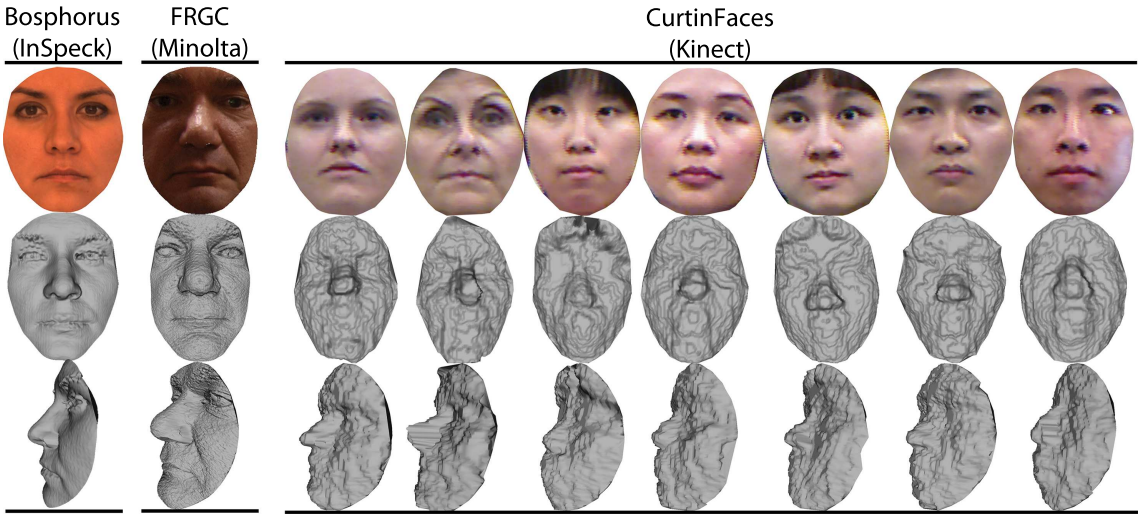


Figure 6.2: Texture and 3D face models acquired with Minolta Phillips *et al.* (2005), InSpeck Savran *et al.* (2008) and Kinect sensors. Top row: 3D faces with texture maps. Second and third row: 3D faces without texture rendered as smooth surfaces in MeshLab (Cignoni, 2012).

In terms of data acquisition, the Kinect sensor includes a standard RGB camera, an infra-red projector and an infra-red camera. The projector projects a static infra-red pattern on the scene (face in our case) which is sensed by the infra-red camera. This pattern is used to resolve correspondence between the projector and the camera, and depth is calculated using stereopsis (Khoshelham and Elberink, 2012). It is able to produce 640×480 range image (mapped with RGB texture from the standard camera) at 30 frames per second.

The depth accuracy of Kinect decreases dramatically from 1.5mm to 50mm when the object is further away form the camera. However, since we use Kinect at a distance of 1m, we are operating at a depth resolution very close to 1.5mm. Nevertheless, the 3D data acquired with the Kinect sensor is very noisy. Some sample face images acquired with the Kinect sensor are shown in Figure 6.2. The 3D faces rendered without texture are hardly recognizable as human faces. One of the main objectives of this chapter is to justify the feasibility of face recognition on such noisy Kinect data, so that we can take advantage of its high speed and low cost.

6.3 CurtinFaces database

This section gives details of the instruments used, the data acquired and the participating subjects during the creation of the CurtinFaces database. This database is available for download at `http://impca.curtin.edu.au/downloads/datasets.cfm`.

6.3.1 Instrument Setup

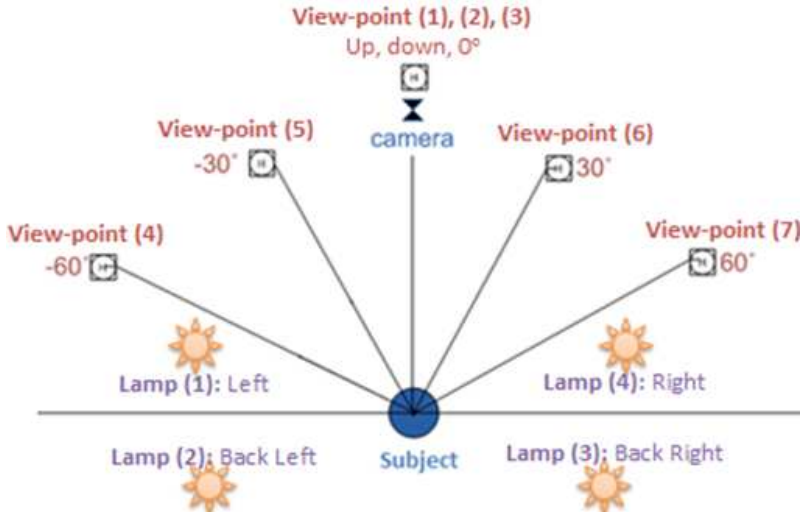


Figure 6.3: Instruments setup.

The main instruments involved were a Kinect sensor, a standard digital camera, five 18W fluorescent lamps and a standard desktop computer used to communicate with the Kinect. The Kinect sensor produces 640×480 RGB image and a depth map (range image) at a frame rate of about 30 per seconds. We have developed a program using the APIs provided

by OpenNI [1] to stream the video frames from Kinect, and store the required frames as still images in PNG format. A standard digital camera, the Lumix-DMC-FT1 model from Panasonic, was used to capture still RGB Images with 4000×3000 resolution (stored in JPEG format) for comparison with the RGB camera of the Kinect sensor. The position of the cameras and lamps are shown in Figure 6.3. The cameras and lamps were located about 1m from the floor and 1m from the subject, while the subject was asked to sit on a chair so that his/her face is about 0.5m from the floor.

6.3.2 Data Acquisition and Organization

Each subject was imaged under different combinations of seven facial expressions, seven poses, five illuminations and occlusions, resulting in a total of 97 variations. For each of these 97 conditions, we obtained two images where one was captured by Kinect sensor and the other was captured by the Panasonic digital camera at almost the same time.

The seven expressions are neutral, happy, disgust, anger, sad, surprise and fear. This expression set is widely used as a standard for research on facial expression recognition (Lucey *et al.*, 2010; Shan *et al.*, 2009). This expression set involves a rich set of facial Action Units (AU) (P. Ekman, 1978) and can, therefore, simulate wide range of facial variations in real application such as talking or laughing under surveillance. Although we do not consider expression recognition in this work, the CurtinFaces database can be used by other researchers for facial expression recognition.

The complete image capturing routine for one subject can be divided into four parts. Each subject was labeled with a unique subject ID (*sid*) while images of the same subject was labeled with a unique image ID (*mid*). A specific image in the database can be uniquely identified by the subject and image ID together, i.e. (sid, mid).

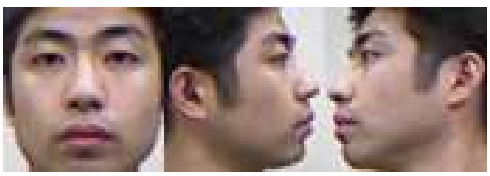


Figure 6.4: Sample images in part 1 of the CurtinFaces database, which contains three controlled shots.

In part 1, a total of 3 controlled shots were taken with frontal pose, 90 degree left profile and right profile, as shown in Figure 6.4. Only for this part, the subjects were required to

[1] We used the Matlab wrapper (`www.mathworks.com/matlabcentral/fileexchange/30242`).

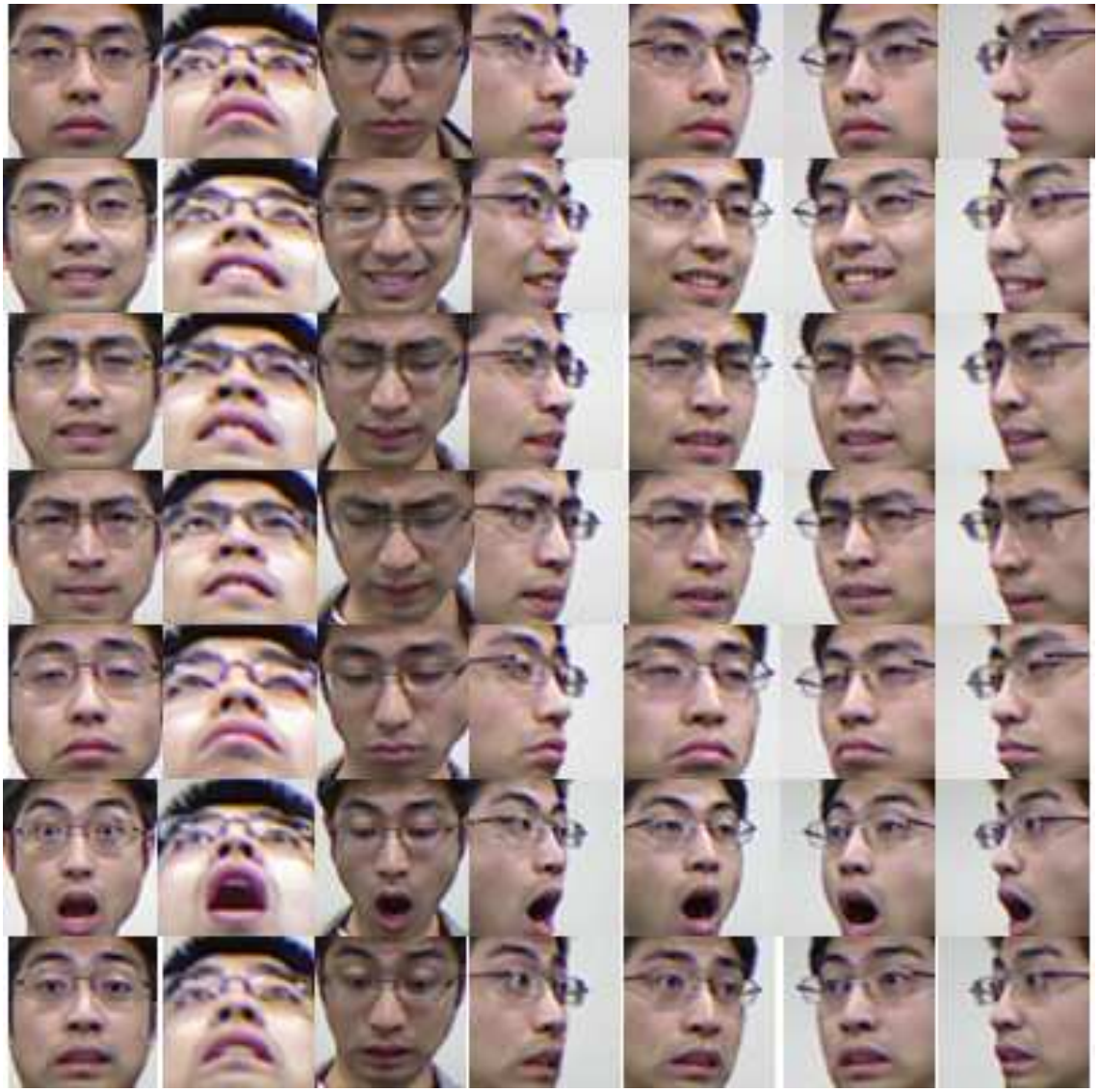


Figure 6.5: Sample images in part 2 of the CurtinFaces database, which contains variations in expression and pose.

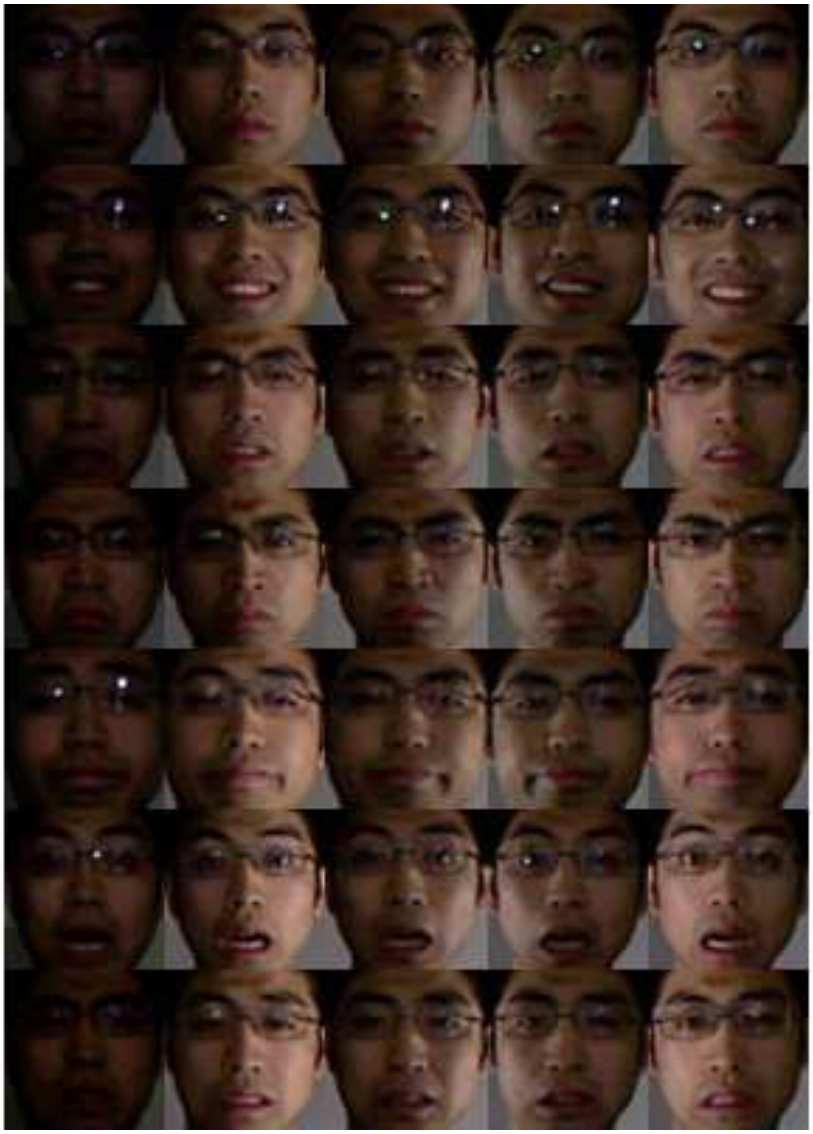


Figure 6.6: Sample images in part 3 of the CurtinFaces database, which contains variations in expression and illumination.

Figure 6.7: Sample images in part 4 of the CurtinFaces database, which contains occluded images.

take off their glasses in case they were wearing them. These three controlled shots can be used to reconstruct a complete 3D face model of the subject for 3D modeling applications. We labeled these three images with $(sid, 1)$, $(sid, 2)$ and $(sid, 3)$ respectively.

In part 2, images were acquired under the seven different expressions and the seven different poses resulting in a total of 49 images per subject (see Figure 6.5). The seven poses, as depicted in Figure 6.3, are 0^o-frontal, 45^o-up, 45^o-down, 60^o-left, 30^o-left, 30^o-right and 60^o-right, respectively. At first, the subject was requested to perform one of the 7 facial expressions. Then for each expression, the subject was requested to move their head to the specific pre-measured position for the different pose captures, while the camera stay still. Therefore, these 49 images were captured at different time. None of the lamps were turned on in this part, while the lab lights were kept on to simulate ambient lighting condition. The labeling of them was in order of poses, then expressions. Specifically, images in first row of Figure 6.5 were labeled with $(sid, 4)$ to $(sid, 10)$ from left to right, while images in second row had ID $(sid, 11)$ to $(sid, 17)$, etc, until $(sid, 52)$.

In part 3, images were acquired under the seven different expressions and five different illuminations resulting in a total of 35 images per subject (see Figure 6.6). The five different illuminations were generated by five fluorescent lamps positioned at different locations as shown in Figure 6.3. After turning off all laboratory lights, the fluorescent lamps were turned on one by one. Under each lighting condition, the subject was requested to perform one of the seven facial expressions. Thus each facial expression was captured under five different lighting conditions in the following sequence: low ambient lighting (from monitors etc.), left lamp, back left lamp, back right lamp and right lamp. Only frontal pose was considered in this part. Images in the first row of Figure 6.6 were labeled with $(sid, 53)$ to $(sid, 57)$ from left to right, while images in second row had ID $(sid, 58)$ to $(sid, 62)$, etc, until $(sid, 87)$.

In part 4, images were captured under two types of occlusions. For each type, 5 conditions were considered: 3 different poses and 2 different lightings, respectively. There were a total of 10 images as shown in Figure 6.7. The first type of occlusion was designed to cover the eyes. Subjects were requested to put on a sunglass. Then with the lab lights switched on but none of the other lamps, the subject was requested to move his/her head to 0^o-frontal, 30^o-left, and 60^o-right. Next, we switched off the lab lights, with the subject looking frontal, we turned on only the left lamp and then only the right lamp. The second type of occlusion was designed to cover around the mouth. Subject was requested to take off the sunglass and cover their mouth with his/her right hand. Then the same 3 poses, 2 lightings procedures were executed. Images in first row of Figure 6.7 were labeled with $(sid, 88)$ to $(sid, 92)$ from left to right, while images in second row had ID $(sid, 93)$ to $(sid, 97)$.

6.3.3 Subjects Detail

The participating subjects were students and staff members from our university and represent different demographics. The participants represent different races including Caucasians, Chinese and Indians. There were 10 females and 42 males. The ages of the participants ranged from 20 to 60 years. 21 of the subjects were also wearing glasses. Subjects were labeled with subject IDs in order.

6.4 Face Recognition using MCTD

In this section, we propose an algorithm that uses only the colored 2D data Kinect provided for robust face recognition. The proposed algorithm namely the Multilinear Color Tensor Discriminant (MCTD) model, integrates two state-of-the-art 2D methods, i.e. the MPCA-PS (Rana *et al.*, 2009) and TDCS (Wang *et al.*, 2011). As a result, MCTD compensates their weakness and retain their strength. This method is composed of novel image representation, color feature extraction and multilinear based classifier. Each component will be detailed in this section after a brief review on MPCA-PS and TDCS.

6.4.1 MPCA-PS and TDCS

MPCA-PS and TDCS are introduced in Section 2.3.2 and Section 2.5.5. A brief review is given here for sake of completeness and contrasting purpose.

MPCA-PS uses a single tensor to represent the whole training data. Assumes that the training data contains samples of N_p people with N_l lighting conditions being captured under N_v viewpoints. MPCA-PS represents all of them as a single fourth-order tensor T:

$$T \in \mathbb{R}^{N_p \times N_l \times N_v \times N_x}, \tag{6.1}$$

where

$$T(i_p, i_l, i_v) \in \mathbb{R}^{N_x}, \tag{6.2}$$

denotes an image vector for the i_p-th persion at i_l-th lighting condition and i_v-th viewpoint with N_x pixels. In conventional face identification, a query image q must be one of the people in training set T. With this assumption, MPCA-PS classifies a query q to be a person k with the minimum value of the following optimization problem:

$$\min_{k,u_l,u_v} \|q - C \times_1 u_p^k \times_2 u_l \times_3 u_v \times_4 U_X\|_2, \tag{6.3}$$

where C is the core tensor, u_p^k is the k-th row of U_p, for $k = 1, \ldots, N_p$. u_l and u_v are two free variables used to reconstruct the lighting and viewpoint modes respectively.

The main idea of MPCA-PS is to create the core tensor C as a multilinear principal subspace, such that any face image can be reconstructed approximately using C from different lighting and pose perspective. Therefore, a query image can be classified as person k, by finding the corresponding coefficients: u_p^k in U_p, u_l and u_v in Eq. 6.3 that yield the least reconstruction error. This approach generalize well to deal with variation factors that are unseen in the training set. However, it is designed only for gray-scale images. How to reformulate it for color images is an open problem.

TDCS represents each color image as a tensor. Thus, if there are N RGB training images of size $I_1 \times I_2$, they will be represented by N tensors $\{A_i \in \mathbb{R}^{I_1 \times I_2 \times I_3}\}$, for $i = 1, \ldots, N$ and $I_3 = 3$. Each A_i can be transformed into its feature tensor D_i as follows:

$$D_i = A_i \times_1 W_1^T \times_2 W_2^T \times_3 W_3^T, \tag{6.4}$$

where W_1, W_2 and W_3 are projection matrices for row-mode, column-mode and color-mode respectively. They are solved to maximize the classes separability in D-space, i.e.:

$$\max_{W_1,W_2,W_3} \quad \frac{tr(S_b)}{tr(S_w)}. \tag{6.5}$$

where, S_b and S_w are the between and within class scatter matrices in D-space. In (Wang *et al.*, 2011), an iterative solution is proposed to solve W_1, W_2 and W_3 alternatively. Finally, a query image is projected to TDCS and the Nearest Neighbor (NN) classifier is used for classification.

The main idea of TDCS model is to extract features in row space, column space and color space with an aim of achieving the maximum class discrimination. In fact the factors of lighting and pose variations are not considered properly.

In summary, there are two significant differences between MPCA and TDCS. The first one is their data representation: MPCA-PS represents each image as a vector and organizes all training image vectors as a single tensor in Eq. 6.2, while TDCS model represents each image as one individual tensor as in Eq. 6.4 instead of converting them to vector. The second difference is their objectives: MPCA-PS focuses on creating a good core tensor such that a query image can be reconstructed with the least error, while TDCS focuses on extracting discriminative features such that the maximum class separability can be achieved. The first drawback of MPCA-PS is that it converts each facial image to vector mechanically, and this may discard some important facial structure information. Secondly, MPCA-PS applies the classical Eigenface approach on the pixel mode, which does not maximize class separability, hence lacking discrimination power. Lastly, MPCA-PS is designed to work only on gray-scale images, thus it loses the advantage of color information. On the other hand, the main disadvantage for TDCS model is that it does not consider factors like pose and illumination properly, hence lacking robustness when dealing with uncontrolled query images with large pose and lighting variations.

6.4.2 The Proposed MCTD Model

Based on above observations, we propose the Multilinear Color Tensor Discriminant (MCTD) model in this section, which integrates MPCA-PS and TDCS as an novel representation for training images. With this new framework, we can extract powerful discriminant features based on color information and also can handle pose and lighting variations properly. This will result in a multilinear based classifier for robust color face recognition.

6.4.2.1 Image Representation

An important issue of the proposed MCTD model is its new data representation. In detail, we first represent each color image as a tensor, to retain its underlying matrix structure, and then organize all training image tensors into one single tensor for multilinear analysis. As a result, N RGB training images can be represented by a tensor M:

$$M \in \mathbb{R}^{N_p \times N_l \times N_v \times I_1 \times I_2 \times I_3}, \tag{6.6}$$

where $N_p \times N_l \times N_v = N$ and each of the N sub-tensors

$$M(i_p, i_l, i_v) \in \mathbb{R}^{I_1 \times I_2 \times I_3} \tag{6.7}$$

denotes a color image, where I_1 is the number of rows, I_2 is the number of columns and $I_3 = 3$ is the number of color components. In contrast to the image $T(i_p, i_l, i_v)$ in Eq. 6.2 which is converted to a vector, $M(i_p, i_l, i_v)$ is a color image represented naturally as a 3-rd order tensor. Consequently, the mean image of the i_p-th person is defined as

$$\overline{M}^{i_p} = \frac{1}{N_l \times N_v} \sum_{i_l=1}^{N_l} \sum_{i_v=1}^{N_v} M(i_p, i_l, i_v), \tag{6.8}$$

and the total mean image of all images is computed as

$$\overline{M} = \frac{1}{N} \sum_{i_p=1}^{N} \overline{M}^{i_p}. \tag{6.9}$$

6.4.2.2 Feature Extraction

In order to extract the discriminant features from M, while preserving its matrix structure, we need to find two discriminant projection matrices $W_1 \in \mathbb{R}^{I_1 \times I_1'}$, $W_2 \in \mathbb{R}^{I_2 \times I_2'}$ and a color space transformation matrix $W_3 \in \mathbb{R}^{I_3 \times I_3'}$ (usually $I_1' < I_1$, $I_2' < I_2$ and $I_3' \leq I_3$), and obtain the feature tensor $F \in \mathbb{R}^{N_p \times N_l \times N_v \times I_1' \times I_2' \times I_3'}$:

$$F(i_p, i_l, i_v) = M(i_p, i_l, i_v) \times_1 W_1^T \times_2 W_2^T \times_3 W_3^T. \tag{6.10}$$

Instead of following the procedure in (Wang *et al.*, 2011) directly, we need to consider the pose and lighting variations here. For such purpose, let $T_{(n)}$ denote mode-n tensor unfolding, $\|T\|$ denote the Frobenius norm for a tensor and $tr(.)$ denote the trace of a matrix. Then the n-mode between-class scatter matrix $\Psi_b^{(n)}$ in the feature space can be

computed as follows:

$$\begin{aligned}
\Psi_b^{(n)} &= \Sigma_{i_p=1}^{N_p} \|\overline{F}^{i_p} - \overline{F}\|^2_{(n)} \\
&= \Sigma_{i_p=1}^{N_p} \|(\overline{M}^{i_p} - \overline{M}) \times_1 W_1^T \times_2 W_2^T \times_3 W_3^T\|^2_{(n)} \\
&= \Sigma_{i_p=1}^{N_p} \|W_n^T(\overline{M}_{(n)}^{i_p} - \overline{M}_{(n)})\tilde{W}_n\| \\
&= \Sigma_{i_p=1}^{N_p} tr\Big[W_n^T(\overline{M}_{(n)}^{i_p} - \overline{M}_{(n)})\tilde{W}_n\tilde{W}_n^T \\
&\qquad\qquad \times (\overline{M}_{(n)}^{i_p} - \overline{M}_{(n)})^T W_n\Big] \\
&= tr(W_n^T S_b^{(n)} W_n),
\end{aligned} \tag{6.11}$$

where $\tilde{W}_n = W_d \otimes \cdots \otimes W_{n+1} \otimes W_{n-1} \otimes \cdots \otimes W_1, n = 1, 2, \ldots, d$ and $d = 3$, while $\otimes$ denotes the Kronecker product. Similarly, the n-mode within-class scatter matrix $\Psi_w^{(n)}$ in the feature space can be defined as :

$$\begin{aligned}
\Psi_w^{(n)} &= \Sigma_{i_p=1}^{N_p}\Sigma_{i_l=1}^{N_l}\Sigma_{i_v=1}^{N_v} \|F(i_p, i_l, i_v) - \overline{F}^{i_p}\|^2_{(n)} \\
&= tr\Bigg\{W_n^T\Big[\Sigma_{i_p=1}^{N_p}\Sigma_{i_l=1}^{N_l}\Sigma_{i_v=1}^{N_v} \\
&\qquad (M(i_p, i_l, i_v)_{(n)} - \overline{M}_{(n)}^{i_p})\tilde{W}_n\tilde{W}_n^T \\
&\qquad \times (M(i_p, i_l, i_v)_{(n)} - \overline{A}_{(n)}^{i_p})^T\Big]W_n\Bigg\} \\
&= tr(W_n^T S_w^{(n)} W_n).
\end{aligned} \tag{6.12}$$

The objective here is to maximize the class separability in the feature space, hence defining the following MCTD criterion:

$$\max J(W_n) = \frac{\Psi_b^{(n)}}{\Psi_w^{(n)}} = \frac{tr(W_n^T S_b^{(n)} W_n)}{tr(W_n^T S_w^{(n)} W_n)}, (n = 1, 2, 3). \tag{6.13}$$

One can see that Eq. 6.13 consists of three variables, it can be optimized alternatively by solving its corresponding generalized eigenvalue decomposition problem while fixing any two of the variables. In fact, the approach for obtaining solutions W_i is actually the same as in (Wang *et al.*, 2011). However the representation will be useful for the classification in next step.

6.4.2.3 Classification

To harness the power of multilinear analysis, we use the multilinear technique on the feature tensors F in Eq. 6.10 to create a core tensor C. In this case, F can be reconstructed

as

$$F \approx C \times_1 U^P \times_2 U^L \times_3 U^V \times_4 U^x \times_5 U^y \times_6 U^z, \tag{6.14}$$

where C is the core tensor computed by Eq. 2.17, U^P, U^L and U^V are the eigen-person, eigen-lighting and eigen-viewpoint, respectively. Compared to the eigen-pixel U^x in MPCA-PS, here we have three items U^x, U^y and U^z. They are the eigen-row-feature, eigen-column-feature and eigen-DCS, respectively. One can see that the images here are still in tensor form without conversion to vectors.

Now given a RGB query image represented as a third-order tensor $Q \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, we first project Q to the feature space with W_n and obtain $Q' \in \mathbb{R}^{I'_1 \times I'_2 \times I'_3}$.

Refereing to Eq. 6.14, let u_p^k denotes the k-th row of U_p and $B_k = C \times_1 u_p^k \times_4 U^x \times_5 U^y \times_6 U^z$. We classify Q as person k by solving the following:

$$\min_{k,u_l,u_v} \|Q' - B_k \times_2 u_l \times_3 u_v\|_2. \tag{6.15}$$

Let $B(k, e_l, e_v) \in \mathbb{R}^{I'_1 \times I'_2 \times I'_3}$ denote the image-feature tensor of the k-th person at e_l-th eigen-lighting and e_v-th eigen-viewpoint. Following the rule of mode multiplication, we can rewrite the following term as matrix multiplication:

$$\begin{aligned} &B_k \times_2 u_l \times_3 u_v \\ &= \Sigma_{e_v}^{N'_v} u_v(e_v) \Sigma_{e_l}^{N'_l} u_l(e_l) \; . \; B(k, e_l, e_v) \equiv g^k \; . \; E_k, \end{aligned} \tag{6.16}$$

where $g^k \in \mathbb{R}^{N'_l N'_v}$ is a row vector of combined coefficients for person k, and $E_k \in \mathbb{R}^{(N'_l . N'_v) \times (I'_1 . I'_2 . I'_3)}$ is a matrix that each of its rows is obtained by concatenating all feature tensors of the k-th person $B(k, 1, \ldots, N'_l, 1, \ldots, N'_v)$ to an augmented row vector. Since Eq. 6.15 is minimized when $Q' = (g^k . E_k)$, thus we can derive the least square solution for g^k as follows:

$$g^k = Q' \; . \; E_k^+, \tag{6.17}$$

where $(.)^+$ denote the Moore-Penrose pseudoinverse operator.

Ultimately, we can classify Q as person k by solving:

$$\min_k \|Q' - g^k \; . \; E_k\|_2. \tag{6.18}$$

6.4.3 Evaluation on MCTD

In this section, we evaluate MCTD on both CMU-PIE and CurtinFaces databases. All images used here are cropped to 32×32 with eyes and mouth aligned to same locations.

PIE (Sim *et al.*, 2001) dataset contains color face images of 68 subjects. The "illumination" set, which has 273 images (at 13 poses $\times$ 21 lighting conditions) for each subject (subject 04039 is excluded from our test), is used for our experiments. The four experiment sets consist of different number of randomly selected training samples (pose $\times$ lighting) per subject, which are: 15(3×5), 60(6×10), 135(9×15) and 209(11×19) respectively, while the remaining images are used for testing.

CurtinFaces, as described previously, consists of 2D and 3D face samples with large variation and disguise. Only a subset of 2D data is used to evaluate MCTD, whereas the 3D data is ignored here. This subset consists of 4368 color images of 52 individuals with variations in poses(V), lighting(L) conditions and facial expressions(E). Each individual has a total of 84 images, where 49 images are composed of $7V \times 7E$ and 35 images are composed of $5L \times 7E$. Four experiment training sets are constructed by randomly selecting images of: $5V \times 5E$, $6V \times 6E$, $3L \times 5E$ and $4L \times 6E$, per subject, respectively and using the rest for testing.

Four methods are implemented here: PCA+NN(Gray) (Turk and Pentland, 1991), MPCA-PS(Gray) (Rana *et al.*, 2009), TDCS(color) (Wang *et al.*, 2011) and MCTD(Proposed). For methods that involving PCA or MPCA, the numbers of eigenvector retained are chosen to preserve 99% of energy. Following (Wang *et al.*, 2011), for TDCS and MCTD feature extraction, 10 row features, 10 column features and 3 discriminant color combinations are retained, with iteration-stop threshold set to 0.1. For each of the eight experiments, the rank-one identification rate (Biometrics, 2006) averaged over 25 repetitions and the corresponding standard-deviation (rate%$\pm$std) are reported in Table 6.2.

Table 6.2: Recognition rates (% $\pm$ std)

PIE database				
Training	PCA+NN	MPCA-PS	TDCS	MCTD
15	36.3 ± 5	46.5 ± 7	46.4 ± 7	$\mathbf{48.0} \pm 8$
60	56.8 ± 5	65.6 ± 7	69.2 ± 5	$\mathbf{73.3} \pm 7$
135	70.3 ± 5	75.3 ± 6	75.3 ± 4	$\mathbf{85.4} \pm 6$
209	70.9 ± 10	76.7 ± 10	77.0 ± 6	$\mathbf{88.6} \pm 10$
CurtinFaces database				
Training	PCA+NN	MPCA-PS	TDCS	MCTD
$5V \times 5E$	52.2 ± 6	56.7 ± 7	74.1 ± 9	$\mathbf{78.2} \pm 10$
$6V \times 6E$	55.4 ± 10	57.9 ± 4	74.7 ± 2	$\mathbf{79.0} \pm 2$
$3L \times 5E$	64.8 ± 3	72.9 ± 2	88.4 ± 3	$\mathbf{90.1} \pm 4$
$4L \times 6E$	80.2 ± 17	88.2 ± 19	97.6 ± 17	$\mathbf{98.5} \pm 16$

From Table 6.2, one can observe that the proposed MCTD model always achieve the best results in comparison with all other approaches in all cases. For PIE database, the identification rate increases with increasing number of training samples, which demonstrates consistency in our experimental setup. The identification rate in CurtinFaces database is lower with pose variations than with lighting variations, which shows that the pose problem is more challenging. Note that for all methods, the standard deviations can be as high as 19%, however, this is reasonable in the situation of having random unseen factors in the testing sets. Despite having large deviation, the proposed MCTD model achieves comparable stability to others. Moreover, both MPCA-PS and TDCS outperform the baseline PCA, which once again demonstrates the power of multilinear analysis and color information. Nevertheless, MPCA-PS is not compared to MPCA since MPCA-PS is proven to be better in Rana *et al.* (2009).

As all 8 experiments are repeated 25 times, there are a total of 200 single evaluations. The proposed MCTD consistently outperforms both MPCA-PS and TDCS, proving that it utilizes the advantage of both methods. Therefore we are confident to conclude that, MCTD model achieves the new state-of-the-art performance for robust face identification using only 2D data that involve large and unseen factor variations.

While MCTD seems to be a reasonable solution to robust face recognition problem, it is not completely reliable, as it sometime has large standard deviation. Its performance drops dramatically especially when more unseen poses appear in the test set. For example, its performance decreases to 48% on PIE when only 15 images per person is used for training and the rest for testing. Similarly in CurtinFaces, the lowest recognition rate is observed for the training set $5V \times 5E$, where the corresponding test set consists of two unseen poses. In next section, we propose an algorithm that also makes use of Kinect depth data. We expect that the Kinect depth data can improve recognition robustness even though it is very noisy.

6.5 Face Recognition using FFF

Similar to most 3D scanners, the Kinect provides both RGB texture image and depth map of the scene at the same time. In terms of data format, there are two common options namely range image or real world coordinates. The range image is similar to an intensity image except that each pixel value represents the depth of the scene point measured from the camera. Using the camera intrinsic parameters, the depth map can be converted to real world coordinates comprising the x, y and z coordinates of the pixels. These are

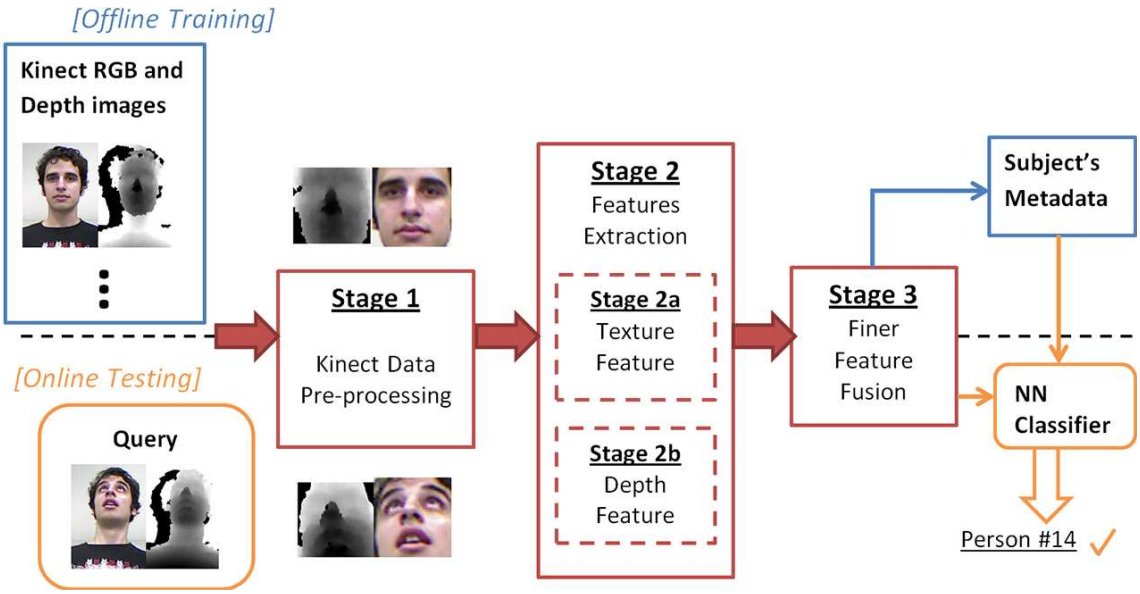


Figure 6.8: The proposed framework.

also referred to as the "point cloud". Using the OpenNI API, it is possible to obtain Kinect data in both formats. We use the range image format in our experiments but the CurtinFaces database contains both formats.

As mentioned, the depth data provided by Kinect has a relatively lower accuracy. In order to show the effectiveness of Kinect data for face recognition, we propose an effective algorithm based on Kinect depth map. An overview of the framework is depicted in Figure 6.8. The framework can be divided into three stages: data preprocessing, feature extraction and finer feature fusion. Finally, the Nearest Neighbor (NN) rule is used for classification.

6.5.1 Stage 1: Data Preprocessing

Data preprocessing is the first stage of our framework. The steps involved are: calibration, face cropping, face alignment, down-sampling and normalization. In particular, we calibrate the RGB and depth image using the parameters provided by ROS.org. After that, the eyes, nose and mouth are located. Then the face image and depth map are cropped with the facial components aligned. Similar to previous chapters, we follow a tradition alignment procedure which first manually locates the two eyes and the lip on the image, then transits and scales the image and depth map such that these three identified points are aligned to the same pixel. They are then downsampled to 32×32. Finally, the depth

map is normalized such that the closest pixel has value 0 (which is usually the nose tip for frontal faces).

Camera calibration is required since Kinect captures RGB and depth image using different cameras. Despite that both images are 640×480, their pixels cannot be directly mapped to each other. This is because the RGB and depth cameras have different extrinsic and intrinsic parameters.

Facial components alignment is an important preprocessing step for the success of appearance based template matching methods such as LDA. Images of the same face under different poses have very different facial component locations which can lead to the failure of face recognition methods. Some 3D face recognition methods align poses in 3D space based on their 3D model (Wang *et al.*, 2010; Spreeuwers, 2011). However, the quality of the data provided by Kinect is not high enough to benefit form such a 3D alignment approach. For the sake of simplicity and efficiency, we adopt a 2D based face alignment approach. Ultimately, our objective is just to show the effectiveness of Kinect data for face recognition.

6.5.2 Stage 2: Feature Extraction

In stage 2, various useful features are extracted from both the preprocessed RGB and range images. In particular, the DCS (Discriminant Color Space) (Yang *et al.*, 2010b) method is applied to the RGB image whereas three different shape descriptors are used to represent the depth map. These shape descriptors are LBP (Local Binary Patterns) (Ahonen *et al.*, 2004), Haar (Viola and Jones, 2001) features and Gabor features (Liu and Wechsler, 2002). Details of the features are given below.

(Stage 2a) Texture Feature: The RGB image captured by Kinect has the same structure as any other standard camera. Therefore, color face recognition algorithms can be applied directly to extract color face features. As detailed in Section 2.5.3, the Discriminant Color Space (DCS) (Yang *et al.*, 2010b) method is one of the simplest, yet efficient and robust approaches. It seeks 3 linear combinations of R, G and B color components to transfer the image from RGB color space to DCS, such that the within class distance is minimized while the between class distance is maximized.

To extract *DCS-LDA features*, the image is first converted from RGB to DCS format. Each of the 3 DCS color components are then normalized to zero mean and unit standard deviation to avoid variance domination in one component over the others. After

concatenating the 3 DCS color components into an augmented vector, LDA is applied.

(Stage 2b) Depth Feature: The raw depth map itself provides very little discriminative information due to its low quality. However, we can extract useful shape cues from it. Wang *et al.* (2010) showed that the Local Binary Patterns (LBP), Haar-like features and Gabor features provide complementary discriminative shape cues for face recognition. Tenllado *et al.* (2010) proved that the original image and its Gabor filtered images also exhibit complementary information. In our framework, we adopt four different shape representations: Range, LBP, Haar and Gabor features. After transforming to these representations, a classical LDA is employed for feature extraction. This step is important, since the dimension for Haar and Gabor representations are too high. More importantly, as we are considering the context of robust face recognition, we need to transform features into a more discriminant space which is robust to intra-class variations.

To extract *Range-LDA features*, the preprocessed range image is scaled from 0 to 255 and stored as a gray-scale image [2]. The range image is converted into a vector form by concatenating its columns before the application of LDA.

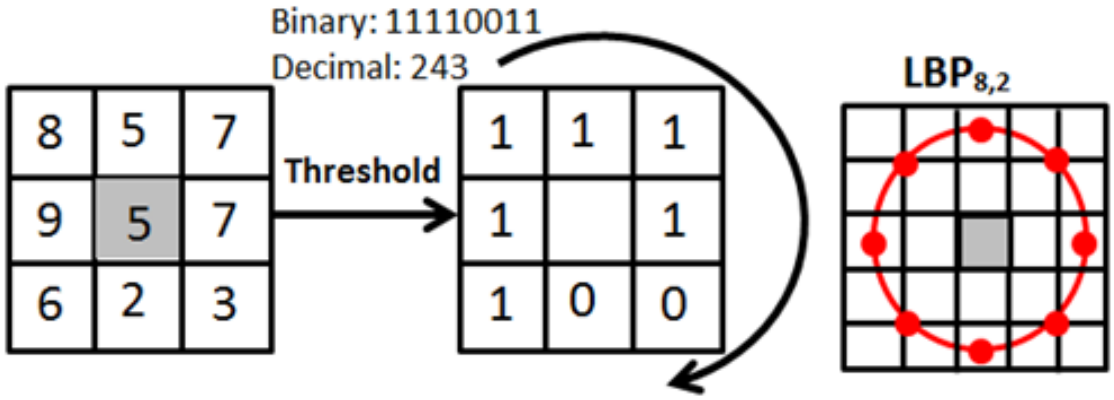


Figure 6.9: The LBP operators. (Left) Gray-scale image block. (Middle) LBP representation. (Right) Extended LBP operator with 8 samples on a circle of radius 2.

To extract *LBP-LDA features*, LBP is applied on the original range image scaled from 0 to 255. LBP is a powerful descriptor for face recognition and we use its improved version (Ahonen *et al.*, 2004) which computes the representation on image patches. It uses circular neighborhoods and emphasizes on uniform patterns. An LBP is uniform if there are at most two bitwise transitions from 0 to 1 or vice versa. For the neighbor pixels selection, 8 pixels are sampled on a circle of radius 2 pixels as shown in Figure 6.9. For histogram computation, instead of computing one for the whole image, the image is divided into 16

[2]Although more than 10 bits are used to store the original depth data, 256 intensity levels is enough to retain all the information as most of the faces only have around 50 mm depth ranges.

non-overlapping patches of 8×8 each and local histograms are computed for each local patch. There are a total of 58 LBPs that is uniform and we use a 59-bin histogram for each patch, where the extra bin stores all non-uniform LBPs. By concatenating all local histograms, we obtain an LBP representation for the 32×32 range image as a vector of 944 dimensions (59×16). Finally, LDA is applied for dimensionality reduction.

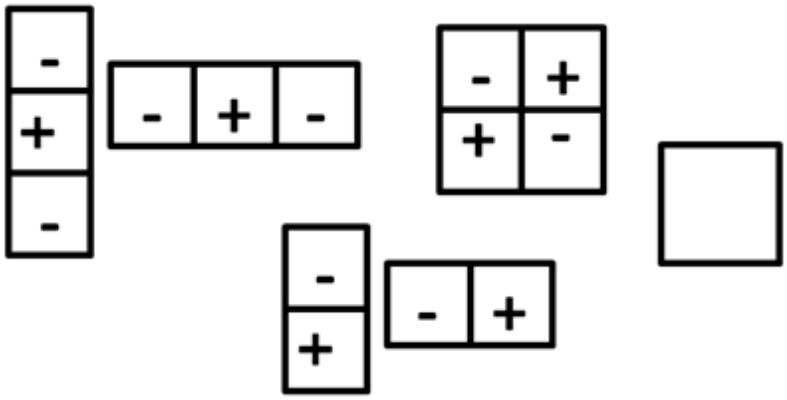


Figure 6.10: The Haar operators. The square on the right denotes the average of a region.

To extract *Haar-LDA features*, the Haar operators are applied on the range image. Haar features effectively describe the local shape differences. The Haar operators we used are shown in Figure 6.10. These operators have multiple possible sizes which may lead to a very high dimensional feature vector. Therefore, we only use squares or rectangles with width and height equal to 6 or 7 pixels. This is a typical size of a facial component (e.g. eyes and mouth) in a 32×32 range image. By combining the output of all operators, we obtain a Haar feature vector of 29230 dimensions. We use LDA to reduce the dimensionality of this feature.

To extract *Gabor-LDA features*, Gabor filters with different scales and orientations are applied on the range image to capture the spatial locality characteristic. The Gabor kernels are defined as

$$\Psi_{u,v}(z) = \frac{\|k_{u,v}\|^2}{\sigma^2} e^{-\|k_{u,v}\|^2 \|z\|^2/(2\sigma^2)} (e^{ik_{u,v}z} - e^{-\sigma^2/2}), \tag{6.19}$$

where $z = (x, y)$, $\|\cdot\|$ is the norm operator, u and v is the orientation and scale respectively. $k_{u,v}$ is the wave vector defined as

$$k_{u,v} = \frac{k_{max}}{f^v} e^{i\pi u/8}, \tag{6.20}$$

where f^v is the spacing factor between kernels and k_{max} is the maximum frequency. We

extract Gabor features at 8 orientations and 5 scales. Each Gabor output is downsampled to 16×16 to avoid the high dimensionality and normalized to zero mean with unit standard deviation. By combining these downsampled outputs, we obtain a Gabor representation vector of 10240 ($40 \times 16 \times 16$) dimensions which is then fed to LDA for dimensionality reduction.

Note that in the case of each feature, before applying LDA, PCA is used to reduce the dimensionality of the feature vectors by retaining 99% of the energy. LDA further reduces the dimensions of all individual features to 51.

6.5.3 Stage 3: Finer Feature Fusion

After obtaining Range-LDA, LBP-LDA, Haar-LDA, Gabor-LDA and DCS-LDA features, an effective fusion mechanism is required, as these features have their own weaknesses and advantages. A feature-level fusion mechanism can be used. We can first normalize each feature vector and then concatenate them into one augmented vector for Nearest Neighbor (NN) classification. However, since LBP, Haar and Gabor decode shape cues, the augmented feature vector is expected to carry redundant information. Performing NN classification on this feature vector, with high dimensionality and redundancy, is ineffective.

Based on this analysis, we propose a novel fusion strategy by applying LDA one more time to extract finer LDA features. Specifically, each type of feature is normalized to zero mean and unit standard deviation. Then they are concatenated to form an augmented feature vector of size ($51 \times 5 = 255$) for the subsequent application of LDA.

There are two advantages of this fusion mechanism. Firstly, the subsequent LDA further reduces the dimension of the augmented feature vector and therefore, removes redundant information. Secondly, LDA extracts features such that they maximize class separability. Therefore, if the training samples cover sufficient variations such as varying poses, illuminations and expressions, then a subsequent application of LDA will extract finer features that are more robust to various imaging conditions.

6.5.4 Evaluation on FFF

In this section, we propose a standard training and test set for the CurtinFaces database. We will evaluate the proposed framework using these standard partitions. We aim to demonstrate the effectiveness of kinect data for face recognition and to report benchmark performances on this newly acquired data set. Although the CurtinFaces database consists of two sets of images, which are captured by Kinect sensor and Panasonic[3] camera respectively, we only consider Kinect data in this work. For all experiments that involve LDA feature extraction, the dimension is set to be 51 (which is the maximum).

6.5.4.1 Data Partitioning and Performance Evaluation

We propose a standard training set for CurtinFaces and use it in all our experiments in this section. Nevertheless, the effect of different training sizes is also examined later. A fair training set should contain sufficient images to cover a reasonable range of variations. Therefore, we select a subset of the database, containing 18 images per subject. More precisely, we select 7 viewpoints, 6 facial expressions and 5 lighting conditions without interaction between these factors i.e., when one factor changes, the other two factors remain fixed. The IDs of the training images are 4 to 11, 18, 25, 32, 39, 46 and 53 to 57. Thus the total number of training images are 936. Moreover, since the Kinect camera can capture video with a frame rate of 30fps, collection of this training set is feasible in real applications during the enrollment stage with the help of a suitable flashing system.

The testing set used for all experiments in this work includes all other unoccluded images that are not used for training. There are a total of 60 images per subject involving 6×6 different viewpoints$\times$expressions and 5×6 different lightings$\times$expressions. Precisely, the IDs of the test images are 4 to 87 excluding the IDs appeared in training set. Thus in overall we have a total of 3120 test images. Note that we do not use images 1,2 and 3 for training or testing. However, other researchers can consider their use for profile face recognition.

CurtinFaces is more suitable for the analysis of face identification, such as surveillance or access control systems. This problem requires the system to identify the identity of the query image from the database. The rank-one identification rate on the Cumulative Match Curve (CMC) is used as the performance benchmark. CMC plots the correct identification rate against the rank. Rank-one means only choosing one identity, which is usually the

[3]The Lumix-DMC-FT1 model digital camera.

requirement for most of the face identification systems. Rank-k means choosing k identities which means to allow certain false accept rate. Since there are 52 subjects in CurtinFaces, thus the maximum value for k is 52.

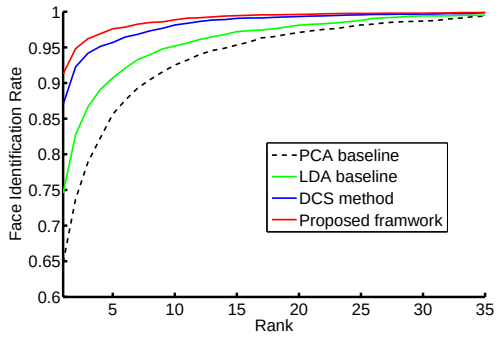


Figure 6.11: The CMC curve up to rank-35 for PCA, LDA, DCS and the proposed framework.

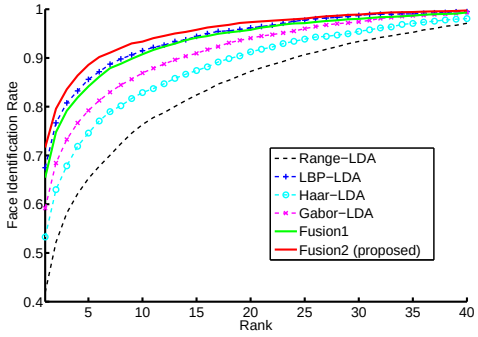


Figure 6.12: The CMC curve up to rank-35 for various shape features and fusion mechanisms.

Table 6.3: Rank-1 identification rates (%) for the proposed framework with different color spaces. The notation "+Depth" in the second row denotes the inclusion of the four shape features.

	Gray	RGB	nRGB	$I_1I_2I_3$	DCS
Texture only	74.8	82.0	85.6	86.9	87.0
Texture+Depth	82.3	87.2	89.7	90.0	91.3
Improvement	7.5	5.1	4.1	3.1	4.3

6.5.4.2 Benchmark Performance Overview

In this experiment, we aim to show some benchmark performances. In order to examine the difficulty level of CurtinFaces, we implemented the EigenFace (PCA) (Turk and Pentland, 1991) and FisherFace (LDA) (Belhumeur *et al.*, 1997) algorithms as the baselines. Both algorithms are based on gray-scale images. Nevertheless, the proposed framework is also evaluated against the DCS (Yang *et al.*, 2010b) method on color images, which is closely related to our framework. The corresponding CMC curve is plotted in Figure 6.11.

In terms of rank-1 identification rate, the PCA method achieves only 64.8% showing the difficulty level of this data set. Although LDA achieves 74.6%, this accuracy is still unacceptable for many applications. The proposed framework achieves 91.3%, which improves the baselines by a large magnitude. By excluding depth data from our framework, the

Table 6.4: Rank-1 identification rates (%) using different training sizes.

	Number of training images per subject				
	6	9	12	15	18
LDA baseline	44.5 ± 4.4	55.2 ± 4.0	62.9 ± 2.4	69.1 ± 2.2	74.6
RGB+LDA	48.6 ± 4.9	61.0 ± 5.1	70.0 ± 3.7	76.8 ± 2.5	82.0
DCS method	57.1 ± 6.1	70.2 ± 5.3	78.4 ± 3.4	82.7 ± 3.0	87.0
Proposed	62.1 ± 5.0	74.3 ± 5.6	82.5 ± 3.3	86.4 ± 3.7	91.3
Improvement over DCS	5.0	4.1	4.1	3.8	4.3

Table 6.5: Recognition time in milliseconds for the complete testing set and a single query (average time).

	Whole Set	Single Query
Preprocessing	175.5ms	0.051 ms
Range image	25.0 ms	0.007 ms
LBP	2299.9 ms	0.670 ms
Haar	4529.3 ms	1.320 ms
Gabor	34172.6 ms	9.957 ms
DCS	233.2 ms	0.080 ms
Fine features	31.0 ms	0.009 ms
NN Classification	75.5 ms	0.022 ms
Total	41466.5 ms	12.082 ms

DCS method itself achieves 87% by using texture image only. As a result, the inclusion of depth data in our framework increases the accuracy by 4.3% over the texture data, justifying its importance. On the CMC curve, our proposed framework is consistently on top of the others, while the performance of all methods converge at around rank-35. These results reveal the difficulty level of face recognition on CurtinFaces and set the current state-of-art performance for it.

6.5.4.3 Recognition with Kinect Depth Map

In this experiment, we perform face recognition using the depth map only. No texture information is involved. In order to justify the effectiveness of our proposed usage of depth data, we investigated the performance of individual shape features and two different fusion mechanisms. As described in Section 6.5.2, the shape features are *Range-LDA* , *LBP-LDA*, *Haar-LDA* and *Gabor-LDA*. We also implemented the common feature-level fusion mechanism mentioned in Section 6.5.3, which is denoted as Funsion1 in Figure 6.12. This method simply concatenates each normalized feature and then performs NN classification directly on the augmented feature. Our proposed mechanism that applies LDA on the augmented feature vector for *finer feature fusion (FFF)* is denoted as Funsion2.

As shown in Figure 6.12, applying LDA on the original range image achieves only 41.9% rank-1 accuracy, but after fusing all shape features by Fusion2, it becomes 72.5%. The other 3 shape features are all performing better than Range-LDA. Among them, LBP-LDA is the most powerful in this case which achieved 67.5%. However, Fusion1 decreases the performance to 66%, showing ineffectiveness of this fusion strategy. One reason is due to a large number of variation factors in our database. The subsequent LDA step in FFF (Funsion2) takes advantage by extracting invariant and complementary *finer features* from all the shape cues. Thus, the proposed Fusion2 is able to further increase the performance by 5% over LBP-LDA.

It is important to emphasize that we achieved a 72.5% recognition rate using the depth data alone from Kinect which is the highest recognition rate reported for Kinect so far.

6.5.4.4 Recognition with Various Color Spaces

In this experiment, we investigate the use of color texture information provided by the Kinect sensor. The color texture image provided by Kinect is encoded using the RGB

color model. However, RGB is a weak color space for face recognition. Other color spaces such as the normalized RGB (denoted as nRGB) using CSN-I proposed in (Yang *et al.*, 2010a), the $I_1I_2I_3$ (Ohta, 1985) and the Discriminant Color Space (DCS) (Yang *et al.*, 2010b) are proven to be better than RGB. Nevertheless, it is important to use a color space that has complementary information to the shape features. The performance of the proposed framework with different color spaces are summarized in Table 6.3.

For gray-scale and all color spaces considered in Table 6.3, performances can be improved when combined with shape features. This observation shows that the shape features encode information that is complementary to the texture features. The candidates that benefit most from shape features are gray-scale and RGB. However, their performances are the lowest due to the fact that both gray-scale and RGB are not a good representation for color image in terms of separability for different identities. Although $I_1I_2I_3$ performs similar to DCS, it benefits the least from depth feature. Therefore, DCS is the best color space in this case.

6.5.4.5 Robustness to Unseen Variations

One may argue that the success of LDA in our experiment can be due to the fact that, the standard training set has covered all poses, illuminations and expressions. Although interaction between these factors (i.e. the testing set) are not seen, it may be sufficient for LDA to extract invariant features. In this section, we aim to show how our framework generalizes against unseen variations. In addition, we also aim to show some baseline performances for CurtinFaces under different number of training samples.

From 18 images per subject in the standard training set, we create 4 smaller sets with 6, 9, 12 and 15 images per subject. These images are randomly chosen and repeated 50 times. Therefore, each training set contains some missing factors with 50 random alternations. We evaluate the performance of LDA baseline (gray-scale), RGB+LDA, DCS method and our proposed framework on the same standard testing set introduced in Section 6.5.4.1. Note that these methods, except the proposed framework, only work on the texture image, not considering any depth data. The averaged rank-1 identification rate and standard deviation for the 50 repetitions are reported in Table 6.4.

From Table 6.4, the performance of LDA baseline drops dramatically to 44.5% with 6 training images per subject. As expected, color is better than gray-scale image, and is even better when depth data is also used. Despite having unseen factors in smaller training sets, the proposed framework still improves over DCS, in all cases. These results demonstrate

the importance of Kinect depth data, as well as the effectiveness of our framework.

6.5.4.6 Time Complexity

Haar and Gabor features are known to be slow due to their high dimensionality. In this section, we justify the applicability of our framework for real time systems. Only testing time is considered, as training is an offline process. Our implementation is based on 64-bit Matlab using a computer with Intel Core2 Quad CPU @ 3GHz and 4GB RAM. The time required in each step of our framework is reported in Table 6.5. As shown in Table 6.5, only around 12 milliseconds are required to recognize a single query image. Actually, most of the time is spent on Gabor feature extraction which can be even faster when implemented optimally in a faster programming language. However, the current implementation is already sufficient for real time applications.

6.6 Summary

The contributions made in this chapter are as follows. Firstly, we construct a new face database namely CurtinFaces. This database is the first publicly available face database that captured by Kinect with large variations. Some standard experimental protocol are proposed. Secondly, we propose the MCTD method for 2D color face recognition. This method outperforms several other state-of-the-art 2D methods on PIE and CurtinFaces. Thirdly, we propose the FFF method that utilizes the RGB-D data Kinect provided. We show that although the Kinect depth data is noisy, it can be used to improve face recognition performance significantly.

A shortcoming of MCTD algorithm is that it dose not harness 3D information. Although we have proposed the FFF algorithm to utilize the depth data from Kinect, it dose not utilize the absolute measurement advantage of 3D data. Moreover, both MCTD and FFF do not harness sparse coding. As a result, they have limited robustness against uncontrolled factors such as pose, noise and disguise. Addressing all these weakness, a sophisticated RGB-D face recognition method is proposed in Chapter 7. This method can perform reliable face recognition under simultaneous variations in pose, illumination, expression and disguise, regardless of whether the 3D resolution is high or low.

Chapter 7

Utilizing Color and Depth for Robust Face Recognition

Three dimensional face recognition has attracted significant research interest in the past decade due to its broad applications. Face recognition can be performed in a non-intrusive way and sometimes without the user's knowledge or explicit cooperation. However, facial images captured in an uncontrolled environment can have combinations of variations such as varying pose, facial expressions, illumination and disguise. Since the type of variations are unknown for a given image, it becomes critical to design a face recognition algorithm that can handle all these factors simultaneously.

Simultaneously dealing with multiple variations is a challenging task for face recognition. Traditional approaches have tried to tackle one challenge at a time using optical 2D images or texture. For example, the illumination cone method (Georghiades *et al.*, 2001) models illumination changes linearly. The authors prove that the set of all images of a face under the same pose but different illuminations lies on a low dimensional convex cone which can be learned from a few training images. Although this technique can be used to generate facial images under novel illuminations, it assumes that faces are convex and requires training images to be taken with a point light source. The Spare Representation Classifier (SRC) (Wright *et al.*, 2009) and its extension, the Robust Sparse Coding (RSC) (Yang *et al.*, 2011), can handle face images with disguise (e.g. wearing sunglasses) and noise, by removing or correcting the outlier pixels. However, some outlier pixels may have similar texture intensity to the human face and thus can not be identified. Some researchers have also tried to solve the pose problem using 2D images. For example, Gross *et al.* (2004) construct the Eigen-light fields which are the 2D appearance models of a face from all viewpoints. This method requires many training images under different poses and dense correspondences between them which are difficult to achieve. Sharma and Jacobs (2011) use Partial Least Squares (PLS) to linearly map facial images in different poses to a common linear subspace where they are highly correlated. However, such a linear subspace may not exist. In fact, pose variations are highly non-linear and can not be modeled by linear methods. This is why the performance of the above methods drops dramatically with extreme pose variations.

As discussed in Chapter 1, limitations of 2D face recognition, especially sensitivity to pose variation, can be overcome by using 3D face data. Facial geometry is invariant to illumination whereas 2D images are a direct function of the lighting conditions. Although, the 3D imaging process can be influenced by lighting, the 3D data itself is illumination invariant. Facial images under different illumination conditions can be generated using a 3D face model (Toderici *et al.*, 2010). Additionally, it can be used to correct the facial pose or to generate infinite novel poses.

Although existing 3D methods (Mian *et al.*, 2007; Queirolo *et al.*, 2010; Spreeuwers, 2011; Lei *et al.*, 2013) can achieve very high accuracy even under challenging experiments such as the Face Recognition Grand Challenge (FRGC) (Phillips *et al.*, 2005), they all assume the availability of high resolution 3D face scanners. Such scanners are costly, bulky in size and have slow acquisition speed which limit their applications. As we can see from Table 6.1 in previous chapter, most 3D devices that have less than 0.5mm depth accuracy require more than one second to acquire one 3D sample. Consequently, subjects must sit still in front of the sensor for the duration of scanning which implicitly means that the user is cooperative. Therefore, the advantage of non-intrusiveness for face recognition is compromised. Although high speed 3D acquisition devices are available such as Kinect, they provide only low resolution and noisy 3D data. In this chapter, we design an algorithm that performs equally well on low and high resolution 3D data for robust face recognition.

The rest of the chapter is organized as follows. Section 7.1 reviews some existing 3D face recognition methods and discuss their limitation in case of low resolution 3D data. Section 7.2 presents an overview of the proposed algorithm and Section 7.3 to 7.5 detail the proposal, which includes canonical preprocessing, multi-channel discriminant transform and multi-channel weighted sparse coding. Section 7.6 introduces the CurtinFaces database and the portion used in our experiment. Section 7.7 describes the experiment setting. Section 7.8 to 7.11 report and analyze the experimental results on four datasets. Section 7.12 discusses the time complexity of the system. A summary is given at the end in Section 7.13.

7.1 Current 3D Face Recognition Methods

Many methods have been proposed for 3D face recognition with increasing performance and sophistication, but they are not designed for noisy data such as from Kinect. It would be interesting to see the performance of existing 3D face recognition techniques on Kinect data. Bowyer *et al.* (2006) gave a comprehensive survey of 3D face recognition methods

in 2006 and recent developments until 2012 are covered in the literature review section of a recent paper by Lei *et al.* (2013). Here, we complement these surveys and discus the limitations of some representative techniques, specifically in the context of uncontrolled face images acquired with a noisy sensor such as the Kinect.

The Iterative Closest Point (ICP) algorithm was proposed by Besl and McKay (1992) for the registration and comparison of rigid surfaces. It has been used by many researchers for pose normalization and comparison of 3D faces. It finds the optimal rigid transformation that minimizes the distance between the corresponding (nearest) points of two 3D datasets. The final registration error is generally used as a classification criterion. ICP and its variants have been used for 3D face recognition (Mian *et al.*, 2007; Faltemier *et al.*, 2008). The point-to-point error of ICP is sensitive to expression variations and incorrect point-to-point correspondences may lead to a local minimum. These two problems are more likely to occur when ICP is applied on a pair of noisy 3D faces acquired by Kinect and therefore, the result can be highly inaccurate.

Bronstein *et al.* (2007) proposed an expression-invariant representation of the facial surfaces based on isometric deformations. Matching was done by computing distance between the canonical forms of two faces in their embedded subspace. This algorithm assumes all faces are frontal and does not perform any pose correction. Imaging artifacts caused by disguise changing the facial surface deformation can also affect the canonical representation.

Mian *et al.* (2007) proposed multi-modal (2D + 3D) hybrid (holistic + part-base) approach for robust face recognition. An iterative PCA based algorithm was used for pose correction. Matching of two faces was done using several heuristics: similarity of the SIFT and spherical feature on the holistic 2D and 3D faces respectively, and ICP registration errors of the segmented 3D nose and eyes-forehead components. They have shown that these segmented parts are robust against expression variations. However, the segmentation requires automatic detection of the inflection points around the nose, which may not be achievable on the noisy Kinect data. Faltemier *et al.* (2008) divided a face into 28 overlapping regions in order to improve recognition robustness. ICP method is applied on each region and matching of two faces was performed by fusing the regional registration errors. The subdivision idea is very effective for high resolution scans. Over 80% matching accuracy was reported on some of the small individual region of just 25-45mm spherical radius. However, small regions in a low resolution data may not be sufficiently discriminative for face recognition.

Queirolo *et al.* (2010) proposed a Simulated Annealing (SA) approach for range image

registration and Surface Interpenetration Measure (SIM) for similarity. Matching was done by fusing the SIM for elliptical regions around the nose, forehead and the holistic face. Although, impressive results were reported on the FRGC database, their algorithm requires six landmark points including eye and nose corners, which can not be detected, even manually, on the noisy Kinect 3D face data. Kakadiaris *et al.* (2007) proposed the Annotated Face Model (AFM) to register the input 3D face to an expression-invariant deformable model. After fitting onto the AFM, several features were extracted on the geometric and normal map for matching. Recently, Passalis *et al.* (2011) further extended the AFM method with facial symmetry to handle missing data caused by self-occlusion in non-frontal poses. Two fitted AFM were generated for matching by mirroring the AFM external forces from one side to the other. As a result, their method can handle pose and expression variations at the same time. However, the fitting of AFM requires eight landmarks over the 3D face, which is even more difficult than Queirolo *et al.* (2010)'s method to be applicable on Kinect data. Even on high resolution scans, some landmark detection errors can be greater than 10mm, especially on 60 degree side scans.

Wang *et al.* (2010) proposed a novel representation namely the Signed Shape Difference Map (SSDM). They fused several features extracted on the SSDM based on a boosting algorithm for face recognition. Pose correction was done by aligning the normal of the symmetry plane, nose tip and direction of nose bridge. The symmetry plane was determined, by registering the 3D face with its mirrored version using ICP, and fitting a plane to the resulting registered faces. This pose correction method is more efficient than ICP since it avoids registration to every gallery face. The proposed SSDM was created by taking the direct pixel differences between two depth images after pose correction. However, searching for the symmetric plane on a profile view leads to non-convergence of ICP. Additionally, the SSDM may not be effective for Kinect data where depth map differences can also be due to noise.

Alyuz *et al.* (2012) is one of the very few works that address the disguise problem in 3D face recognition. They registered the probe face to a generic face model by applying ICP to the area around the nose. After registration, points that were far away from the generic face model were treated as outliers and removed. Missing data was then restored using "Gappy PCA". Matching was done by dividing the restored face into 30 regions and fusing the regional similarity scores obtained from multiple local Linear Discriminant Analysis (LDA) classifiers. Although they achieved 76% to 94% in their experiments on the Bosphorus 3D face dataset (Savran *et al.*, 2008), they have only considered frontal views with neutral expression and have not compared with other 3D methods.

A summary of the aforementioned methods is presented in Table 7.1. The main limitations

Table 7.1: Summary of some 3D methods on their required resolution, landmarks and the main variation they addressed, i.e. Pose(P), Illumination(I), Expression(E) and Disguise(D).

	Resolution	Landmark	Variation[1]
Bronstein *et al.* (2007)	high	2	E
Mian *et al.* (2007)	high	5	E
Faltemier *et al.* (2008)	high	Nose tip	E
Queirolo *et al.* (2010)	high	6	E
Wang *et al.* (2010)	high	Nose tip	E
Passalis *et al.* (2011)	high	8	E, P
Alyuz *et al.* (2012)	high	Nose tip	D
Proposed	low	Nose tip	P, I, E, D[2]

[1] Main variation addressed.

[2] We also consider some combinations of variations.

of these techniques are as follows.

- All methods assume the availability of high resolution 3D scans and therefore, may not work well for low resolution data.

- Some methods require more than one landmarks which may not be accurately detectable on low resolution 3D data.

- Most techniques rely on face segmentation or region sub-division to ensure robustness. This idea works well for high resolution scans because these smaller parts themselves are discriminative enough to separate different identities. However, such an approach is not feasible for noisy Kinect data, where even the completed 3D face is hardly recognizable.

- Most algorithms focus on 3D data alone and ignore the 2D texture. However, 3D data alone is insufficient for robust face recognition especially when acquired with a low resolution sensor. Although, pure 3D techniques can be extended to multi-modal (2D+3D) in a straight forward way using score level fusion, a more sophisticated approach that considers interaction between 2D and 3D data will be more reliable and accurate.

- Most methods are optimized for the FRGC data alone and have not been tested on other datasets. Their high performance could well be due to overfitting on this data since faces in the FRGC data are all near-frontal and without disguise. In realistic applications, uncontrolled images can be acquired in arbitrary variations of pose, illumination, expression and disguise. None of the existing methods are evaluated against all of these factors.

7.2 Proposed Method Overview

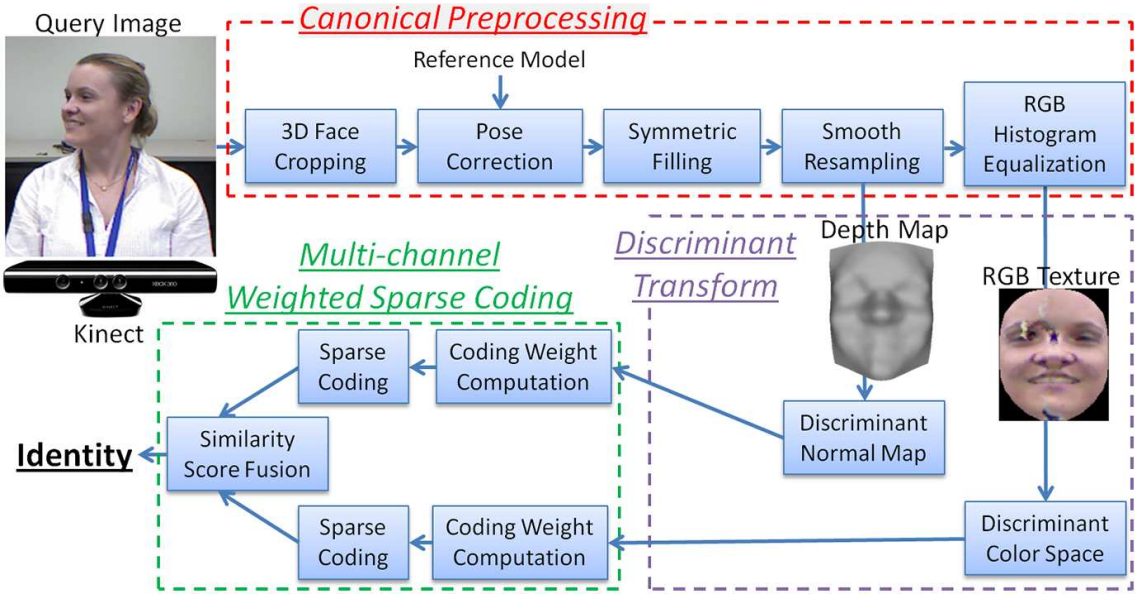


Figure 7.1: Overview of the proposed method.

An overview of our proposed method is shown in Figure 7.1. Our algorithm is designed specifically for robust face recognition using Kinect and when the query image is acquired without the user's explicit cooperation. The following imaging conditions can be expected in this context:

- Multiple training images of the enrolled subjects are available. This is possible with Kinect's high acquisition speed.

- The query image is uncontrolled since it can be acquired under arbitrary poses, illuminations, expressions, and possibly with disguise.

- Both 2D (RGB) and 3D (XYZ) data are available. The 3D data are in low resolution

and noisy.

Our results show that the proposed algorithm performs equally well on high and low resolution data. Moreover, it can use a single gallery image per subject or exploit the presence of multiple gallery images to perform recognition. Although multiple query images can also be acquired during recognition, we restrict our experiments to recognition based on a single query image to make our results consistent with standard face recognition protocols.

7.3 Canonical Preprocessing

The input to our canonical preprocessing algorithm is a 6D (XYZ-RGB) point cloud and the output is a canonicalized depth map and registered RGB texture image of the face. Unlike common range data preprocessing which only removes spikes and fills holes, the proposed algorithm additionally corrects the facial pose so that it is view-point invariant and completes missing data due to self-occlusion. In fact, most data obtained from the Kinect sensor do not have spikes[1]. Holes are filled during a resampling step. Details of each preprocessing step are given below.

7.3.1 3D Face Cropping and Pose Correction

Due to the level of noise in Kinect depth data (as illustrated in Figure 6.2), the nose tip is the most reliable landmark that can be located on the 3D face. In this work, we assume that the approximate nose tip location has been detected. Since the nose tip is required only for rough alignment and face cropping, the algorithm works as long as the detected nose tip is close enough to the true location. Given the nose tip position, we first translate the point cloud such that the nose tip is at the origin. Then a sphere of 80mm radius centered at the nose tip is used to crop the face. As a result, a 6D point cloud (XYZ-RGB) of only the face area is obtained.

The Iterative Closest Point (ICP) algorithm (Besl and McKay, 1992) is an accurate technique for aligning two 3D point clouds. However, it is known to be computationally expensive, and hence registering the query face to every frontal gallery face in search of the best alignment is not feasible. Instead, we register the query (XYZ only) to a reference model. Since different subjects have different face shape, the reference face model must be

[1]It is possible that filtering is done inside the Kinect hardware or API.

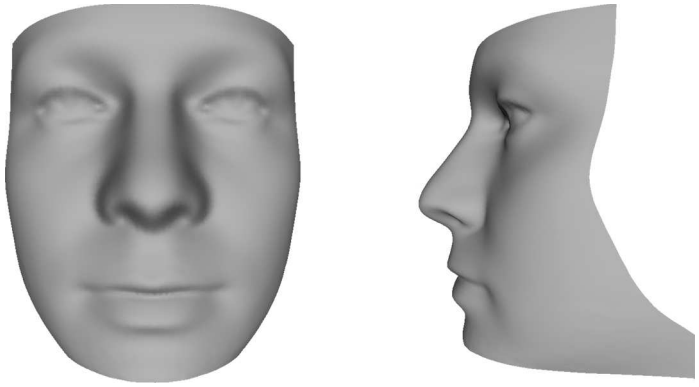


Figure 7.2: The reference face model.

a reliable representation of common 3D faces. Such a reference face can not be constructed from the noisy Kinect data. Therefore, we build the reference using face models (with no expression) from the FRGC (Phillips *et al.*, 2005) and the UWA database (Mian, 2011). The reference face is constructed by aligning the scans, resampling them on a uniform 128x128 grid and then taking their mean. The reference face has 64 points between the centers of the eyes. The number of points from the center of the lip to the line joining the eyes is also 64. Figure 7.2 shows the reference face used in our experiments. Both training and test data are registered to this reference face using up to 30 iterations of ICP. In each iteration, we do not consider point correspondences further than 16mm apart. Such a setting allows us to correct poses up to $\pm 90^o$. An example registration of a profile face to our reference face is shown in Figure 7.3. In this case, only points around the nose can establish correspondences in first few iterations. More correspondences are found in subsequent iterations until the two faces are correctly aligned.

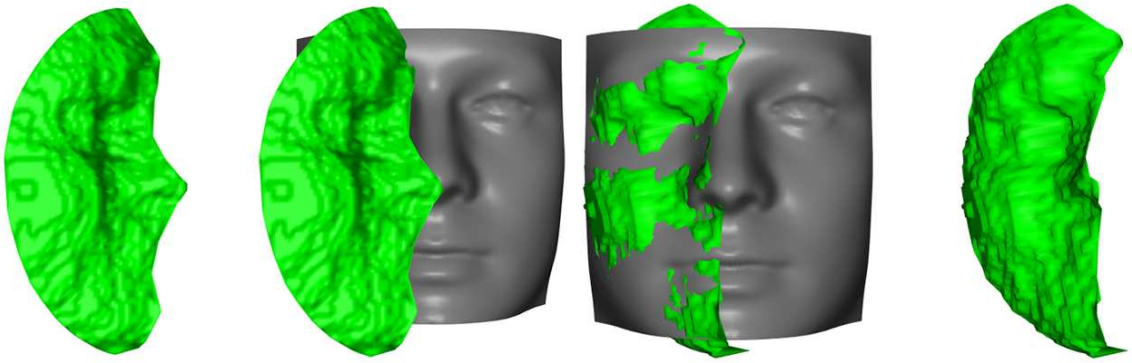


Figure 7.3: First column show the profile face before ICP and last two columns show the result after ICP converge.

7.3.2 Symmetric Filling and Resampling

After pose correction, some data may be missing due to self occlusion for non-frontal views. This missing data can be estimated based on facial symmetry. Despite the fact that human faces are not perfectly symmetric, the variations caused by facial asymmetry are less than the variations caused by different identities (Passalis *et al.*, 2011). Unlike the work in (Passalis *et al.*, 2011) which mirrors the AFM external forces from one side to the other and then generates two different fitted AFMs for recognition, we utilize facial symmetry in the preprocessing stage at the point cloud level. Specifically, a mirrored point cloud is created by replacing the X values in the original point cloud by their opposite numbers (-X). However, not all the mirrored points are useful as we only want to fill in the missing data. Ideally, no point should be added on a frontal face, while all points should be mirrored on a profile view. To this end, for each mirrored point, we compute its Euclidean distance using (XY values only) to the closest point in the original point cloud. If this distance is less than δ, the mirrored point is removed. The idea is to add the mirrored point only if there is no neighboring point at that location. Note that Z is not used when calculating the distance, because the difference in Z is usually caused by facial asymmetry rather than missing data. The remaining mirrored points are then combined with the original point cloud before resampling. A sample symmetric filling can be seen in Figure 7.4. The threshold δ can be chosen based on the spatial resolution of the sensor or the point cloud itself. In our experiments, it was user defined. Depending on the original sample density, high values of δ will lead to a noisy surfaces while values too low will not help in symmetric filling. We empirically found that a good balance can be achieved with $\delta = 2$mm, however, the performance is not affected much when setting δ to values between 1-5mm.

Resampling is the final step in our preprocessing algorithm, which is done by fitting a smooth surface to the point cloud (XYZ) using an approximation approach[2]. This algorithm fits a surface to the points with a smoothing (or stiffness) constrain that does not allow it to bend abruptly and thereby alleviating the effects of noise and outliers. Since surface fitting is done after symmetric filling, the added mirror points will also contribute to the surface. This is especially helpful to stabilize the noisy Kinect data. For each face, 161×161 points are re-sampled uniformly from its minimum to the maximum X and Y values. The advantage of re-sampling from min to max is that it aligns faces on a 2D grid. Notice that we do not smooth the RGB texture since it is not noisy and smoothing will only blur it. Instead, we just re-sample it to the same XY location with interpolation. After re-sampling, the X and Y grids are discarded and the Z depth map is converted

[2] `mathworks.com/matlabcentral/fileexchange/8998`

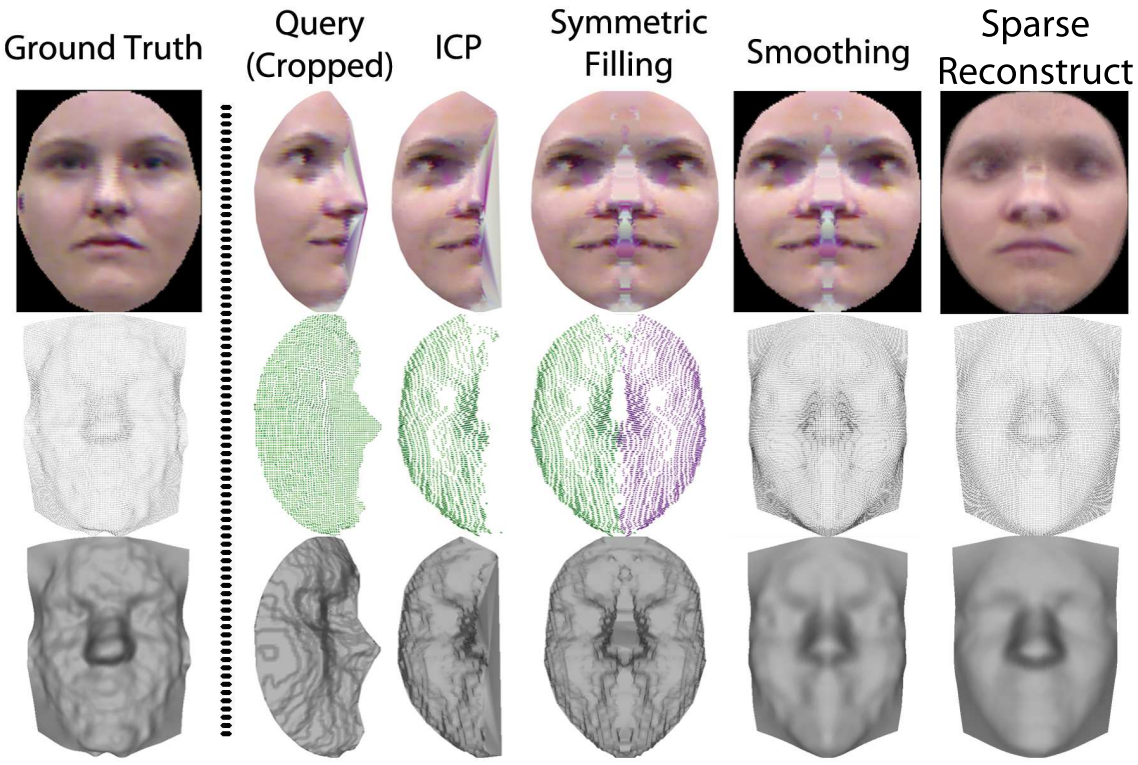


Figure 7.4: Example canonical preprocessing and sparse reconstruction on profile view probe image.

to its surface normal map. Finally, six 161×161 matrices (RGB$N_xN_yN_z$) are obtained. These matrices are down-sampled to 32×32 for further processing. Some sample output images of the proposed canonical preprocessing algorithm are shown in Figure 7.9b.

7.3.3 RGB Histogram Equalization

Figure 7.5: Result of RGB histogram equalization.

The above preprocessing steps also gives the RGB texture of the face that is registered with the depth map. However, texture is easily affected by illumination. Therefore, the last step of our proposed preprocessing algorithm aims to enhance the reliability of the texture map. Since we only want to enhance the illumination contrast, the RGB color image is first converted to the CIE $L^*a^*b^*$ color space, where the three channels representing the lightness of color (L^*), position between red and green (a^*) and position between yellow and blue (b^*), respectively. The Contrast-Limited Adaptive Histogram Equalization (CLAHE) method (Zuiderveld, 1994) is applied on the L^* component only, using a 2×2 window, to enhance local contrast. This method applies histogram equalization on each patch of the image with a specified contrast limit to avoid amplification of noise. Neighboring patches are then combined using bilinear interpolation to eliminate artificially induced boundaries. The standard histogram equalization method is then applied again on L^* channel of the holistic image. Finally, the resulting image is converted back to RGB space. Some examples are shown in Figure 7.5. The top row are original images, while the bottom row are processed images. Compared with the original images, the facial details are more apparent after preprocessing.

7.4 Multi-channel Discriminant Transform

Since both the color map (RGB) and normal map ($N_xN_yN_z$) images consist of multiple channels, we can improve their discriminative power by applying a transformation that is learned from labeled training data similar to the idea of Discriminant Color Space (DCS) (Yang *et al.*, 2010b). Most existing multi-modal (2D+3D) face recognition methods convert the color image to gray-scale first (Mian *et al.*, 2007; Al-Osaimi *et al.*, 2012). However, color information is proven to be useful especially when the shape cue is noisy (Yip and Sinha, 2002). Therefore, color cue is likely to be very useful in the case of Kinect where the 3D data is noisy and low resolution.

Color images are usually modeled in the RGB space which is not a discriminant space due to high inter-component correlation (Yang *et al.*, 2010a). Although Yang *et al.* (2010a) have proposed Color Space Normalization (CSN) to reduce correlation, CSN does not consider class separability. Recently, optimal color spaces, that are learned from the training data to maximize class separability, are proposed. The Discriminant Color Space (DCS) (Yang *et al.*, 2010b) method finds a set of linear combinations for the R, G and B components in order to maximize class separability similar to the idea of LDA. The Color Image Discriminant Model (CID) (Yang and Liu, 2008a) seeks the optimal color space and feature subspace simultaneously. The Tensor Discriminant Color Space (TDCS) (Wang *et al.*, 2011) method models the color image as tensor and seeks the color space transformation and two feature subspaces respectively along the row and column directions. However, both CID and TDCS methods do not have closed-form solutions and are solved iteratively which can lead to local minima or non-convergence. Our experience shows that DCS is a reliable color space for face recognition.

Similar to RGB image, the normal map also has three channels ($N_xN_yN_z$). A discriminant transform similar to DCS can be derived to increase its discriminative power. To this end, we propose a Multi-channel Discriminant Transform (MDT) method, which is a generalization of the DCS method to work on multi-channel data of any order.

Suppose there are a total of M training samples of C classes. Each d dimensional training sample with h channels is denoted by $U_j = [u_j^1, u_j^2, \cdots, u_j^h] \in \mathbb{R}^{d \times h}$, where $j = 1, 2, \cdots, M$. We can define the following linear transform:

$$V = a_1 u^1 + a_2 u^2 + \cdots + a_3 u^h = U_j \cdot A, \tag{7.1}$$

where $A = [a_1, a_2, \cdots, a_h]^T$ is a vector of the transformation coefficients. A good transformation vector should point to the direction that maximizes class separability. Let S_b^v and S_w^v be the scatter matrices to describe the between-class and within-class variance

in the transformed V-space, then A can be found by maximizing the following objective function:

$$J(A) = \frac{tr(S_b^v)}{tr(S_w^v)}, \tag{7.2}$$

Let P_i be the label of the i-th class (i=1,2,...,C), $\overline{U}_i$ be the mean sample of class P_i, $\overline{U}$ be the grand mean for all data, and M_i be the number of samples in class P_i, we can derive the following:

$$\begin{aligned} tr(S_b^v) &= A^T \left(\sum_{i=1}^{C} M_i (\overline{U_i} - \overline{U})^T (\overline{U_i} - \overline{U}) \right) A \\ &= A^T S_b^u A, \end{aligned} \tag{7.3}$$

$$\begin{aligned} tr(S_w^v) &= A^T \left(\sum_{i=1}^{C} \sum_{U_j \in P_i} (U_j - \overline{U_i})^T (U_j - \overline{U_i}) \right) A \\ &= A^T S_w^u A. \end{aligned} \tag{7.4}$$

The objective junction in Eq. 7.2 can be re-written as:

$$J(A) = \frac{A^T S_b^u A}{A^T S_w^u A}, \tag{7.5}$$

Since S_b^u and S_w^u are both $h \times h$ nonnegative definite matrices in general (where h is usually small), the optimal solution can be obtained by solving the equivalent generalized eigenvalue problem:

$$S_b^u A = \lambda S_w^u A, \tag{7.6}$$

such that $A = [A^1, A^2, \cdots, A^h] \in \mathbb{R}^{h \times h}$ are the eigenvectors arranged in descending order of their corresponding eigenvalues.

In the proposed approach, the RGB map is ordered as a 3-channel sample (i.e. $[R, G, B] \in \mathbb{R}^{d \times 3}$). We apply MDT to obtain the corresponding discriminative transformation matrix $A = [A^1, A^2, A^3] \in \mathbb{R}^{3 \times 3}$. Although A^1 is the most discriminative projection, usually all three eigenvectors are required to achieve maximum recognition performance. After the transformation, the texture map is converted from RGB space to Discriminant Color Space (DCS). Similarly, by applying MDT on the normal map which consist of three channels (i.e. $[N_x, N_y, N_z] \in \mathbb{R}^{d \times 3}$), a Discriminant Normal Map (DNM) is obtained. Both DCS and DNM consist of three discriminant channels. Each channel is then normalized to zero mean and unit standard deviation to avoid magnitude domination of one channel over the others.

Figure 7.6 shows some sample images after transformation and a plot of Euclidean distances between images of different subjects of the FRGC dataset. One can observe that the images exhibit a greater color contrast after MDT. Similarly, distances between images of different subjects also increase i.e. the red circles shift upwards.

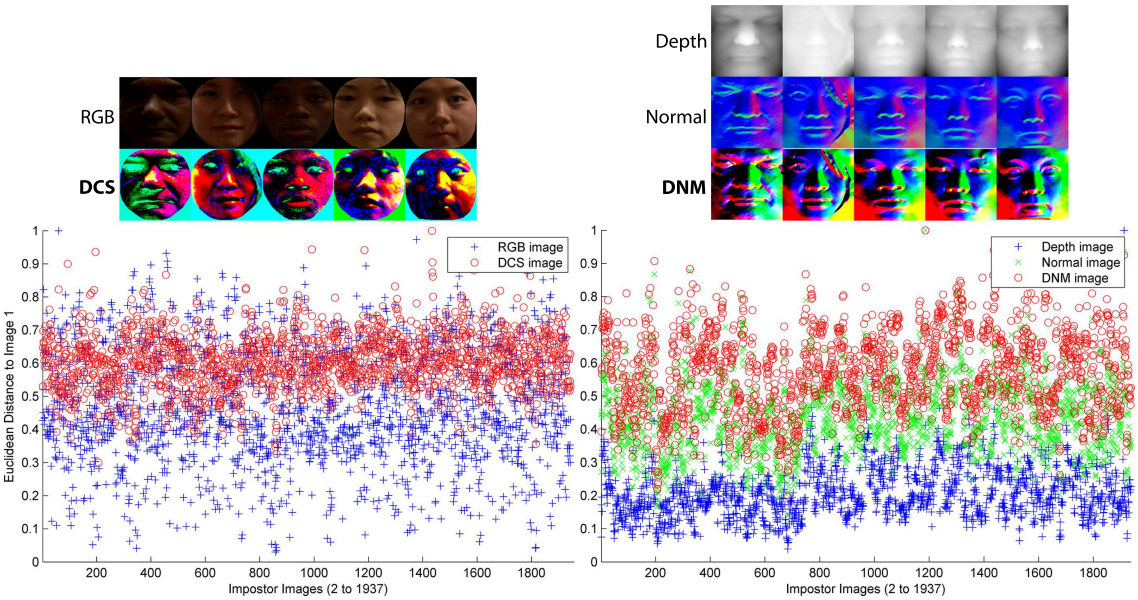


Figure 7.6: Some sample images from the FRGC dataset. Different subjects are more discriminative after Multi-channel Discriminant Transform (MDT).

7.5 Multi-channel Weighted Sparse Coding

Given multiple gallery images per subject, one way to utilize them effectively is by allowing their sparse linear combinations to be matched with a query image (Wright *et al.*, 2009). Although, our preprocessing algorithm removes some level of noise and completes missing data due to occlusions, it cannot perfectly reconstruct a frontal view from profile views. This is because missing data can not be estimated when there are no reference points for mirroring. See Figure 7.4 as an example, which shows an error line in the middle of the resulting canonical face image. A robust approach such as weighted sparse coding (Yang *et al.*, 2011; He *et al.*, 2010), which are shown to be robust against outliers and missing data, can be employed to overcome this problem. In this chapter, we propose a novel face recognition method namely Multi-channel Weighted Sparse Coding (MWSC), which extends Yang et al.'s Robust Sparse Coding (Yang *et al.*, 2011) method to work effectively on multiple channels (e.g. R, G and B) and multiple modalities (2D + 3D). First we formulate the sparse coding as the weighted Lasso problem with ℓ_1 penalty:

$$x = \underset{x}{\operatorname{argmin}} \|W(Ax - y)\|_2 + \lambda \|x\|_1 \tag{7.7}$$

where A is the dictionary i.e. the training samples in our case, y is the query face, x is the coding parameters vector,λ is a constant that controls the coding sparsity and W is a vector consisting of weights for each variable in A. Yang et al. (Yang *et al.*, 2011) have

shown that a robust W can be estimated by this function:

$$W = \frac{\exp(\mu\delta - \mu(e)^2)}{(1 + \exp(\mu\delta - \mu(e)^2))} \tag{7.8}$$

where $e = Ax - y$ is a vector of reconstruction residuals, μ and δ are user defined parameters controlling the rate of decrease and the location of demarcation point respectively. $W^{(1)}$ is initialized as the residual to the mean dictionary atom $e^{(1)} = \overline{A} - y$. Equation 7.7 is then iteratively solved for x and W. The iterations stop at the t-th iteration when the change in W is smaller than ε, i.e:

$$\|W^{(t)} - W^{(t-1)}\|_2 / \|W^{(t-1)}\|_2 < \varepsilon \tag{7.9}$$

In our case, the d-pixels query image $Y = [c^1, c^2, c^3] \in \mathbb{R}^{d\times 3}$ consists of three channels: c^1, c^2 and c^3 (which can be either the three channels of DCS or DNM). In order to apply Eq. 7.7, we convert Y to a column vector $y = [c^1 c^2 c^3]^T \in \mathbb{R}^{3d}$ by stacking the three channels. The dictionary $A \in \mathbb{R}^{3d\times m}$ with m training samples is also arranged in a similar way. While this is a standard data-level fusion strategy, we propose a more effective way to compute the weights W.

We use three channels instead of gray scale images because outliers have more chances of getting detected with multi-channels compared to a single channel. Consider a toy example where a pure red face (RGB1: [1 0 0]) is wearing a pure blue scarf (RGB2: [0 0 1]). Although the facial part and scarf have different colors, they have the same gray-scale intensity of 1/3. Since the residual vector e is computed by direct pixel difference, both red and blue pixels are likely to contribute equally in a grayscale image. Based on this observation, we treat each multi-channel pixel as a h-dimensional vector ($h = 3$ in our case), and compute the residual for each pixel using Euclidean distance in the h-dimensional space.

In each iteration, after the coding vector x is computed using Eq. 7.7, the reconstructed image vector $\tilde{y} = Ax \in \mathbb{R}^{3d}$ is rearranged back to image matrix $\widetilde{Y} = [\tilde{c}^1, \tilde{c}^2, \tilde{c}^3] \in \mathbb{R}^{d\times 3}$. The query image vector y is also rearranged back to image matrix $Y = [c^1, c^2, c^3] \in \mathbb{R}^{d\times 3}$. The d-dimensional residual vector is computed pixel-wise by

$$e_j = \|\widetilde{Y}_j - Y_j\|_2 \quad (j = 1, \ldots, d). \tag{7.10}$$

In our approach, two sets of weights (W_{tex} and W_{dep}) are computed for the DCS and DNM images respectively. Similarly, separate coefficient vectors (x_{tex} and x_{dep}) are computed by sparse coding the DCS and DNM images using Eq. 7.7. For C classes, two sets of

class-wise similarity scores (S_{tex} and S_{dep}) are computed based on the class-wise weighted reconstruction residual:

$$\begin{aligned} S^i_{tex} &= -\|W_{tex}(A^{\in P_i}_{tex} x^{\in P_i}_{tex} - y_{tex})\|_2 \\ S^i_{dep} &= -\|W_{dep}(A^{\in P_i}_{dep} x^{\in P_i}_{dep} - y_{dep})\|_2 \end{aligned} \tag{7.11}$$

where i=1,...,C, and P_i is the label for class i. The two scores, S_{tex} and S_{dep}, are then individually normalized using z-score technique (Jain *et al.*, 2005) and then summed before final decision. The query Y is recognized as the person with the highest final similarity score.

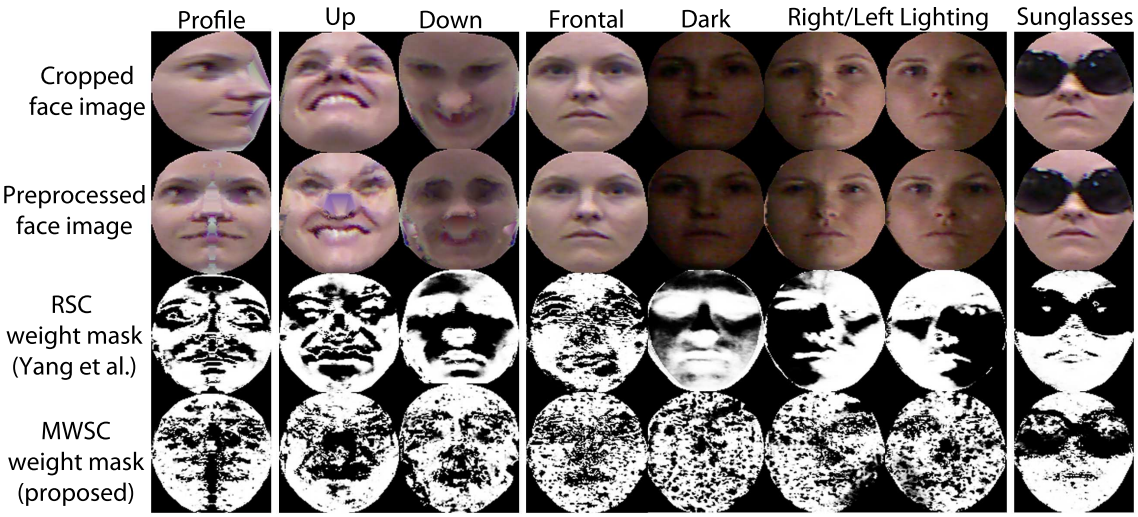


Figure 7.7: Weight masks computed for some probes in CurtinFaces (without histogram equalization). The proposed MWSC works better in terms of masking out outlier pixels.

Figure 7.7 shows some sample weight matrices obtained using Yang *et al.* (2011)'s Robust Sparse Coding (RSC) method and our proposed MWSC approach. The weight matrices are computed using the texture images in CurtinFaces and are shown as 256-level intensity images where brighter pixels represent higher weights. One can see that, for the preprocessed profile face (column 1), a few error lines appear in the middle caused by missing data. These lines are masked out better by our MWSC approach compared to RSC. For a smiling face that is looking up or down (columns 2 and 3), MWSC assigns lower weights to the mouth area, which is the non-rigid part of the face that easily deforms under facial expression. The most apparent advantage of MWSC over RSC is that it is more robust against illumination. RSC assigns low weights to shaded pixels under non-frontal lighting, while MWSC computes a sparse weight mask consistently for frontal faces (column 4-7). In the last column, both RSC and MWSC correctly mask out the pixels associated with sunglasses.

For the choice of parameters, we set λ to 0.001 and ε to 0.05. Following Yang *et al.* (2011),

μ is set as c/δ and δ is chosen as the ℓ-th smallest pixel residual, where ℓ is computed by $\lfloor \tau n \rfloor$. We set c to 8 and τ to 0.6. We empirically found out that this setting yields the best trade-off between robustness, accuracy and speed. To show that these parameters are not tuned to overfit a specific situation, this setting is used through out this chapter in all the experiments using different datasets.

7.6 CurtinFaces: A Kinect Face Database

Table 7.2: Some publicly available 3D databases.

Name	Year	Res.	Acq.	Variation[(1)]
CASIA (CASIA, 2004)	2004	high	4624	P, I, E [(2)]
FRGCv2 (Phillips *et al.*, 2005)	2005	high	4007	E, I
BU-3DFE (Yin *et al.*, 2006)	2006	high	2500	E
ND2006 (Faltemier *et al.*, 2007)	2007	high	13450	E
Bosphorus (Savran *et al.*, 2008)	2008	high	4652	P, I, E, D
CurtinFaces	2012	low	5044	P×E, P×I, D

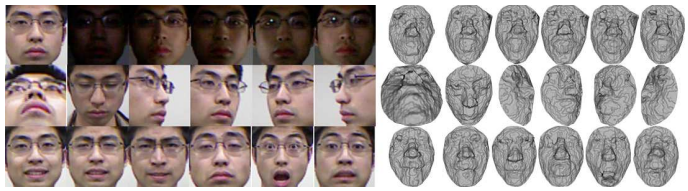
[(1)]Main ones only: {P(pose), I(illum.), E(exp.), D(Disguise)}.

[(2)]CASIA also contains some E+I and P+E.

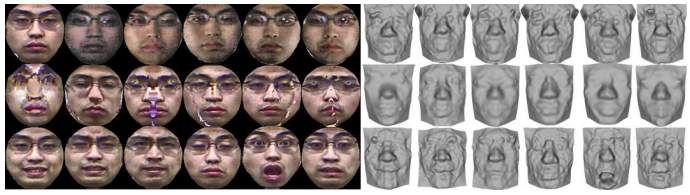
Table 7.2 lists some publicly available datasets. To the best of our knowledge, all existing 3D databases are acquired using high resolution scanners. In addition, none of them consider extensive combination of variation factors. Therefore, we construct our own dataset namely CurtinFaces which is available to the research community[3]. This dataset contains over 5000 images of 52 subjects acquired using Kinect. In this chapter, we use a subset which consists of 4784 images of 52 individuals with variations in poses (P), illumination (I), facial expressions (E) and sunglasses disguise. The database contains facial images with and without glasses. For each subject, three images in the front, left and right profile view are without glasses. Additionally, for each subject, there are 49 images at $7E \times 7P$ and 35 images at $7E \times 5I$ i.e. combinations of 7 expressions with 7 poses and 5 illuminations. Images with sunglasses are under five conditions (i.e. $3P$ and $2L$). The full set of images per person are 92.

Out of the 92, 18 images per subject (see Figure 7.8) are used as the training/gallery set. Each of these 18 images contain only one of the three variations (I, P or E). They are used to learn the DCS and DNM transformations and as the coding dictionary, after

[3]`impca.curtin.edu.au/downloads/datasets.cfm`

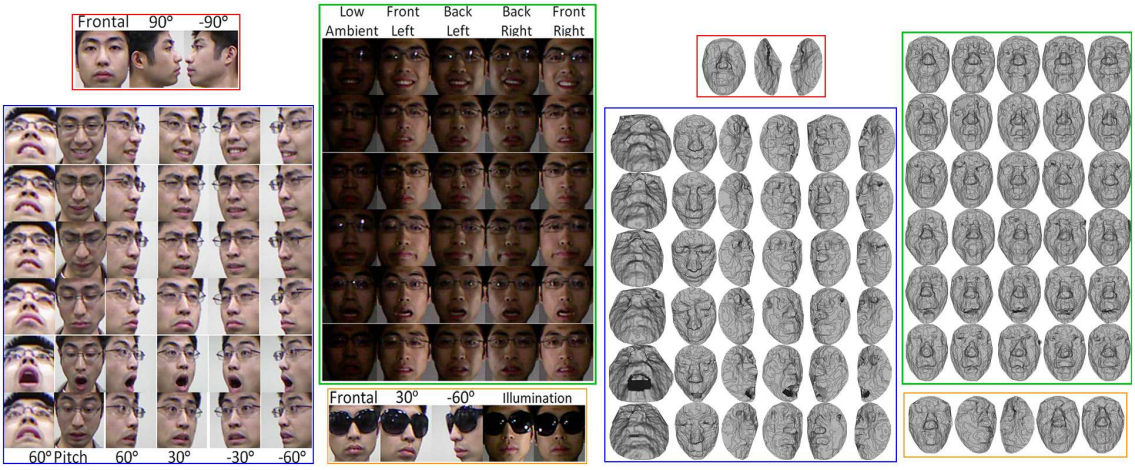


(a) Before preprocessing.

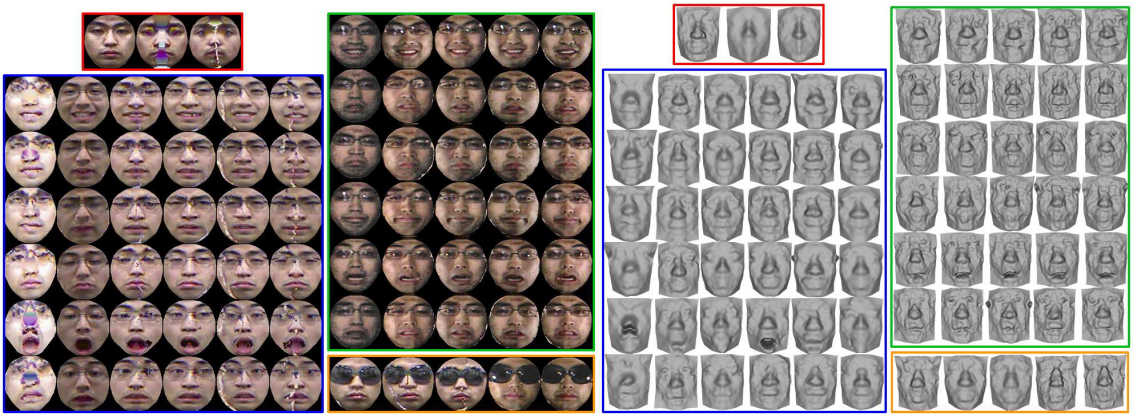


(b) After preprocessing.

Figure 7.8: Sample enrollment images of one subject.



(a) Before preprocessing.



(b) After preprocessing.

Figure 7.9: Sample test images of one subject.

preprocessing. The 74 remaining images per subject are used as test images (see Figure 7.9). For all images, the nose tip is manually detected.

7.7 Experiment Setup

Face recognition can be divided into verification and identification. The former verifies if two face images are from the same subject, while the latter finds the identity of a query face image within a database. Verification problems usually assume certain level of user cooperation (such as displaying the passport photo and looking at the camera). However, as discussed previously, this work focuses on applications where user cooperation may not be achievable. Therefore, we evaluate our system under face identification protocol. The proposed method applies canonical preprocessing on both training and test data as detailed in Section 7.2. The multi-channel Discriminant Transform (mDT) is applied afterwards. The multi-channel Weighted Sparse Coding (MWSC) method is then used to identify the probe. We also compare the performance of the proposed method with the following methods from existing literature:

- Mian *et al.* (2007) (2D+3D)
- RSC (Yang *et al.*, 2011) (RGB-D)
- SRC (Wright *et al.*, 2009) (RGB-D)
- SVM-rbf (Chang and Lin, 2011) (RGB-D)

Although some of these methods were proposed for 2D gray-scale images, we extend them and tune them for RGB-D data for a fair comparison. First, we scale and translate every face image such that the eyes and mouths are aligned on the same pixels. Then a bounding box is used to crop the face area. These procedures are applied on both the RGB and depth data. Afterwards, they are resized to 32×32 and converted to two vectors by stacking their columns. The resulting 1024D depth vector and 3072D RGB vector are input into the aforementioned 2D algorithms separately, except for SVM, where PCA is applied first to reduce the data dimension retaining 99% energy (around 350D). Two different similarity scores obtained from RGB and depth are then normalized using z-score (Jain *et al.*, 2005) and summed for final decision.

7.8 Robust Identification using Kinect

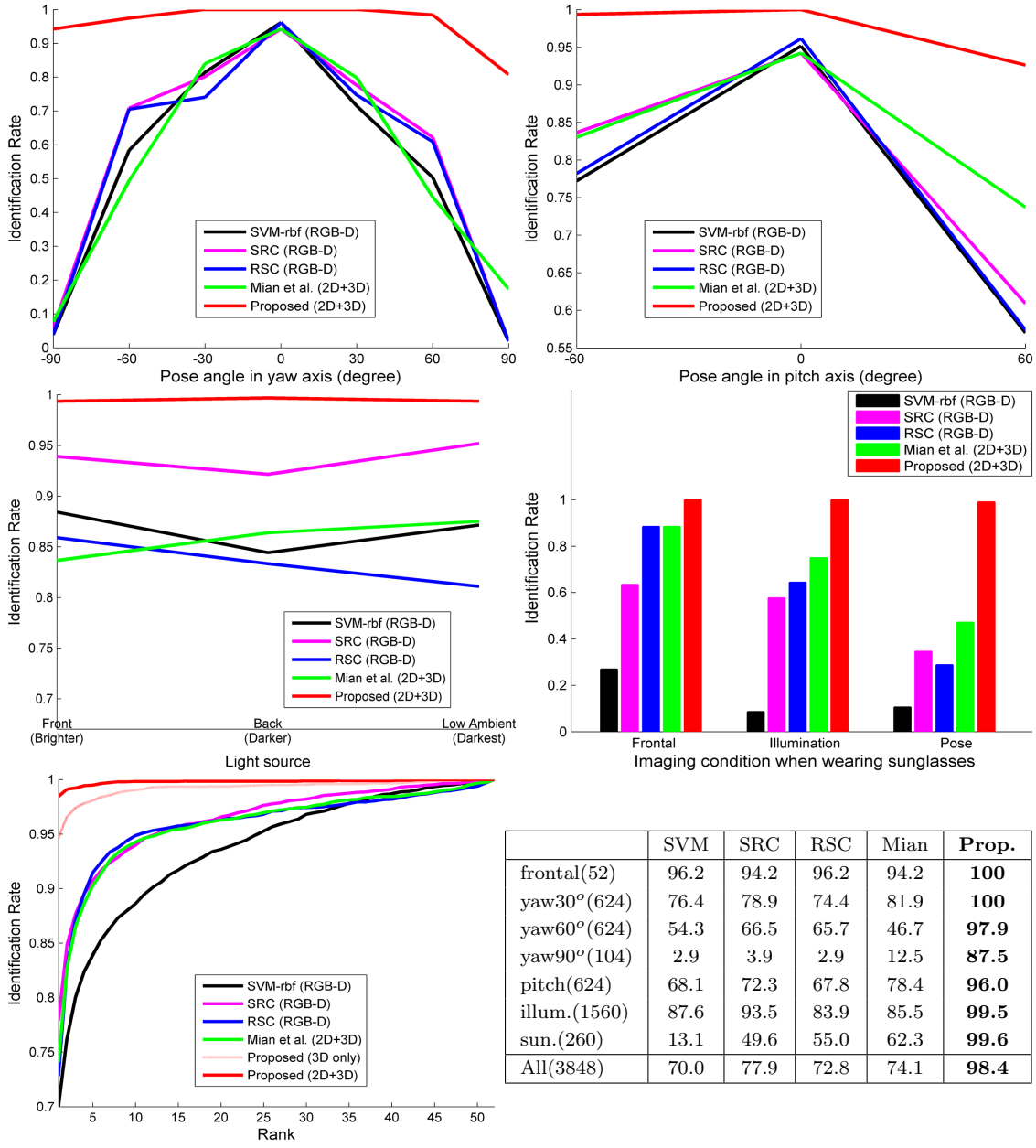
Identification results are reported in Figure 7.10 and labels are defined in Figure 7.9a. For simultaneous variation in pose and expression (P×E, top two plots), the performances of other methods decrease dramatically with larger pose variations (both yaw and pitch) to the extent that accuracy is $< 15\%$ on profile query faces ($\pm 90^o$). Although Main et al.'s MMH method performs pose correction using 3D data, their method is not designed to handle missing data caused by self-occlusion on non-frontal views. One can see that our proposed method outperforms all four techniques and is more robust to variations.

Observe from (left middle plot) the result under simultaneous variations in illumination and expression (I×E) that most methods are not affected by illumination. This is because of the fusion of depth data which is less sensitive to illumination. Also notice that RSC performs the worst in this case. The reason is that, as illustrated in Figure 7.7, the RSC's weight mask computation is sensitive to extreme outliers such as texture pixels that are in cast shadows. Although our method also uses texture data, it is not affected by illumination and is able to maintain consistent performance across different lighting conditions.

The results of sunglasses disguise is presented in the bottom right figure. As expected, SVM achieves very low performance because it is not designed to handel occlusion. RSC performs better than SRC on the average, as it can correctly mask out the sunglasses pixels. Mian et al.'s method can handle disguise to some extends as it segments the face into two parts and the nose part is completely un-occluded by sunglasses. However, our method achieves the best performance. The main reason is that the small nose region used by MMH is not discriminative enough due to Kinect's low resolution.

The main reason why methods like MMH that have been tested on the FRGC data are not suitable for Kinect is because they require face segmentation or landmark identification which can not be accurately performed on low resolution data. Observe that MMH only achieves 94.2% on frontal views with expressions. This suggests that the idea of using rigid face segments to deal with expressions is not very effective on low resolution data. This is due to the fact that smaller regions themselves are not sufficiently discriminative in low resolution. Moreover, some errors are also caused by the failure of landmark detection leading to incorrect face segmentation.

Furthermore, MMH achieves only 85.5% for test images under illumination variations (frontal views with expressions). This significant drop in performance is caused by the



	SVM	SRC	RSC	Mian	**Prop.**
frontal(52)	96.2	94.2	96.2	94.2	**100**
yaw30^o(624)	76.4	78.9	74.4	81.9	**100**
yaw60^o(624)	54.3	66.5	65.7	46.7	**97.9**
yaw90^o(104)	2.9	3.9	2.9	12.5	**87.5**
pitch(624)	68.1	72.3	67.8	78.4	**96.0**
illum.(1560)	87.6	93.5	83.9	85.5	**99.5**
sun.(260)	13.1	49.6	55.0	62.3	**99.6**
All(3848)	70.0	77.9	72.8	74.1	**98.4**

Figure 7.10: Identification results on **CurtinFaces**. The top two plots show that the proposed method outperforms all others and is robust to pose variations in yaw and pitch. The middle two plots show that the proposed algorithm is robust to illumination and occlusions. CMC curves are given in the bottom left plot and rank-1 identification rates are summarized in the table. The proposed method achieves the overall best results.

false rejection in the first stage of MMH where candidates are eliminated by a low cost rejection classifier which is partly based on the the SIFT features extracted from the texture image. Removing the rejection step or setting a higher threshold to reject less faces may not be feasible for identification problem due to the fact that the matching engine of MMH is based on ICP which is computationally very expensive. Matching a single query to a large gallery may take several hours.

In summary, our proposed algorithm achieves an overall average of 96.5% and 94.6% rank-1 identification rates respectively using only the DCS texture and DNM depth images alone. Considering the high levels of noise in Kinect 3D data, 94.6% accuracy is a significant achievement. Combining DCS and DNM, our proposed approach is able to achieve an average of 98.4% identification rate. Even under the extreme case of profile view, our accuracy is 87.5%. To the best of our knowledge, these are the best identification rates reported for low resolution 3D data acquired under challenging conditions. Our results justify that noisy Kinect data is useful for face recognition.

7.9 Evaluation on Bosphorus Database

To show that the proposed algorithm works equally well on high resolution scans, we evaluate it on the Bosphorus 3D face database (Savran *et al.*, 2008) which was acquired with a high resolution scanner namely InSpeck. This database contains 4666 scans of 105 individuals. Each subject has up to 54 scans out of which up to 35 are with expression variations. There are six standard emotional expressions and 28 expressions are performed according to the Facial Action Coding System (FACS) (Savran *et al.*, 2012). Besides expressions, the database also contains 13 poses with different degrees of yaw, pitch and a combination of both up to 90^o. Each person also has up to four types of occlusions or disguises. Some sample images are shown in Figure 7.11. The amount of uncontrolled factors in this database makes it very suitable for our study on non-intrusive face recognition.

All images in the Bosphorus database are labeled with up to 24 landmarks. However, our algorithm requires only the nose tip location. For training, we use the FRGC dataset to compute the DCS and DNM transformation matrices. For testing, we follow the first versus all experimental protocol similar to Li et al.(Li *et al.*, 2011). This protocol uses the first scan of each subject (105) as the gallery and the rest (4561) for testing. In fact, this setting is not favorable for sparse coding which exploits linear combinations of multiple gallery images per subject. To overcome this limitation, we generate multiple samples from each gallery image. Since our approach works on images as low in resolution as 32×32, multiple

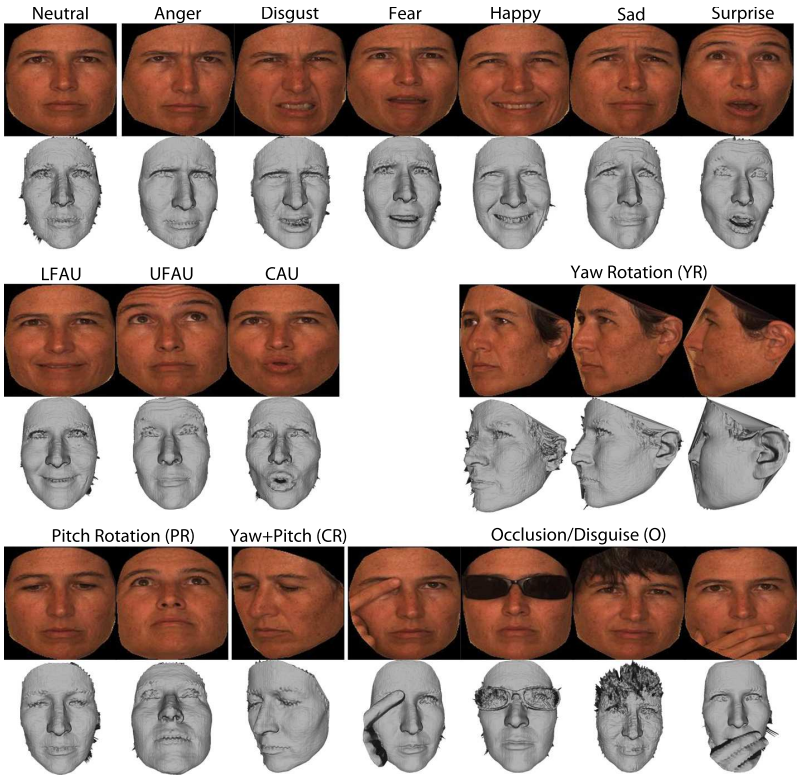


Figure 7.11: Sample images in **Bosphorus**.

independent (to some extent) samples can be generated by downsampling the original high-resolution image. More specifically, we downsample the original preprocessed image from 161×161 to 64 × 64, and then generate four 32×32 images by taking its alternate (even or odd) rows and columns. The fifth image is obtained by directly downsampling the 64 × 64 image to 32×32 using interpolation. These five samples mimic images taken by five independent low resolution cameras with slight translational shifts. Our claim is backed up by the increased identification rates we achieved using this approach.

Furthermore, these generated samples also increase the tolerance to minor translational errors caused by misalignments. Note that the use of synthesized samples do not violate Li's protocol since they are generated from a single gallery image. Therefore, our performance comparison to Li's work in the first versus all scenario is fair. A comparison of rank-1 identification rates is reported in Table 7.3. Note that Mian's approach does not achieve good performance in this dataset because it was not designed to handle pose and occlusion variations. Our proposed method outperforms its competitors in all cases except with occlusion. This is mainly because of the point-to-point ICP registration which tries to register the hand-occluded surface to the reference face. In fact, our performance for occlusion can be increased if ICP step is skipped since all occluded faces are frontal.

Table 7.3: Results on **Bosphorus** using first-neutral (105) vs. all (4561) protocol.

	SVM	SRC	RSC	Mian *et al.* (2007)	Li *et al.* (2011)	**Proposed**
Expression by Emotion						
Neural(194)	91.8	89.7	90.7	100	100	**100**
Anger(71)	67.6	64.8	81.7	93.0	88.7	**98.6**
Disgust(69)	52.2	75.4	88.0	88.4	76.8	**100**
Fear(70)	40.0	58.6	81.4	95.7	92.9	**100**
Happy(106)	38.7	73.6	88.7	87.7	95.3	**97.2**
Sad(66)	68.2	77.3	89.4	98.5	95.5	**100**
Surprise(71)	31.0	66.2	90.1	95.8	98.6	**100**
Expression by Facial Action Unit						
LFAU(1549)	72.8	83.5	91.7	94.3	97.2	**99.5**
UFAU(432)	81.9	85.7	92.8	98.8	99.1	**100**
CAU(169)	60.4	81.1	94.1	96.5	98.8	**100**
Poses in Yaw, Pitch and Combination						
YR(735)	7.8	18.2	23.8	50.9	78.0	**92.4**
PR(419)	50.8	59.2	73.5	98.1	98.8	**99.5**
CR(419)	4.7	13.7	19.0	62.6	94.3	**96.7**
Disguise/Occlusion						
O(381)	28.9	51.70	74.0	77.7	**99.2**	91.1
All Probes						
All (4561)	52.3	63.9	73.9	87.9	94.1	**97.6**

However, we assume we do not have previous knowledge about the query face and the same algorithm is applied on every database. The 91.1% identification rate shows that our system does not fail even in the difficult case of hand occlusion.

Interestingly, without the use of synthesized multiple enrollments, our system still achieves an overall average 96.6% identification rate on the Bosphorus database. This is because sparse coding is performed collaboratively over all gallery images. Faces of different subjects share some parts in common which can help stabilize the sparse coding results (also see Zhang *et al.* (2011)). Due to the high discriminativeness of DCS and DNM feature, the sparse coding solution always assign a large coefficient to the image of correct identity, hence recognizing most of the query faces correctly.

In summary, our proposed approach achieves the highest average performance of 97.6% which is, to the best of our knowledge, the highest identification rate reported for the Bosphorus database. It is important to emphasize that the performances reported in the table are obtained by employing exactly the same algorithm and parameters as those used for CurtinFaces in Section 7.8. Our proposed method performs consistently and robustly across different datasets with arbitrary uncontrolled conditions.

7.10 Evaluation on CASIA

This dataset (CASIA, 2004) is also acquired with a high resolution 3D scanner namely Minolta. It contains a total of 4624 scans of 123 individuals. We consider this dataset suitable for our experiments because of two reasons. Firstly, it contains separate variations in expressions (E), poses (P) up to 90^o and illumination (I). Secondly, it contains two types of combined variations: expression variations with illumination from the right side (E+I) and pose variations with a smiling expression (E+P). Some sample images are shown in Figure 7.12. Notice that due to the use of fast mode of Minolta, the 3D models are not as accurate as those in Bosphorus or FRGC databases. However, they are far better than those acquired by Kinect.

All images in the CASIA database with $\leq 60^o$ pose and without glasses are labeled with nose tip position using their nose tip detection algorithm. We manually label the nose tip in the remanning images. We follow Xu et al.'s (Xu *et al.*, 2009) experimental protocol and use 759 images from the last 23 subjects for training and the first images of the remaining 100 subjects for gallery. Unlike Xu et al. we use all the remaining images as probes whereas Xu et al. exclude probes with $> 60^o$ pose and those wearing glasses. We

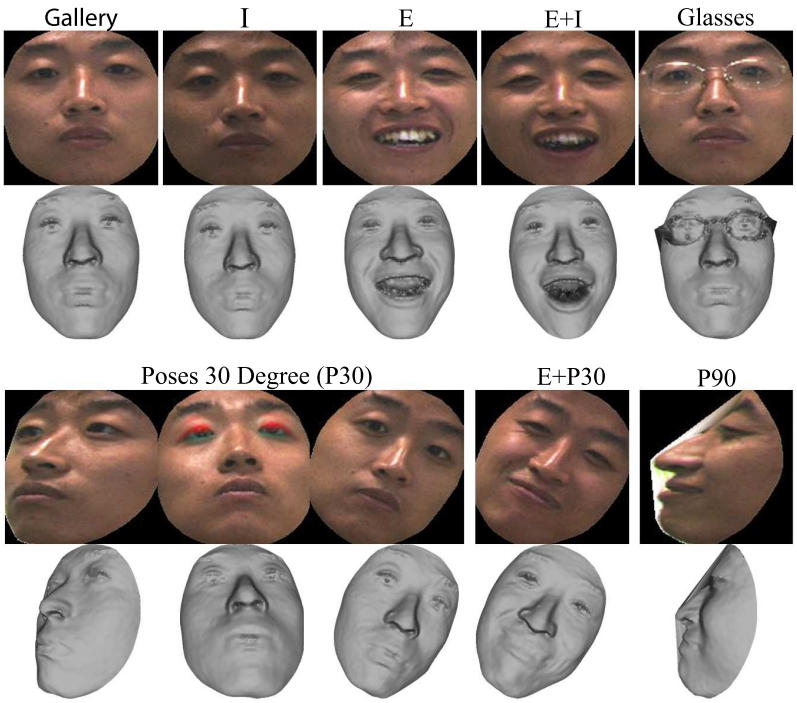


Figure 7.12: Sample images in **CASIA**. Note that images are shown after contrast enhancement for better visualization.

Table 7.4: Results on **CASIA** using first-neutral (100) vs. all (3663) protocol.

	SVM	SRC	RSC	Mian *et al.* (2007)	Xu *et al.* (2009)	**Proposed**
I(400)	80.5	86.0	88.5	99.8	98.3	**99.8**
E(500)	57.6	74.4	76.6	94.2	90.0	**98.0**
E+I(500)	58.0	73.8	77.8	93.2	93.3	**97.8**
P30^o(700)	20.3	26.7	36.3	96.1	91.0	**98.3**
P60^o(200)	1.0	4	6	47	91.0	**96.5**
P90^o(200)	1.0	0.5	1.5	5	-	**81.0**
E+P30^o(700)	19.0	26.4	34.9	93.7	87.9	**96.0**
E+P60^o(200)	1.0	4	5.5	45.5	79.0	**95.0**
E+P90^o(200)	1.0	2.5	2.0	4.5	-	**77.0**
Glasses(63)	54.0	79.4	82.5	95.3	-	**100**
All(3663)	33.2	41.7	46.6	80.0	-	**95.6**
All Xu's(3200)	36.8	46.0	51.5	89.1	90.7	**97.5**

use the training set to derive DCS and DNM transforms and the gallery set to form the coding dictionary. Similar to the case of Bosphorus database, we synthesize five samples from each gallery image as described in Section 7.9. The rank-1 identification rates are reported in Table 7.4 and a similar pattern to former experimental results can be observed.

The proposed algorithm outperforms all other methods and achieves an average of 95.6% identification rate using all the probes and 97.5% when using Xu et al.'s limited probe set. No obvious drop in performance is observed across most of the variations. Even under the challenging case of profile view, we can maintain an accuracy of 81% without expression and 77% with a smiling expression. To the best of our knowledge, these are the best results reported for the CASIA databases under the first versus all protocol.

7.11 Evaluation on FRGC

This database contains 4007 images of 465 subjects[4] captured across multiple sessions and with various expressions. The de-facto standard in FRGC for identification is the first versus all protocol. Figure 7.13 shows the CMC curve of our algorithm. Although synthesized gallery images are generated in a similar way (see Section 7.9), we only use them in the last iteration of our multi-channel weighted sparse coding to avoid the computational cost associated with large dictionary ($465 \times 5 = 2325$).

The proposed algorithm achieved a rank-1 identification rate of 95.2% which is comparable to the state-of-the-art (Mian *et al.*, 2007; Faltemier *et al.*, 2008; Queirolo *et al.*, 2010; Spreeuwers, 2011). Although some existing 3D methods in the literature report higher performance compared to our algorithm, most of them are evaluated only on the FRGC dataset. These methods may have been optimized for high resolution data and it is difficult to say how they will scale to low resolution data. Moreover, the FRGC database mostly contain expression variations, no occlusions and only minor pose variations. Therefore, it is difficult to perceive the robustness of these algorithms to occlusions and pose variations. Furthermore, the de-facto standard protocol allowing only a single gallery image per subject does not favor our proposed algorithm. In practical applications, multiple 3D scans of the same subject can easily be obtained during enrollment. Nevertheless, our results show that our method does not fail even under the unfavorable situation of single gallery image per subject.

[4]Confirmed in (Queirolo *et al.*, 2010) and by the FRGC organizers that subject ID 04643 and 04783 are the same person.

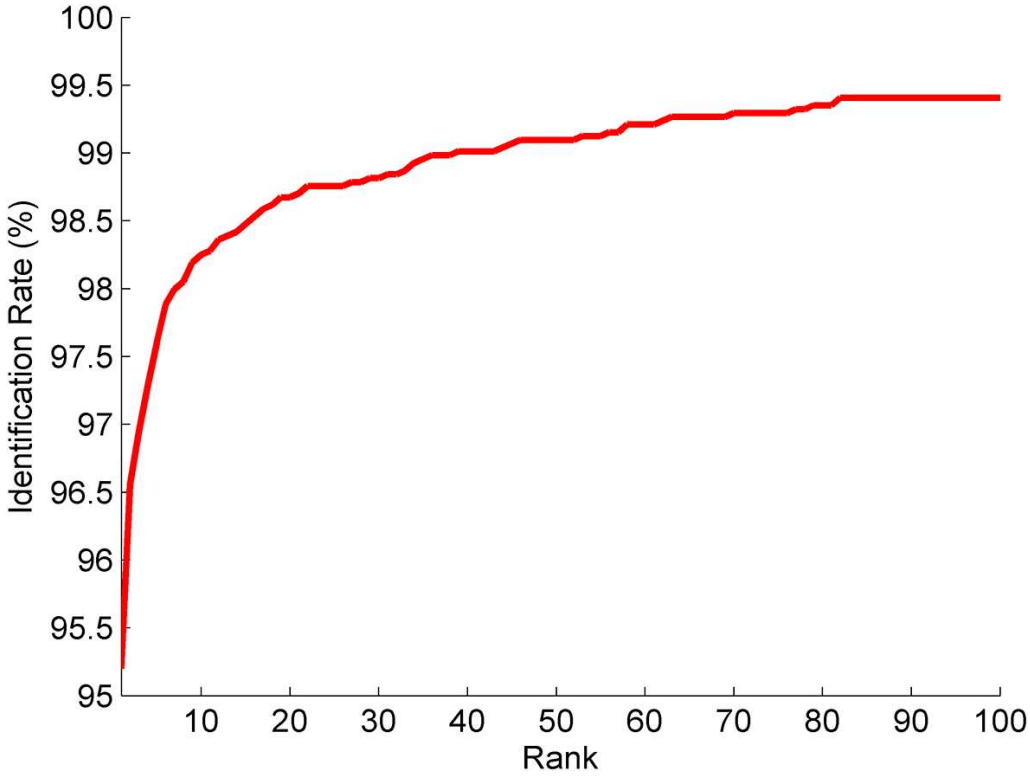


Figure 7.13: CMC curve for first-neutral (465) vs. all (3542) protocol on **FRGC**.

7.12 Time Complexity

The proposed algorithm was implemented on an Intel Core2 Quad 3GHz CPU with 4GB RAM using a 64-bit Matlab. It takes a total of 15 seconds to identify one query face from a gallery of 100 images. The time increases to 25 seconds for a larger dictionary of 2325 images. The average time per match is less than 0.5 seconds. Most of the time is taken by ICP based registration. We used up to 30 ICP iterations to achieve fine registration in our experiments which takes 10 seconds on the average. This can be speeded up to just 2 seconds by using only 5 ICP iterations which causes little drop in accuracy and can correct the pose even in the case of profile faces. Symmetric filling step requires searching for nearest neighbor points which takes 0.5 seconds. The histogram equalization takes 2 seconds. The DCS and DNM are linear transformations and take less than a millisecond.

Sparse coding is performed using the SPAMS (Mairal *et al.*, 2010) package with Matlab interface. Despite the overhead of Matlab function calls, SPAMS is able to return the solution in 0.05 seconds on a dictionary of 100 images with 3072 ($32\times32\times3$) feature dimension. Note that our Multi-channel Weighted Sparse Coding (MWSC) computes a mask of weights for each pixel iteratively. With our proposed parameter setting, about

40% of pixels receive a weight <= 0.001 and 10% of the pixels are located outside of the face after resampling to a square grid. These pixels are removed and therefore, the effective feature dimension is around 1100. Furthermore, about 4.5 iterations are required on the average for MWSC to converge and we apply MWSC on both 2D and 3D images. Taking all these into account, a total of 0.3 seconds are required to recognize one face in the Bosphorus and CASIA databases with a gallery size of 105 and 100 respectively. When synthesized images are generated, the dictionary size increases five times and the algorithm converges in 2 seconds. For CurtinFaces with 936 gallery images (dictionary size), a total of 3 seconds are required. For FRGC, as mentioned in Section 7.11, the synthesized gallery images are used only in last iteration. A total of 9.5 seconds are needed because the last iteration on a gallery of 2325 images consumes most of the time (about 8 seconds). Lastly, computation of the class-wise distance requires less than 0.05 seconds in all cases.

Note that excluding the sparse coding step, our algorithm is implemented fully in Matlab which makes it slow. Moreover, for consistency and comparison with existing 3D face recognition techniques, we focused on achieving higher accuracy at the cost of computational complexity. Implementation in a faster programming language will considerably increase the speed of our algorithm given the iterative nature of many components of our algorithm. Further, higher speed can also be achieved by algorithmic changes such as fewer ICP iterations and smaller dictionary sizes albeit at the cost of minor drop in accuracy.

7.13 Summary

In this chapter, we have made the following contributions. Firstly, we propose a practical algorithm for robust face recognition that works equally well on low and high resolution RGB-D data. These algorithm consists of multiple novel components. The proposed Canonical Preprocessing procedures can correct poses and estimate the full frontal-view from side-view of a face. The proposed multi-channel Discriminant Transform (MDT) can increase the discriminative power of any multi-channel data. The proposed multi-channel Weighted Sparse Coding (MWSC) method, which computes varies weighting for sparse coding using multiple channels, is better than single channel in terms of robustness to variation in imaging conditions. Secondly, we analyzed and experimentally showed that existing 3D face recognition methods designed for high resolution 3D data are not suitable for low resolution Kinect data. However, using our proposed approach, accurate face recognition can still be performed reliably with Kinect. Lastly, our method outperforms

existing techniques under challenging conditions. Specifically, we can maintain consistent performance even under simultaneous variations in pose, expression, illumination and disguise. Our method can also handle, to some extend, faces in the extreme case of profile-view. Moreover, no parameter tuning is required to achieve satisfactory performance in arbitrary cases. State-of-the-art results on the CurtinFaces, Bosphours and CASIA databases are reported.

Nevertheless, our proposed method inherited some problems from the original sparse coding framework. In order to perform sparse coding classification, the person of the probe image must have previously enrolled into the coding dictionary. Therefore, it can be used for face identification application only, but not for verification problem. The objective of verification problem is to determine whether two images are of the same identity, where this identity may not appear in the training set, and thus sparse coding can not be used.

In future research, we will try to generalize the sparse coding framework to work for verification problem. One possible direction is to derive personal signature using sparse coding. Given a image, it can be sparsely coded using a referencing dictionary. This referencing dictionary must not contain the identity of the image to be coded. We expect that the same person will have similar sparse code pattern while different people will have very different one and therefore can be used as an unique biometric signature. Verification can then be done by comparing two signatures. The advantage of of sparse coding based signature is that it is robust to noise and disguise. Another advantage of this proposal is that we can control the time complexity by controlling the size of referencing dictionary, whereas in tradition framework, time complexity increases dramatically with increasing number of people, since the dictionary size is also increased.

Chapter Acknowledgements

This work was partially supported by the Australian Research Council (ARC) Discovery Grant DP110102399. Portion of the research in this chapter use the CASIA-3D FaceV1 collected by the Chinese Academy of Sciences' Institute of Automation (CASIA).

Chapter 8

Conclusions and Future Directions

This thesis addresses the problem of automatic face recognition where user cooperation is not possible. We demonstrate that in such context, color and 3D information are essential to ensure recognition robustness of the system. Starting from existing color face recognition methods, we have shown some of their weaknesses and proposed corresponding improvements. We also investigate into the feasibility of low resolution 3D data for face recognition. Finally, an algorithm that utilizes color and 3D data for robust face recognition is proposed. A few conclusions can be drawn through out our research which can inspire some future research directions. These will be discussed in the perspective of color spaces, color methods and RGB-D methods in this chapter.

8.1 Color Spaces

We can conclude from Chapter 3 that using only one Discriminant Color Space (DCS) for the holistic human face image is suboptimal. Since human faces display different colors at different locations, performance can always be improved by deriving different color spaces that are locally optimal for each patch of the image. This idea was realised into two successful algorithms namely the Pixel-level Discriminant Color Space (PLDCS) and Block-wise Discriminant Color Space (BWDCS). The BWDCS method is actually a generalization of the DCS method with adjustable block sizes, while PLDCS is an extreme case of BWDCS with block size of one pixel. In our experiments involving five databases with six test sets, the proposed BWDCS can improve the recognition accuracy of subspace methods by about 7% over DCS. The projection vectors found by PLDCS are different to the one found by DCS on an average of 41 degrees in angle. PLDCS also exhibits lower inter-component correlation and hence higher discriminative power. Therefore, when seeking color spaces for face recognition, it may be more desirable to operate on local patches of the image instead of the holistic image.

As shown in Chapter 4 that complementary information for face recognition resides in multiple color components across different linear and non-linear color spaces, however

most of the existing color spaces consist of only three color components and are linearly transformed from RGB. We proposed the Multiple Color Fusion (MCF) algorithm that obtains a color model for face recognition with more than three color components from both linear and non-linear color spaces. This algorithm searches for the optimal color combinations offline using the training samples with a greedy approach. In our experiments, the proposed system adopted a color model with 12 color components and achieved 80% recognition accuracy on the FRGC database, outperforming other state-of-the-art color spaces by 3% to 8%. On AR database, 8 color components were adopted and 96% accuracy was achieved, while other competing color spaces achieved only 91% to 93%. We observed that although using a random combination of 12 and 8 color components on FRGC and AR databases respectively can sometimes achieve high performance as well, but this performance is not stable and is still lower than the proposed method. We have also shown that different color components have different variation distribution across the face image. For example, some color has higher variance near the eyes while some has higher variance around the mouths. We believe that this is the reason why performance can be improved when fusing multiple color components. Therefore, when seeking color representation for face data, it may be better not to limit to only three color components.

Based on the above findings, a few future directions will be possible. For example, we can expect further improvement of BWDCS by using overlapping blocks, as well as fusing the decisions from serval block sizes, instead of using one fixed block size. The MCF method can be improved by replacing the greedy search with some state-of-art feature selection methods. There are other research done on decision score level fusion which can also be employed to replace the sum rule technique used by MCF, or we may formulate the multiple color selection problem as an optimization problem to derive a more elegant solution. Lastly, many other recently proposed methods are still being developed using gray-scale images, therefore the state of the art can be advanced by reformulating them using the color spaces or model we proposed.

8.2 Color Recognition Methods

Although the popular sparse coding technique is claimed to be feature invariant for face recognition, we have shown in Chapter 5 that its performance can be affected by color. We have presented an algorithm to formulate the sparse coding method on color images based on image level fusion, which has achieved the state-of-the-art performance. We also proposed the concept of *Correctness* and *Discriminativeness* (DIS) which describes the discriminative power of a sparse coding solution. We found that color can always increase

DIS and thus increases the recognition performance, regardless of the choice of dimensions and features. In addition, the choice of color space can greatly affect the performance. Furthermore, formulating sparse coding on color image also facilitates error correction and is more robust in case of random pixel corruption. When recognizing occluded face images, the Correntropy Sparse Representation (CESR), which is a sparse coding method with pixel weighting, is an effective solution. Its performance can be further increased with DCS, because more accurate pixel weighting is derived. In particular, we can achieve 100% for sunglasses occlusion and 95% for scarves occlusion, while Wright *et al.* (2009) achieved only 87% and 59.5% respectively using the popular Sparse Representation Classifier (SRC) method.

When dealing with images that have large variations, the multilinear technique is very promising. We have shown in Chapter 6 that the performance of Rana *et al.* (2009)'s MPCA-PS method can be improved further when integrated with Wang *et al.* (2011)'s Tensor Discriminant Color Space (TDCS) method. We found that the formulation of MPCA-PS requires vectorization of images, which discard important structural information. On the other hand, the TDCS method has linear formulation, which can not deal with non-linear variation like poses. We proposed the Multilinear Color Tensor Discriminant (MCTD) model, which utilizes the advantage of both methods. As a result, MCTD outperforms MPCA-PS and TDCS on test data that consist of large variation in poses, expressions and lighting.

One of the the future directions that may be interesting is to statistically derive optimal color and feature space for sparse coding in terms of the maximum discriminativeness. We need an alternative discriminativeness measurement that can be computed without recovering the sparse representation, so that it can be used as the optimization criterion. Another possible future direction is related to MCTD. Although MCTD tries to preserve the color image structure using tensor, the unfolding operation is still mechanical rather than mathematical. To fully preserve the image structure, a mathematical representation may be needed such as using the quaternion matrix (Sun *et al.*, 2011).

8.3 Recognition with Color and Depth

Existing commercial 3D acquisition devices, that are high in resolution, are slow in acquisition speed, high in cost and bulky in size, which limit their applications. Although the recent release of Kinect overcame these problems, we found that it has low depth data quality. Nevertheless, Chapter 6 shows that such low quality data is still useful for face

recognition when utilized effectively. An algorithm was presented that utilizes the RGB-D (Red, Green, Blue and Depth) data from Kinect, which extracts multiple features and fusing them at the feature level using a novel technique namely the Finer Feature Fusion. We found that the performance of our algorithm was decreased by about 3% to 7% when the depth data is ignored. On the other hand, when utilizing the depth data alone, a high recognition rate of 91.3% was achieved on the CurtinFaces database, which has test image of 52 subjects in different poses, expressions and lighting conditions. The high recognition rate suggests that 3D face recognition can be performed well using Kinect even though the data quality is low, and therefore we can take advantage of its high speed and low cost.

In Chapter 7, we proposed a robust face recognition algorithm that performs equally well on low and high resolution 3D data. During preprocessing, facial symmetry is exploited at the 3D point cloud level to obtain a canonical frontal view irrespective of the initial pose. Depth data is converted to XYZ normal maps. We proposed the Multi-channel Discriminant Transforms (MDT) which converts RGB to DCS (Discriminant Color Space) and normal maps to DNM (Discriminant Normal Maps). A Multi-channel Robust Sparse Coding method is proposed that codes the multiple channels (DCS or DNM) of a test image as a sparse combination of training samples with different pixel weighting. Weights are calculated dynamically in an iterative process to achieve robustness to variations in pose, illumination, facial expressions and disguise. In contrast to existing techniques, our multi-channel approach is more robust to variations. Reconstruction errors of the test image (DCS and DNM) are normalized and fused to decide its identity. The proposed algorithm was evaluated on four public databases. We achieved 98.4% identification rate on CurtinFaces, a Kinect database with 4784 RGB-D images of 52 subjects. Using a first versus all protocol on the Bosphorus, CASIA and FRGC v2 databases, we achieved 97.6%, 95.6% and 95.2% identification rates respectively. To the best of our knowledge, for CurtinFaces, Bosphorus and CASIA, these are the highest identification rates reported to date.

In future work, we will improve our 3D algorithm in each stages. First of all, although the canonical preprocessing can estimate full frontal poses by utilizing facial symmetry, the illumination correction is by histogram equalization on the 2D texture. With 3D information, surface reflection can be modelled and we can perform a more accurate illumination correction. Secondly, the current system is based on DCS. This can be improved by our previous works on color spaces. We can derive the color space block-wise using more than three color components to further improve the performance. The block-wise idea can also be applied to the DNM. Moreover, instead of fixed blocks, we can divide the face into components using 3D information and perform component-wise operations.

Lastly, as mentioned in Section 7.13, the sparse coding based recognition method used in our algorithm only support face identification problems, since the coding dictionary is required to contain images of all users. We want to derive a personal signature based on sparse coding, then both identification and verification can be done by comparing the signature.

Bibliography

Ahonen, T., Hadid, A., and Pietikainen, M. (2004). Face recognition with local binary patterns. *Lecture Notes in Computer Science*, **3021**, 469–481.

Al-Osaimi, F., Bennamoun, M., and Mian, A. (2012). Spatially optimized data-level fusion of texture and shape for face recognition. *IEEE Trans. on Image Processing*, **21**(2), 859–872.

Alyuz, N., Gokberk, B., Spreeuwers, L., Veldhuis, R., and Akarun, L. (2012). Robust 3D face recognition in the presence of realistic occlusions. In *IAPR Int'l Conf. on Biometrics*, pages 111–118.

Belhumeur, P. N., Hespanha, J. a. P., and Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on Pattern Anal. and Machine Intel.*, **19**(7), 711–720.

Besl, P. and McKay, N. (1992). A method for registration of 3-d shapes. *IEEE Trans. on Pattern Anal. and Machine Intel.*, **14**(2), 239–256.

Biometrics, N. S. O. (2006). Biometrics testing and statistics.

Boehnen, C. and Flynn, P. (2005). Accuracy of 3D scanning technologies in a face scanning scenario. In *Int'l Conf. on 3-D Digital Imaging and Modeling*, pages 310–317.

Bowyer, K., Chang, K., and Flynn, P. (2006). A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition. *Computer Vision and Image Understanding*, **101**(1), 1–15.

Bronstein, A., Bronstein, M., and Kimmel, R. (2007). Expression-invariant representations of faces. *IEEE Trans. on Image Processing,*, **16**(1), 188–197.

Buchsbaum, W. (1975). *Color TV Servicing, third ed.* Prentice-Hall, Englewood Cliffs, NJ, USA.

CASIA (2004). CASIA-3D Face V1. `http://biometrics.idealtest.org/`.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Trans. on Intel. Systems and Technology*, **2**, 27:1–27:27.

Cignoni, P. (2012). Meshlab. `www.meshlab.sourceforge.net`.

Deng, W., Hu, J., Guo, J., Cai, W., and Feng, D. (2010a). Robust, accurate and efficient face recognition from a single training image: A uniform pursuit approach. *Pattern Recogn.*, **43**(5), 1748–1762.

Deng, Z.-X., Dai, D.-Q., and Li, X.-X. (2010b). Low-resolution face recognition via color information and regularized coupled mappings. *Chinese Conference on Pattern Recogn.*, pages 779–783.

Faltemier, T., Bowyer, K., and Flynn, P. (2007). Using a multi-instance enrollment representation to improve 3D face recognition. In *IEEE Int'l Conf. on Biometrics: Theory, Applications, and Systems*, pages 1–6.

Faltemier, T., Bowyer, K., and Flynn, P. (2008). A region ensemble for 3-d face recognition. *IEEE Trans. on Information Forensics and Security*, **3**(1), 62–73.

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

Gross, R., Matthews, I., and Baker, S. (2004). Appearance-based face recognition and light-fields. *IEEE Trans. on Pattern Anal. and Machine Intel.*, **26**(4), 449–465.

Hancock, P. (2004). The psychological image collection at stirling (pics). `http://pics.psych.stir.ac.uk/`.

He, R., Zheng, W., and Hu, B. (2010). Maximum correntropy criterion for robust face recognition. *IEEE Trans. on Pattern Anal. and Machine Intel.*, **33**(8), 1561–1576.

He, X., Yan, S., Hu, Y., Niyogi, P., and Zhang, H.-J. (2005). Face recognition using laplacianfaces. *IEEE Trans. Pattern Anal. Machine Intell.*, **27**(3), 328–340.

J. Phillips, H. Moon, S. R. P. R. (2000). Evaluation methodology for face-recognition algorithms. *IEEE Trans. on Pattern Anal. and Machine Intel.*, **22**, 1090–1104.

Jain, A., Nandakumar, K., and Ross, A. (2005). Score normalization in multimodal biometric systems. *Pattern Recogn.*, **38**, 2270–2285.

Jia, C.-C., Wang, S.-J., Peng, X.-J., Pang, W., Zhang, C.-Y., Zhou, C.-G., and Yu, Z.-Z. (2012). Incremental multi-linear discriminant analysis using canonical correlations for action recognition. *Neurocomputing*, **83**, 56–63.

Jiang, X., Mandal, B., and Kot, A. (2008). Eigenfeature regularization and extraction in face recognition. *IEEE Trans. on Pattern Anal. and Machine Intel.*, **30**, 383–394.

Jing, X., Liu, Q., Lan, C., Man, J., Li, S., and Zhang, D. (2010). Holistic orthogonal analysis of discriminant transforms for color face recognition. In *IEEE Int'l Conf. on Image Processing*, pages 3841 –3844.

Kakadiaris, I., Passalis, G., Toderici, G., Murtuza, M., Lu, Y., Karampatziakis, N., and Theoharis, T. (2007). Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *IEEE Trans. on Pattern Anal. and Machine Intel.*, **29**(4), 640–649.

Khoshelham, K. and Elberink, S. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, **12**(2), 1437–1454.

Kim, J., Choi, J., Yi, J., and Turk, M. (2005). Effective representation using ica for face recognition robust to local distortion and partial occlusion. *IEEE Trans. on Pattern Anal. and Machine Intel.*, **27**, 1977–1981.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, **401**(6755), 788–791.

Lei, Y., Bennamoun, M., and El-Sallam, A. (2013). An efficient 3D face recognition approach based on the fusion of novel local low-level features. *Pattern Recogn.*, **46**(1), 24–37.

Li, H., Huang, D., Lemaire, P., Morvan, J., and Chen, L. (2011). Expression robust 3D face recognition via mesh-based histograms of multiple order surface differential quantities. In *IEEE Int'l Conf. on Image Processing*, pages 3053–3056.

Li, S., Hou, X., Zhang, H., and Cheng, Q. (2001). Learning spatially localized, parts-based representation. *IEEE Int'l Conf. on Computer Vision and Pattern Recogn.*, pages 1–6.

Li, Z., Lin, D., and Tang, X. (2009). Nonparametric discriminant analysis for face recognition. *IEEE Trans. on Pattern Anal. and Machine Intel.*, **31**, 755 –761.

Liu, C. (2008). Learning the uncorrelated, independent, and discriminating color spaces for face recognition. *IEEE Trans. on Information Forensics and Security*, **3**(2), 213–222.

Liu, C. and Wechsler, H. (2002). Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Trans. on Image Processing*, **11**(4), 467–476.

Liu, W., Pokharel, P., and Principe, J. (2007). Correntropy: Properties and applications in non-gaussian signal processing. *IEEE Trans. on Signal Processing,*, **55**(11), 5286–5298.

Liu, Z. and Liu, C. (2010). Fusion of color, local spatial and global frequency information for face recognition. *Pattern Recogn.*, **43**(8), 2882–2890.

Lu, J., Plataniotis, K. N., and Venetsanopoulos, A. N. (2005). Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition. *Pattern Recogn. Letter*, **26**(2), 181–191.

Lucey, P., Cohn, J., Kanade, T., Saragih, J., Ambadar, Z., and Matthews, I. (2010). The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *IEEE Workshops on Computer Vision and Pattern Recogn.*, pages 94 –101.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, **11**, 19–60.

Martinez, A. and Benavente, R. (1998). The AR face database. CVC Technical Report #24.

Mian, A. (2011). Illumination invariant recognition and 3d reconstruction of faces using desktop optics. *Opt. Express*, **19**(8), 7491–7506.

Mian, A., Bennamoun, M., and Owens, R. (2007). An efficient multimodal 2D-3D hybrid approach to automatic face recognition. *IEEE Trans. on Pattern Anal. and Machine Intel.*, **29**(11), 1927–1943.

Naseem, I., Togneri, R., and Bennamoun, M. (2010). Linear regression for face recognition. *IEEE Trans. on Pattern Anal. and Machine Intel.*, **32**(11), 2106–2112.

Nefian, A. V. (2007). Georgia Tech face database. `http://www.anefian.com/research/face_reco.htm`.

Ohta, Y. (1985). *Knowledge-based interpretation of outdoor natural color scenes*. Pitman Publishing, Inc., Marshfield, MA, USA.

P. Ekman, W. F. (1978). Facial action coding system (FACS): Manual. *Palo Alto: Consulting Psychologists Press*.

Passalis, G., Perakis, P., Theoharis, T., and Kakadiaris, I. (2011). Using facial symmetry to handle pose variations in real-world 3D face recognition. *IEEE Trans. on Pattern Anal. and Machine Intel.*, **33**(10), 1938–1951.

Phillips, P., Flynn, P., Scruggs, T., Bowyer, K., Chang, J., Hoffman, K., Marques, J., Min, J., and Worek, W. (2005). Overview of the face recognition grand challenge. In *IEEE Conf. on Computer Vision and Pattern Recogn.*, pages 947–954.

Queirolo, C., Silva, L., Bellon, O., and Segundo, M. (2010). 3D face recognition using simulated annealing and the surface interpenetration measure. *IEEE Trans. on Pattern Anal. and Machine Intel.*, **32**(2), 206–219.

Rajapakse, M., Tan, J., and Rajapakse, J. (2004). Color channel encoding with nmf for face recognition. In *Int'l Conf. on Image Processing*, volume 3, pages 2007–2010.

Rana, S., Liu, W., Lazarescu, M., and Venkatesh, S. (2009). A unified tensor framework for face recognition. *Pattern Recogn.*, **42**, 2850–2862.

Ross, A. and Jain, A. (2003). Information fusion in biometrics. *Pattern Recogn. Letters*, **24**, 2115–2125.

Savran, A., Alyüz, N., Dibeklioğlu, H., Çeliktutan, O., Gökberk, B., Sankur, B., and Akarun, L. (2008). Bosphorus database for 3D face analysis. *Biometrics and Identity Management*, pages 47–56.

Savran, A., Sankur, B., and Taha Bilge, M. (2012). Comparative evaluation of 3D vs. 2D modality for automatic detection of facial action units. *Pattern Recogn.*, **45**(2), 767–782.

Shan, C., Gong, S., and McOwan, P. W. (2009). Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, **27**(6), 803 – 816.

Sharma, A. and Jacobs, D. (2011). Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. In *IEEE Conf. on Computer Vision and Pattern Recogn.*, pages 593–600.

Sharon, Y., Wright, J., and Ma, Y. (2007). Computation and relaxation of conditions for equivalence between l 1 and l 0 minimization. Technical report.

Sharon, Y., Wright, J., and Ma, Y. (2009). Minimum sum of distances estimator: Robustness and stability. In *American Control Conference*, pages 524–530.

Shih, P. and Liu, C. (2006). Improving the face recognition grand challenge baseline performance using color configurations across color spaces. *Proceedings - Int'l Conf. on Image Processing, ICIP*, pages 1001–1004.

Sim, T., Baker, S., and Bsat, M. (2001). The CMU pose, illumination, and expression (PIE) database of human faces. Technical Report CMU-RI-TR-01-02, The Robotics Institute, Carnegie Mellon University.

Spreeuwers, L. (2011). Fast and accurate 3D face recognition. *Int'l Journal of Computer Vision*, **93**(3), 389–414.

Sun, Y., Chen, S., and Yin, B. (2011). Color face recognition based on quaternion matrix representation. *Pattern Recogn. Letters*, **32**(4), 597–605.

Tan, X. and Triggs, B. (2010). Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Trans. Image Processing*, **19**(6), 1635–1650.

Tenllado, C., Gómez, J. I., Setoain, J., Mora, D., and Prieto, M. (2010). Improving face recognition by combination of natural and gabor faces. *Pattern Recogn. Letters*, **31**, 1453–1460.

Thomas, M., Kambhamettu, C., and Kumar, S. (2008). Face recognition using a color subspace lda approach. *Proceedings - Int'l Conf. on Tools with Artificial Intelligence, ICTAI*, **1**, 231–235.

Tibshirani, R. (1994). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, **58**, 267–288.

Toderici, G., Passalis, G., Zafeiriou, S., Tzimiropoulos, G., Petrou, M., Theoharis, T., and Kakadiaris, I. (2010). Bidirectional relighting for 3D-aided 2d face recognition. In *IEEE Conf. on Computer Vision and Pattern Recogn.*, pages 2721–2728.

Turk, M. and Pentland, A. (1991). Face recognition using eigenfaces. In *IEEE Conf. on Computer Vision and Pattern Recogn.*, pages 586–591.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.

Vasilescu, M. A. O. and Terzopoulos, D. (2002). Multilinear Image Analysis for Facial Recognition. In *Int'l Conf. on Pattern Recogn.*, volume 2, pages 511–514, Quebec City, Canada.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE Conf. on Computer Vision and Pattern Recogn.*, pages I511–I518.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *Int'l Journal of Computer Vision*, **57**(2), 137–154.

Wang, S.-J., Yang, J., Zhang, N., and Zhou, C.-G. (2011). Tensor discriminant color space for face recognition. *IEEE Trans. on Image Proccessing*, **20**(9), 2490–2501.

Wang, Y. and Wu, Y. (2010). Face recognition using intrinsicfaces. *Pattern Recogn.*, **43**(10), 3580–3590.

Wang, Y., Jia, Y., Hu, C., and Turk, M. (2005). Non-negative matrix factorization framework for face recognition. *Int'l Journal of Pattern Recogn. and Artificial Intel.*, **19**(4), 495511.

Wang, Y., Liu, J., and Tang, X. (2010). Robust 3D face recognition by local shape difference boosting. *IEEE Trans. on Pattern Anal. and Machine Intel.*, **32**(10), 1858–1870.

Weeks, A. (1996). *Fundamentals of Electronic Image Processing*. SPIE Optical Engineering Press, Washington, USA.

Wright, J., Yang, A., Ganesh, A., Sastry, S., and Ma, Y. (2009). Robust face recognition via sparse representation. *IEEE Trans. on Pattern Anal. and Machine Intel.*, **31**(2), 210–227.

Xu, C., Li, S., Tan, T., and Quan, L. (2009). Automatic 3D face recognition from depth and intensity gabor features. *Pattern Recogn.*, **42**(9), 1895–1905.

Yang, J. and Liu, C. (2008a). Color image discriminant models and algorithms for face recognition. *IEEE Trans. on Neural Networks*, **19**(12), 2088–2098.

Yang, J. and Liu, C. (2008b). A discriminant color space method for face representation and verification on a large-scale database. In *Int'l Conf. on Pattern Recogn*, pages 1 –4.

Yang, J., Liu, C., and Zhang, L. (2010a). Color space normalization: Enhancing the discriminating power of color spaces for face recognition. *Pattern Recogn.*, **43**(4), 1454–1466.

Yang, J., Liu, C., and Yang, J.-Y. (2010b). What kind of color spaces is suitable for color face recognition? *Neurocomputing*, **73**(10-12), 2140–2146.

Yang, M., Zhang, L., Yang, J., and Zhang, D. (2011). Robust sparse coding for face recognition. In *IEEE Conf. on Computer Vision and Pattern Recogn.*, pages 625–632.

Yao, Y.-F., Jing, X.-Y., and Wong, H.-S. (2007). Face and palmprint feature level fusion for single sample biometrics recognition. *Neurocomputing*, **70**(7-9), 1582 – 1586.

Yin, L., Wei, X., Sun, Y., Wang, J., and Rosato, M. (2006). A 3D facial expression database for facial behavior research. In *Int'l Conf. on Automatic Face and Gesture Recognition*, pages 211–216.

Yip, A. and Sinha, P. (2002). Role of color in face recognition. *Journal of Vision*, **2**(7).

Zhang, L., Yang, M., and Feng, X. (2011). Sparse representation or collaborative representation: Which helps face recognition? In *IEEE Int'l Conf. on Computer Vision*, pages 471 –478.

Zheng, Z., Zhao, J., and Yang, J. (2006). Gabor feature based face recognition using supervised locality preserving projection. In *Int'l conf. on Advanced Concepts For Intell. Vision Systems*, pages 644–653.

Zuiderveld, K. (1994). Graphics gems iv. chapter Contrast limited adaptive histogram equalization, pages 474–485. Academic Press Professional, Inc., San Diego, CA, USA.

Zuo, W., Wang, K., Zhang, D., and Zhang, H. (2007). Combination of two novel lda-based methods for face recognition. *Neurocomputing*, **70**(4-6), 735 – 742.