


2013

Time And Space Efficient Techniques For Facial Recognition

Waleed Alrasheed
University of Central Florida

 Part of the [Electrical and Electronics Commons](#)
Find similar works at: <https://stars.library.ucf.edu/etd>
University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Alrasheed, Waleed, "Time And Space Efficient Techniques For Facial Recognition" (2013). *Electronic Theses and Dissertations, 2004-2019*. 3008.
<https://stars.library.ucf.edu/etd/3008>

TIME AND SPACE EFFICIENT TECHNIQUES FOR FACIAL RECOGNITION

by

WALEED ALRASHEED
B.S. King Saud University, 1999
M.S. University of Central Florida, 2008

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2013

Major Professor: Wasfy B. Mikhael

© 2013 Waleed Alrasheed

ABSTRACT

In recent years, there has been an increasing interest in face recognition. As a result, many new facial recognition techniques have been introduced. Recent developments in the field of face recognition have led to an increase in the number of available face recognition commercial products. However, Face recognition techniques are currently constrained by three main factors: recognition accuracy, computational complexity, and storage requirements. The problem is that most of the current face recognition techniques succeed in improving one or two of these factors at the expense of the others.

In this dissertation, four novel face recognition techniques that improve the storage and computational requirements of face recognition systems are presented and analyzed.

Three of the four novel face recognition techniques to be introduced, namely, Quantized/truncated Transform Domain (QTD), Frequency Domain Thresholding and Quantization (FD-TQ), and Normalized Transform Domain (NTD). All the three techniques utilize the Two-dimensional Discrete Cosine Transform (DCT-II), which reduces the dimensionality of facial feature images, thereby reducing the computational complexity. The fourth novel face recognition technique is introduced, namely, the Normalized Histogram Intensity (NHI). It is based on utilizing the pixel intensity histogram of poses' subimages, which reduces the computational complexity and the needed storage requirements.

Various simulation experiments using MATLAB were conducted to test the proposed methods. For the purpose of benchmarking the performance of the proposed methods, the simulation experiments were performed using current state-of-the-art face recognition techniques, namely, Two Dimensional Principal Component Analysis (2DPCA), Two-Directional Two-Dimensional Principal Component Analysis ((2D)²PCA), and Transform Domain Two Dimensional Principal Component

Analysis (TD2DPCA). The experiments were applied to the ORL, Yale, and FERET databases.

The experimental results for the proposed techniques confirm that the use of any of the four novel techniques examined in this study results in a significant reduction in computational complexity and storage requirements compared to the state-of-the-art techniques without sacrificing the recognition accuracy.

To Fatima Alkhalefah, Khaled Alrasheed,
and my little family Alanoud Alfawaz, Lamya, and Yousef.

ACKNOWLEDGMENTS

First, and foremost, all praise and thanks are to Allah for his persistent bounty and blessings. Then, I would like to convey my deep gratitude and appreciation to my supervisor, Dr. Wasfy Mikhael, for his unwavering encouragement, guidance and support.

I would like to express my sincere appreciation to Dr. Ronald DeMara, Dr. Michael Haralambous, Dr. Brent Myers, and Dr. Lei Wei for serving on my dissertation committee and providing insightful comments and stimulating thoughtful discussion.

I would like to express my sincere love and gratitude to my father Khaled, my mother Fatima, my lovely wife Alanoud, my daughter Lamya, and my son Yousef. Their love, encouragement and support have been the root of this success.

TABLE OF CONTENTS

| | |
|--|------|
| LIST OF FIGURES | xii |
| LIST OF TABLES | xiii |
| CHAPTER 1: INTRODUCTION | 1 |
| 1.1 Face Recognition | 2 |
| 1.2 Problem | 3 |
| 1.3 Contributions of Our Research | 4 |
| 1.4 Organization of This Dissertation | 5 |
| CHAPTER 2: LITERATURE REVIEW | 7 |
| 2.1 Introduction | 7 |
| 2.2 Principal Component Analysis | 7 |
| 2.3 Two-dimensional Principal Component Analysis | 8 |
| 2.3.1 Training Mode | 8 |
| 2.3.2 Testing Mode | 9 |
| 2.4 Two-directional Two-dimensional Principle Component Analysis | 10 |

| | | |
|---|---|----|
| 2.4.1 | Training Mode | 10 |
| 2.4.2 | Testing Mode | 11 |
| 2.5 | Transform Domain Two-Dimensional Principal Component Analysis | 12 |
| 2.5.1 | Training Mode | 13 |
| 2.5.2 | Testing Mode | 14 |
| | | |
| CHAPTER 3: QUANTIZED/TRUNCATED TRANSFORM DOMAIN TECHNIQUE (QTD) | | |
| | 15 | |
| 3.1 | Introduction | 15 |
| 3.2 | The QTD Algorithm | 15 |
| 3.2.1 | Training Mode | 15 |
| 3.2.2 | Testing Mode | 16 |
| 3.3 | Experimental Results | 20 |
| 3.3.1 | Experimental Results Using ORL Database | 24 |
| 3.3.2 | Experimental Results Using Yale Database | 27 |
| 3.3.3 | Experimental Results Using FERET Database | 30 |
| 3.4 | Conclusion | 33 |
| | | |
| CHAPTER 4: FREQUENCY DOMAIN THRESHOLDING AND QUANTIZATION AL- | | |

| | | |
|---|---|----|
| | GORITHM (FD-TQ) | 34 |
| 4.1 | Introduction | 34 |
| 4.2 | The FD-TQ Algorithm | 34 |
| 4.2.1 | Training Mode | 34 |
| 4.2.2 | Testing Mode | 36 |
| 4.3 | Experimental Results | 40 |
| 4.3.1 | Experimental Results Using ORL Database | 40 |
| 4.3.2 | Experimental Results Using Yale Database | 43 |
| 4.3.3 | Experimental Results Using FERET Database | 46 |
| 4.4 | Conclusion | 49 |
| CHAPTER 5: NORMALIZED HISTOGRAM INTENSITY ALGORITHM (NHI) | | 50 |
| 5.1 | Introduction | 50 |
| 5.2 | The NHI Algorithm | 51 |
| 5.2.1 | Training Mode | 51 |
| 5.2.2 | Testing Mode | 52 |
| 5.3 | Experimental Results | 55 |
| 5.3.1 | Experimental Results Using ORL Database | 55 |

| | | |
|--|---|----|
| 5.3.2 | Experimental Results Using Yale Database | 58 |
| 5.3.3 | Experimental Results Using FERET Database | 61 |
| 5.4 | Conclusion | 64 |
| CHAPTER 6: NORMALIZED TRANSFORM DOMAIN ALGORITHM (NTD) | | 65 |
| 6.1 | Introduction | 65 |
| 6.2 | The NTD Algorithm | 66 |
| 6.2.1 | Training Mode | 66 |
| 6.2.2 | Testing Mode | 66 |
| 6.3 | Experimental Results | 70 |
| 6.3.1 | Experimental Results Using ORL Database | 70 |
| 6.3.2 | Experimental Results Using Yale Database | 73 |
| 6.3.3 | Experimental Results Using FERET Database | 76 |
| 6.4 | Conclusion | 79 |
| CHAPTER 7: SUMMARY AND FUTURE WORK | | 80 |
| 7.1 | Summary | 80 |
| 7.2 | Future Work | 81 |

| | |
|--|-----|
| APPENDIX : SOURCE CODE OF DEVELOPED ALGORITHMS | 83 |
| LIST OF REFERENCES | 105 |

LIST OF FIGURES

| | |
|--|----|
| Figure 3.1: QTD Training mode flow-chart. | 18 |
| Figure 3.2: QTD Testing mode flow-chart. | 19 |
| Figure 3.3: Sample of different images of one person in the ORL database. | 21 |
| Figure 3.4: Sample of different images of one person in the Yale database. | 22 |
| Figure 3.5: Sample of the (fa) and (fb) images of 5 individuals in the FERET database. . . | 23 |
| Figure 4.1: FD-TQ Training mode flow-chart. | 38 |
| Figure 4.2: FD-TQ Testing mode flow-chart. | 39 |
| Figure 5.1: NHI Training mode flow-chart. | 53 |
| Figure 5.2: NHI Testing mode flow-chart. | 54 |
| Figure 6.1: NTD Training mode flow-chart. | 68 |
| Figure 6.2: NTD Testing mode flow-chart. | 69 |

LIST OF TABLES

| | |
|--|----|
| Table 3.1: Recognition accuracy in Experiments I and II for ORL database using the proposed technique (QTD) and existing algorithms (2DPCA, (2D) ² PCA, and TD2DPCA). | 25 |
| Table 3.2: Storage requirements and computational complexity in Experiments I and II for ORL database using the proposed technique (QTD) and existing algorithms (2DPCA, (2D) ² PCA, and TD2DPCA). | 26 |
| Table 3.3: Recognition accuracy in Experiments III and IV for Yale database using the proposed technique (QTD) and existing algorithms (2DPCA, (2D) ² PCA, and TD2DPCA). | 28 |
| Table 3.4: Storage requirements and computational complexity in Experiments III and IV for Yale database using the proposed technique (QTD) and existing algorithms (2DPCA, (2D) ² PCA, and TD2DPCA). | 29 |
| Table 3.5: Recognition accuracy in Experiment V for FERET database using the proposed technique (QTD) and existing algorithms (2DPCA, (2D) ² PCA, and TD2DPCA). | 31 |
| Table 3.6: Storage requirements and computational complexity in Experiment V for FERET database using the proposed technique (QTD) and existing algorithms (2DPCA, (2D) ² PCA, and TD2DPCA). | 32 |

| | |
|--|----|
| Table 4.1: Recognition accuracy in Experiments I and II for ORL database using the proposed technique (FD-TQ) and existing algorithms (2DPCA, (2D) ² PCA, and TD2DPCA). | 41 |
| Table 4.2: Storage requirements and computational complexity in Experiments I and II for ORL database using the proposed technique (FD-TQ) and existing algorithms (2DPCA, (2D) ² PCA, and TD2DPCA). | 42 |
| Table 4.3: Recognition accuracy in Experiments III and IV for Yale database using the proposed technique (FD-TQ) and existing algorithms (2DPCA, (2D) ² PCA, and TD2DPCA). | 44 |
| Table 4.4: Storage requirements and computational complexity in Experiments III and IV for Yale database using the proposed technique (FD-TQ) and existing algorithms (2DPCA, (2D) ² PCA, and TD2DPCA). | 45 |
| Table 4.5: Recognition accuracy in Experiment V for FERET database using the proposed technique (FD-TQ) and existing algorithms (2DPCA, (2D) ² PCA, and TD2DPCA). | 47 |
| Table 4.6: Storage requirements and computational complexity in Experiment V for FERET database using the proposed technique (FD-TQ) and existing algorithms (2DPCA, (2D) ² PCA, and TD2DPCA). | 48 |
| Table 5.1: Recognition accuracy in Experiments I and II for ORL database using the proposed technique (NHI) and existing algorithms (2DPCA, (2D) ² PCA, and TD2DPCA). | 56 |

| | |
|---|----|
| Table 5.2: Storage requirements and computational complexity in Experiments I and II for ORL database using the proposed technique (NHI) and existing algorithms (2DPCA, $(2D)^2PCA$, and TD2DPCA). | 57 |
| Table 5.3: Recognition accuracy in Experiments III and IV for Yale database using the proposed technique (NHI) and existing algorithms (2DPCA, $(2D)^2PCA$, and TD2DPCA). | 59 |
| Table 5.4: Storage requirements and computational complexity in Experiments III and IV for Yale database using the proposed technique (NHI) and existing algorithms (2DPCA, $(2D)^2PCA$, and TD2DPCA). | 60 |
| Table 5.5: Recognition accuracy in Experiment V for FERET database using the proposed technique (NHI) and existing algorithms (2DPCA, $(2D)^2PCA$, and TD2DPCA). | 62 |
| Table 5.6: Storage requirements and computational complexity in Experiment V for FERET database using the proposed technique (NHI) and existing algorithms (2DPCA, $(2D)^2PCA$, and TD2DPCA). | 63 |
| Table 6.1: Recognition accuracy in Experiments I and II for ORL database using the proposed technique (NTD) and existing algorithms (2DPCA, $(2D)^2PCA$, and TD2DPCA). | 71 |
| Table 6.2: Storage requirements and computational complexity in Experiments I and II for ORL database using the proposed technique (NTD) and existing algorithms (2DPCA, $(2D)^2PCA$, and TD2DPCA). | 72 |

| | |
|---|----|
| Table 6.3: Recognition accuracy in Experiments III and IV for Yale database using the proposed technique (NTD) and existing algorithms (2DPCA, $(2D)^2PCA$, and TD2DPCA). | 74 |
| Table 6.4: Storage requirements and computational complexity in Experiments III and IV for Yale database using the proposed technique (NTD) and existing algorithms (2DPCA, $(2D)^2PCA$, and TD2DPCA). | 75 |
| Table 6.5: Recognition accuracy in Experiment V for FERET database using the proposed technique (NTD) and existing algorithms (2DPCA, $(2D)^2PCA$, and TD2DPCA). | 77 |
| Table 6.6: Storage requirements and computational complexity in Experiment V for FERET database using the proposed technique (NTD) and existing algorithms (2DPCA, $(2D)^2PCA$, and TD2DPCA). | 78 |

CHAPTER 1: INTRODUCTION

Face recognition is a daily activity that humans, and even most animals, do smoothly and quickly most of the time. Due to software and hardware improvements in computers and Integrated Circuits, there is an increased interest in automated face image recognition that mimics humans' face recognition skill. Applications of automated face image recognition include: biometric research, security systems, authentication processes, automated surveillance systems, human-computer interfaces. Several commercial face recognition systems have been implemented such as FaceIt, Viisage [1], Cognitec [2], and Facekey [3].

Face recognition is considered to be one type of biometric indicators. Biometric indicators may fit into one of two categories: physiological appearance or behavioral attitude. Both signature recognition and voice recognition are considered behavioral biometric measures. Physiological appearance biometrics, on the other hand, include iris recognition, fingerprint recognition, Deoxyribonucleic Acid (DNA) analysis, and face recognition [4]. However, each biometric indicator is used to fulfill certain application requirements. For example, the need for identifying people led the United States (US) law enforcement in the beginning of the last century to start using fingerprint recognition. Later, fingerprint recognition was developed to be fully automated; nowadays, US passport control officers use an automated fingerprint recognition system to authenticate all the arriving travelers. Nevertheless, two or more biometrics could be used together to achieve a certain level of recognition accuracy.

1.1 Face Recognition

Recently, there has been an increasing interest in face recognition from researchers in the areas of computer vision, biometrics, and pattern recognition [5–8]. Our amazing capability to recognize people and the need for that human activity everyday has motivated researchers from different fields to have a common interest in developing face recognition techniques. Also, the use of face recognition technologies is required in a large number of security and commercial applications.

Face recognition scenarios can be classified into two types: face verification (or authentication) and face identification (or recognition) [9]. Face verification ("Am I who I say I am?") is a one-to-one match that compares a query face image against a template face image when someone's identity is being claimed. Face identification ("Who am I?") is a one-to-many matching process that compares a query face image against all the template images in a face database to determine the identity of the query face. The identification of the test image is done by locating the image in the database which has the closest similarity to the test image [10]. The unknown face image that needs to be identified is called a test image. Thus, the procedure of preparing the test image for a certain algorithm is called the "testing mode procedure. The face database images are called the trained images. Therefore, the procedure of preparing the training images for a certain algorithm is called the "training mode procedure.

In this dissertation, all the experiments are conducted in the face identification scenario.

1.2 Problem

Recently, there has been an increased interest in facial recognition techniques. The success of these techniques is determined by three main factors: recognition accuracy, computational complexity and storage requirements. However, a major problem with most of the existing facial algorithms techniques is one or two of these factors can be improved only at the expense of the other(s).

Principal Component Analysis (PCA) was used in facial recognition by Turk and Pentland [11,12]. PCA starts with transforming the feature images matrices from two dimensional (2D) to one dimensional (1D). Then, all of the 1D feature images matrices are catenated into the image covariance matrix. Thus, the size of the covariance matrix increases as the number of the training images increase. Since the PCA algorithm is based on computing the eigenvectors of the large covariance matrix, the computational complexity of the PCA is high. Therefore, as the PCA succeeds in achieving high recognition accuracy with low storage requirements, the PCA approach requires a high computational complexity.

To solve this weakness, several algorithms have been introduced. One of the successful algorithms is the Two-Dimensional Principal Component Analysis (2DPCA) [13]. It succeeds in reducing the computational complexity by reducing the covariance matrix to be a square matrix with a size equal to the width of the images. Thus, the computation of the eigenvectors of the covariance matrix is made less complex in the 2DPCA than in the PCA. In addition to having reduced the computational complexity compared with the PCA, the 2DPCA algorithm yields higher recognition accuracy [13]. Nevertheless, the storage requirements for the implementation of the 2DPCA is even greater than those of the PCA.

To overcome this drawback, another algorithm called the Two-Directional Two-Dimensional Principal Component Analysis ((2D)²PCA) was developed [14]. It is based on the fact that while the 2DPCA works on either row or column direction, the (2D)²PCA works in both directions at the same time. (2D)²PCA has reduced storage requirements compare to the 2DPCA while maintaining the improved recognition accuracy. Unfortunately, (2D)²PCAs achievements are at the expense of increased computational complexity compared to the 2DPCA.

1.3 Contributions of Our Research

The main contributions of our research can be summarized as follows:

- In this work, we introduce a new facial recognition technique which we call the Quantized/truncated Transform Domain technique (QTD) [15], suitable particularly for large databases. The algorithm has attractive properties with respect to storage requirements and computational complexity in both the training and testing modes. QTD uses the Two-dimensional Discrete Cosine Transform (DCT-II) [16], then truncates the significant DCT coefficients and quantizes those coefficients based on their average energy. The experimental results confirm a significant reduction in the storage and computational requirements and retaining of a high recognition accuracy rate.
- In this dissertation, we introduce a new facial recognition technique which we call Frequency Domain Thresholding and Quantization (FD-TQ) [17]. The algorithm has attractive properties with respect to storage requirements and computational complexity in both the training and testing modes, which make the technique particularly suitable for large databases. The new algorithm is applied to ORL, Yale and FERET databases. The experimental results confirm a significant reduction in the storage and computational requirements compared with

other recently reported techniques, without sacrificing the recognition accuracy.

- A facial recognition technique employing Subimages Normalized Histogram Intensity (NHI) [18] is presented in this work. The algorithm has attractive properties with respect to storage requirements and computational complexity in both the training and testing modes, which make the technique particularly suitable for large databases. The new algorithm is applied to ORL, Yale and FERET databases. The experimental results confirm the significant reduction in the storage and computational requirements compared with recently reported techniques, without sacrificing the recognition accuracy.
- In this work, we introduce a new facial recognition technique which we call the Normalized Transform Domain technique (NTD) [19], suitable particularly for large databases. The algorithm has attractive properties with respect to storage requirements and computational complexity in both the training and testing modes. NTD uses the Two-dimensional Discrete Cosine Transform (DCT-II) [16], then truncates the significant DCT coefficients and normalized those coefficients to get 8-bit representation of the DCT. The experimental results confirm a significant reduction in the storage and computational requirements and retaining of a high recognition accuracy rate.

1.4 Organization of This Dissertation

This dissertation consists of five chapters:

- **Chapter 2** reviews relevant appearance-based face recognition algorithms. This chapter provides background on the state-of-the-art algorithms selected for use in this dissertation.
- **Chapter 3** introduces the Quantized/truncated Transform algorithm (QTD). The different experiments applied to the proposed method are discussed in this chapter. Also, the ex-

perimental results for the QTD are compared to the results for existing face recognition algorithms.

- **Chapter 4** introduces the Frequency Domain Thresholding and Quantization algorithm (FD-TQ). The different experiments applied to the proposed method are discussed in this chapter. Also, the experimental results for the FD-QT are compared to the results for existing face recognition algorithms.
- **Chapter 5** introduces the Normalized Histogram Intensity algorithm (NHI). The different experiments applied to the proposed method are discussed in this chapter. Also, the experimental results for the NHI are compared to the results for existing face recognition algorithms.
- **Chapter 6** introduces the Normalized Transform Domain algorithm (NTD). The different experiments applied to the proposed method are discussed in this chapter. Also, the experimental results for the NTD are compared to the results for existing face recognition algorithms.
- **Chapter 7** summarizes the important points made in this dissertation and indicates future research directions.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

In this chapter a background is provided on appearance-based face recognition to provide context for our study. Descriptions include the training and testing modes procedures for most of them. Also, the strengths and the weaknesses of each algorithm are discussed briefly. The relevant appearance-based algorithms are: (i) Principal Component Analysis (PCA), (ii) Two-dimensional Principal component analysis (2DPCA), (iii) Two-directional Two-dimensional Principal Component Analysis (2D²PCA), (iv) Transform Domain Two-dimensional Principal Component Analysis (TD2DPCA). The results from simulation experiments of these algorithms are used in this dissertation as benchmarks to evaluate the performance of our proposed algorithms.

2.2 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical method that has a key advantage: data dimensionality reduction. PCA is based on Karhunen-Loeve Transformation (KLT) [20,21]. Since 1991, when Pentland and Turk proposed using PCA, namely, the eigenfaces algorithm, for face recognition [11,12], and PCA have turned out to be the algorithm most studied and used in image recognition. Thus, the success of the PCA algorithm has resulted in using it as a base for many techniques, such as [22–30].

PCA starts with transforming the feature images matrices from two dimensions (2D) to one dimension (1D). Then, all 1D feature images matrices are catenated into an image covariance matrix. As the number of training images increases, so does the size of the covariance matrix. Since the PCA

algorithm is based on computing the eigenvectors of the covariance matrix, the computational complexity of the PCA is high. Therefore, though the PCA succeeds in achieving high recognition accuracy with low storage requirements, the PCA algorithm requires a high computational complexity.

2.3 Two-dimensional Principal Component Analysis

The Two-Dimensional Principal Component Analysis (2DPCA) algorithm was introduced to reduce the computational complexity of the PCA. The 2DPCA algorithm represents the images and their covariance matrix. This results in a considerable reduction in the number of coefficients required to represent the images. Thus, the success of the 2DPCA algorithm has resulted in using it as a base for many techniques, such as [31–43]. The algorithm is described in the following section.

2.3.1 Training Mode

In the training mode, the features of the database are extracted and stored as described by steps 1 through 4.

Step 1: The covariance matrix S for N training images is calculated as follows.

$$S = \frac{1}{N} \sum_{i=1}^N (A_i - \bar{A})^T (A_i - \bar{A}) \quad (2.1)$$

Where \bar{A} is the mean matrix of all the training images.

Step 2: A set of k eigenvectors, $V = [V_1, V_2 \dots V_k]$, of size n corresponding to the largest k eigenvalues is obtained for S .

Step 3: The projections of the training images on the set of the dominant eigenvectors V are calculated, yielding the projected feature vectors $Y_{j,i}$.

$$Y_{j,i} = A_i V_j \quad (2.2)$$

where $j = 1, 2, \dots, k$ and $i = 1, 2, \dots, N$

Step 4: The feature matrices of the training images B_i ($i = 1$ to N) are calculated

$$B_i = [Y_{1,i}, Y_{2,i}, \dots, Y_{k,i}] \quad (2.3)$$

and the B_i matrices are stored.

2.3.2 *Testing Mode*

In the testing mode a facial image A_t is presented to the system to be identified. The following steps are followed.

Step 1: The projection of the test image on the set of the dominant eigenvectors V is calculated, yielding the projected feature vectors $Y_{j,t}$.

$$Y_{j,t} = A_t V_j \quad (2.4)$$

where $j = 1, 2, \dots, k$

Step 2: The feature matrix B_t for the testing image is calculated.

$$B_t = [Y_{1,t}, Y_{2,t}, \dots, Y_{k,t}] \quad (2.5)$$

Step 3: Distance measures, such as the Euclidean distances, between the feature matrix of the testing image and the feature matrices of the training images are measured. The minimum distance represents the image to be identified.

2.4 Two-directional Two-dimensional Principle Component Analysis

The Two-directional Two-dimensional Principle Component Analysis ((2D)²PCA) algorithm was introduced to improve on the storage requirements of the 2DPCA. According to [14], 2DPCA works only in one direction (row direction) of the images to compute the covariance matrix S . (2D)²PCA, however, works in two directions (row and column directions) to form two covariance matrices: (i) row direction covariance matrix, S_r , and (ii) column direction covariance matrix, S_c . Consequently, the computational complexity of the (2D)²PCA algorithm is much greater than its of the 2DPCA algorithm. However, the training and testing modes procedures for the (2D)²PCA algorithm are illustrated below.

2.4.1 Training Mode

In the training mode, the features of the database are extracted and stored as described by steps 1 through 5.

Step 1: The row direction covariance matrix S_r for N training images is calculated as follows.

$$S_r = \frac{1}{N} \sum_{i=1}^N (A_i - \bar{A})^T (A_i - \bar{A}) \quad (2.6)$$

Where \bar{A} is the mean matrix of all the training images.

Step 2: The column direction covariance matrix S_r for N training images is calculated as follows.

$$S_c = \frac{1}{N} \sum_{i=1}^N (A_i - \bar{A})(A_i - \bar{A})^T \quad (2.7)$$

Where \bar{A} is the mean matrix of all the training images.

Step 3: Two sets, V_r and V_c , of the k dominant eigenvectors, of sizes n and m respectively, for both covariance matrices S_r and S_c , are obtained, Where $V_r = [V_{r1}, V_{r2}, \dots, V_{rk}]$ and $V_c = [V_{c1}, V_{c2}, \dots, V_{ck}]$.

Step 4: V_r and V_c are used for feature extraction for every training image A_i , yielding the projected feature vectors $Y_{j,i}$.

$$Y_{j,i} = V_c^T A_i V_j \quad (2.8)$$

where $j = 1, 2, \dots, k$ and $i = 1, 2, \dots, N$.

Step 5: The feature matrices of the training images B_i ($i = 1$ to N) are calculated as follows.

$$B_i = [Y_{1,i}, Y_{2,i}, \dots, Y_{k,i}] \quad (2.9)$$

The B_i matrices are stored. It important to note that the size of the feature matrix B_i of each training image A_i is $k \times k$. Thus, the storage requirements of the $(2D)^2$ PCA algorithm are smaller than those of the 2DPCA Algorithm.

2.4.2 Testing Mode

In the testing mode a facial image A_t is presented to the system to be identified. The following steps are followed.

Step 1: The projection of the test image on the V_r and V_c is calculated, yielding the projected feature vectors $Y_{j,t}$.

$$Y_{j,t} = V_c^T A_t V_j \quad (2.10)$$

where $j = 1, 2, \dots, k$.

Step 2: The feature matrix B_t for the testing image is calculated

$$B_t = [Y_{1,t}, Y_{2,t}, \dots, Y_{k,t}] \quad (2.11)$$

Step 3: Distance measures, such as the Euclidean distances, between the feature matrix of the testing image and the feature matrices of the training images are measured. The minimum distance represents the image to be identified.

2.5 Transform Domain Two-Dimensional Principal Component Analysis

The Transform Domain Two-Dimensional Principal Component Analysis (TD2DPCA) algorithm was introduced to reduce the computational complexity of the 2DPCA. The TD2DPCA algorithm represents the images and their covariance matrix in the Transform domain, where energy is compacted into a small number of coefficients. This results in a considerable reduction in the number of coefficients required to represent the images [44, 45]. The algorithm is described in the following section.

2.5.1 Training Mode

In the training mode, the features of the database are extracted and stored as described by steps 1 through 5 below.

Step 1: A suitable transform (Tr), such as DCT-II, is applied to each $m \times n$ image A_i of the training images, yielding T_i ($i = 1$ to N).

$$T_i = Tr\{A_i - \bar{A}\} \quad (2.12)$$

where \bar{A} is the mean matrix of all the training images.

Step 2: A transform is chosen such that the significant coefficients of T_i are contained in a submatrix, T'_i (upper left part of T_i) of dimension $n' \times n'$. T'_i is then used to replace A_i in our algorithm.

Step 3: The covariance matrix S for N training images is calculated as follows.

$$S = \frac{1}{N} \sum_{i=1}^N (T'_i)^T (T'_i) \quad (2.13)$$

Step 4: A set of k eigenvectors, $V = [V_1, V_2 \dots V_k]$, of size n' , corresponding to the largest k eigenvalues, is obtained for S .

Step 5: The feature matrices of the training images B_i ($i = 1$ to N) are calculated as follows.

$$B_i = [Y_{1,i}, Y_{2,i}, \dots Y_{k,i}] \quad (2.14)$$

where $Y_{j,i} = T'_i V_j$ and $j = 1, 2, \dots k$.

And the B_i matrices are stored.

2.5.2 Testing Mode

In the testing mode a facial image A_t is presented to the system to be identified. The following steps are taken.

Step 1: The same transform used in the training mode is applied to A_t , which yield T_t .

Step 2: The submatrix T'_t containing the significant coefficients is obtained (dimension $n' \times n'$.)

Step 3: The feature matrix B_t for the testing image is calculated

$$B_t = [Y_{1,t}, Y_{2,t}, \dots Y_{k,t}] \quad (2.15)$$

where $Y_{j,t} = T'_t V_j$ and $j = 1, 2, \dots k$.

Step 4: Distance measures, such as the Euclidean distances, between the feature matrix of the testing image and the feature matrices of the training images are measured. The minimum distance represents the image to be identified.

CHAPTER 3: QUANTIZED/TRUNCATED TRANSFORM DOMAIN TECHNIQUE (QTD)

3.1 Introduction

The proposed algorithm depends on representing the images in the transform domain. Mainly, the energy in facial images is concentrated in low spatial frequencies. In addition, a criterion is proposed to decide which coefficients to retain in the low spatial frequency ranges. Combining these two simple, but effective, constraints lead to considerable reduction in number of coefficients required to represent the images. Consequently, the computational and storage requirements are simplified, while excellent recognition accuracy is obtained, as will be shown later. The algorithm is described in the following section.

3.2 The QTD Algorithm

The following subsections illustrate step-by-step the procedures of the QTD algorithm in both the training and the testing modes.

3.2.1 Training Mode

In the training mode, the features of the database are extracted and stored as described by steps 1 through 4, Figure 3.1.

Step 1: A suitable transform (Tr), such as DCT-II, is applied to each $m \times n$ image A_i of the training

images, yielding $T_i (i = 1 \text{ to } N)$.

$$T_i = Tr\{A_i\} \quad (3.1)$$

Step 2: The mean energy per pixel, M_i , of the transformed image, T_i , is calculated, where $T_{i,pq}$ is its pq element.

$$M_i = \frac{1}{m} \frac{1}{n} \sum_{p=1}^m \sum_{q=1}^n T_{i,pq}^2 \quad (3.2)$$

Step 3: A transform is chosen such that the significant coefficients of T_i are contained in a submatrix, T'_i , (upper left part of T_i) of dimension $n' \times n'$. T'_i is then used to replace A_i in the algorithm.

Step 4: All the elements of the submatrix T'_i , $T'_{i,jk}$ ($j = 1 \text{ to } n'$ and $k = 1 \text{ to } n'$), which have an energy less than M_i are set to zero, yielding the feature matrix B_i , where $B_{i,jk}$ is its jk element.

$$B_{i,jk} = \begin{cases} T'_{i,jk} & \text{when } T_{i,jk}^2 \geq M_i \\ 0 & \text{elsewhere} \end{cases} \quad (3.3)$$

3.2.2 Testing Mode

In the testing mode, a facial test image, A_t , is presented to the system to be identified. The following steps are taken, Figure 3.2.

Step 1: The same transform used in the training mode is applied to A_t , which yields T_t .

$$T_t = Tr\{A_t\} \quad (3.4)$$

Step 2: The mean energy per pixel, M_t , of the transformed image, T_t is calculated, where $T_{t,pq}$ is

its pq element.

$$M_t = \frac{1}{m} \frac{1}{n} \sum_{p=1}^m \sum_{q=1}^n T_{t,pq}^2 \quad (3.5)$$

Step 3: The submatrix T'_t , dimension $n' \times n'$, containing the significant coefficients, is obtained.

Step 4: All the elements of the submatrix T'_t , $T'_{t,jk}$ ($j = 1$ to n' and $k = 1$ to n'), which have an energy less than M_t are set to zero, yielding the feature matrix B_t , where $B_{t,jk}$ is its jk element.

$$B_{t,jk} = \begin{cases} T'_{t,jk} & \text{when } T'^2_{t,jk} \geq M_t \\ 0 & \text{elsewhere} \end{cases} \quad (3.6)$$

Step 5: Distance measures, such as the Euclidean distance, between the feature matrix of the testing image and the feature matrices of the training images are computed. The image in the training database corresponding to the minimum distance defines the image to be identified.

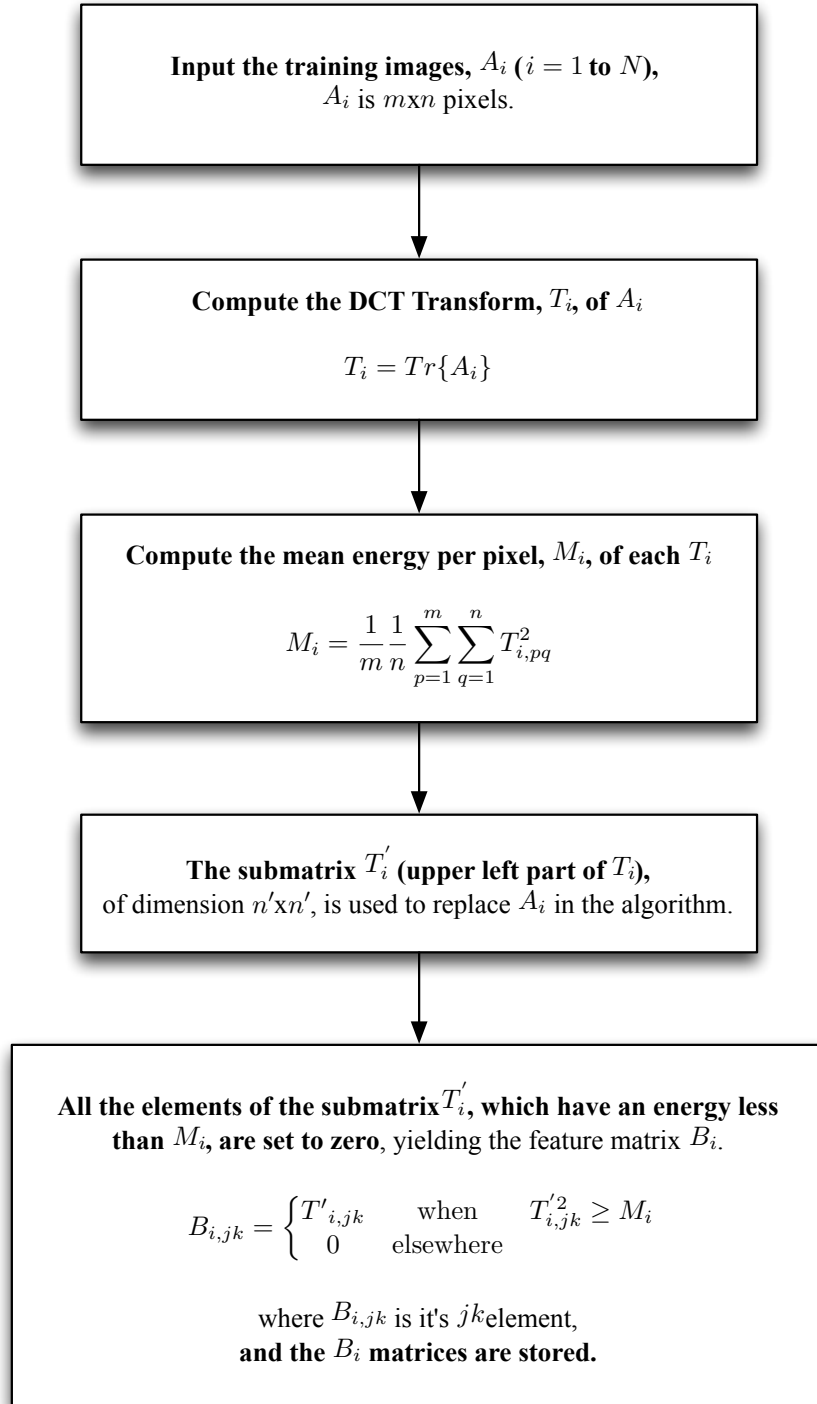


Figure 3.1: QTD Training mode flow-chart.

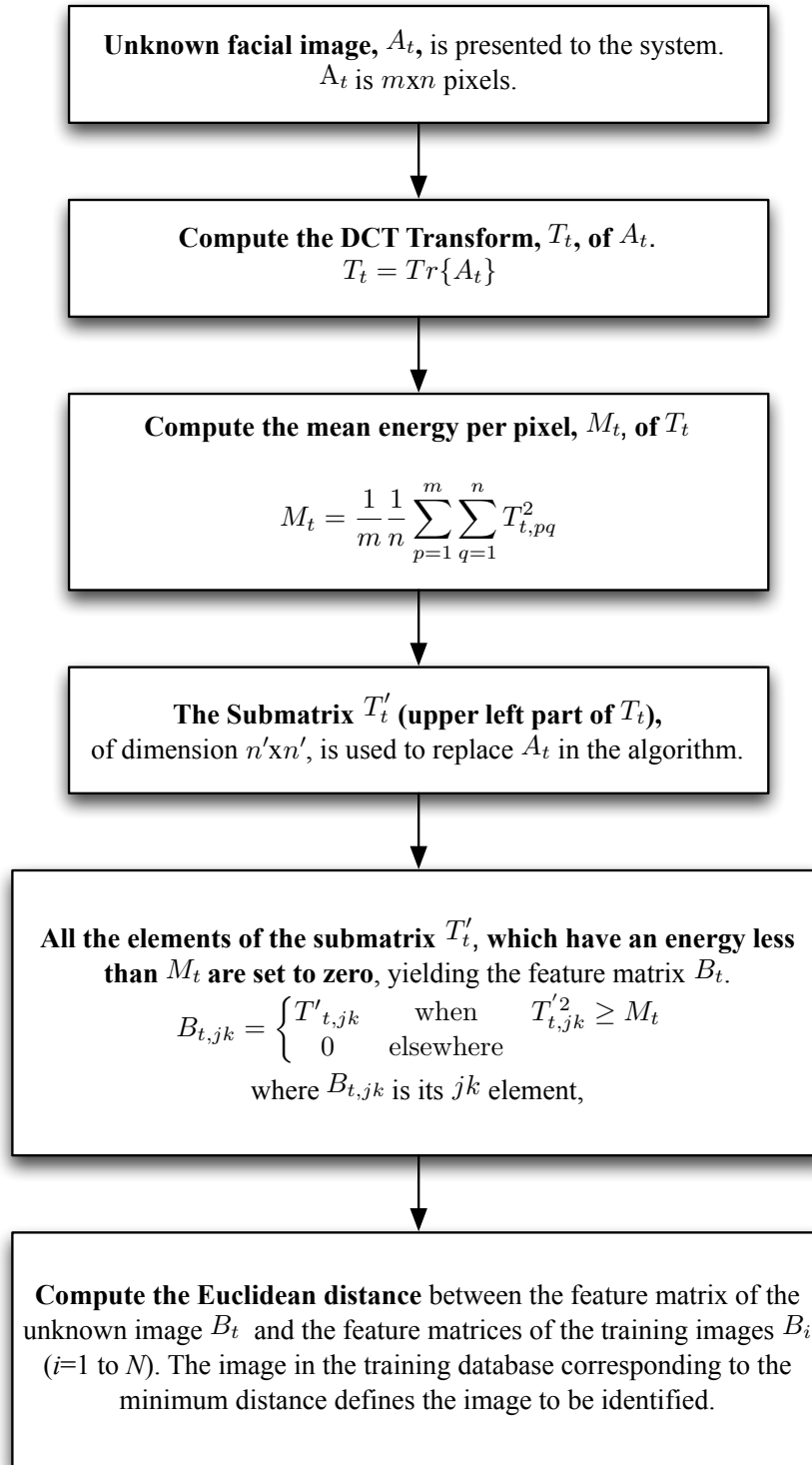


Figure 3.2: QTD Testing mode flow-chart.

3.3 Experimental Results

The QTD technique was tested using the ORL and Yale databases [46, 47]. The ORL database consists of 40 individuals, with ten images for each individual in various poses and with various facial expressions, Figure 3.3. The Yale database consists of 15 individuals, with 11 images for each individual in various poses and with various facial expressions, Figure 3.4. The part FERET database consists of 400 grayscale images for 200 individuals, individual in 2 different frontal poses, fa and fb, Figure 3.5.

Face recognition results were compared with those of existing techniques, namely, the 2DPCA, the $(2D)^2$ PCA and the TD2DPCA. The procedures for the compared techniques were taken from [13, 14, 44], while the procedures for the QTD technique are presented in section 3.2.



Figure 3.3: Sample of different images of one person in the ORL database.



Figure 3.4: Sample of different images of one person in the Yale database.



Figure 3.5: Sample of the (fa) and (fb) images of 5 individuals in the FERET database.

3.3.1 Experimental Results Using ORL Database

Two experiments, I and II, were been applied to the ORL database, in which all the images are grayscale with 112 x 92 pixels each. In Experiment I, the first five images per individual were used for training, and the remaining 200 images are used for testing. For both the feature image and the covariance matrix [13, 14, 44], the dimensions that give maximum recognition accuracy were selected. Thus, the recognition accuracy values in Table 3.1 are the maximum achievable for all compared algorithms. Results are listed in Tables 3.1 and 3.2.

In Experiment II, one image per individual was used for training, and the remaining 360 images are used for testing. The dimensions of feature image and covariance matrix [13, 14, 44] are the same as those used in Experiment I. Results are listed in Table 3.1.

Table 3.1: Recognition accuracy in Experiments I and II for ORL database using the proposed technique (QTD) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Recognition accuracy for Experiment I[*] | Recognition accuracy for Experiment II^{**} |
|-----------------------|--|--|
| QTD (proposed) | 94.00% | 71.10% |
| 2DPCA | 93.00% | 71.68% |
| (2D) ² PCA | 92.50% | 71.68% |
| TD2DPCA | 93.50% | 71.10% |

* Training with the first 5 poses per individual, and testing with the remaining poses.

** Training with the first pose per individual, and testing with the remaining poses.

Table 3.2: Storage requirements and computational complexity in Experiments I and II for ORL database using the proposed technique (QTD) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Dim. of feature matrix per image | Number of multiplications for | | Storage require- ments |
|-----------------------|---|--|--------------|---------------------------------------|
| | | training mode | testing mode | |
| QTD (proposed) | 13x13 | 159,804 xN [*] | 159,804 | 169 xN [*] |
| 2DPCA | 112x7 | 1,020,096 xN [*] | 72,128 | 784 xN [*] |
| (2D) ² PCA | 10x10 | 2,214,256 xN [*] | 112,240 | 100 xN [*] |
| TD2DPCA | 15x12 | 181,335 xN [*] | 177,960 | 180 xN [*] |

* N is the number of the training images, with N=200 in Experiment I and N=40 in Experiment II.

Table 3.1 shows that the proposed algorithm yields good recognition accuracy compared with the 2DPCA, (2D)²PCA and TD2DPCA methods.

Table 3.2 illustrates the computational requirements in the training and testing modes in terms of the number of multiplications. The QTD reduces the computational requirements the training mode. However, the storage requirements for the (2D)²PCA algorithm, in terms of the dimensions of the feature matrix, are lower compared with the QTD algorithm.

3.3.2 Experimental Results Using Yale Database

Two experiments, III and IV, were applied to the Yale database, in which all the images are grayscale with 243x320 pixels each. In Experiment III, the first five images per individual were used for training, and the remaining 90 images were used for testing. For both the feature image and the covariance matrix [13, 14, 44], the dimensions that give maximum recognition accuracy were selected. Thus, the recognition accuracy values in Table 3.3 are the maximum achievable for all compared algorithms. Results are listed in Tables 3.3 and 3.4.

In Experiment IV, one image per individual was used for training and the remaining 150 images were used for testing. The dimensions of feature image and covariance matrix [13, 14, 44] were the same as those used in Experiment III. Results are listed in Table 3.4.

Table 3.3: Recognition accuracy in Experiments III and IV for Yale database using the proposed technique (QTD) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Recognition accuracy for Experiment III[*] | Recognition accuracy for Experiment IV^{**} |
|-----------------------|--|--|
| QTD (proposed) | 97.78% | 62.00% |
| 2DPCA | 97.78% | 62.70% |
| (2D) ² PCA | 91.1% | 56.70% |
| TD2DPCA | 97.78% | 59.30% |

* Training with the first 5 poses per individual, and testing with the remaining poses.

** Training with the first pose per individual, and testing with the remaining poses.

Table 3.4: Storage requirements and computational complexity in Experiments III and IV for Yale database using the proposed technique (QTD) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Dim. of feature matrix per image | Number of multiplications for | | Storage requirements |
|-----------------------|----------------------------------|-------------------------------|--------------|----------------------|
| | | training mode | testing mode | |
| QTD (proposed) | 10x10 | 879,660 xN* | 879,660 | 100 xN* |
| 2DPCA | 320x9 | 19,595,520 xN* | 699,840 | 2,880 xN* |
| (2D) ² PCA | 16x16 | 45,104,960 xN* | 1,326,080 | 256 xN* |
| TD2DPCA | 15x11 | 1,244,250 xN* | 1,240,875 | 165 xN* |

* N is the number of the training images, with N=75 in Experiment III and N=15 in Experiment IV.

Table 3.3 shows that the proposed algorithm yields good recognition accuracy compared with the 2DPCA, (2D)²PCA and TD2DPCA methods.

Table 3.4 illustrates the computational requirements in the training and testing modes in terms of the number of multiplications. The QTD reduces the computational requirements in both the training mode and the testing mode. Furthermore, it is worthwhile to note that the storage requirements for the QTD, in terms of the dimensions of the feature matrix, are also reduced by 40% compared with the algorithm which performed most closely to QTD, namely TD2DPCA.

3.3.3 *Experimental Results Using FERET Database*

Experiment V was conducted using part of the FERET database, in which all the images are grayscale with 384 x 256 pixels each. In Experiment V, one image (fa) per individual was used for training, and one image (fb) per individual was used for testing. The images of the first 200 individuals of the FERET database were used for Experiment V. The dimensions of both the feature image and the covariance matrix [13, 14, 44] that yielded the highest recognition accuracy were selected. Thus, the recognition accuracy values in Table 3.5 are the maximum achievable for all compared algorithms. Results are listed in Tables 3.5 and 3.6.

Table 3.5: Recognition accuracy in Experiment V for FERET database using the proposed technique (QTD) and existing algorithms (2DPCA, $(2D)^2PCA$, and TD2DPCA).

| | Recognition accuracy[*] |
|----------------|---|
| QTD (proposed) | 94.00% |
| 2DPCA | 90.00% |
| $(2D)^2PCA$ | 91.50% |
| TD2DPCA | 92.50% |

* Training with one pose per individual, and testing with another one pose per individual.

Table 3.6: Storage requirements and computational complexity in Experiment V for FERET database using the proposed technique (QTD) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Dim. of feature matrix per image | Number of multiplications for | | Storage requirements (in bits) |
|-----------------------|----------------------------------|-------------------------------|--------------|--------------------------------|
| | | training mode | testing mode | |
| QTD (proposed) | 5x5 | 596,224 xN* | 596,224 | 25 xN* |
| 2DPCA | 256x4 | 38,141,952 xN* | 393,216 | 1,024 xN* |
| (2D) ² PCA | 8x8 | 63,725,568 xN* | 811,008 | 64 xN* |
| TD2DPCA | 15x5 | 1,536,660 xN* | 1,533,285 | 75 xN* |

* N is the number of the training images, with N=200.

Table 3.5 shows that the proposed algorithm yields good recognition accuracy compared with the 2DPCA, $(2D)^2PCA$ and TD2DPCA methods.

Table 3.6 illustrates the computational requirements in the training and testing modes in terms of the number of multiplications. The QTD reduces the computational requirements in both the training mode and the testing mode. Furthermore, it is worthwhile to note that the storage requirements for the QTD, in terms of the dimensions of the feature matrix, are also reduced by a factor of 2.5 compared with the algorithm which performed most closely to QTD, namely $(2D)^2PCA$.

3.4 Conclusion

Recently, several facial recognition methods with high recognition accuracy have been reported. In this dissertation, a new algorithm, namely QTD, has been presented. Our study of QTD shows that it reduces both the storage requirements and the computational complexity in comparison with other recently reported methods, while maintaining high recognition accuracy. Thus, experimental results confirm the excellent properties of the proposed technique.

CHAPTER 4: FREQUENCY DOMAIN THRESHOLDING AND QUANTIZATION ALGORITHM (FD-TQ)

4.1 Introduction

With the Frequency Domain Thresholding and Quantization Algorithm (FD-TQ), first, images are transformed to a domain that concentrates the energy in the low spatial frequencies region. In this dissertation, the Two-dimensional Discrete Cosine Transform (DCT-II) was used for this task. In addition, a criterion is proposed to quantize the coefficients to only 256 levels. Combining these two simple, but effective, operations leads to a considerable reduction in the number of coefficients required to represent the images. Consequently, the computational and storage requirements are simplified, while excellent recognition accuracy is achieved, as will be shown later. The algorithm is described in the following section.

4.2 The FD-TQ Algorithm

The following subsections illustrate step-by-step the procedures of the FD-TQ algorithm in both training and testing modes.

4.2.1 *Training Mode*

In the training mode, the features of the database are extracted and stored as described by steps 1 through 6, Figure 4.1.

Step 1: A suitable transform (Tr), such as DCT-II, is applied to each $m \times n$ image A_i of the training

images, yielding $T_i (i = 1 \text{ to } N)$.

$$T_i = Tr\{A_i\} \quad (4.1)$$

Step 2: A transform is chosen such that the significant coefficients of T_i are contained in a submatrix, T'_i , (upper left part of T_i) of dimensions $n' \times n'$. T'_i is then used to replace A_i in the algorithm.

Step 3: The value of the DC component of the submatrix, T'_i , $T'_{i,11}$ ($j = 1$ and $k = 1$), is set to zero.

Step 4: The upper boundary threshold, α_u , and the lower boundary threshold, α_l , are calculated as follows:

$$\alpha_u = \mu_i + 3\sigma_i \quad (4.2)$$

$$\alpha_l = \mu_i - 3\sigma_i \quad (4.3)$$

Where μ_i is the mean of the values of all elements in all the submatrices, T'_i , in the training images, and σ_i is the standard deviation for the values of all elements in all the submatrices, T'_i , in the training images.

Step 5: The threshold matrix, T_{it} , is computed, where its jk element, $T_{it,jk}$, is given by:

$$T_{it,jk} = \begin{cases} \alpha_u & \text{when } T'_{i,jk} \geq \alpha_u \\ \alpha_l & \text{when } T'_{i,jk} \leq \alpha_l \\ T'_{i,jk} & \text{else} \end{cases} \quad (4.4)$$

It is worthwhile to note that clipping the extreme values improves the accuracy of representing the majority of the data in Step 6.

Step 6: The threshold matrix, T_{it} , is normalized such that all its elements fall between 0 and 1. Then the elements are multiplied by 255 and linearly quantized to get the 8-bit representation of the normalized submatrix, T_{it} . This yields the feature matrix, B_i , where its jk element, $B_{i,jk}$, is given by:

$$B_{i,jk} = \text{floor}(255 * T_{it,jk}) \quad (4.5)$$

4.2.2 Testing Mode

In the testing mode, a facial test image, A_t , is presented to the system to be identified. The following steps are followed, Figure 4.2.

Step 1: The same transform used in the training mode is applied to A_t which yields T_t .

Step 2: The submatrix, T'_t , of dimensions $n' \times n'$, containing the significant coefficients is obtained, as described in the training mode procedure.

Step 3: The value of the DC component of the submatrix, T'_t , $T'_{t,11}$ ($j = 1$ and $k = 1$), is set to zero.

Step 4: The threshold matrix, T_{tt} , is computed, where its jk element, $T_{tt,jk}$, is given by:

$$T_{tt,jk} = \begin{cases} \alpha_u & \text{when } T'_{t,jk} \geq \alpha_u \\ \alpha_l & \text{when } T'_{t,jk} \leq \alpha_l \\ T'_{t,jk} & \text{else} \end{cases} \quad (4.6)$$

Step 5: The threshold matrix, T_{tt} , is normalized such that all its elements fall between 0 and 1.

Then the elements are multiplied by 255 and linearly quantized to get the 8-bit representation of the normalized submatrix, T_{tt} . This yields the feature matrix, B_t , where its jk element, $B_{t,jk}$, is given by:

$$B_{t,jk} = \text{floor}(255 * T_{tt,jk}) \quad (4.7)$$

Step 6: Distance measures, such as the Euclidean distance, between the feature matrix of the testing image and the feature matrices of the training images are computed. The image in the training database corresponding to the minimum distance defines the image to be identified.

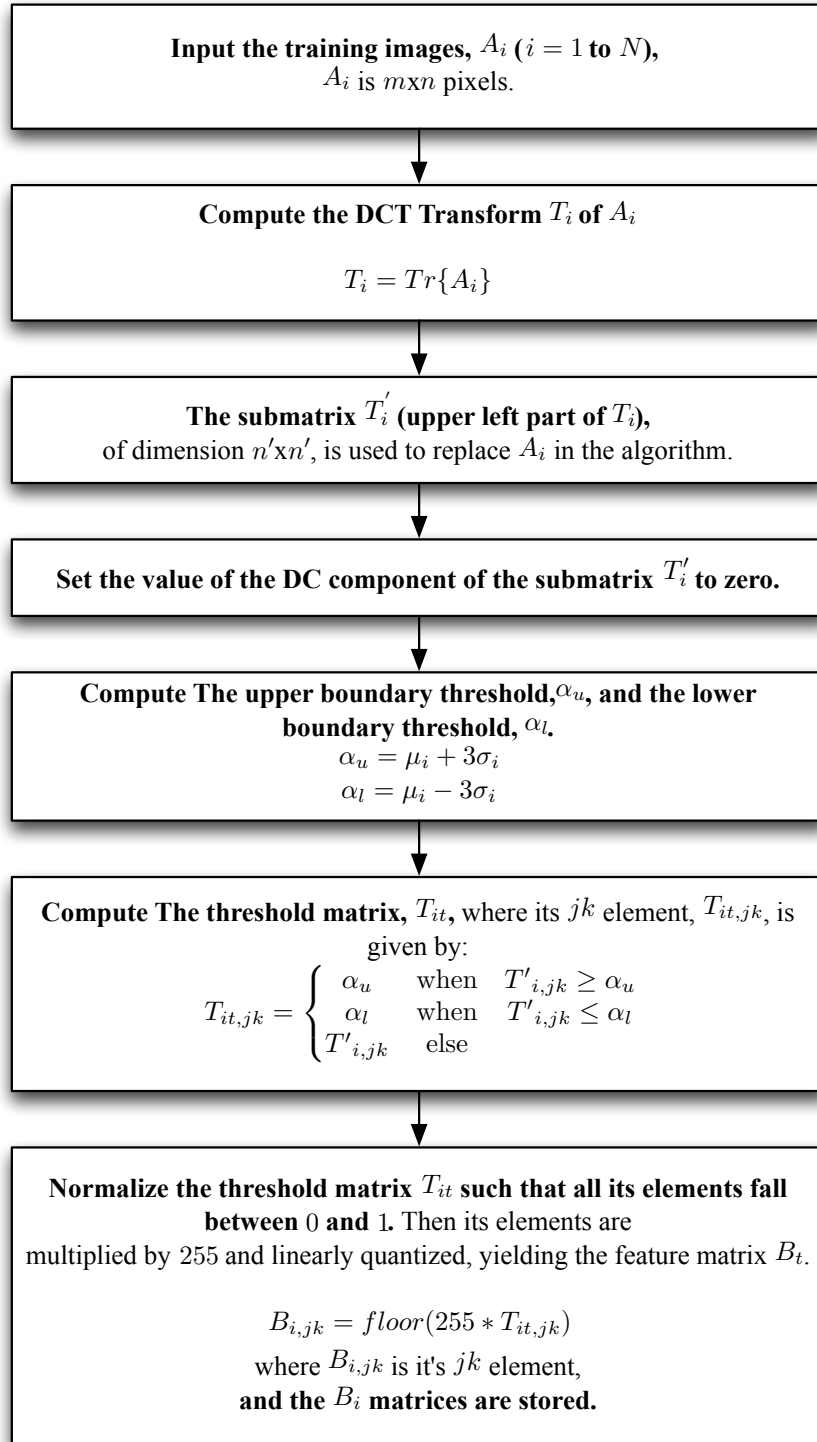


Figure 4.1: FD-TQ Training mode flow-chart.

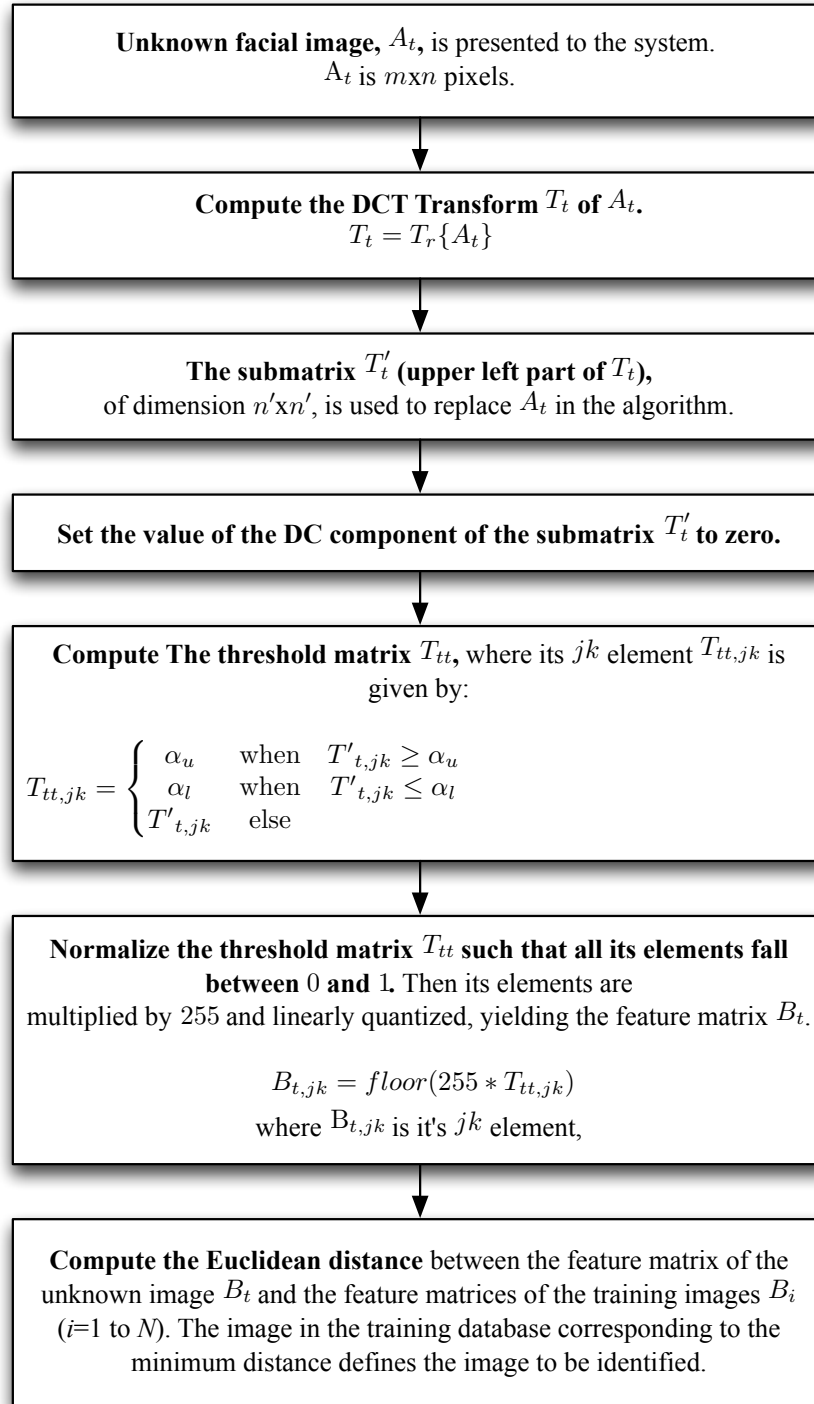


Figure 4.2: FD-TQ Testing mode flow-chart.

4.3 Experimental Results

The FD-TQ technique was tested using the ORL, the Yale and part of the FERET databases [46–48]. The ORL database consists of 40 individuals, with ten images for each individual in various poses and with various facial expressions, Figure 3.3. The Yale database consists of 15 individuals, with 11 images for each individual in various poses and with various facial expressions, Figure 3.4. The part FERET database consists of 400 grayscale images for 200 individuals, individual in 2 different frontal poses, fa and fb, Figure 3.5.

Face recognition results of the FD-TQ technique were compared with those of existing techniques, namely, the 2DPCA, the $(2D)^2PCA$ and the TD2DPCA. The procedures for the compared techniques were taken from [13, 14, 44], while the procedures for the FD-TQ technique are presented in section 4.2.

4.3.1 Experimental Results Using ORL Database

Two experiments, I and II, were conducted using the ORL database, in which all the images are grayscale with 112 x 92 pixels each. In Experiment I, the first five images per individual were used for training, and the remaining 200 images were used for testing. The dimensions of both the feature images and the covariance matrix [13, 14, 44] that yielded the highest recognition accuracy were selected. Thus, the recognition accuracy values in Table 4.1 are the maximum achievable for all the compared algorithms. Results are listed in Tables 4.1 and 4.2.

In Experiment II, the first image per individual was used for training, and the remaining 360 images were used for testing. The dimensions of the feature images and the covariance matrix [13, 14, 44] were the same as those used in Experiment I. Results are listed in Table 4.1.

Table 4.1: Recognition accuracy in Experiments I and II for ORL database using the proposed technique (FD-TQ) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Recognition accuracy for Experiment I[*] | Recognition accuracy for Experiment II^{**} |
|-----------------------|--|--|
| FD-TQ (proposed) | 92.00% | 72.78% |
| 2DPCA | 93.00% | 71.68% |
| (2D) ² PCA | 92.50% | 71.68% |
| TD2DPCA | 93.50% | 71.10% |

* Training with the first 5 poses per individual, and testing with the remaining poses.

** Training with the first pose per individual, and testing with the remaining poses.

Table 4.2: Storage requirements and computational complexity in Experiments I and II for ORL database using the proposed technique (FD-TQ) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Dim. of feature matrix per image | Number of multiplications for | | Storage requirements (in bits) |
|-----------------------|----------------------------------|-------------------------------|--------------|--------------------------------|
| | | training mode | testing mode | |
| FD-TQ (proposed) | 5x5 | 53,845 xN* | 53,845 | 200 xN* |
| 2DPCA | 112x7 | 1,020,096 xN* | 72,128 | 25,088** xN* |
| (2D) ² PCA | 10x10 | 2,214,256 xN* | 112,240 | 3,200** xN* |
| TD2DPCA | 15x12 | 181,335 xN* | 177,960 | 5,760** xN* |

* N is the number of the training images, with N=200 in Experiment I and N=40 in Experiment II.

** 32-bit is typically needed to represent the data in this technique.

Table 4.1 shows that the proposed algorithm yields good recognition accuracy compared with the 2DPCA, $(2D)^2PCA$ and TD2DPCA methods.

Table 4.2 illustrates the computational requirements in the training and testing modes in terms of the number of multiplications. The FD-TQ reduces the computational requirements in both the training mode and the testing mode. Furthermore, it is worthwhile to note that the storage requirements for the FD-TQ, in terms of the dimensions of the feature matrix, are also reduced by a factor of 16 compared with the algorithm which performed most closely to FD-TQ, namely $(2D)^2PCA$.

4.3.2 Experimental Results Using Yale Database

Two experiments, III and IV, were performed using the Yale database, in which all the images are grayscale with 243×320 pixels each. In Experiment III, the first five images per individual were used for training, and the remaining 90 images were used for testing. The dimensions of both the feature images and the covariance matrix [13, 14, 44] that yielded the highest recognition accuracy were selected. Thus, the recognition accuracy values in Table 4.3 are the maximum achievable for all the compared algorithms. Results are listed in Tables 4.3 and 4.4.

In Experiment IV, the first image per individual was used for training and the remaining 150 images were used for testing. The dimensions of the feature images and the covariance matrix [13, 14, 44] were the same as those used in Experiment III. Results are listed in Table 4.3.

Table 4.3: Recognition accuracy in Experiments III and IV for Yale database using the proposed technique (FD-TQ) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Recognition accuracy for Experiment III[*] | Recognition accuracy for Experiment IV^{**} |
|-----------------------|--|--|
| FD-TQ (proposed) | 97.78% | 61.33% |
| 2DPCA | 97.78% | 62.70% |
| (2D) ² PCA | 91.10% | 56.70% |
| TD2DPCA | 97.78% | 59.30% |

* Training with the first 5 poses per individual, and testing with the remaining poses.

** Training with the first pose per individual, and testing with the remaining poses.

Table 4.4: Storage requirements and computational complexity in Experiments III and IV for Yale database using the proposed technique (FD-TQ) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Dim. of feature matrix per image | Number of multiplications for | | Storage requirements (in bits) |
|-----------------------|----------------------------------|-------------------------------|--------------|--------------------------------|
| | | training mode | testing mode | |
| FD-TQ (proposed) | 12x12 | 968,256 xN* | 968,256 | 1,152 xN* |
| 2DPCA | 320x9 | 19,595,520 xN* | 699,840 | 92,160** xN* |
| (2D) ² PCA | 16x16 | 45,104,960 xN* | 1,326,080 | 8,192** xN* |
| TD2DPCA | 15x11 | 1,244,250 xN* | 1,240,875 | 5,280** xN* |

* N is the number of the training images, with N=75 in Experiment III and N=15 in Experiment IV.

** 32-bit is typically needed to represent the data in this technique.

Table 4.3 shows that the proposed algorithm yields good recognition accuracy compared with the 2DPCA, (2D)²PCA and TD2DPCA methods.

Table 4.4 illustrates the computational requirements in the training and testing modes in terms of the number of multiplications. The FD-TQ reduces the computational requirements in both the training mode and the testing mode. Furthermore, it is worthwhile to note that the storage requirements for the FD-TQ, in terms of the dimensions of the feature matrix, are also reduced by a factor of 4.5 compared with the algorithm which performed most closely to FD-TQ, namely TD2DPCA.

4.3.3 *Experimental Results Using FERET Database*

Experiment V was conducted using part of the FERET database, in which all the images are grayscale with 384 x 256 pixels each. In Experiment V, one image (fa) per individual was used for training, and one image (fb) per individual was used for testing. The images of the first 200 individuals of the FERET database were used for Experiment V. The dimensions of both the feature image and the covariance matrix [13, 14, 44] that yielded the highest recognition accuracy were selected. Thus, the recognition accuracy values in Table 4.5 are the maximum achievable for all compared algorithms. Results are listed in Tables 4.5 and 4.6.

Table 4.5: Recognition accuracy in Experiment V for FERET database using the proposed technique (FD-TQ) and existing algorithms (2DPCA, $(2D)^2PCA$, and TD2DPCA).

| | Recognition accuracy[*] |
|------------------|---|
| FD-TQ (proposed) | 93.00% |
| 2DPCA | 90.00% |
| $(2D)^2PCA$ | 91.50% |
| TD2DPCA | 92.50% |

* Training with one pose per individual, and testing with another one pose per individual.

Table 4.6: Storage requirements and computational complexity in Experiment V for FERET database using the proposed technique (FD-TQ) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Dim. of feature matrix per image | Number of multiplications for | | Storage requirements (in bits) |
|-----------------------|----------------------------------|-------------------------------|--------------|--------------------------------|
| | | training mode | testing mode | |
| FD-TQ (proposed) | 5x5 | 497,945 xN* | 497,945 | 200 xN* |
| 2DPCA | 256x4 | 38,141,952 xN* | 393,216 | 32,768** xN* |
| (2D) ² PCA | 8x8 | 63,725,568 xN* | 811,008 | 2,048** xN* |
| TD2DPCA | 15x5 | 1,536,660 xN* | 1,533,285 | 2,400** xN* |

* N is the number of the training images, with N=200.

** 32-bit is typically needed to represent the data in this technique.

Table 4.5 shows that the proposed algorithm yields good recognition accuracy compared with the 2DPCA, $(2D)^2PCA$ and TD2DPCA methods.

Table 4.6 illustrates the computational requirements in the training and testing modes in terms of the number of multiplications. The FD-TQ reduces the computational requirements in both the training mode and the testing mode. Furthermore, it is worthwhile to note that the storage requirements for the FD-TQ, in terms of the dimensions of the feature matrix, are also reduced by a factor of 10 compared with the algorithm which performed most closely to FD-TQ, namely $(2D)^2PCA$.

4.4 Conclusion

Recently, several facial recognition methods with high recognition accuracy have been reported. In this dissertation, a new algorithm, namely FD-TQ, is presented. Our study of FD-TQ shows that it reduces both the storage requirements and the computational complexity in comparison with the recently reported methods, while maintaining the achieved recognition accuracy. Experimental results confirm the excellent properties of the proposed technique.

CHAPTER 5: NORMALIZED HISTOGRAM INTENSITY ALGORITHM (NHI)

5.1 Introduction

An image histogram is a statistical calculation that provides counts of pixels with a particular range of amplitude values. In other words, an image histogram represents the frequency of the occurrence of the intensity values in an image. The histogram provides useful information to many image processing applications, such as those that deal with image enhancement [49–52]. Furthermore, the histogram statistical information have been utilized in many face recognition techniques, such as [53–58].

In this chapter, a facial recognition technique employing Subimages Normalized Histogram Intensity (NHI) is presented. This NHI algorithm calculates histogram intensity of poses' subimages, follow by quantization of the histogram intensity coefficients to only 256 levels, thereby reducing the storage requirements of the NHI algorithm. The algorithm has attractive properties with respect to storage requirements and computational complexity in both the training and testing modes, which makes the technique particularly suitable for large databases.

For this study, the new algorithm is applied to the ORL, Yale and FERET databases. The experimental results confirm a significant reduction in the storage and computational requirements compared with other recently reported techniques, without any sacrifices necessary in terms of recognition accuracy, as will be shown later. The algorithm is described in the following section.

5.2 The NHI Algorithm

The following subsections illustrate step-by-step the procedures of the NHI algorithm in both training and testing modes.

5.2.1 Training Mode

In the training mode, the features of the database are extracted and stored as described by steps 1 through 4, Figure 5.1.

Step 1: Each $m \times n$ image A_i of the N training images is divided into $b \times b$ sub-matrix blocks, yielding image sub-matrix, A_{ijk} , where jk represents the sub-matrix block location. A block size of 32×32 is used in this study.

Step 2: Histogram intensity is computed for each $b \times b$ sub-matrix blocks, A_{ijk} , of the training images. Using a certain number of histogram levels (6 histogram levels are used in this study) yields histogram sub-matrix H_{ijk} where jk represents the sub-matrix block location.

Step 3: All histogram intensity sub-matrices H_{ijk} are combined, yielding the histogram intensity matrix H_i ($i = 1$ to N).

Step 4: The histogram intensity matrix, H_i , is normalized such that all its elements fall between 0 and 1. Then the elements are multiplied by 255 and linearly quantized to get the 8-bit representation of the normalized histogram intensity matrix H_i . This yields the feature matrix, B_i , where its jk element, $B_{i,jk}$, is given by:

$$B_{i,jk} = \text{floor}(255 * H_{i,jk}) \quad (5.1)$$

5.2.2 Testing Mode

In the testing mode, a facial test image, A_t , is presented to the system to be identified. The following steps are followed, Figure 5.2.

Step 1: An $m \times n$ image A_t is divided into $b \times b$ sub-matrix blocks, yielding image sub-matrix A_{tjk} , where jk represents the sub-matrix block location. As described in the training procedure, a block size of 32×32 will be used.

Step 2: Histogram intensity is computed for each $b \times b$ sub-matrix block A_{tjk} of the test image. Using a certain number of histogram levels (6 histogram levels are used in this work) yields histogram sub-matrix H_{tjk} , where jk represents the sub-matrix block location.

Step 3: All histogram intensity sub-matrices H_{tjk} are combined yielding histogram intensity matrix H_t .

Step 4: The histogram intensity matrix H_t is normalized such that all its elements fall between 0 and 1. Then the elements are multiplied by 255 and linearly quantized to get an 8-bit representation of the normalized histogram intensity matrix H_t . This yields the feature matrix B_t , where its jk element $B_{t,jk}$ is given by:

$$B_{t,jk} = \text{floor}(255 * H_{t,jk}) \quad (5.2)$$

Step 5: Distance measures, such as the Euclidean distance, between the feature matrix of the testing image and the feature matrices of the training images are computed. The image in the training database corresponding to the minimum distance defines the image to be identified.

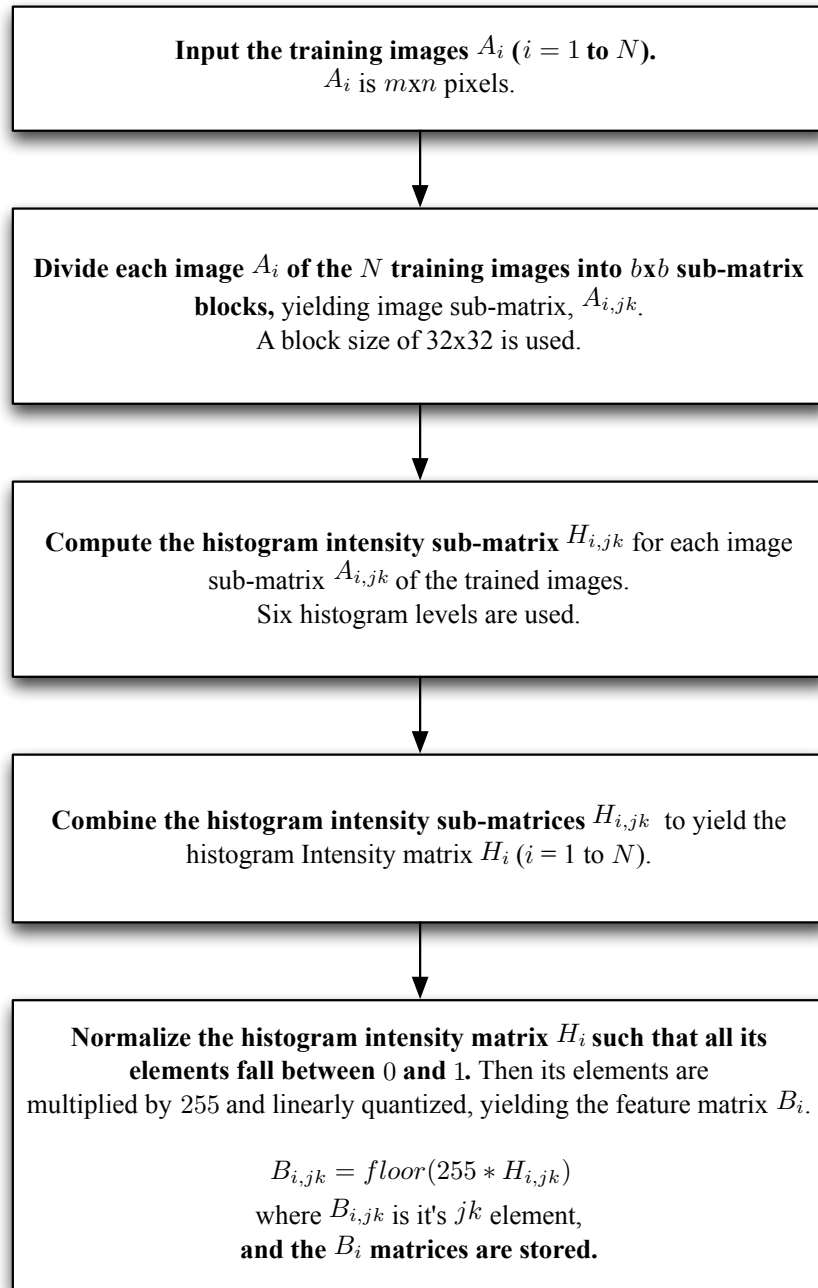


Figure 5.1: NHI Training mode flow-chart.

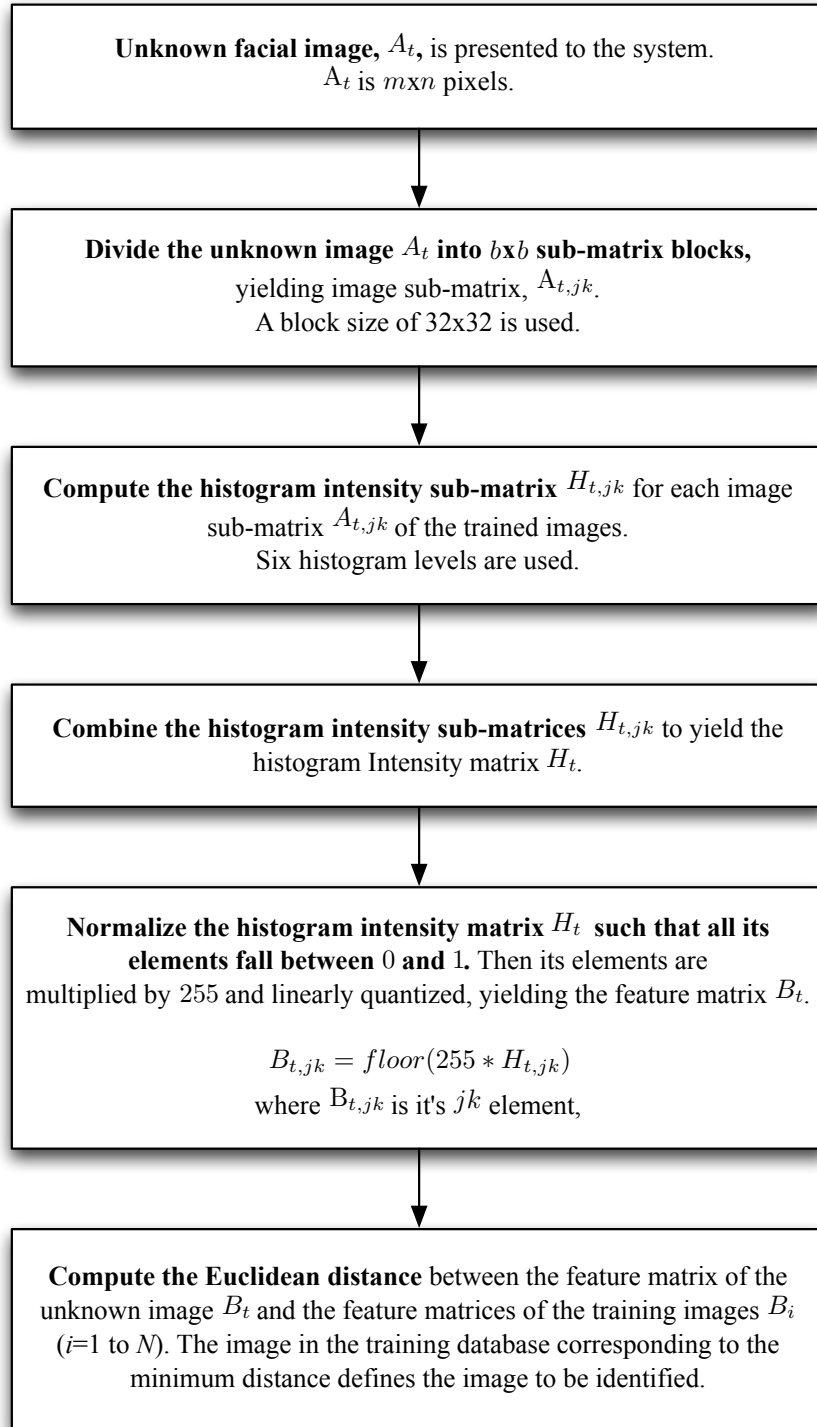


Figure 5.2: NHI Testing mode flow-chart.

5.3 Experimental Results

The NHI technique was tested using the ORL, the Yale and part of the FERET databases [46–48]. The ORL database consists of 40 individuals, with ten images for each individual in various poses and with various facial expressions, Figure 3.3. The Yale database consists of 15 individuals, with 11 images for each individual in various poses and with various facial expressions, Figure 3.4. The part of the FERET database consists of 400 grayscale images for 200 individuals, with individual in 2 different frontal poses, fa and fb, Figure 3.5.

Face recognition results of the NHI technique were compared with those of existing techniques, namely, the 2DPCA, $(2D)^2$ PCA, and TD2DPCA. The procedures for the compared techniques were taken from [13, 14, 44], while the procedures for the NHI technique are presented in section 6.2.

5.3.1 *Experimental Results Using ORL Database*

Two experiments, I and II, were conducted using the ORL database, in which all the images are grayscale with 112 x 92 pixels each. In Experiment I, the first five images per individual were used for training, and the remaining 200 images were used for testing. The dimensions of both the feature images and the covariance matrix [13, 14, 44] that yielded the highest recognition accuracy were selected. Thus, the recognition accuracy values in Table 5.1 are the maximum achievable for all the compared algorithms. Results are listed in Tables 5.1 and 5.2.

In Experiment II, the first image per individual was used for training, and the remaining 360 images were used for testing. The dimensions of the feature images and the covariance matrix [13, 14, 44] were the same as those used in Experiment I. Results are listed in Table 5.1.

Table 5.1: Recognition accuracy in Experiments I and II for ORL database using the proposed technique (NHI) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Recognition accuracy for Experiment I[*] | Recognition accuracy for Experiment II^{**} |
|-----------------------|--|--|
| NHI (proposed) | 95.50% | 72.22% |
| 2DPCA | 93.00% | 71.68% |
| (2D) ² PCA | 92.50% | 71.68% |
| TD2DPCA | 93.50% | 71.10% |

* Training with the first 5 poses per individual, and testing with the remaining poses.

** Training with the first pose per individual, and testing with the remaining poses.

Table 5.2: Storage requirements and computational complexity in Experiments I and II for ORL database using the proposed technique (NHI) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Dim. of feature matrix per image | Number of multiplications for | | Storage requirements (in bits) |
|-----------------------|----------------------------------|-------------------------------|--------------|--------------------------------------|
| | | training mode | testing mode | |
| NHI (proposed) | 24x3 | 72 xN [*] | 72 | 576 xN [*] |
| 2DPCA | 112x7 | 1,020,096 xN [*] | 72,128 | 25,088 ^{**} xN [*] |
| (2D) ² PCA | 10x10 | 2,214,256 xN [*] | 112,240 | 3,200 ^{**} xN [*] |
| TD2DPCA | 15x12 | 181,335 xN [*] | 177,960 | 5,760 ^{**} xN [*] |

* N is the number of the training images, with N=200 in Experiment I and N=40 in Experiment II.

** 32-bit is typically needed to represent the data in this technique.

Table 5.1 shows that the proposed algorithm yields good recognition accuracy compared with the 2DPCA, $(2D)^2PCA$ and TD2DPCA methods.

Table 5.2 illustrates the computational requirements in the training and testing modes in terms of the number of multiplications. The NHI reduces the computational requirements in both the training mode and the testing mode. Furthermore, it is worthwhile to note that the storage requirements for the NHI, in terms of the dimensions of the feature matrix, are also reduced by a factor of 5 compared with the algorithm which performed most closely to NHI, namely $(2D)^2PCA$.

5.3.2 *Experimental Results Using Yale Database*

Two experiments, III and IV, were performed using the Yale database, in which all the images are grayscale with 243×320 pixels each. In Experiment III, the first five images per individual were used for training, and the remaining 90 images were used for testing. The dimensions of both the feature images and the covariance matrix [13, 14, 44] that yielded the highest recognition accuracy were selected. Thus, the recognition accuracy values in Table 5.3 are the maximum achievable for all the compared algorithms. Results are listed in Tables 5.3 and 5.4.

In Experiment IV, the first image per individual was used for training and the remaining 150 images were used for testing. The dimensions of the feature images and the covariance matrix [13, 14, 44] were the same as those used in Experiment III. Results are listed in Table 5.3.

Table 5.3: Recognition accuracy in Experiments III and IV for Yale database using the proposed technique (NHI) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Recognition accuracy for Experiment III[*] | Recognition accuracy for Experiment IV^{**} |
|-----------------------|--|--|
| NHI (proposed) | 94.44% | 59.33% |
| 2DPCA | 97.78% | 62.70% |
| (2D) ² PCA | 91.10% | 56.70% |
| TD2DPCA | 97.78% | 59.30% |

* Training with the first 5 poses per individual, and testing with the remaining poses.

** Training with the first pose per individual, and testing with the remaining poses.

Table 5.4: Storage requirements and computational complexity in Experiments III and IV for Yale database using the proposed technique (NHI) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Dim. of feature matrix per image | Number of multiplications for | | Storage requirements (in bits) |
|-----------------------|----------------------------------|-------------------------------|--------------|--------------------------------|
| | | training mode | testing mode | |
| NHI (proposed) | 48x10 | 480 xN* | 480 | 3,840 xN* |
| 2DPCA | 320x9 | 19,595,520 xN* | 699,840 | 92,160** xN* |
| (2D) ² PCA | 16x16 | 45,104,960 xN* | 1,326,080 | 8,192** xN* |
| TD2DPCA | 15x11 | 1,244,250 xN* | 1,240,875 | 5,280** xN* |

* N is the number of the training images, with N=75 in Experiment III and N=15 in Experiment IV.

** 32-bit is typically needed to represent the data in this technique.

Table 5.3 shows that the proposed algorithm yields good recognition accuracy compared with the 2DPCA, (2D)²PCA and TD2DPCA methods.

Table 5.4 illustrates the computational requirements in the training and testing modes in terms of the number of multiplications. The NHI reduces the computational requirements in both the training mode and the testing mode. Furthermore, it is worthwhile to note that the storage requirements for the NHI, in terms of the dimensions of the feature matrix, are also reduced by 27% compared with the algorithm which performed most closely to NHI, namely TD2DPCA.

5.3.3 *Experimental Results Using FERET Database*

Experiment V was conducted using part of the FERET database, in which all the images are grayscale with 384 x 256 pixels each. In Experiment V, one image (fa) per individual was used for training, and one image (fb) per individual was used for testing. The images of the first 200 individuals of the FERET database were used for Experiment V. The dimensions of both the feature image and the covariance matrix [13, 14, 44] that yielded the highest recognition accuracy were selected. Thus, the recognition accuracy values in Table 5.5 are the maximum achievable for all compared algorithms. Results are listed in Tables 5.5 and 5.6.

Table 5.5: Recognition accuracy in Experiment V for FERET database using the proposed technique (NHI) and existing algorithms (2DPCA, $(2D)^2PCA$, and TD2DPCA).

| | Recognition accuracy[*] |
|----------------|---|
| NHI (proposed) | 95.00% |
| 2DPCA | 90.00% |
| $(2D)^2PCA$ | 91.50% |
| TD2DPCA | 92.50% |

* Training with one pose per individual, and testing with another one pose per individual.

Table 5.6: Storage requirements and computational complexity in Experiment V for FERET database using the proposed technique (NHI) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Dim. of feature matrix per image | Number of multiplications for | | Storage require- ments (in bits) |
|-----------------------|---|----------------------------------|--------------|---|
| | | training mode | testing mode | |
| NHI (proposed) | 72x8 | 576 xN [*] | 576 | 4,608 xN [*] |
| 2DPCA | 256x4 | 38,141,952 xN [*] | 393,216 | 32,768 ^{**} xN [*] |
| (2D) ² PCA | 8x8 | 63,725,568 xN [*] | 811,008 | 2,048 ^{**} xN [*] |
| TD2DPCA | 15x5 | 1,536,660 xN [*] | 1,533,285 | 2,400 ^{**} xN [*] |

* N is the number of the training images, with N=200.

** 32-bit is typically needed to represent the data in this technique.

Table 5.5 shows that the proposed algorithm yields good recognition accuracy compared with the 2DPCA, (2D)²PCA and TD2DPCA methods.

Table 5.6 illustrates the computational requirements in the training and testing modes in terms of the number of multiplications. The NHI reduces the computational requirements in both the training mode and the testing mode. However, the storage requirements for the TD2DPCA and (2D)²PCA algorithms, in terms of the dimensions of the feature matrix, are lower compared with the NHI algorithm.

5.4 Conclusion

Recently, several facial recognition methods with high recognition accuracy have been investigated. In this dissertation, a new facial recognition algorithm, namely NHI, is introduced. Our study of NHI shows that both the storage requirements and the computational complexity required for facial recognition calculations were reduced compared with results from other methods recently investigated, while maintaining the achieved recognition accuracy. Experimental results, therefore, confirm the excellent properties of the proposed technique.

CHAPTER 6: NORMALIZED TRANSFORM DOMAIN ALGORITHM (NTD)

6.1 Introduction

In this chapter, a facial recognition algorithm employing Normalized Transform Domain (NTD) is presented. This NTD algorithm transforms the training and test images to a domain that concentrates the energy in the low spatial frequencies region. In this dissertation, the Two-dimensional Discrete Cosine Transform (DCT-II) [16] was used for this task. In addition, a criterion is proposed to normalize the coefficients to only 256 levels. Combining these two simple, but effective, operations leads to a considerable reduction in the number of coefficients required to represent the images.

For this study, the new algorithm is applied to the ORL, Yale, and FERET databases. The experimental results confirm a significant reduction in the storage and computational requirements compared with other recently reported techniques, without any sacrifices necessary in terms of recognition accuracy, as will be shown later. The algorithm is described in the following section.

6.2 The NTD Algorithm

The following subsections illustrate step-by-step the procedures of the NTD algorithm in both training and testing modes.

6.2.1 Training Mode

In the training mode, the features of the database are extracted and stored as described by steps 1 through 3, Figure 6.1.

Step 1: A suitable transform (Tr), such as DCT-II, is applied to each $m \times n$ image A_i of the training images, yielding $T_i (i = 1 \text{ to } N)$.

$$T_i = Tr\{A_i\} \quad (6.1)$$

Step 2: A transform is chosen such that the significant coefficients of T_i are contained in a submatrix T'_i (upper left part of T_i) of dimensions $n' \times n'$. T'_i is then used to replace A_i in the algorithm.

Step 3: The submatrix T'_i is normalized such that all its elements fall between 0 and 1. Then the elements are multiplied by 255 and linearly quantized to get the 8-bit representation of the normalized submatrix T'_i . This yields the feature matrix B_i , where its jk element, $B_{i,jk}$, is given by:

$$B_{i,jk} = floor(255 * T'_{i,jk}) \quad (6.2)$$

6.2.2 Testing Mode

In the testing mode, a facial test image, A_t , is presented to the system to be identified. The following steps are followed, Figure 6.2.

Step 1: The same transform used in the training mode is applied to A_t which yields T_t .

Step 2: The submatrix T'_t of dimensions $n' \times n'$, containing the significant coefficients is obtained, as described in the training mode procedure.

Step 3: The submatrix T'_t is normalized such that all its elements fall between 0 and 1. Then the elements are multiplied by 255 and linearly quantized to get the 8-bit representation of the normalized submatrix T'_t . This yields the feature matrix B_t , where its jk element, $B_{t,jk}$, is given by:

$$B_{t,jk} = \text{floor}(255 * T'_{t,jk}) \quad (6.3)$$

Step 4: Distance measures, such as the Euclidean distance, between the feature matrix of the testing image and the feature matrices of the training images are computed. The image in the training database corresponding to the minimum distance defines the image to be identified.

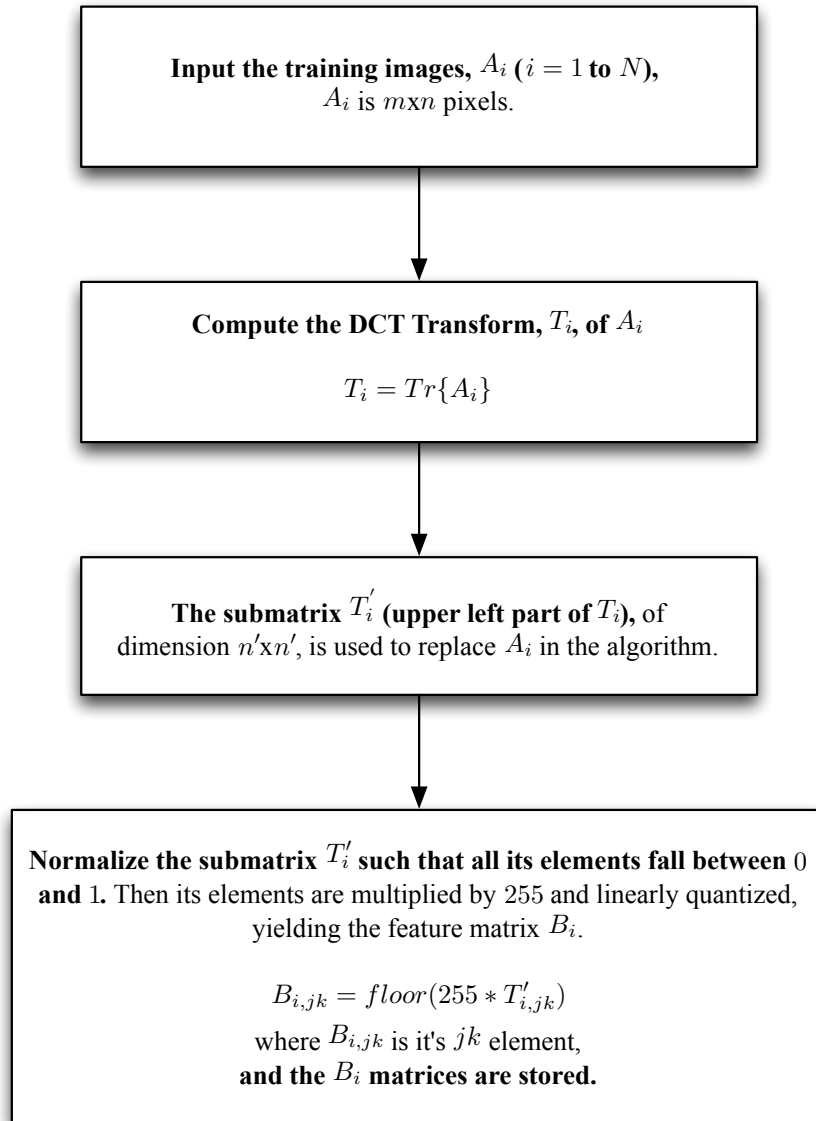


Figure 6.1: NTD Training mode flow-chart.

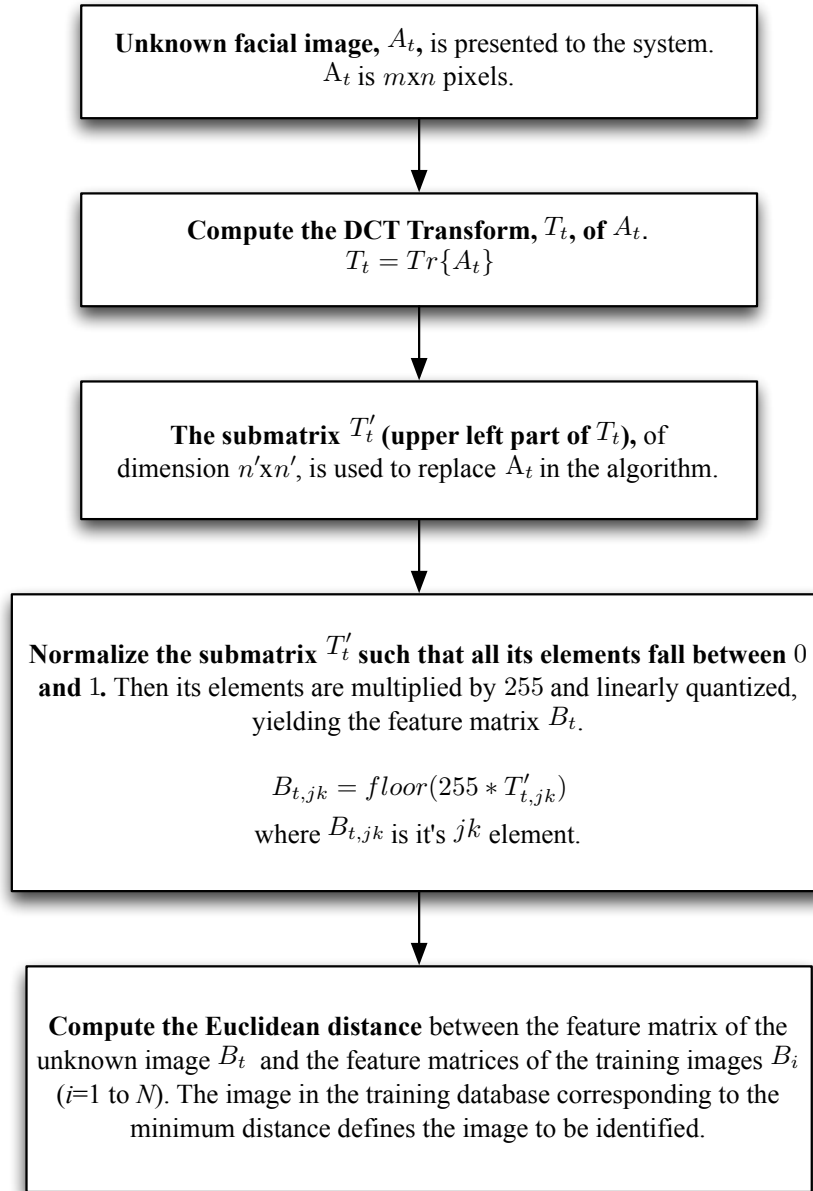


Figure 6.2: NTD Testing mode flow-chart.

6.3 Experimental Results

The NTD technique was tested using the ORL, Yale and part of FERET databases [46–48]. The ORL database consists of 40 individuals, with ten images for each individual in various poses and with various facial expressions, Figure 3.3. The Yale database consists of 15 individuals, with 11 images for each individual in various poses and with various facial expressions, Figure 3.4. The part of the FERET database consists of 400 grayscale images for 200 individuals, with individual in 2 different frontal poses, fa and fb, Figure 3.5.

Face recognition results of the NTD technique were compared with those of existing techniques, namely, the 2DPCA, (2D)²PCA, and TD2DPCA. The procedures for the compared techniques were taken from [13, 14, 44], while the procedures for the NTD technique are presented in section 6.2.

6.3.1 Experimental Results Using ORL Database

Two experiments, I and II, were conducted using the ORL database, in which all the images are grayscale with 112 x 92 pixels each. In Experiment I, the first five images per individual were used for training, and the remaining 200 images were used for testing. The dimensions of both the feature images and the covariance matrix [13, 14, 44] that yielded the highest recognition accuracy were selected. Thus, the recognition accuracy values in Table 6.1 are the maximum achievable for all the compared algorithms. Results are listed in Tables 6.1 and 6.2.

In Experiment II, the first image per individual was used for training, and the remaining 360 images were used for testing. The dimensions of the feature images and the covariance matrix [13, 14, 44] were the same as those used in Experiment I. Results are listed in Table 6.1.

Table 6.1: Recognition accuracy in Experiments I and II for ORL database using the proposed technique (NTD) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Recognition accuracy for Experiment I[*] | Recognition accuracy for Experiment II^{**} |
|-----------------------|--|--|
| NTD (proposed) | 93.50% | 71.39% |
| 2DPCA | 93.00% | 71.68% |
| (2D) ² PCA | 92.50% | 71.68% |
| TD2DPCA | 93.50% | 71.10% |

* Training with the first 5 poses per individual, and testing with the remaining poses.

** Training with the first pose per individual, and testing with the remaining poses.

Table 6.2: Storage requirements and computational complexity in Experiments I and II for ORL database using the proposed technique (NTD) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Dim. of feature matrix per image | Number of multiplications for | | Storage requirements (in bits) |
|-----------------------|----------------------------------|-------------------------------|--------------|--------------------------------------|
| | | training mode | testing mode | |
| NTD (proposed) | 5x5 | 53,845 xN [*] | 53,845 | 200 xN [*] |
| 2DPCA | 112x7 | 1,020,096 xN [*] | 72,128 | 25,088 ^{**} xN [*] |
| (2D) ² PCA | 10x10 | 2,214,256 xN [*] | 112,240 | 3,200 ^{**} xN [*] |
| TD2DPCA | 15x12 | 181,335 xN [*] | 177,960 | 5,760 ^{**} xN [*] |

* N is the number of the training images, with N=200 in Experiment I and N=40 in Experiment II.

** 32-bit is typically needed to represent the data in this technique.

Table 6.1 shows that the proposed algorithm yields good recognition accuracy compared with the 2DPCA, (2D)²PCA and TD2DPCA methods.

Table 6.2 illustrates the computational requirements in the training and testing modes in terms of the number of multiplications. The NTD reduces the computational requirements in both the training mode and the testing mode. Furthermore, it is worthwhile to note that the storage requirements for the NTD, in terms of the dimensions of the feature matrix, are also reduced by a factor of 16 compared with the algorithm which performed most closely to NTD, namely (2D)²PCA.

6.3.2 *Experimental Results Using Yale Database*

Two experiments, III and IV, were performed using the Yale database, in which all the images are grayscale with 243x320 pixels each. In Experiment III, the first five images per individual were used for training, and the remaining 90 images were used for testing. The dimensions of both the feature images and the covariance matrix [13, 14, 44] that yielded the highest recognition accuracy were selected. Thus, the recognition accuracy values in Table 6.3 are the maximum achievable for all the compared algorithms. Results are listed in Tables 6.3 and 6.4.

In Experiment IV, the first image per individual was used for training and the remaining 150 images were used for testing. The dimensions of the feature images and the covariance matrix [13, 14, 44] were the same as those used in Experiment III. Results are listed in Table 5.3.

Table 6.3: Recognition accuracy in Experiments III and IV for Yale database using the proposed technique (NTD) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Recognition accuracy for Experiment III[*] | Recognition accuracy for Experiment IV^{**} |
|-----------------------|--|--|
| NTD (proposed) | 96.67% | 63.33% |
| 2DPCA | 97.78% | 62.70% |
| (2D) ² PCA | 91.10% | 56.70% |
| TD2DPCA | 97.78% | 59.30% |

* Training with the first 5 poses per individual, and testing with the remaining poses.

** Training with the first pose per individual, and testing with the remaining poses.

Table 6.4: Storage requirements and computational complexity in Experiments III and IV for Yale database using the proposed technique (NTD) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Dim. of feature matrix per image | Number of multiplications for | | Storage require- ments (in bits) |
|-----------------------|---|--|--------------|---|
| | | training mode | testing mode | |
| NTD (proposed) | 10x10 | 809,700 xN* | 809,700 | 800 xN* |
| 2DPCA | 320x9 | 19,595,520 xN* | 699,840 | 92,160** xN* |
| (2D) ² PCA | 16x16 | 45,104,960 xN* | 1,326,080 | 8,192** xN* |
| TD2DPCA | 15x11 | 1,244,250 xN* | 1,240,875 | 5,280** xN* |

* N is the number of the training images, with N=75 in Experiment III and N=15 in Experiment IV.

** 32-bit is typically needed to represent the data in this technique.

Table 6.3 shows that the proposed algorithm yields good recognition accuracy compared with the 2DPCA, (2D)²PCA and TD2DPCA methods.

Table 6.4 illustrates the computational requirements in the training and testing modes in terms of the number of multiplications. The NTD reduces the computational requirements in both the training mode and the testing mode. Furthermore, it is worthwhile to note that the storage requirements for the NTD, in terms of the dimensions of the feature matrix, are also reduced by a factor of 6 compared with the algorithm which performed most closely to NTD, namely TD2DPCA.

6.3.3 Experimental Results Using FERET Database

Experiment V was conducted using part of the FERET database, in which all the images are grayscale with 384 x 256 pixels each. In Experiment V, one image (fa) per individual was used for training, and one image (fb) per individual was used for testing. The images of the first 200 individuals of the FERET database were used for Experiment V. The dimensions of both the feature image and the covariance matrix [13, 14, 44] that yielded the highest recognition accuracy were selected. Thus, the recognition accuracy values in Table 6.5 are the maximum achievable for all compared algorithms. Results are listed in Tables 6.5 and 6.6.

Table 6.5: Recognition accuracy in Experiment V for FERET database using the proposed technique (NTD) and existing algorithms (2DPCA, $(2D)^2PCA$, and TD2DPCA).

| | Recognition accuracy[*] |
|----------------|---|
| NTD (proposed) | 93.00% |
| 2DPCA | 90.00% |
| $(2D)^2PCA$ | 91.50% |
| TD2DPCA | 92.50% |

* Training with one pose per individual, and testing with another one pose per individual.

Table 6.6: Storage requirements and computational complexity in Experiment V for FERET database using the proposed technique (NTD) and existing algorithms (2DPCA, (2D)²PCA, and TD2DPCA).

| | Dim. of feature matrix per image | Number of multiplications for | | Storage requirements (in bits) |
|-----------------------|----------------------------------|-------------------------------|--------------|--------------------------------|
| | | training mode | testing mode | |
| NTD (proposed) | 5x5 | 497,945 xN* | 497,945 | 200 xN* |
| 2DPCA | 256x4 | 38,141,952 xN* | 393,216 | 32,768** xN* |
| (2D) ² PCA | 8x8 | 63,725,568 xN* | 811,008 | 2,048** xN* |
| TD2DPCA | 15x5 | 1,536,660 xN* | 1,533,285 | 2,400** xN* |

* N is the number of the training images, with N=200.

** 32-bit is typically needed to represent the data in this technique.

Table 6.5 shows that the proposed algorithm yields good recognition accuracy compared with the 2DPCA, $(2D)^2PCA$ and TD2DPCA methods.

Table 6.6 illustrates the computational requirements in the training and testing modes in terms of the number of multiplications. The NTD reduces the computational requirements in both the training mode and the testing mode. Furthermore, it is worthwhile to note that the storage requirements for the NTD, in terms of the dimensions of the feature matrix, are also reduced by a factor of 10 compared with the algorithm which performed most closely to NTD, namely $(2D)^2PCA$.

6.4 Conclusion

Recently, several facial recognition methods with high recognition accuracy have been investigated. In this dissertation, a new facial recognition algorithm, namely NTD, is introduced. Our study of NTD shows that it reduces both the storage requirements and the computational complexity required for facial recognition calculations in comparison with the recently reported methods, while maintaining the achieved recognition accuracy. Experimental results, therefore, confirm the excellent properties of the proposed technique.

CHAPTER 7: SUMMARY AND FUTURE WORK

7.1 Summary

The performance of any face image recognition algorithm is commonly measured by looking at three characteristics: (i) the recognition accuracy achieved, (ii) the storage requirements needed, and (iii) the computational complexity involved. The problem with most existing face recognition algorithms is the focus on achieving a higher recognition accuracy while de-emphasizing the increase of the algorithm's computational complexity and/or storage requirements.

Our proposed algorithms, QTD, FD-TQ, NHI, and NTD, focus on simplifying the computational complexity and reducing the storage requirements of the face recognition algorithm without sacrificing the level of recognition accuracy achieved by the existing algorithms.

To measure the performance of our proposed algorithms, experiments were performed on the proposed algorithms and on the state-of-the-art algorithms, namely, 2DPCA, $(2D)^2$ PCA, and TD2DPCA. The experiments were conducted on the famous face recognition databases, namely, ORL, Yale, and FERET.

In effect, the results confirm that our proposed algorithms reduce the computational complexity and the storage requirements compared to the existing algorithms. Also, the results show that our proposed algorithms maintained the recognition accuracy of the compared algorithms.

Finally, the reduction in both computational complexity and storage requirements of our proposed algorithms demonstrates their suitability for use with large databases.

7.2 Future Work

In this section, a number of ideas for new face recognition algorithms will be discussed to provide direction for future research.

- The usage of the transform domain, such as DCT, in face recognition showed an advantage towards reducing the storage requirements of the facial recognition system. On the other hand, utilizing statistical information of the image, such as pixel intensity histogram, demonstrated an advantage towards reducing the computational complexity of the facial recognition system. Because of that, we would like to investigate combining both the DCT transform and the histogram distribution in a face recognition technique.
- Utilizing Vector Quantization (VQ) techniques [59] in a face recognition algorithm could add advantages in term of reducing algorithm's storage requirements and needed computational operations. A good example of this is shown in [60]. Therefore, we believe that combining vector quantization techniques with one of our proposed algorithms could improve performance of the algorithm.
- Throughout our experiments, the noise effect on the performance of our proposed algorithm was not considered. We would like to investigate the impact of adding noise to the training images on the performance of our proposed algorithms.
- A multicriteria approach may be useful for face recognition. Using a proper voting scheme to manage the decision of the multicriteria approach could achieve 100% recognition accuracy. An example of the performance of multicriteria is in [61]. However, we would like to investigate suitable ways to use the multicriteria approach in face recognition without sacrificing in term of computational complexity and storage requirements.

- Continue publishing and presenting our contributions at international refereed conferences and submitting paper(s) to refereed journals.

APPENDIX : SOURCE CODE OF DEVELOPED ALGORITHMS

```

% The code for QTD algorithm
%
% Written By:waleed@knights.ucf.edu
% ORL Database

clear all

close all

clc

G=40; % number of persons
H=5; % number of poses per person for training
CM=zeros(G*H,2);
temp3=0;
NTR1=zeros(G*H,1);
iter1=0;
% Training Mode
for i= 1:G
    for j=1:H
        iter1=iter1+1;
        xs=imread(strcat('/s',int2str(i),'/', int2str(j)),'pgm');
        xi(:, :, iter1)=xs;
        x=double(xs);
        xd=dct2(x);
        I(:, :, iter1)=x;
        Tid(:, :, iter1)=xd;
        CM(iter1,:)=[iter1,i];
    end
end

tfmin=5;

```

```

tfmax=20;

recog=zeros(tfmax-tfmin+1,4);

for tf=tfmin:tfmax

    clear Vi1;

    clear Vtd1;

    clear Dd1;

    temp3=0;

    NTR1=zeros(G*H,1);

    i=1;j=1;k=1;L=1;

for i=1:iter1

    xtemp1=Tid(:, :, i);

    xtemp=xtemp1(1:tf,1:tf);

    xtemphalfenr=mean(mean(xtemp1.^2)); % half power of Vi1

    for ii=1:tf

        for jj=1:tf

            if xtemp(ii,jj)^2 <= xtemphalfenr

                xtemp(ii,jj)= zeros;

            end

        end

    end

    Vi1(:, :, i)=xtemp;

    clear xtemp

end

correct1=0;

wrong1=0;

iter2=0;

ST=H+1;

```

```

EN=10;

for i= 1:G

    for j=ST:EN

        iter2=iter2+1;

        NTR=zeros(iter1,1);

        xta=imread(strcat('/s',int2str(i),'/', int2str(j)),'pgm');

        xtar(:, :, iter2)=xta;

        xt=double(xta);

        It(:, :, iter2)=xt;

        Tt=dct2(xt);

        Ttd(:, :, iter2)=Tt;

        Vt1=Tt(1:tf,1:tf);

        Vt1halfenr=mean(mean(Tt.^2)); % half power of Vt1

        for ii=1:tf

            for jj=1:tf

                if Vt1(ii,jj)^2 <= Vt1halfenr

                    Vt1(ii,jj)= zeros;

                end

            end

        end

        for L=1:iter1

            D1=Vi1(:, :, L)-Vt1;

            Dd1(:, :, L)=D1;

            for k=1:tf

                temp3= temp3+norm(D1(:,k));

                NT1=temp3;

            end

        end

    end

end

```

```

        NTR1(L)=NT1;
        temp3=0;
    end
    [Si1,Hi1] = min(NTR1);
    NTRF1(:, :, iter2)=NTR1;
        if CM(Hi1,2) == i
        correct1=correct1+1;
        rescl(correct1, :)= [iter2, Hi1, Si1];
        else
        wrong1=wrong1+1;
        resl(wrong1, :)= [iter2, Hi1, Si1];
        NTRFF1(:, :, wrong1)=NTR1;
        end
    end
end
end
recog(tf-tfmin+1,1)=tf;
recog(tf-tfmin+1,2)=correct1;
recog(tf-tfmin+1,3)=wrong1;
recog(tf-tfmin+1,4)=correct1*100/(wrong1+correct1);
end
ymin=min(min(recog(:,4))-1);
ymax=max(max(recog(:,4))+0.5);
titlestr = ['ORL Database Varying the No. Of columns',...
    ' Of Test & Train Poses Matrix With ',int2str(iter1) ,...
    ' Training & ',int2str(iter2), ' Testing'];
figure
plot(recog(:,1),recog(:,4),'-ro','LineWidth',2,...

```



```

        'MarkerEdgeColor','k',...
        'MarkerFaceColor','g',...
        'MarkerSize',10)
axis([tfmin tf ymin ymax])
title(titlestr)
xlabel('No. Of columns Of Pose Matrix')
ylabel('% of Accuracy')
hleg1 = legend('QTD');
set(hleg1,'Location','south')
grid on

clear Ttd;

```

```

% The code for FD-TQ algorithm
%
% Written By:waleed@knights.ucf.edu
% Yale Database

clear all
close all
clc

load Yale

G=15; % number of persons
Ht=11; % total number per person in the database
H=1; % number of poses per person for training
train=[1:5];
testing=[6:11];

% Training Mode
tfmin=12;
tfmax=12;

for tf=tfmin:tfmax

    Tidm=mean2(Tid(1:tf,1:tf,1:G,1:5));

    Tidstd=std2(Tid(1:tf,1:tf,1:G,1:5));

    %Tidmax=max(max(max
    msteps=[Tidstd,sqrt(2)*Tidstd,2*Tidstd,3*Tidstd,4*Tidstd,...
            5*Tidstd,6*Tidstd, 7*Tidstd];

    iter3=0;

%     recog=zeros(tfmax-tfmin+1,4);

for step=msteps
    xmax=step+Tidm;
    xmin=-1*step+Tidm;

```

```

for i=1:G
    for j= 1:11
        xd=Tid(:,:,i,j);
        xn= (xd-xmin)./(xmax-xmin);

        for k=1: numel(xn)
            if xn(k) > 1
                xn(k)= 1;
            elseif xn(k) < 0
                xn(k)=0;
            end
        end

        end

        xq=floor(xn.*255);
        Tiq(:,:,i,j)=xq;
        clear xn xq
    end
end

iter3=iter3+1;
clear Vi1;
clear Vtd1;
clear Dd1;
temp3=0;
NTR1=zeros(G*H,3);
i=1;j=1;k=1;L=1;
iter1=0;
for i=1:G
    for j=train
        xtemp1=Tiq(:,:,i,j);

```

```

    xtemp=xtemp1(1:tf,1:tf);

    iter1=iter1+1;

    Vi1(:, :, i, j)=xtemp;

    clear xtemp

end

end

% Testing Mode

correct1=0;

wrong1=0;

iter2=0;

for i= 1:G

    for j=testing

        iter2=iter2+1;

        xtemp2=Tiq(:, :, i, j);

        Vt1=xtemp2(1:tf,1:tf);

        clear xtemp2

L=0;

        for m=1:G

            for n= train

                D1=Vi1(:, :, m, n)-Vt1;

                L=L+1;

                Dd1(:, :, L)=D1;

                for k=1:tf

                    temp3= temp3+norm(D1(:, k));

                    NT1=temp3;

                end

                NTR1(L, 1)=NT1;NTR1(L, 2)=m;NTR1(L, 3)=n;

```

```

        temp3=0;
    end
end
[Si1,Hi1] = min(NTR1(:,1));
NTRF1(:, :, iter2)=NTR1;
    if NTR1(Hi1,2) == i
        correct1=correct1+1;
        rescl(correct1, :)= [iter2, Hi1, Si1, NTR1(Hi1,2), NTR1(Hi1,3)];

    else
        wrong1=wrong1+1;
        resl(wrong1, :)= [iter2, Hi1, Si1, NTR1(Hi1,2), NTR1(Hi1,3)];
        NTRFF1(:, :, wrong1)=NTR1;
    end
end
end
recog(iter3,1)=xmax;
recog(iter3,2)=correct1;
recog(iter3,3)=wrong1;
recog(iter3,4)=correct1*100/(wrong1+correct1);
end
ymin=min(min(recog(:,4))-1);
% ymin=80;
ymax=max(max(recog(:,4))+0.5);
titlestr = ['Yale Normalized and Quant Vo. of coulmsns of pose ', ...
    'matrix = ', int2str(tf), 'Varying quantiazation limit', ...
    ' Of Test & Train ', int2str(iter1) , ...

```

```

        ' Training & ',int2str(iter2), ' Testing'];
figure;
plot(recog(:,1),recog(:,4),'-ro','LineWidth',2,...
      'MarkerEdgeColor','k',...
      'MarkerFaceColor','g',...
      'MarkerSize',10)
      axis([100 xmax ymin ymax])
      title(titlestr)
      xlabel('No. of columns of pose Matrix')
      ylabel('% of Accuracy')
      hleg1 = legend('FD-TQ');
      set(hleg1,'Location','south')
      grid on
end

```

```

% The code for NHI algorithm
%
% Written By:waleed@knights.ucf.edu
% FERET Database

clear all

close all

clc

G=200; % number of persons

Ht=2; % total number per person in the database

train=[1];

testing=[2];

[z1,testp]=size(testing);

[z2,trainp]=size(train);

% Training Mode

    clear Vi1;

    clear Vtd1;

    clear Dd1;

    temp3=0;

    i=1;j=1;k=1;L=1;

bit=4;

blk=8;

while (blk < 33)

    for lev = 6:6

        clear wrong1 correct1 Vi1 Dd1

        correct1=0;

        wrong1=0;

        iter2=0;

```

```

iter1=0;

    for i=1:G

        for j=train

            iter1=iter1+1;

xs=double(imread(strcat('/Users/waleed/Pictures/fa_new/',...
            int2str(i),'_', int2str(j), '.tif'),'tiff'));

            xd=histbitform(xs,blk,lev,bit);

            I(:, :, i, j)=xs;

            Vi1(:, :, i, j)=xd;

            clear  xd xs

        end

    end

[m,tf]=size(Vi1(:, :, 1, 1));

% Testing Mode

for i= 1:G

    for j=testing

        iter2=iter2+1;

        xd=double(imread(strcat('/fa_new/',...
            int2str(i),'_', int2str(j), '.tif'),'tiff'));

        Vt1=histbitform(xd,blk,lev,bit);

        I(:, :, i, j)=xd;

        clear  xd xs

L=0;

    for m=1:G

        for n= train

            D1=Vi1(:, :, m, n)-Vt1;

            L=L+1;

```



```

Dd1(:, :, L) = D1;

    for k = 1:tf
        temp3 = temp3 + norm(D1(:, k));
        NT1 = temp3;
    end

    NTR1(L, 1) = NT1; NTR1(L, 2) = m; NTR1(L, 3) = n;
    temp3 = 0;
end

end

[Si1, Hi1] = min(NTR1(:, 1));
NTRF1(:, :, iter2) = NTR1;

    if NTR1(Hi1, 2) == i
        correct1 = correct1 + 1;
        rescl(correct1, :) = [iter2, Hi1, Si1, NTR1(Hi1, 2), NTR1(Hi1, 3)];
    else
        wrong1 = wrong1 + 1;
        resl(wrong1, :) = [iter2, Hi1, Si1, NTR1(Hi1, 2), NTR1(Hi1, 3)];
        NTRFF1(:, :, wrong1) = NTR1;
    end

end

end

recog(lev, 1, blk) = lev;
recog(lev, 2, blk) = correct1;
recog(lev, 3, blk) = wrong1;
recog(lev, 4, blk) = correct1 * 100 / (wrong1 + correct1);
end

blk = blk * 2;

```

```

end

%ymin=min(min(recog(:,4))-1);

ymin=80;

%ymax=max(max(recog(:,4,blk))+0.5);

ymax=98;

xmax=lev+1;

titlestr = [int2str(bit),' bit Histogram on FERET with ',...
    'Varying Block size And Levels',...
    '   while Training with ',int2str(trainp) ,...
    ' Pose & Testing with ',int2str(testp), ' Pose'];

%figure;

set(0,'DefaultAxesColorOrder',[0 0 0],...
    'DefaultAxesLineStyleOrder','-|--|:|-.')

plot(recog(:,1,32),recog(:,4,8),...
    recog(:,1,32),recog(:,4,16),recog(:,1,32),recog(:,4,32))

    axis([0 xmax ymin ymax])

    title(titlestr)

    xlabel('No. of Levels that been used in the Hisogram')

    ylabel('% of Accuracy')

    hleg1 = legend('NHI with block size = 8',...
        'NHI with block size = 16',...
        'NHI with block size = 32');

    set(hleg1,'Location','south')

    grid on

    hold on

```

```

% This function used in NHI Algorithm
%
% Written By:waleed@knights.ucf.edu
function Y = histbitform(M,blk,lev,bit)
bits=2^bit-1;
bs=[blk blk];% block size
szM=size(M);
nb = floor(szM ./ bs);
eb = rem(szM,bs);
e1=eb(1);
e2=eb(2);
if e1 == 0
    e1=[];
end
if e2 == 0
    e2=[];
end
C = mat2cell(M,[repmat(bs(1),1,nb(1)) e1], ...
    [repmat(bs(2),1,nb(2)) e2]);
C2 = cellfun(@(x) histlevelsum(x(:),lev), C, 'un', 0);
C3= cell2mat(C2);
%Y= floor(bits*C3);
Y = floor(bits*C3/(blk^2));

```

```

% This function used in NHI Algorithm
%
% Written By:waleed@knights.ucf.edu
function T = histlevelsum(I,lev)
hm=[];
step=ceil(256/lev);
for i= 0: lev-1
    hlst=sum(sum(histc(I, i*step:(i+1)*step-1)));
    hm=[hm; hlst];
end
T=hm;

```

```

% The code for NTD algorithm
%
% Written By:waleed@knights.ucf.edu
% ORL Database

clear all
close all
clc

G=40; % number of persons
H=5 ; % number of poses per person for training
[xsrow xscol]=size(imread(strcat('/s',...
                                int2str(1),'/', int2str(1)),'pgm')));
dctmxtrow=dctmtx(xsrow);
dctmxtcol=dctmtx(xscol)';
dctmxtrow20=dctmxtrow(1:20,:);
dctmxtcol20=dctmxtcol(:,1:20);
CM=zeros(G*H,2);
temp3=0;
NTR1=zeros(G*H,1);
iter1=0;
% Training Mode
for i= 1:G
    for j=1:H
        iter1=iter1+1;
        xs=double(imread(strcat('/s',...
                                int2str(i),'/', int2str(j)),'pgm')));
        xi(:, :, iter1)=xs;
        xd1=dctmxtrow20*xs;
    end
end

```

```

        xd=xd1*dctmxtcol20;

        Tid(:, :, iter1)=xd;

        CM(iter1, :)= [iter1, i];

    end

end

tfmin=2;

tfmax=20;

recog=zeros (tfmax-tfmin+1, 4);

for bit= 6:1:8 %16000: 1000: 24000

    clear Ttd recog

for tf=tfmin:tfmax

    clear Viln;

    clear Vtd1;

    clear Dd1;

    temp3=0;

    NTR1=zeros (G*H, 1);

    i=1; j=1; k=1; L=1;

for i=1:iter1

    xtemp1=Tid(:, :, i);

    xtemp=xtemp1(1:tf, 1:tf);

    xtempn= ( (xtemp) ./ (max (max (xtemp)) -min (min (xtemp)) ) )+.35;

    if xtempn(1,1) >= 1

        xtempn(1,1) = 1;

    end

    Viln(:, :, i)=floor(xtempn.*(2^bit-1));

    clear xtempn xtemp

end

```

```

correct1=0;
wrong1=0;
iter2=0;
% Testing Mode
ST=H+1;
EN=10;
for i= 1:G
    for j=ST:EN
        iter2=iter2+1;
        NTR=zeros(iter1,1);
        xta= double(imread(strcat('/Users/waleed/Pictures/att_faces/s',...
            int2str(i),'/', int2str(j)),'pgm'));
        xtar(:, :, iter2)=xta;
        xt=double(xta);
        It(:, :, iter2)=xt;
        Tt1=dctmxtrow20*xt;
        Tt=Tt1*dctmxtcol20;
        Ttd(:, :, iter2)=Tt;
        Vt1=Tt(1:tf, 1:tf);
        deft=max(max(Vt1))-min(min(Vt1));
        Vt1n=(Vt1)./deft+.35;
        if Vt1n(1,1) > 1
            Vt1n(1,1) = 1;
        end
        Vt1n=floor(Vt1n.*(2^bit-1));
        for L=1:iter1
            D1=Vi1n(:, :, L)-Vt1n;

```

```

Dd1 (:, :, L) = D1;

for k = 1:tf
    temp3 = temp3 + norm(D1(:, k));
    NT1 = temp3;
end

NTR1(L) = NT1;

temp3 = 0;

end

[Si1, Hi1] = min(NTR1);

NTRF1(:, :, iter2) = NTR1;

if CM(Hi1, 2) == i
    correct1 = correct1 + 1;
    rescl(correct1, :) = [iter2, Hi1, Si1];
else
    wrong1 = wrong1 + 1;
    resl(wrong1, :) = [iter2, Hi1, Si1];
    NTRFF1(:, :, wrong1) = NTR1;
end

end

end

recog(tf - tfmin + 1, 1) = tf;
recog(tf - tfmin + 1, 2) = correct1;
recog(tf - tfmin + 1, 3) = wrong1;
recog(tf - tfmin + 1, 4) = correct1 * 100 / (wrong1 + correct1);

end

ymin = min(min(recog(:, 4)) - 1);
ymax = max(max(recog(:, 4)) + 0.5);

```



```

titlestr = ['Varying the No. Of columns',...
            ' Of Test & Train Poses Matrix with ',...
            int2str(bit),' bits using ORL Database With ',...
            int2str(iter1),' Training & ',int2str(iter2), ' Testing'];
figure
plot(recog(:,1),recog(:,4),'-ro','LineWidth',2,...
      'MarkerEdgeColor','k',...
      'MarkerFaceColor','g',...
      'MarkerSize',10)
axis([tfmin tf ymin ymax])
title(titlestr)
xlabel('No. Of columns Of Pose Matrix')
ylabel('% of Accuracy')
hleg1 = legend('NTD');
set(hleg1,'Location','south')
grid on
end

```

LIST OF REFERENCES

- [1] “Morphotrust, USA, inc.” <http://www.morphotrust.com>.
- [2] “Cognitec systems GmbH, Germany,” <http://www.cognitec-systems.de>.
- [3] “Facekey corporation, USA,” <http://www.facekey.com>.
- [4] F. L. Podio and J. S. Dunn, “Biometric authentication technology: from the movies to your desktop,” in *ITL Bulletin, May 2001*, 2001.
- [5] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, Dec. 2003. [Online]. Available: <http://doi.acm.org/10.1145/954339.954342>
- [6] R. Chellappa, C. Wilson, and S. Sirohey, “Human and machine recognition of faces: a survey,” *Proceedings of the IEEE*, vol. 83, no. 5, pp. 705–741, 1995.
- [7] H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman Soulié, and T. S. Huang, *Face recognition: From theory to applications*. Springer Verlag, 1998, no. 163.
- [8] S. Gong, S. J. McKenna, and A. Psarrou, *Dynamic vision: From Images to Face Recognition*. Imperial College Press, 2000.
- [9] P. Phillips, P. Grother, R. Micheals, D. Blackburn, E. Tabassi, and J. Bone, “Face recognition vendor test (frvt) 2002: Overview and summary,” *Online*: http://biometrics.nist.gov/cs_links/face/frvt/FRVT_2002_Overview_and_Summary.pdf, 2003.
- [10] K. Peng, L. Chen, and S. Ruan, “A smart identification card system using facial biometric: From architecture to application,” in *Ubiquitous Intelligence and Computing*, ser. Lecture Notes in Computer Science, J. Ma, H. Jin, L. Yang, and J.-P. Tsai,

- Eds. Springer Berlin Heidelberg, 2006, vol. 4159, pp. 61–70. [Online]. Available: http://dx.doi.org/10.1007/11833529_7
- [11] M. Turk and A. Pentland, “Eigenfaces for recognition,” *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, Jan. 1991. [Online]. Available: <http://dx.doi.org/10.1162/jocn.1991.3.1.71>
- [12] M. Turk and A. Pentland, “Face recognition using eigenfaces,” in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, 1991, pp. 586–591.
- [13] J. Yang, D. Zhang, A. Frangi, and J.-Y. Yang, “Two-dimensional pca: a new approach to appearance-based face representation and recognition,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 1, pp. 131–137, 2004.
- [14] D. Zhang and Z.-H. Zhou, “Letters: (2d)2pca: Two-directional two-dimensional pca for efficient face representation and recognition,” *Neurocomput.*, vol. 69, no. 1-3, pp. 224–231, Dec. 2005. [Online]. Available: <http://dx.doi.org/10.1016/j.neucom.2005.06.004>
- [15] W. Alrasheed and W. Mikhael, “A quantized/truncated transform domain technique (qtd) for fast facial recognition,” in *Circuits and Systems (MWSCAS), 2011 IEEE 54th International Midwest Symposium on*, 2011, pp. 1–3.
- [16] N. Ahmed, T. Natarajan, and K. Rao, “Discrete cosine transform,” *Computers, IEEE Transactions on*, vol. C-23, no. 1, pp. 90–93, 1974.
- [17] W. Alrasheed and W. Mikhael, “A facial recognition technique employing frequency domain thresholding and quantization (fd-tq),” in *Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on*, 2012, pp. 1032–1035.

- [18] W. Alrasheed and W. Mikhael, "A facial recognition technique employing normalized histogram intensity," 2013, presented at the IEEE 56th International Midwest Symposium on Circuits And Systems (MWSCAS).
- [19] W. Alrasheed and W. Mikhael, "An efficient facial recognition technique based on normalized transform domain (ntd)," 2013, in preparation for the IET Electronics Letters.
- [20] M. Kirby and L. Sirovich, "Application of the karhunen-loeve procedure for the characterization of human faces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 1, pp. 103–108, 1990.
- [21] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *JOSA A*, vol. 4, no. 3, pp. 519–524, 1987.
- [22] C. Wang, L. Lan, Y. Zhang, and M. Gu, "Face recognition based on principle component analysis and support vector machine," in *Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on*, 2011, pp. 1–4.
- [23] Y. Choi, T. Tokumoto, M. Lee, and S. Ozawa, "Incremental two-dimensional two-directional principal component analysis (i(2d)2pca) for face recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 2011, pp. 1493–1496.
- [24] Z. Lin, Z. Wenrui, S. Li, and F. Zhijun, "Infrared face recognition based on compressive sensing and pca," in *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on*, vol. 2, 2011, pp. 51–54.
- [25] G. Sun, L. Zhang, and H. Sun, "Face recognition based on symmetrical weighted pca," in *Computer Science and Service System (CSSS), 2011 International Conference on*, 2011, pp. 2249–2252.

- [26] G.-C. Luh, "Face recognition using pca based immune networks with single training sample per person," in *Machine Learning and Cybernetics (ICMLC), 2011 International Conference on*, vol. 4, 2011, pp. 1773–1779.
- [27] A. Mohammed, R. Minhas, Q. Wu, and M. Sid-Ahmed, "A face portion based recognition system using multidimensional pca," in *Circuits and Systems (MWSCAS), 2011 IEEE 54th International Midwest Symposium on*, 2011, pp. 1–4.
- [28] Z. Liu and X. Lu, "Face recognition based on fractional fourier transform and pca," in *Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC), 2011*, vol. 2, 2011, pp. 1406–1409.
- [29] D. Zhang, S. Mabu, F. Wen, and K. Hirasawa, "A supervised learning framework for pca-based face recognition using gnp fuzzy data mining," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, 2011, pp. 516–520.
- [30] S. Bag and G. Sanyal, "An efficient face recognition approach using pca and minimum distance classifier," in *Image Information Processing (ICIIP), 2011 International Conference on*, 2011, pp. 1–6.
- [31] Y. Pang, D. Tao, Y. Yuan, and X. Li, "Binary two-dimensional pca," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 38, no. 4, pp. 1176–1180, 2008.
- [32] J. Yang and C. Liu, "Horizontal and vertical 2dpca-based discriminant analysis for face verification on a large-scale database," *Information Forensics and Security, IEEE Transactions on*, vol. 2, no. 4, pp. 781–792, 2007.

- [33] A. Xu, X. Jin, Y. Jiang, and P. Guo, "Complete two-dimensional pca for face recognition," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3, 2006, pp. 481–484.
- [34] C. Wang, B. Yin, X. Bai, and Y. Sun, "Color face recognition based on 2dpca," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1–4.
- [35] J. Wang and H. Yang, "Face detection based on template matching and 2dpca algorithm," in *Image and Signal Processing, 2008. CISP '08. Congress on*, vol. 4, 2008, pp. 575–579.
- [36] M. Safayani, M.-T. Manzuri-Shalmani, and M. Khademi, "Extended two-dimensional pca for efficient face representation and recognition," in *Intelligent Computer Communication and Processing, 2008. ICCP 2008. 4th International Conference on*, 2008, pp. 295–298.
- [37] H. Kong, X. Li, L. Wang, E. K. Teoh, J.-G. Wang, and R. Venkateswarlu, "Generalized 2d principal component analysis," in *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 1, 2005, pp. 108–113 vol. 1.
- [38] W. Sun and Q. Ruan, "Two-dimension pca for facial expression recognition," in *Signal Processing, 2006 8th International Conference on*, vol. 3, 2006, pp. –.
- [39] W. Zuo, K. Wang, and D. Zhang, "Bi-directional pca with assembled matrix distance metric," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 2, 2005, pp. II–958–61.
- [40] R. M. Mutelo, L. C. Khor, W. Woo, and S. Dlay, "A novel fisher discriminant for biometrics recognition: 2dpca plus 2dfld," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, 2006, pp. 4 pp.–.

- [41] P. Sanguansat, W. Asdornwised, S. Jitapunkul, and S. Marukatat, "Class-specific subspace-based two-dimensional principal component analysis for face recognition," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 2, 2006, pp. 1246–1249.
- [42] V. D. M. Nhat and S. Lee, "Kernel-based 2dpca for face recognition," in *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, 2007, pp. 35–39.
- [43] X. Chunming, J. Haibo, and Y. Jianjiang, "Robust two-dimensional principle component analysis," in *Control Conference, 2008. CCC 2008. 27th Chinese*, 2008, pp. 452–455.
- [44] M. Abdelwahab and W. Mikhael, "Recognition of noisy facial images employing transform - domain two-dimensional principal component analysis," in *Circuits and Systems, 2006. MWSCAS '06. 49th IEEE International Midwest Symposium on*, vol. 1, 2006, pp. 596–599.
- [45] M. M. Abdelwahab and W. B. Mikhael, "An efficient frequency domain algorithm for face recognition based on two-dimensional principal component analysis," in *Proceedings of the 4th WSEAS international conference on Electronics, control and signal processing*. World Scientific and Engineering Academy and Society (WSEAS), 2005, pp. 240–244.
- [46] "ORL data base," <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>.
- [47] "Yale data base," <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>.
- [48] "FERET data base," <http://www.itl.nist.gov/iad/humanid/feret/>.
- [49] R. C. Gonzalez and E. Richard, "Woods, digital image processing," *ed: Prentice Hall Press, ISBN 0-201-18075-8*, 2002.
- [50] W. K. Pratt, "Digital image processing," 2001.
- [51] A. K. Jain, *Fundamentals of digital image processing*. Prentice-Hall Englewood Cliffs, 1989, vol. 3.

- [52] W. Burger and M. J. Burge, *Digital image processing: an algorithmic introduction using Java*. Springer, 2008.
- [53] H. Demirel and G. Anbarjafari, “A new face recognition system based on color histogram matching,” in *Signal Processing, Communication and Applications Conference, 2008. SIU 2008. IEEE 16th*, 2008, pp. 1–4.
- [54] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, “Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition,” in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, 2005, pp. 786–791 Vol. 1.
- [55] B. Zhang, S. Shan, X. Chen, and W. Gao, “Histogram of gabor phase patterns (hgpp): A novel object representation approach for face recognition,” *Image Processing, IEEE Transactions on*, vol. 16, no. 1, pp. 57–68, 2007.
- [56] H. Tang, Y. Sun, B. Yin, and Y. Ge, “Face recognition based on haar lbp histogram,” in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, vol. 6, 2010, pp. V6–235–V6–238.
- [57] C.-H. Chan and J. Kittler, “Sparse representation of (multiscale) histograms for face recognition robust to registration and illumination problems,” in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, 2010, pp. 2441–2444.
- [58] A. Sevcenco and W.-S. Lu, “Histogram-enhanced principal component analysis for face recognition,” in *Communications, Computers and Signal Processing, 2009. PacRim 2009. IEEE Pacific Rim Conference on*, 2009, pp. 175–180.
- [59] A. Gersho and R. M. Gray, *Vector quantization and signal compression*, 1991. Kluwer Academic Publishers.

- [60] M. Abdelwahab and W. Mikhael, "Efficient feature representation employing pca and vq in the transform domain for facial recognition," in *Circuits and Systems, 2007. MWSCAS 2007. 50th Midwest Symposium on*, 2007, pp. 281–284.
- [61] M. Abdelwahab and W. Mikhael, "Neural network pattern recognition employing multicriteria extracted from signal projections in multiple transform domains," in *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on*, 2001, pp. 40–43.