A Comparative Study of DCT and Kekre’s Median Code Book Generation Algorithm for Face Recognition

H B Kekre¹, T K Sarode², V A Bharadi³
¹Sr. Professor, ²³Ph.D. Scholar, MPSTME, NMIMS University, Vileparle (w), Mumbai-56, India
²Assistant Professor, TSEC, Mumbai-50, India.
³Assistant Professor, TCET, Mumbai-101, India.
¹hbkekre@yahoo.com

ABSTRACT
In this paper we propose novel Face Recognition method based on vector quantization (VQ) using Kekre's Median Code Book Generation (KMCG) algorithm. The performance of the proposed method is compared with the well known face recognition method based on Discrete Cosine Transform (DCT). Both the methods are implemented on Georgia Tech Database of 750 images consisting of 15 images for each of 50 individuals. From the results it is observe that our proposed method gives 92.67 % accuracy as compared to DCT. Further it is observed that KMCG requires 99.45% computation less than DCT.

Categories and Subject Descriptors
I.4.7 Image Processing and Computer vision

General Terms

Keywords
Biometrics, Face recognition, DCT, KMCG, VQ.

1. INTRODUCTION
Biometrics is the science of identifying a human being by its measurable physical and behavioral traits. Biometrics is now inevitable in next generation security systems, as these systems are exploiting the uniqueness of human traits for identification and authorization [1]. Biometrics is not new for the mankind, we have been using signature for decades for authorization of documents etc. Another such example is the use of human fingerprints in forensic as well as commercial domains. With the advancement in engineering we have more powerful computers, which enable us to explore various biometric identification mechanisms in detail.

The biometrics has a significant advantage over traditional authentication techniques (namely passwords, PIN numbers, smartcards etc.) due to the fact that biometric characteristics of the individual are not easily transferable, are unique of every person, and cannot be lost, stolen or broken. The choice of one of the biometric solutions depends on several factors [1]:

- User acceptance
- Level of security required
- Accuracy
- Cost and implementation time

Fingerprints, Palmprints, signature, voice, iris, retina, face, gait and DNA sequence are amongst the most widely used biometric traits.

Among all biometrics listed above, face biometric is unique because face is the only biometric belonging to both physiological and behavioral categories. While the physiological part of the face biometric has been widely researched in the literature, the behavioral part is not yet fully investigated. In addition, as reported in [2], face has advantage over other biometrics because it is a natural, non-intrusive, and easy-to-use biometric. For example [2], among the biometrics of face, finger, hand, voice, eye, DNA and signature, the face biometric ranks first in the compatibility evaluation of a machine readable travel document (MRTD) system on the basis of six criteria: enrollment, renewal, machine-assisted identity verification requirements, redundancy, public perception, and storage requirements and performance.

Probably the most important feature of a biometric is its ability to collect the signature from non-cooperating subjects. Besides applications related to identification and verification such as access control, law enforcement, ID and licensing, surveillance, etc., face recognition is also useful in human-computer interaction, virtual reality, database retrieval, multimedia, computer entertainment, etc. [3]

Face recognition mainly involves the following three tasks [4]:

- Verification - The recognition system determines if the query face image and the claimed identity match.
- Identification - The recognition system determines the identity of the query face image by matching it with a database of images with known identities, assuming that the identity of the quest face image is inside the database.
- Watch list - The recognition system first determines if the identity of the query face image is in the stored watch list and, if yes, then identifies the individual.

In this paper we discuss face recognition using DCT (Transform) & vector quantization (VQ) based approaches and the performance of both the systems is compared. In the next section we discuss some of the existing approaches related to our work.
2. EXISTING FACE RECOGNITION METHODS

Besides VQ & DCT many approached are used for face recognition, main factor to build a face recognition system is to extract a personal facial feature vector, which will be used for matching faces. A lot of feature vector extraction methods are available, in [5] Turk et. al. have used eigenface based feature vector extraction for face recognition. Another approach is based on Local feature analysis (LFA) [6] both of these approaches are PCA based approaches.

Gabor filters are also widely used for extracting facial feature vectors. Zhang et. al.[7] have used local Gabor binary patterns. They used a reduced set of local histograms based on Local Gabor Binary Patterns (LGBP). In [8] Gonzalez & Castro have proposed another method based on Gabor filter which uses Shape driven Gabor Jets for face description and Authentication.

Kotani and Quiu [9] have used vector Quantization of face to generate a codevector histogram, the codebook is defined as a 33 different variation in grey levels. They generated a codevector histogram and matched them. This method is robust towards grey level intensity variations.

DCT has been used as a feature extraction step in various cases on face recognition. Up to now, either DCT features have been used in a holistic appearance-based sense [10], or local appearance-based sense which ignores spatial information during the classification step.

2.1 DCT Feature Extraction

The DCT algorithm that we have used for our study on face recognition is as shown below:

1) Read the database image (Size=100x130x3)
2) Extract the Red, Green and Blue component of that image such that each is of size 100x130.
3) Apply DCT to each component and append in a new column the result for each Red, Green and Blue in matrix form. So we get 100x390 entries. This is the Feature Vector (F.V) of that image.
4) Repeat steps 1 through 3 for every database image.
5) Read the Query image.
6) Repeat step 2 and 3 for the query image so as to obtain its Feature Vector.
7) For every Database image ‘i’ and a Query image ‘q’ Calculate the Mean Squared Error (MSE) using the Equation 1.

\[
MSE(i) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [FV_i(m,n) - FV_q(m,n)]^2
\]

Where
- \(M=100\)
- \(N=390\)
- \(FV_i\) = Feature vector of \(i^{th}\) database image
- \(FV_q\) = Feature vector of \(q^{th}\) query image

Determine the minimum MSE and corresponding image matching it. We discuss the accuracy of this method in the results section, next we discuss the Vector Quantization based feature vector generation.

3. VECTOR QUANTIZATION

Vector Quantization (VQ) [11]-[18] is an efficient technique for data compression and has been successfully used in variety of research fields such as speech recognition and face detection [23], image segmentation [19][20][21], speech data compression [22], content based image retrieval CBIR [26][27] and face recognition [9][24].

Vector Quantization (VQ) techniques employ the process of clustering. Various VQ algorithms differ from one another on the basis of the approach employed for cluster formations. VQ is a technique in which a codebook is generated for each image. A codebook is a representation of the entire image containing a definite pixel pattern which is computed according to a specific VQ algorithm. The image is divided into fixed sized blocks that form the training vector. The generation of the training vector is the first step to cluster formation. The method most commonly used to generate codebook is the Linde-Buzo-Gray (LBG) algorithm which is also called as Generalized Lloyd Algorithm (GLA) [11]. In this paper, we carry out face Recognition using KMCG.

3.1 Proposed Face Recognition Technique

Vector Quantization is basically the clustering algorithm, where image is divided into pixel windows. Here we have divided image in 1x2 pixel and the windows and then converted to vector of size 6. The Kekre’s Median Codebook Generation (KMCG) [15] algorithm is applied on this set to get image signature (feature vector).

3.2 Feature Extraction

Here the feature space has 128*6 number of elements. This is obtained using following steps of Kekre’s Median Codebook Generation (KMCG) algorithm

1. Image is divided into the windows of size 1x2 pixels (each pixel consisting of red, green and blue components).
2. These are put in a row to get 6 values per vector. Collection of these vectors is a training set.
3. The training set is sorted with respect to first column. The centre value of first column is used to divide the training set in two parts.
4. Further each part is then separately sorted with respect to second column to get two centre values.
5. The process of sorting is repeated till we get 128 centre values.
6. Using these centre values as codewectors, Codebook of size 128*6 is generated.
7. The codebook is stored as the feature vector for the image. Thus the feature vector database is generated.

3.3 Query Execution

Here the codebook of size 128*6 for the query image is extracted using Kekre’s Median Codebook Generation Algorithm and the feature vector of query image is obtained. Then this feature set is compared with other feature sets in feature database using Euclidian distance as similarity measure. As compared to taking complete DCT of the image, this proposed VQ based techniques saves tremendous number of computations.

4. RESULTS

In the above implemented methods, we have not done any pre-processing on the face images in the database or the query images. We have Georgia Tech Database [25] of 750 images consisting of 50. Each individual has 15 images. We have resized each image to
a 100 x 130 colour pixels. Thus, we have a 3- dimensional image sized 100 x 130 x 3. Each of these methods are implemented using MATLAB 7.0 on Intel Pentium Dual Core Processor (2.01 GHz), 2GB RAM on Windows XP Professional SP3.

Figure 1 shows the sample images for two individuals, 15 different images with different poses for each.

Here we divided the database of 750 into two parts.
1. 400 images i.e. 8 images per individual as a training set, and remaining 7 images per individual as a test set, making test set of size equal to 350.
2. 450 images i.e. 9 images per individual as a training set, and remaining 6 images per individual as a test set, making test set of size equal to 300.

(a)
(b)

Figure 1. Database Images (a) and (b) represents 15 images of 2 subjects in the database

4.1 Accuracy Comparison / Analysis
Accuracy shown here implies the number of correctly identified faces to the total number of images queried for recognition. Table 1 shows the percentage accuracy of the DCT and KMCG method for face recognition applied for two cases:
1. Training set 400, Test set 350
2. Training set 450, Test set 300

<table>
<thead>
<tr>
<th>Training Set</th>
<th>DCT</th>
<th>KMCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>400, 350</td>
<td>75.43%</td>
<td>86.86%</td>
</tr>
<tr>
<td>450, 300</td>
<td>90.33%</td>
<td>92.67%</td>
</tr>
</tbody>
</table>

Table 1. Comparison of the Different Algorithms tested

From Table 1, it is observed that KMCG gives 86.86% and 92.67% accuracy for training set sizes 400 and 450 respectively outperforming DCT with less percentage accuracy 75.43 and 90.33 for training set sizes 400 and 450 respectively.

4.2 Complexity Analysis
Let M be the total number of training vectors, k be the vector dimension,
N be the codebook size,
P = \log_2 N
1 CPU unit is required for addition of 8 bit numbers
For multiplication of two 8 bits number 8 CPU units are required.

Table 2. KMCG Algorithm with respect to total number of comparisons, Total No. of ED Computations & Total CPU Units required.

<table>
<thead>
<tr>
<th>Complexity Parameters</th>
<th>KMCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Comparisons</td>
<td>( M \sum_{i=0}^{P} \log_2 (M / 2^i) )</td>
</tr>
<tr>
<td>Total No. of ED</td>
<td>0</td>
</tr>
<tr>
<td>Total CPU units</td>
<td>( M \sum_{i=0}^{P} \log_2 (M / 2^i) )</td>
</tr>
</tbody>
</table>

For full 2-Dimensional DCT for an PxQx3 image the number of multiplications required are 3xPxQx(P+Q) and number of additions required are 3xPxQx(P+Q-2). Total CPU units required for full 2-Dimensional DCT = 8x3xPxQx(P+Q) + 3xPxQx(P+Q-2).

Table 3. Total CPU units required for DCT and KMCG.

<table>
<thead>
<tr>
<th>Complexity Parameters</th>
<th>DCT</th>
<th>KMCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total CPU Units</td>
<td>80652000</td>
<td>476640</td>
</tr>
</tbody>
</table>

As we can see in Table 3, it is observed that DCT requires 169.02 times more computations than KMCG.

5. CONCLUSIONS
In this paper we have discussed face recognition using DCT and KMCG. These methods are implemented on face database without any pre-processing. We have obtained high accuracy of 86% using KMCG against 75.43% of DCT on a database of 400 training images (8 per person) and 92.67% in KMCG against 90.33% of DCT on a database of 450 training images (9 images per person). Further it is observed that even in the worst case KMCG requires 99.45% computation less than DCT.

6. REFERENCES


