# Principal Component Analysis Based Image Recognition

[1]J.ASHOK, [2]DR.E.G.RAJAN

[1]Professor and Head , Department of Information Technology, GCET, Hyderabad,India.
[2]President, Pentagram Research Center, Hyderabad, India

E-Mail: [1]jammiashok@yahoo.co.in,[2]rajaneg@yahoo.co.in

***Abstract***

The aim of this paper is to recognize a query image from a database of images. This process involves finding the principal component of the image, which distinguishes it from the other images. Principal Component Analysis (PCA) is a classical statistical method and is widely used in data analysis. The main use of PCA is to reduce the dimensionality of a data set while retaining as much information as possible. This paper uses the concept of PCA to recognize images by extracting their principal components. In this I have used 32X32, 24 bit gray scale bitmap images. For extraction of image attributes, the data file is extracted from the bitmap image file format. Discrete Cosine Transform is used to reduce the size of the data set so that the most relevant intensity information of the query image is contained in a first few lower order frequency components. PCA is used to extract the unique characteristic of the query image, which distinguishes it from the other images. Hence comparison of the query image with the database of images will result in an exact match.

***Keywords:*** Principal Component Analysis, bitmap, Common Factor Analysis, Discrete Cosine Transform

## 1. INTRODUCTION

Basic recognition system involving image database techniques would require the following modules

a) Creating a database of images such that certain points of uniqueness are extracted.
b) Selecting a suitable recognition algorithm as per the application.
c) Providing a query image.
d) Extracting the unique feature sets of the image provided as the query.
e) Comparing the feature sets of the query with those in the available database using a suitable comparison algorithm.
f) Determining the authenticity from the results obtained.

## 2. BITMAPS

Bitmaps are a standard devised jointly by Microsoft and IBM for storing images on IBM compatible PC's running the Windows software. A bitmap is logically divided into two basic portions, the header section and the data section. The size of the header is 54 bytes. The header usually contains a type field and an offset field. The type field legitimizes the fact that the file is actually a bitmap. The offset field tells us where the data section of the bitmap begins relative to the Start of File. The information regarding image attributes such as the image resolution, height, width, and the number of bits per pixel are contained in the information part of the header, termed as the information header for the sake of convenience. This portion of the header constitutes 40 bytes of the total 54 bytes of the header.

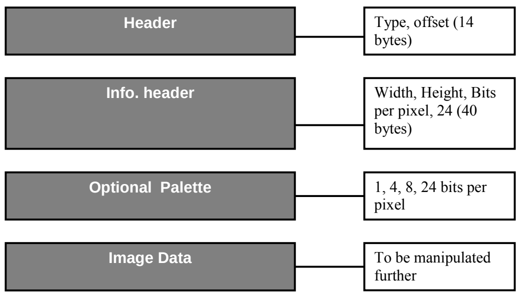



FIG. BIT MAPS FORMAT

Bitmaps are simple to store and the format is easy to understand. This makes the extraction and eventually the manipulation of the data easy. Performing an image recognition technique calls for reading of its pixel values into a separate file so that required features can be extracted. Once image

values have been extracted, their unique feature sets need to be identified to perform the image recognition procedure. This may be done by choosing a proper recognition algorithm. Several algorithms may be used for the recognition process. Some of them are Multidimensional Scaling, Singular Value Decomposition (SVD) and Factor analysis (FA).MDS provides a visual representation of the patterns of proximities in the data set. It is applied on a matrix containing distance measures between the variables of the data set. It cannot cope with reflections of variables. Increased number of computations makes it undesirable for applications such as image recognition. SVD and FA, both attempt to highlight the variances amongst a given set of data. SVD can be applied only to singular matrices. It attempts to represent a singular matrix S as the product of an m x n orthogonal matrix, U, an
n x n diagonal matrix, W, and the transpose of an n x n orthogonal matrix, V. The diagonal elements of the matrix W give the singular values of the matrix S. Factor analysis aims to highlight the variances amongst a large number of observable data through a set of unobservable data called factors. FA can be approached in two ways. They are

a) Principal Component Analysis

b) Common Factor Analysis

Common factor analysis considers the common variances of the variables concerned. It results in a testable model that explains intercorrelations amongst the variables. PCA considers the total variance of the data set. Hence given an image input, PCA would summarize the total variance in image values. It aims to fit this total variance into a reduced number of variables thus ***reducing*** the data size. It results in a *principal component*, which summarizes or represents the entire data set with the relevant information retained. This is important as the image information is large in size. A reduced data set makes computations easier. PCA has the advantage of being a simple algorithm, based on direct user defined inputs. It can accept both subjective and objective attributes and always results in a unique solution. It can be applied to a semantic environment making its use desirable.
All these advantages make PCA the most desirable procedure to be used as a recognition algorithm.

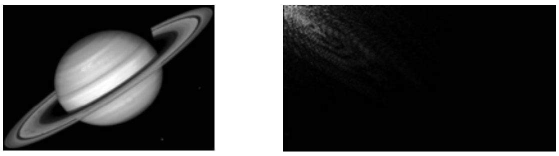
## 3. DISCRETE COSINE TRANSFORM

Transform coding constitutes an integral component of contemporary image/video processing applications. Transform coding relies on the premise that pixels in an image exhibit a certain level of correlation with their neighbouring pixels. A transformation is, therefore, defined to map this spatial (correlated) data into transformed

(uncorrelated) coefficients. Clearly, the transformation should utilize the fact that the information content of an individual pixel is relatively small i.e., to a large extent visual contribution of a pixel can be predicted using its neighbours. Some properties of the DCT, which are of particular value to image processing applications:

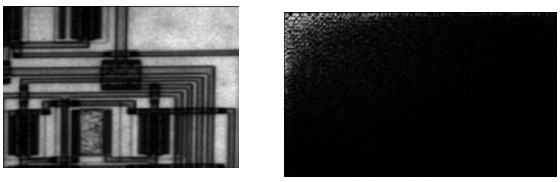
**De correlation:** The principle advantage of image transformation is the removal of redundancy between neighbouring pixels. This property helps us to manipulate the uncorrelated transform coefficients as they can be operated upon independently.

**Energy Compaction:** This property can be explained by using a few examples given below with figs

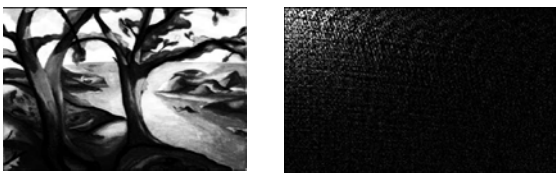


**(a) SATURN AND ITS DCT**

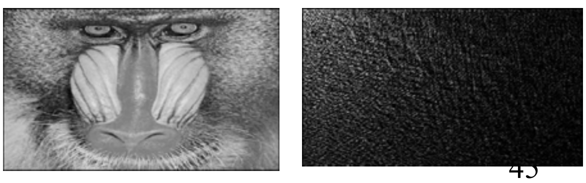
**(b) Child and its DCT**



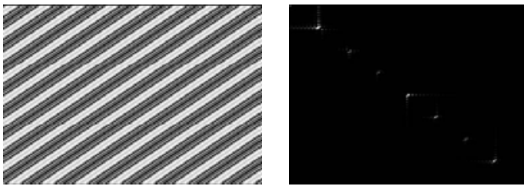
**(c) Circuit and its DCT**



**(d) Trees and its DCT**



**(e ) Baboon and its DCT**



**(f) A sine wave and its DCT.**

The example comprises of four broad image classes. Figure (a) and (b) contain large areas of slowly varying intensities. These images can be classified as low frequency images with low spatial details. A DCT operation on these images provides very good energy compaction in the low frequency region of the transformed image. Figure (c) contains a number of edges (i.e., sharp intensity variations) and therefore can be classified as a high frequency image with low spatial content. However, the image data exhibits high correlation, which is exploited by the DCT algorithm to provide good energy compaction. Figure (d) and (e) are images with progressively high frequency and spatial content. Consequently, the transform coefficients are spread over low and high frequencies. Figure (e) shows periodicity therefore the DCT contains impulses with amplitudes proportional to the weight of a particular frequency in the original waveform. The other (relatively insignificant) harmonics of the sine wave can also be observed by closer examination of its DCT image.

Hence, from the preceding discussion it can be inferred that DCT renders excellent energy compaction for correlated images. Studies have shown that the energy compaction performance of DCT approaches optimality as image correlation approaches one i.e., DCT provides (almost) optimal decorrelation for such images. Thus, the ease of using DCT lies in its innate ability to ensconce relevant information into a few lower frequency components leading to data reduction. An image is essentially a collection of a large amount of data and reduction of its size necessarily reduces the number of computations required. The general equation for a 2D ($N$ by $M$ image) DCT is defined by the following equation:

$$F(u,v)=\sqrt{(\frac{2}{N})}\sqrt{(\frac{2}{M})}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\Lambda(i).\Lambda(j).\cos\left[\frac{\Pi.u}{2.N}(2i+1)\right]\cos\left[\frac{\Pi.v}{2.M}(2j+1)\right].f(i,j)$$

Where

$$\Lambda(\xi)=\begin{cases}\frac{1}{\sqrt{2}} & \text{for } \xi=0\\ 1 & otherwise\end{cases}$$

The good performance shown by 2D DCT with PCA method is direct result of coupling 2D DCT coefficients with PCA method. It mainly benefits from using 2D discrete cosine transform to eliminate the correlation between rows and columns and get energy concentrated at the same time. Since only selected 2D DCT coefficients formed the feature space, the feature space dimension is greatly reduced.

## 4. COVARIANCE

Covariance is a property that gives us the amount of variation of two dimensions from their mean with respect to each other. This property is used on the DCT-matrix so that a form of distance measure is performed on the image pixel values thus providing their relative intensity measures. The covariance of two variants $X_i$ and $X_j$ provides a measure of how strongly correlated these variables are, and the derived quantity

$$cor(x_i,x_j)\equiv\frac{cov(x_i,x_j)}{\sigma_i\sigma_j},$$

where $\sigma_i$, $\sigma_j$ are the standard deviations, is called statistical Correlation of $x_i$ and $x_j$. The covariance is symmetric since $\operatorname{cov}(x,y)=\operatorname{cov}(y,x)$ Covariance is performed on the extracted 3*3 DCT matrix and covariance matrix is calculated using the following formulae

Given n sets of variants denoted $\{x_1\},\ldots..,\{x_n\}$, the first-order covariance matrix is defined by

$$Vij=cov(x_i,x_j)\equiv\langle(x_i-\mu_i)\ (x_j-\mu_j)\rangle,$$

Where $\mu_i$ is the mean.

Higher order matrices are given by

$$Vi^m j^n=\langle(x_i-\mu_i)^m\,(x_j-\mu_j)^n\rangle$$

An individual matrix element $v_{ij}=\operatorname{cov}(x_i,x_j)$ is called the covariance of $x_i$ and $x_j$.

## 5. CHARACTERISTIC EQUATION

Then a characteristic equation is generated from the covariance matrix. This characteristic equation is a cubic root equation and the maximum root is found out using Cardan's method. The maximum root is the eigen value i.e. the principal component of the data set which uniquely identifies the image.

## 6. EIGEN VALUES AND EIGEN VECTORS

From a symmetric matrix such as the covariance matrix, we can calculate an orthogonal basis by finding its eigen values and eigen vectors. The eigenvectors $e_i$ and the corresponding eigen values $\lambda_i$ are the solutions of the equation

$$C_x e_i = \lambda_i e_i, \; i = 1, 2, \ldots n$$

For simplicity we assume that the eigen values $\lambda_i$ are distinct. These values can be found, for example, by finding the solutions of the characteristic equation

$$|C_x - \lambda I| = 0$$

where the I is the identity matrix having the same order than $C_x$ and the |.| denotes the determinant of the matrix. We are here faced with contradictory goals:

1. On one hand, we should simplify the problem by reducing the dimension of the representation.

2. On the other hand we want to preserve as much as possible of the original information content.

PCA offers a convenient way to control the trade-off between loosing information and simplifying the problem at hand.

## 7. PRINCIPAL COMPONENT ANALYSIS STEPS

The following are the steps needed to perform a principal component analysis (PCA) on a set of data are:

1. Give the input in matrix form (here DCT matrix)

2. Subtract the mean: For PCA to work properly, we have to subtract the mean from each of the data dimensions. The mean subtracted is the average across each dimension. This produces a data set whose mean is zero.

3. Calculate the covariance matrix

4. Calculate the eigen values and eigen vectors of the covariance matrix: Since the covariance matrix is a square matrix, we can calculate the eigen values and eigen vectors for this matrix. These are rather important, as they tell us useful information about our data. But, more importantly, they provide us with information about the patterns in data. So, by this process of taking the Eigen vectors of the covariance matrix, we have been able to extract the lines that characterize data.

5. Choosing components and forming a feature vector: Here is where the notion of data compression and reduced dimensionality comes into it. In fact, it turns out that the Eigen vector with the highest Eigen value is the principal component of the data set. It is the most significant relationship between the data dimensions. In general, once Eigen vectors are found from the covariance matrix, the next step is to order them by Eigen value, highest to lowest. This gives the components in order of significance. Now we can ignore the components of lesser significance. We do lose some information, but if the Eigen values are small, we don't lose much information. So the final data set will have lesser dimensions than the original.

## 8. LEAST MEAN SQUARE ALGORITHM

Having derived the principal component of the query image, it is now left to the program the image recognition process. The database of images is a collection of the Eigen vectors of the different images that correspond to the maximum Eigen value. The Maximum Eigen value obtained from PCA for the query image is substituted in the Covariance matrix that has been used in solving the characteristic equation to get the maximum Eigen Value, to obtain the Eigen Vector corresponding to the query image. The Eigen vectors play an important role in the image recognition process. We need to define stepwise rules that need to be executed to achieve image recognition. Hence we need to adopt an algorithm that is suitable for the environment we work in. There are basically two kinds of environments classified based on the adaptation nature. They are described below:

**1. Static Environment**: The system is said to be static in nature if there is a negative output for a new image given as a input. The system responds negatively even when a existing image in the database is rotated or translated or is subjected to changes in contrast, brightness, scaling factors etc and is fed to the system for recognition process. In short, the statistics are constant during the entire process of running.

**2. Dynamic Environment**: The system is said to be dynamic in nature if there exists a learning process if any image is fed for the recognition process. The system initially responds negatively once a new image is presented. But, it extracts its principal component and stores it in the database so that it can respond positively when the earlier image is given as an input again in future. In short, the statistics vary with time during the process of running. The algorithm should always compare the principal component of the query image with the principal components of the database of images in a sequence.

**Least Mean Square Algorithm:**

Generally, LMS algorithm is implemented using the instantaneous values for the input function. The name itself defines the rough formula, as LMS is a least value of the squares of the means calculated with respect to a maximum Eigen value calculated for a particular Eigen space. The major property of LMS algorithm is that it operates linearly.

**Advantages of LMS algorithm**: The reasons of adopting LMS in our paper are due to its properties like Model independence, Robustness, Optimality in results and Simplicity

**LMS formula**: After we defined a general formula, as we adopted a statistical and matrices concepts, we necessarily define a specific formula as follows:

LMS value = Summation {square [ (Cij-Em)q-(Cij-Em)d] } (i,j)

where

i : row element indicator

j : column element indicator

Cij : Covariance element at ith row and jth column

Em: Maximum Eigen value

Subscript q: query image element

Subscript d: Database image element

We have defined that for LMS value that computes:

1. '0'will give a exact match
2. Otherwise, a "no match".

It is reminded that image recognition using PCA is a not a similarity match.

## 9. ALGORITHM

1. Give the query image as the input.
2. Convert the input from BMP to data file.
3. Perform Discrete Cosine Transform on the dataset.
4. Extract 3X3 matrix based on subjective analysis.
5. Perform covariance on the 3X3 matrix.
6. Generate characteristic equation from the covariance matrix and solve for maximum Eigen value.
7. Perform comparison of Eigen values by Least Mean Square Algorithm (LMS).
8 If LMS=0, recognize and display image; else no match found.

## 10. INFERENCES

For a static environment, the algorithm gives a maximum robustness because we applied matrices and statistics that are constant with respect to time. Unless we subject the image to rotation or translation or scaling factors, the LMS will be effective. For a dynamic environment, the convergence will occur definitely at some point but the rate of convergence will be slow. By convergence, we mean the exact recognition process. Moreover, due to dynamic learning methods, the number of iterations will be high and also the number of computations that require a large memory space and hardware. The sensitivity (S) of the LMS algorithm is calculated on the basis of a Condition Number (CN). CN is defined as the ratio of maximum Eigen value to the minimum Eigen value.

CN=(Max Eigen value) / (Min Eigen value)

The sensitivity S, is directly proportional to the Condition Number CN. If then CN is high, the system is ill conditioned and the sensitivity becomes acute.

## 11. CONCLUSIONS

This paper finds immense applications pertaining to security aspects in government agencies and public places with economical software implementation. Having thoroughly researched this area of Digital Image Processing, certain drawbacks have been observed during the course of the project. Firstly, an image that is already present in the formed database, when rotationally or transactionally scaled is not recognized by the system. Though such a system may prove to be an asset for security purposes it fails when the application involves the classification and categorization of images based on their content. Secondly, an image whose brightness or contrast has been changed from the original, though present in the database will not be recognized by the system. Hence put simply, the system is susceptible to the ageing and discoloration of the image. Thirdly, it is observed that the deployment of the application is flawless when an image of size 32X32 is used. Images of larger sizes would require more number of computations. Also it would not be enough to distinguish an image of a larger size with the help of a single principal component from other images since subjective analysis of an image differs from user to user. Finally, the use of bitmapped images results in the occupation of a significant amount of memory. The rectification of these features would indeed be a great step forward. It is these considerations that form the core of the future prospects of the software. Rotational and translational changes and changes in the brightness and contrast of the image can be accommodated by making use of the concepts of training through neural networks.

## 12. REFERENCES

[1] K. Ramchandran, A. Ortega, K. Metin Uz, and M. Vetterli, "Multiresolution broadcast for digital HDTV using joint source/channel coding." *IEEE Journal on Selected Areas in Communications*, 11(1):6–23, January 1993.

[2] M. W. Garrett and M. Vetterli, "Joint Source/Channel Coding of Statistically Multiplexed Real Time Services on Packet Networks," *IEEE/ACM Transactions on Networking*, 1993.

[3] S. McCanne and M. Vetterli, "Joint source/channel coding for multicast packet video," International Conference on Image Processing (Vol. 1)-Volume 1, October 1995, Washington D.C**.**

[4] K. Sayood and J. C. Borkenhagen, "Use of residual redundancy in the design of joint source/channel coders," *IEEE Transactions on Communications*, 39(6):838-846, June 1991.

[5] J. Modestino, D.G. Daut, and A. Vickers, "Combined source channel coding of images using the block cosine transform," *IEEE Transactions on Communications*, vol. 29, pp.1261-1274, September 1981.

[6] T. Cover, "Broadcast Channels," *IEEE Transactions on Information Theory,* vol. 18, pp. 2- 14, January 1972.

[7] Q. Zhang, Z. Ji, W. Zhu, J. Lu, and Y.-Q. Zhang, "Joint power control and source-channel coding for video communication over wireless," *IEEE VTC'01*, October 2001, New Jersey.

[8] Q. Zhang, W. Zhu, Zu Ji, and Y. Zhang, "A Power-Optimized Joint Source Channel Coding for Scalable Video Streaming over Wireless Channel," *IEEE ISCAS'01*, May, 2001, Sydney, Australia.

[9] T1.523-2001, American National Standard: Telecom Glossary 2000.

[10] Hayder Radha, "Lecture Notes: ECE 802 - Information Theory and Coding," January 2003.

[11] R. C. Gonzalez and P. Wintz, "Digital Image Processing," Reading. MA: Addison-Wesley, 1977.

[12] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-32, pp. 90-93, Jan. 1974. *ECE 802 – 602: Information Theory and Coding*

[13] W. B. Pennebaker and J. L. Mitchell, "JPEG – Still Image Data Compression Standard," Newyork: International Thomsan Publishing, 1993.

[14] G. Strang, "The Discrete Cosine Transform," *SIAM Review*, Volume 41, Number 1, pp. 135-147, 1999.

[15] R. J. Clark, "Transform Coding of Images," New York: Academic Press, 1985.

[16] A. K. Jain, "Fundamentals of Digital Image Processing," New Jersey: Prentice Hall Inc., 1989.

[17] A. C. Hung and TH-Y Meng, "A Comparison of fast DCT algorithms," *Multimedia Systems*, No. 5 Vol. 2, Dec 1994.

[18] G. Aggarwal and D. D. Gajski, "Exploring DCT Implementations," UC Irvine, Technical Report ICS-TR-98-10, March 1998.

[19] J. F. Blinn, "What's the Deal with the DCT," *EEE Computer Graphics and Applications*, July 1993, pp.78-83.

[20] C. T. Chiu and K. J. R. Liu, "Real-Time Parallel and Fully Pipelined 2-D DCT Lattice Structures with Application to HDTV Systems," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 2 pp. 25-37, March 1992.

## 13. BIOGRAPHY

Prof J.Ashok is currently working as Professor and Head of Information Technology at Geethanjali College of Engg. & Technology, Hyderabad, A.P, INDIA. He has received his B.E. Degree from Electronics and Communication Engineering from Osmania University and M.E. with specialization in Computer Technology from SRTMU, Nanded, INDIA. His main research interest includes neural networks, data retrieval process and Artificial Intelligence. He has been involved in the organization of a number of conferences and workshops. He has been published number papers in national and International journals and conferences. He is currently doing his PhD from Anna University.

Prof Dr. E.G. Rajan is an Electronics Engineer and a Professor of signal Processing having about 30 years of experience in teaching, research and administration. He has a number of publications to his credit. He is a Professional member of ACM and he is an editor of the Journal of AMCE, Barcelona, Spain. He received his Ph.D degree in electrical Engineering from Indian Institute of Technology(IIT),Kanpur, U.P., M.E. degree in Applied Electronics from Madras University. His Contribution to the state of the art of Electronic welfare has been well recognized in the Government and industrial sectors. He received distinguished Scientist and man of the Millennium award from who is who Bibliographical records,Cambridge,2000. He authored many books like, symbolic Computing, Signal and Image Processing, Electronic Order of Battle Records of Military Radars, Computers and Information Technology. Two of his Fifteen Ph. D scholars are involved in the design of digital Circuits using Oganic molecules and some of them are working in the Novel area of Genomic Signal Processing. He has brought out more than 25 original concepts.