International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 9, September- 2014

Gesture Controlled Robot

Suresh. J. Rai, Sonal Narayankar
Rijo Rajan, Ronald. M. Laban
St John College of Engineering and Technology
UNIVERSITY OF MUMBAI

Abstract **- This paper presents a simple but efficient method to implement hand gesture recognition using Principal Components Analysis. We first created an Image Database consisting of five different hand gesture images. Before populating the database for an images of various gesture categories in Hand Gesture Recognition system, each image was first processed i.e., the images were converted to 8-bit binary images and filtering was performed to minimize any noise present in the images. The method mentioned above were applied on the input test images captured form the sensor device of the system to find the suitable match form the data base. The methods used were successful to retrieve the correct matches. The results based on speed and accuracy was analyzed.And also according to the particular gesture a specified value is received by the robot microprocessor and according to its programming the robot moves forward, backward, left,right or stops**

Index Terms**— Euclidean, Gradient, Principal Component Analysis (PCA), Rotation Invariant.**

I. INTRODUCTION

This paper is based on the study and implementation of a pattern recognition system that was used to identify digitalimages of hand gestures. In this paper, we identified five different types of hand-gestures. The aim is to study the PCA methods that allow us to implement a hand gesture recognition system. Moreover, therecognition has to be done by one camera and in real time, so that one can operate as fast as he wants to. The sensor device used is an USB web cam. So, this makes it possible for any user to use it in his office or home. The system was developedusing MATLAB 2010b on Windows XP Operating System. Theimages after being captured through the web cam were saved in the database using .bmp format. The images are labeled using integer numbers starting from 1. The database was created using fifty different images for all the five hand gestures.

The method which was studied was:

.

Principal Component Analysis: The goal is to compute and study the Eigenvectors of the different pictures and then to express each image with its principal components (Eigenvectors).

One of the main goals of Hand Gesture Recognition is to identify hand gestures and classify them as accurately as possible. For systems to be successfully implemented, it is critical that their performance is known. To date the performance of most algorithms has only been reported on identification tasks, which imply that characterization on identification tasks holds for verification. For Hand gesture recognition systems to successfully meet the demands of verification applications it is necessary to develop testing and scoring procedures that specifically address these applications this process would probably start with image processing techniques such as noise removal, followed by (low-level) feature extraction to locate lines, regions and possibly areas with certain textures.

II. PRINCIPAL COMPONENT ANALYSIS (PCA) METHOD

In this section, we will study the hand gesture recognition through Principal Components Analysis, but we will need some mathematical background to understand the method. This method is called: PCA or Eigenfaces [1-3]. It is a useful statistical technique that has found application in different fields (such as face recognition and imagecompression). This is also a common technique for finding patterns in data of high dimension too. Before realizing a description of this method, we will first introduce mathematical concepts that will be used in PCA.

A. Mathematical Backgrounds:

1) Standard Deviation:

In statistics, we generally use samples of population to realize the measurements. For the notation, we will use the symbol X to refer to the entire sample and we will use the symbol Xi to indicate a specific data of the sample.

$$\overline{X} = \frac{\sum_{i=0}^{n} X_i}{n}$$

2) Standard deviation s,

$$\sigma = \sqrt{\frac{\sum_{i=0}^{n}(X_i - \overline{X})^2}{n}}$$

3) Variance

Variance is another measure of the spread out of data in a set. In fact it is quite the same as the standard deviation.

$$\sigma^2 = \frac{\sum_{i=0}^{n}(X_i - \overline{X})^2}{n}$$

4) Covariance

Covariance can be expressed as:

$$\mathrm{cov}(X) = \frac{\sum_{i=0}^{n}(X_i - \overline{X})(X_i - \overline{X})^T}{n}$$

5) Eigenvectors

The eigenvector of a linear operator are non-vectors which, when operated on by the operator, result in a scalar

multiple of themselves. The scalar is then called the Eigenvalue associated with the eigenvectors.

6) Eigenvalue

Each eigenvector is associated to an Eigenvalue. The Eigenvalue could give us some information about the importance of the eigenvector. The Eigenvalues are really important in the PCA method, because they will permit to realize some threshold to filter the non-significant eigenvectors, so that we can keep just the principal ones.

Images are generally represented as 2D matrices in computers. But to apply PCA, it is convenient to represent images as vector(1D).The conversion of 2D to 1D conversion is discusses later .Once image is converted into vector, each pixel in the image is treated as a dimension in a multidimensional space.

Each image put through PCA is decomposed linearly in terms of the features. Thus, instead storing all reference images in the memory, only weights of a feature need to be stored, which in turn requires less memory and reduces processing time. If the weight of an input image matches or is nearby to ones stored in the memory, the pattern (gesture) is recognized. The set of reference images (i.e. the ones used to form the feature vectors) is also called training set(data base).

B. Main Steps of the method:

First of all, we had to create the data set. The aim is to choose a good number of pictures and a good resolution of these in order to have the best recognition with the smallest database. To obtain the feature set, the common features present in all the images are removed. While applying PCA we are interested only in the unique features of each image, so we remove a part of the redundant data from the input patterns. This is done by subtracting the "average pattern", defined
for the image (i.e. images are mean centered by subtracting the mean image from each image vector).

$$\bar{I} = \frac{1}{m}\sum_{k=1}^{m} I_k$$

$$I_{ck} = I_k - \bar{I}$$

whereI_k= converted input image matrix into column.

The step three is to calculate the covariance matrix of the database. We could not calculate the covariance matrix of the first matrix, because it was too huge. So we had to find a way to find out the principal eigenvectors without calculating the big covariance matrix.

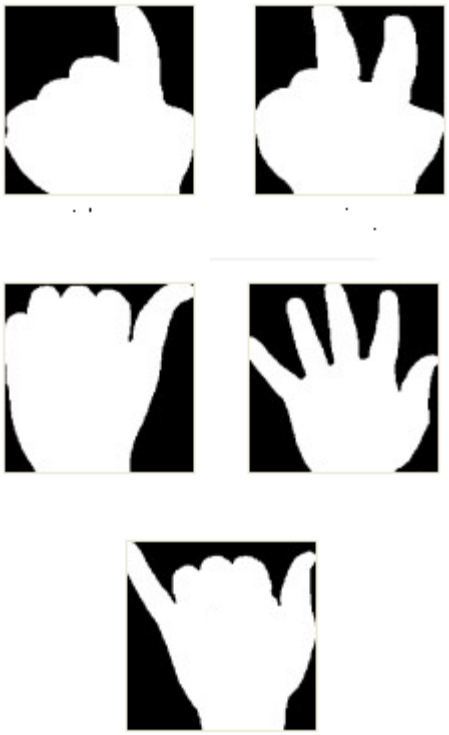

Fig (1): Example of Database

The method consists in choosing a new covariance matrix.

$$C_x = \frac{1}{n}X^T X$$

Instead of $C_x = \frac{1}{n}XX^T$

Then, the eigenvectors and the Eigenvalues of C are the principal components of our data set. The eigenvectors of the covariance matrix found from the product of the matrix X and eigenvectors of matrix L. The dimension of L is $m \times m$ where the values of m generally vary from twenty to a few hundreds, proportionate to the number of training image in the database.This reduces memory utilization.

Now, finally we need to project the image in the database on these eigenvectors and store the projections (dot product magnitude) on them as a vector of weights. But we do not take all eigenvectors for projection the training image since only the ones with large eigenvalues (and hence large variances) form feature vectors which are significant to the data set. Let these weights be stored in a vector named $\Delta_i$where *i*takes values over the range of patterns.

$$\Delta_I = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_R \end{bmatrix}_{R\times n^2} \begin{bmatrix} I_{11} \\ I_{12} \\ \vdots \\ I_{nn} \end{bmatrix}_{n^2\times 1} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_R \end{bmatrix}_{R\times 1}$$

whereP_i are the first R eigenvectors (in decreasing order of eigenvalues) of the covariance matrix as column vectors and I is the gesture to be recognized.

C. Gesture Detection and Recognition:

For gesture detection and recognition we first calculate the weights of the input pattern using the last equation. It is denoted by Δ. Then the Euclidean distance between the weight vectors of the input image and the images in the database, taken one at a time are calculated. If this distance falls below a particular threshold value, set heuristically (based on experimentation, evaluation, or trial-and-error methods) ,an input pattern similar to the class patterns being analyzed is said to have been detected. Formula for Euclidean distance in a multidimensional space is given below;

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}$$
$$= \sqrt{\sum_{i=0}^{n}(p_i - q_i)^2}$$

where p and q are the N dimensional vectors (weight vectors Δ and Δ_I in our case). For recognition the Euclidean distance is calculated and the minimum distance is found. The input image is then said to be represent the training image which gives this minimum distance.

$$\forall\, i \epsilon I \Longrightarrow d(\Delta, \Delta_i) \leq \Theta_{det}$$ Gesture Detection

$$\forall\, i \epsilon I \Longrightarrow min\|d(\Delta, \Delta_i)\| \leq \Theta_{rec}$$ Gesture Recognition

where Θ_{det} and Θ_{rec} are threshold set by user. The whole algorithm is implemented on MATLAB, once the proper gesture is recognized; a particular data is transmitted serially on the robot. Based on the microcontroller code for a particular gesture the robot will move left, right, forward, backward or stopped.

III. RESULT AND CONCLUSION

During this project, we have proposed a promising framework for human gesture recognition for real time. We have validated this approach on the Eigenvalues and Eigen vectors. We believe that the proposed approach is sufficiently robust and flexible to deal with this kind of gestures. Indeed, we have theoretically proved that our learning-classification frameworkcan be adapted to on-line recognition. Moreover, we have proposed several solutionsto improve the efficiency and the effectiveness of our method. For instance, we can learnnormal gestures and consider as abnormal the non-recognized ones. The performance of thealgorithm can be then improved incrementally by increasing database images.
Processing steps consisting of **gesture generation**, **gesture learning** and **gesture classification**. This approach has introduced a novel gesture representation which combineslocal and global motion descriptor advantages.

IV. SUMMARY AND FUTURE SCOPE

Hand Gesture Recognition system is very useful for the physically impaired persons. The system can be trained to help these people to communicate with each other. In this system we have only considered the static gesture, but in real time we need to extract the gesture form the video or moving scene. Therefore the system needs to be upgraded to support dynamic gesture. This system can be further upgraded to give order and control robots. It can also be very helpful for the physically impaired persons. All the above methods can be further enhanced for binary and color images. Some more applications are that this Hand Gesture Recognition system can be in case of games. Instead of using the mouse or keyboard, we can use some pre-defined hand gesture to play any game. Also, this system can be used to operate any electronic devices by just keeping a sensor which recognizes the hand gestures. Another application is that this can be used for security and authorization by keeping anyparticular hand gesture as the password.

REFERENCES

[1] A.Pentland et al., "Eigenfaces for recognition", Journal of Cognitive Neuroscience vol 3,no.1, MIT, 1991.
[2] Lindsay I Smith, et at., "The FERET verification testing protocol for face recognition algorithms"IEEE-FGR, pp 48-55, 1998.
[3] T. Randen et al.,"Filtering for Texture Classification: AComparative Study", IEEE Trans. Pattern Analysis and Machine, Intelligence, vol. 21, pp. 291-310, 1999.
[4]http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial.html
[5] http://en.wikipedia.org/wiki/Eigenface