Hand Gesture Recognition: A Comparative Study

Prateem Chakraborty, Prashant Sarawgi, Ankit Mehrotra, Gaurav Agarwal, Ratika Pradhan

Abstract - **This paper presents four very simple but efficient methods to implement hand gesture recognition namely Subtraction, Gradient, Principal Components Analysis and Rotation Invariant. We first created an Image Database consisting of four 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 grayscale 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.**

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 digital images of hand gestures. In this paper, we identified four different types of hand-gestures- one, two palm and fist. The aim is to study the different methods that allow one to implement a hand gesture recognition system. Moreover, the recognition 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 developed using MATLAB 7.1 on Windows XP Operating System. The images after being captured through the web cam were saved in the database using .jpg format. The images are labeled using integer numbers starting from 1. The database was created using ten different images for all the four hand gestures.

The methods that were studied and implemented are the following:

- Subtraction Method**:** This method involves a simple subtraction between two images, pixel per pixel to compare them.
- Gradient Method: This method involves detecting edges in an image and counting the bright pixels that comprise them for each row or column and then making a comparison.
- 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).
- Rotation Invariant: This method involves two steps basically. First, achieving gray scale and secondly, achieving rotation invariance of an image.

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.

Manuscript received December 10, 2007. This work was supported by Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology, Sikkim, India.

Prateem Chakraborty is a Final yr. B. Tech. student of Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology, Majitar, Rangpo, Sikkim, India. (e-mail: prateem_chakraborty@yahoo.co.in).

Prashant Sarawgi is a Final yr. B. Tech. student of Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology, Majitar, Rangpo, Sikkim, India- 737132 (e-mail: prashant_sarawgi@yahoo.com).

Ankit Mehrotra is a final yr. B. Tech. student of Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology, Majitar, Rangpo, Sikkim, India- 737132 (e-mail: ankit_mehrotra2000@yahoo.co.in).

Gaurav Agarwal is a final yr. B. Tech. student of Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology, Majitar, Rangpo, Sikkim, India-737132. (e-mail: garv_ag@yahoo.com).

Ratika Pradhan is a Reader in Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology, Majitar, Rangpo, Sikkim, India-737132. (e-mail: ratika_pradhan@yahoo.co.in).

II. RELATED WORKS

Many methods for hand gesture recognition using visual analysis have been proposed for hand gesture recognition. Sebastiean Marcel, Oliver Bernier, Jean Emmanuel Viallet and Danieal Collobert have proposed the same using Input-output Hidden Markov Models [1]. Xia Liu and Kikuo Fujimura have proposed the hand gesture recognition using depth data [2]. For hand detection, many approached uses color or motion information [3, 4]. Attila Licsar and Tamas Sziranyi have developed a hand gesture recognition system based on the shape analysis of the static gesture [5]. Another method is proposed by E. Stergiopoulou and N. Papamarkos [6] which says that detection of the hand region can be achieved through color segmentation. Byung-Woo Min, Ho-Sub Yoon, Jung Soh, Yun-Mo Yangc and Toskiaki Ejima have suggested the method of Hand Gesture Recognition using Hidden Markov models [7]. Another very important method is suggested by Meide Zhao, Francis K.H. Quek and Xindong Wu [8]. They have used AQ Family Algorithms and R-MINI Algorithms for the detection of Hand Gestures. There is another efficient technique which uses Fast Multi-Scale Analysis for the recognition of hand gestures as suggested by Yikai Fang, Jian Cheng, Kongqiao Wang and Hanqing Lu [9], but this method is computationally expensive. Chris Joslin et. al. have suggested the method for enabling dynamic gesture recognition for hand gestures [10]. Rotation Invariant method is widely used for texture classification and recognition. Timi Ojala et. al. have suggested the method for texture classification using Local Binary Patterns [11].

III. SUBTRACTION METHOD

This is a very simple method to implement but not a very efficient one since the result generated may be highly inaccurate. This method involves first converting all the images, including the test image, into black and white. This is done by selecting a threshold value for the pixels i.e., if T is the selected threshold value and p is the pixel intensity value then replace p=0 if p<T and p=255 if p>=T. We then perform direct subtraction of each of the pixels in the test image with the corresponding pixels in each of the images in the database and calculate the Euclidean distance between them. The smallest value of 'd' will imply the closest match. Here, 'd' can be defined as

$$d = \sqrt{\sum_{x=0}^{width}\sum_{y=0}^{height}\left(f1(x,y) - f2(x,y)\right)^2}$$

where f1 and f2 are the two images being compared.

IV. GRADIENT METHOD

The steps involved in this process are as follows: First of all, we had to implement the gradient magnitude calculation. The aim is to define where in the picture the biggest gradient magnitudes are. Then, it will be easy to apply a threshold in the gradients in order to keep the really interesting one and to cut all the background noise. To realize this part, the theory is to calculate the magnitude with the formula:

$$magnitude = \sqrt{dx^2 + dy^2}$$

Therefore, we have to calculate the derivative of the image in x and y direction to have the magnitude. We have used Sobel filter to approximate the gradients. The Sobel operator used for gradient-x is as shown below:

-1	-2	-1
0	0	0
1	2	1

The Sobel operator used for gradient-y is as shown below:-

-1	0	-1
-2	0	-2
-1	0	-1

Then, we had to realize a Gaussian filter to blur the image and have a homogeneous picture. It will permit to obtain better results in the gradient magnitude. The goal of this filter is to erase the background defects. It is really important to have a uniform background to avoid noise. We created a gradient magnitude threshold which had to erase the lower levels gradients in order to keep the really interesting ones. This will cut all the noise and regularize the background. This part will be complementary with the Gaussian filter. The Gaussian filter will blur the big defects and the threshold will cut the lowest magnitudes. Then the noise will be quite well cut. Then, the next step was to calculate the Euclidian distance between the vectors of the different images analyzed. This part is made to compare the different pictures, by comparing the different histograms. This is the final step. With this, we are able to recognize the different gestures. To conclude, we can say that this method does not require special mathematical back-grounds.

V. 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 [12-14]. It is a useful statistical technique that has found application in different fields (such as face recognition and image compression). 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 X_i to indicate a specific data of the sample.

$$\overline{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

2) Standard deviation s,

$$s = \sqrt{\frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}{(n-1)}}$$

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.

$$s^2 = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}{(n-1)}$$

4) Covariance

Covariance can be expressed as:

$$var(X) = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(X_i - \overline{X}\right)}{(n-1)}$$

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.

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. Then, the next step is to subtract the mean from each of the data dimensions. The mean subtracted is simply the average across each dimension. 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. The method consists in choosing a new covariance matrix. Our covariance matrix for A was called C and C is defined by:

$$C = A * A'$$

Then, the eigenvectors and the Eigenvalues of C are the principal components of our data set. But as explained before, we could not calculate C. The idea is to say that when we have 12 points in a huge space, the meaningful eigenvectors will be less than the dimension, and the number of the meaningful ones will be the number of points minus 1. So in our case, we can say that we will have 11 meaningful eigenvectors. The remaining eigenvectors will have an Eigenvalue around zero. Then, we calculated the eigenvectors and the Eigenvalues of the covariance matrix. This gave us the principal orientation of the data. With the help of MATLAB we did it easily.

After that, we have chosen the good components and form the feature vector. This is the principal step. We then chose the principal (most important) eigenvectors with which we expressed our data with the lowest information loss. We also had chosen a precise number of eigenvectors to have the less calculation time, but the best recognition. Here, the theory says that we will normally have 11 meaningful eigenvectors. The final step is to make a new data set (that we will call 'Eigenset'). Then, it made possible to realize the last script which could compare the different pictures and class them by resemblance order. To compare the different pictures, we had to express each image of the data set with these principal eigenvectors. The last thing to do is to compare (by calculating the Euclidian distance between the coefficients that are before each eigenvector).

VI. ROTATION INVARIANT METHOD

This method focuses on grayscale and rotation invariant texture classification. In this paper, we propose a theoretically and computationally simple approach which is robust in terms of grayscale variations and which is shown to discriminate a large range of rotated textures efficiently. We present a gray-scale and rotation invariant texture operator based on local binary patterns [14]. Starting from the joint distribution of gray values of a circularly symmetric neighbor set of pixels in a local neighborhood, we derive an operator that is, by definition, invariant against any monotonic transformation of the grayscale. Rotation invariance is achieved by recognizing that this gray-scale invariant operator incorporates a fixed set of rotation invariant patterns. The main contribution of this work lies in recognizing that certain local binary texture patterns termed uniform are fundamental properties of local image texture and in developing a generalized gray-scale and rotation invariant operator for detecting these uniform patterns. The term uniform refers to the uniform appearance of the local binary pattern. The most frequent uniform binary patterns correspond to primitive micro-features, such as edges, corners, and spots; hence, they can be regarded as feature detectors that are triggered by the best matching pattern. The proposed texture operator allows for detecting uniform local binary patterns at circular neighborhoods of any quantization of the angular space and at any spatial resolution. We derive the operator for a general case based on a circularly symmetric neighbor set of P members on a circle of radius R, denoting the operator as $LBP^{riu2}_{P;R}$. Parameter P controls the quantization of the angular space, whereas R determines the spatial resolution of the operator. In addition to evaluating the performance of individual operators of a particular (P,R), we also propose a straightforward approach for multi-resolution analysis, which combines the responses of multiple operators realized with different (P,R). The discrete occurrence histogram of the uniform patterns (i.e., the responses of the $LBP^{riu2}_{P,R}$ operator) computed over an image or a region of image is shown to be a very powerful texture feature. By computing the occurrence histogram, we effectively combine structural and statistical approaches: The local binary pattern detects microstructures (e.g., edges, lines, spots, flat areas) whose underlying distribution is estimated by the histogram. We regard image texture as a two-dimensional phenomenon characterized by two orthogonal properties: spatial structure (pattern) and contrast (the amount of local image texture). In terms of gray-scale and rotation invariant texture description, these two are an interesting pair: Where spatial pattern is affected by rotation, contrast is not, and vice versa, where contrast is affected by the gray scale, spatial pattern is not. Consequently, as long as we want to restrict ourselves to pure gray-scale invariant texture analysis, contrast is of no interest as it depends on the gray-scale. $LBP^{riu2}_{P,R}$ operator is an excellent measure of the spatial structure of local image texture, but it, by definition, discards the other important property of local image texture, i.e., contrast, since it depends on the gray scale. If only rotation invariant texture analysis is desired, i.e., gray-scale invariance is not required; the performance of $LBP^{riu2}_{P,R}$ can be further enhanced by combining it with a rotation invariant variance measure VARP that characterizes the contrast of local image texture. We present the joint distribution of these two complementary operators, $LBP^{riu2}_{P,R} = VAR_{P,R}$ as a powerful tool for rotation invariant texture classification.

A. Achieving Gray-Scale Invariance

As the first step toward gray-scale invariance, we subtract without losing information, the gray value of the center pixel from the gray values of the circularly symmetric neighborhood g_p(p=0,…., P-1). Next, we assume that differences g_p-g_c is independent of g_c, which allows us to factorize. The factorized distribution is only an approximation of the joint distribution. We neglect small loss in information as it allows us to achieve invariance with respect to shifts in gray scale. The distribution t(g_c) describes the over-luminance of the image, which is unrelated to local image texture and, consequently, does not provide useful information for texture analysis. Hence, much of the information in the original joint gray level distribution about the textural characteristics is conveyed by the joint difference distribution .This is a highly discriminative texture operator. It records the occurrences of various patterns in the neighborhood of each pixel in a P-dimensional histogram. For constant regions, the differences are zero in all directions. On a slowly sloped edge, the operator records the highest difference in the gradient direction and zero values along the edge and, for a spot, the differences are high in all directions. Signed differences g_p-g_c is not affected by changes in mean luminance; hence, the joint difference distribution is invariant against gray-scale shifts. We achieve invariance with respect to the scaling of the gray scale by considering just the signs of the differences instead of their exact values:

$$T = t\left(s(g_0 - g_c), s(g_0 - g_c), \ldots\ldots\ldots, s(g_0 - g_c)\right)$$

where s(x) =1, if x>=0 and s(x) =0, if x<0.

By assigning a binomial factor 2^p for each sign s (g_p-g_c), we transform the above expression into a unique $LBP_{P,R}$ number that characterizes the spatial structure of the local image texture:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p$$

The name "Local Binary Pattern" reflects the functionality of the operator, i.e., a local neighborhood is threshold at the gray value of the center pixel into binary pattern. $LBP_{P,R}$ operator is by definition invariant against any monotonic transformation of the gray scale i.e., as long as the order of the gray values in the image stays the same, the output of the $LBP_{P,R}$ operator remains constant. If we set (P =8; R =1), we obtain $LBP_{8,1}$. The two differences between LBP $_{8,1}$ and LBP are: 1) The pixels in the neighbor set are indexed so that they form a circular chain and 2) the gray values of the diagonal pixels are determined by interpolation. Both modifications are necessary to obtain the circularly symmetric neighbor set, which allows for deriving a rotation invariant version of $LBP_{P,R.}$

B. Achieving Rotation Invariance

The $LBP_{P,R}$ operator produces 2^p different output values, corresponding to the 2^p different binary patterns that can be formed by the P pixels in the neighbor set. When the image is rotated, the gray values g_p will correspondingly move along the perimeter of the circle around g_0. Since g_0 is always assigned to be the gray value of element (0, R) to the right of g_c rotating a particular binary pattern naturally results in a different $LBP_{P,R}$ value. This does not apply to patterns comprising of only 0s (or 1s) which remain constant at all rotation angles. To remove the effect of rotation, i.e., to assign a unique identifier to each rotation invariant local binary pattern we define:

$$LBP^{ri}_{P,R} = min\{ROR(LBP_{P,R}, i) \mid i = 0,1,\ldots.,P - 1\}$$

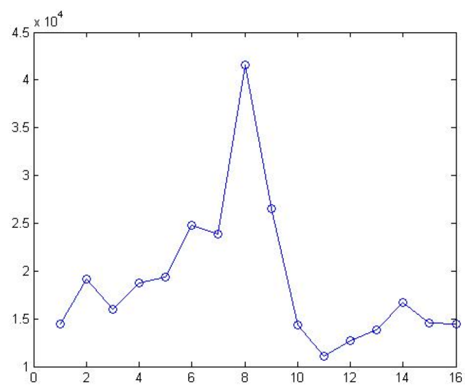
where ROR (x, i) performs a circular bit-wise right shift on the P-bit number x i times. In terms of image pixels, simply corresponds to rotating the neighbor set clockwise so many times that a maximal number of the most significant bits, starting from g_{p-1}, is 0. $LBP^{ri}_{P,R}$ quantifies the occurrence statistics of individual rotation invariant patterns corresponding to certain micro-features in the image; hence, the patterns can be considered as feature detectors.

VII. RESULT AND DISCUSSIONS

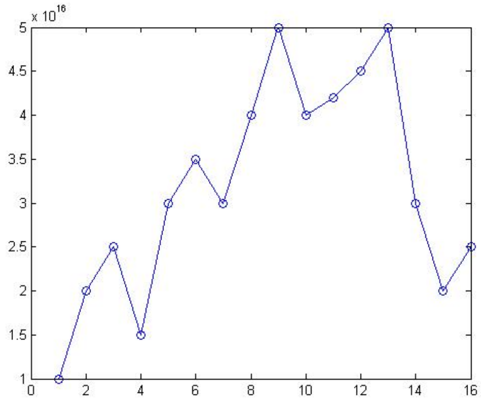
We compared the various algorithms based on the results obtained by testing the MATLAB implementations of the methods with a test database for 4 gestures containing 4 images for each gesture. Three test images for each type of gesture were considered for the comparison. Each test image was processed using the entire four methods one after the other. Then the method that works best for a particular type of gesture was inferred based on the results obtained. We tested the following four gestures namely one, two, palm and fist. The gestures are as shown below

Figure1: Gesture types in clockwise direction- one, two, palm and fist.

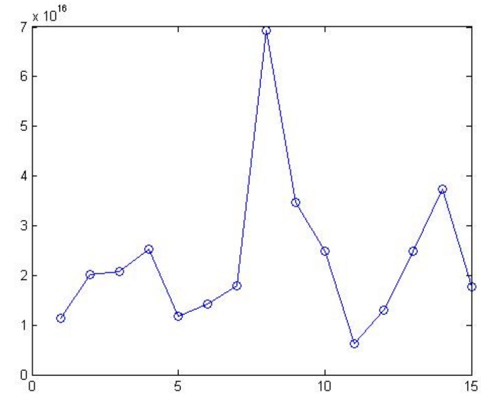
The comparison graphs for the four methods are shown in Figure 1 given below. Then, based on the comparison graphs, we compared the performance of various input images with the database images.



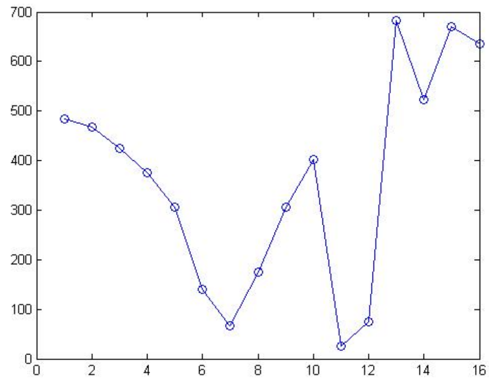
(a)



(b)



(c)



(d)

Figure 2: Euclidean distance computed between the test image and the images in the database for a) subtraction method, b) Gradient method and c) PCA method and the no of pixels, p computed between the test image and the images in the database for d) rotation invariant method.

We also calculated which method gave how many matches for a particular type of gesture. Based on those matches, we made a table which summarizes the no. of database images and the no. of matches for all methods. Those are shown in the table1 given below.

Test image gesture type	Subtraction		Gradient		PCA		Rotation Invariant	
	N	C	N	C	N	C	N	C
One	3	2	3	1	3	3	3	0
Two	3	1	3	2	3	1	3	2
Palm	3	2	3	2	3	1	3	2
Fist	3	1	3	1	3	1	3	2

Table 1: Comparison between the various methods showing the number of matches retrieved by Hand Gesture Recognition system for different gesture type where N denotes number of test images taken and C denotes number of correct matches.

From the inference table2, we come to the conclusion that the results given by different algorithms for different gestures varied greatly. No particular method gave consistent results for all the test gestures. Out of the method described above the following method can give satisfactory result for particular gesture type.

Gesture Type	Best Method
One	Gradient and PCA
Two	Rotation Invariant
Palm	Rotation Invariant
Fist	Subtraction

Table 2: Inference Table.

VIII.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 any particular hand gesture as the password.

IX. ACKNOWLEDGMENTS

We would like to express our sincere thanks to all the Staff of Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology, Sikkim for their support and guidance throughout.

REFERENCES

[1] Sebastian Marcel, Oliver Bernier, Jean Emmanuel Viallet and Daniel Collobert, " Hand Gesture Recognition using Input – Output Hidden Markov Models", Proc. of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 456 - 461 2000.

[2] Xia Liu and Kikuo Fujimura, " Hand Gesture Recognition using Depth Data", Proc. of the Sixth IEEE International conference on automatic Face and Gesture Recognition, pp. 529-534, 2004.

[3] L. Bretzner, I. Laptev, and T. Lindeberg, "Hand Gesture using multi-scale color features, hierarchical models and particle filtering", Proc. of the Fifth International conference on Automatic Face and Gesture Recognition, pp. 423- 428, 2003.

[4] V. Pavlovic, et al. visual interpretation of hand gesture for human-computer interaction: a review, IEEE Trans. On Pattern anal. Mach. Intel. 19(7), pp 677-695, 1997.

[5] Attila Licsar and Tamas Sziranyi, "Supervised training based hand gesture recognition system", Proc. of the 16th International Conference on Pattern Recognition, Vol. 3, pp 30999 – 31003, 2002.

[6] E.Stergiopoulou and N.Papamarkos: "A New Technique on Hand Gesture Recognition", Proc of the IEEE International Conference on Image Processing, 2657-2660, 2006.

[7] Byung-Woo Min, Ho-Sub Yoon, Jung Soh, Yun-Mo Yangc and Toshiaki Ejima : "Hand Gesture Recognition Using Hidden Markov Models", Proc. of the IEEE International conference on Systems, Man and Cybernetics, vol 5, pp. 4232 -4235, 1997.

[8] Meide Zhao, Francis K.H. Quek, Member, IEEE, and Xindong Wu, Senior Member, IEEE : "RIEVL: Recursive Induction Learning in Hand Gesture Recognition", IEEE Transactions on Pattern Analysis and machine intelligence, vol. 20, no. 11, November 1998.

[9] Yikai Fang, Jian Cheng, Kongqiao Wang and Hanqing Lu, "Hand Gesture Recognition Using Fast Multi-scale Analysis", Proc. of the Fourth International Conference on Image and Graphics, pp 694-698, 2007.

[10] Chris Joslin, Ayman El-Sawah, Qing chen, Nicolas Georganas, " Dynamic Gesture Recognition", Proc. of the Instrumental and Measurement Technology Conference, pp 1706-1710, 2005.

[11] Timi Ojala, Matti Pietikainen and Topi Maenpaa, "Multi-resolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, pp. 971-987, 2002.

[12] A. Pentland M. Turk. Paper on Eigenfaces for recognition.

[13] Lindsay I Smith, A tutorial on Principal Components AnalysisSyed A. Rizvi, P. Jonathon Phillips and Hyeonjoon Moon, The FERET verification testing protocol for face recognition algorithms

[14] T. Randen and J.H. Husoy, "Filtering for Texture Classification: A Comparative Study", IEEE Trans. Pattern Analysis and Machine, Intelligence, vol. 21, pp. 291-310, 1999.

[15] R.L. Kashyap and A. Khotanzad, "A Model-Based Method for Rotation Invariant Texture Classification", IEEE Trans. Pattern. Analysis and Machine Intelligence, vol. 8, pp. 472-481, 1986.

[16] J.-L. Chen and A. Kundu, "Rotation and Gray Scale Transform Invariant Texture Identification Using Wavelet Decomposition and Hidden Markov Model", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 16, pp. 208-214, 1994.

[17] M. Porat and Y. Zeevi, "Localized Texture Processing in Vision: Analysis and Synthesis in the Gaborian Space", IEEE Trans. Biomedical Eng., vol. 36, pp. 115-129, 1989.

[18] H. Arof and F. Deravi, "Circular Neighbourhood and 1-D DFT Features for Texture Classification and Segmentation", IEEE Proc.-Vision, Image, and Signal Processing, vol. 145, pp. 167-172, 1998.

[19] L. Wang and G. Healey, "Using Zernike Moments for the Illumination and Geometry Invariant Classification of Multi-Spectral Texture", IEEE Trans. Image Processing, vol. 7, pp. 196-203, 1998.

[20] T.N. Tan, "Scale and Rotation Invariant Texture Classification", IEE Colloquium Texture Classification: Theory and Applications 1994.

[21] Gonzalez and Woods, Digital Image Processing

[22] Chapman, MATLAB Programming For Engineers.