Vision based Robot Navigation for Disaster Scenarios

Sudheesh.P
II M.Tech (Embedded Systems)
Amrita Vishwa Vidyapeetham
Coimbatore, India

Gireesh Kumar.T
Cyber Security
Amrita Vishwa Vidyapeetham
Coimbatore, India

ABSTRACT

This paper aims at providing real time vision based robot-navigation for disaster management scenarios. The task is to navigate a robot in unstructured environment by using gestures. Navigation is an important task that is to be performed while traversing a particular path in disaster management scenario. There are various methodologies like autonomous mapping and SLAM techniques in which the robot is trained itself to create the path by making a map, but training the robot and creating a map itself requires a lot of time and is a tedious process. Meanwhile in this approach a real time video streaming is done by the robot itself that is being transmitted to user who in turn controls the robot using gestures. Apart from streaming the video we also find the closest obstacle distance using IR sensors. For the purpose of performing a particular task for a detected gesture, the robot needs to have intelligence. This intelligence is the algorithm that is loaded into the robot to make it perform the task assigned to it. Here we make use of principle component analysis along with image moments for identifying the gestures and thereby controlling the robot. Real-time implementation is done on iRobot platform.

General Terms
Disaster management, hand gestures, moments, PCA, Navigation.

Keywords
PCA, Image moments, video streaming, Disaster management

1. INTRODUCTION

Cognitive Robotics is currently a highly active research area with huge potential. Service robotics is one such field in which robots directly interact with people. So finding a natural and easy to use user interfaces are of large importance. Many works in the past were focused on issues like navigation and mapping, several of them are provided with interfaces like keyboard, joysticks etc. Only a few robotic systems are incorporated with flexible user interfaces that enable the user to control the robot in an easier manner. It is therefore essential that these robots be equipped with such flexible interfaces that provide a better interaction of robots with the user. Incorporating navigation with service robots is quite a challenging work. This paper considers an approach in which navigation of the robot is incorporated with streaming of video from the robot and finding the closest obstacle distance using the IR sensors. Control over the robot is made using gestures.

Gestures are an easy way to control the robot while it's traversing a path. The research in this field also states that in future the communication interface between robot and user will be even more flexible. The main purpose of gesture recognition is to identify a particular human gesture and convey the command related to the detected gesture for controlling the movement of the robot. Hand gesture recognition can be used to enhance interaction between user and the robot as compared to the traditional interfaces. Robots can be controlled naturally and flexibly with such a mode of communication. Robotic movement can be controlled remotely by hand gestures using wireless module like Bluetooth and zigbee. This paper involves usage of zigbee to provide wireless control over the robot for every detected gesture

The control interface involves a camera that captures the image of the hand at that current state and compares it with the image stored in its database using PCA and image moments, so that a better recognition of the image takes place, for every detected gesture a signal is being sent to the robot that makes it to perform a particular action.

Remaining sections in this paper is organized as follows: In the section 2 involves tracking and detection. Section 3 is to focus on feature extraction and comparison. In section 4 we demonstrate streaming of video from the robot to the user. Hardware implementation of the system is described in section5. In section 6 describes the performance of the system and Conclusions and future enhancement is given in section 7.

2. HAND DETECTION

This section mainly deals about detection of hand in an image so that other background objects are avoided for feature extraction. Detection of hand is done by skin detection [14]. For this an appropriate color space is detected.

2.1 Skin Detection

It is a method of finding skin-color pixels and regions in an image. Here a given pixel gets transformed into a particular color space and the by using a skin classifier, this pixel gets classified as a skin pixel or non-skin pixel. Color space representation is as follows.

$$red = \frac{R}{R+G+B} \qquad green = \frac{G}{R+G+B}$$

$$blue = \frac{B}{R+G+B}$$

2.2 Finding the Centroid and Farthest Distance

Given the segmented hand region, calculation of centroid is very essential to find out the area of hand detected so that a proper tracking and detection can take place. This detected gesture is then segmented to do feature vector extraction

The centroid ($\bar{x}$,$\bar{y}$) is given as follows:

$$\bar{x} = \frac{\sum_{i=0}^{k} x_i}{k} \qquad \bar{y} = \frac{\sum_{i=0}^{k} y_i}{k}$$

Where xi and yi are the ith pixel in the hand region and k is the no. of pixels in the region. After obtaining the centroid, calculate the distance D from the most extreme point in the hand to the Centre. Usually this distance is from the centroid to the tip of the longest finger. Using distance D, draw a rectangular box that covers all the fingers of the hand in the image as shown in fig 1.

Fig 1: Hand detection

3. FEATURE EXTRACTION AND RECOGNITION

As discussed in the previous section hand is tracked and detected, features extraction and comparison is done using PCA [13] and image Moments [8].

3.1 Principle component analysis for feature extraction

PCA is used for feature extraction and comparisons. Training set consists of the features extracted from known hand images that are stored in the database. The algorithm's task is to find out the most similar feature vector of images in the database to the feature vector of a given test image by computing similarity between them. This can be done by using Euclidean distance or mahalanobis distance.

The algorithm is described as follows:

1. Obtain a set I with M face images. Each image is transformed into a vector of size N and placed into the set.

   $I = \{ S_1, S_2, S_3 ... S_M \}$

2. Obtain the mean image Ψ

   $\Psi = \frac{1}{M} \sum_{n=1}^{M} S_M$

3. Find the difference Φ between the input image and the mean image

   $\Phi_i = S_i - \Psi_i$

4. Seek a set of M orthonormal vectors, U_n, which best describes the distribution of the data.

   $$\eta = \frac{1}{M} \sum_{n=1}^{M} (\Phi_n u_k^T)^2$$

   is maximum subjected to

   $$U_l U_k = \delta_{lk} = \begin{cases} 1 & if\ l = k \\ 0 & otherwise \end{cases}$$

   U_k and λ_k are Eigen vectors and Eigen values of covariance matrix C.

5. Obtain covariance matrix

   $C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T$

   $A = \{\Phi_1, \Phi_2, \Phi_3 ... \Phi_M\}$

   $L = \Phi_m^T \Phi_n$

6. Eigen vectors are given by

   $$U_l = \frac{1}{M} \sum_{n=1}^{M} \Phi_k V_{lk}$$

Features in the form of Eigen vectors are then extracted and stored for comparison

3.2 Euclidean Distance for feature comparison

The match between image in the database and the real-time image is done by computing Euclidean distance between their feature vectors. First we compare the input image with mean image and multiply their difference with each eigenvectors of the L matrix. Each value would be represented as a weight and saved on a vector Ω described as follows

$W_k = U_k^T (S - \Psi)$

$\Omega_t = [W_1, W2 ... W_M]$

Euclidean Distance is given by

$\varepsilon_k = \|\Omega - \Omega_k\|$

The input gesture is considered to belong to a class if ε_k lies within a predetermined threshold as shown in fig 2.

Euclidean distance is being computed between the images in the database and the test images. Here we make use of the vectors calculated using PCA for the test image and the images in the database and then compute the similarity between them. Theoretically a Euclidean distance should be zero, but in actual case getting a zero Euclidean distance is difficult. So for the least computed Euclidean distance we identify the particular gesture from the database.

The input gesture is considered to belong to a class if ε_k is below an established threshold. Then the gesture is considered to be a known gesture. If the difference is above the given threshold, but below a second threshold, the image can be determined as an unknown gesture. If the input image is above these two thresholds, the image is determined not to be a gesture.

If a match occurs between test image and image in the database, then a command signal is being given to the robot that in turn helps to control the robot during its navigation.

3.3 Scale Invariance using image moments

This algorithm is also scale invariant, meaning that actual size of hand and its distance from camera do not affect the interpretation and are less affected to changes in illumination.

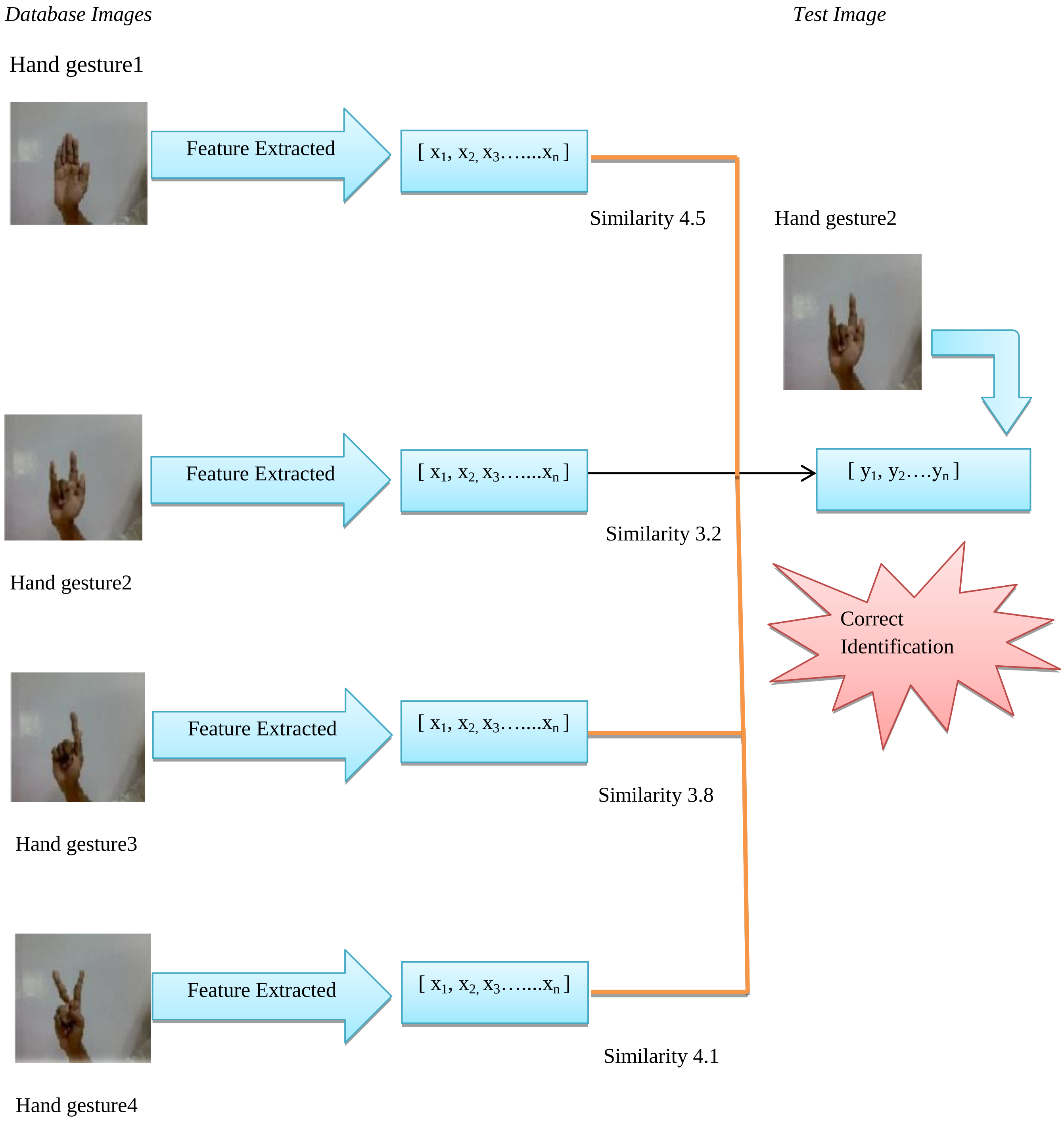

Fig 2: Gesture Recognition Using PCA

This is achieved by incorporating moments [7 8].Image moments are being used in order to achieve this scale invariance. But too much variation in illumination may affect the recognition rates. As when illumination around the image changes, the color space around the image also gets varied.

The scale invariance during gesture recognition is shown in fig.3

Fig 3: Scale invariance in recognition using moments

4. NAVIGATION AND OBSTACLE AVOIDANCE

The navigation is a two-step process, first step involves video streaming of and the other based on the usage of IR sensors that will provide the distance to the closest obstacle thereby allowing us to take proper action in proper time.

4.1 Video streaming

Video streaming involves capturing the video of the environment where the robot moves and transmitting it to the user end, where control is being performed. Here we use the concept of socket programming and create two applications that run on two computers, one placed over robot while the other at the user end. The application at the robot will capture the video through the webcam and transmit it to the PC at user as shown in fig.4

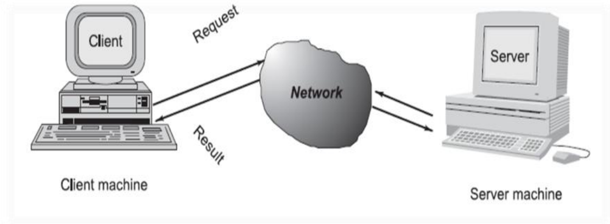

Fig 4: Block diagram for video streaming using sockets

The system over the robot is server machine and the system over user end is client. Client keeps on listening to the server so when server starts transmitting the video it's instantly received by the user.

Sockets are API used for interprocess communication based on TCP/IP [11] and RTP protocol for establishing communication. Once connection is established the video is being divided to frames and these frames are then transmitted from server to client in a continuous manner. These frames then keep on getting overwritten consequently so that when seen in a continuous manner it resembles a video at the user end.

4.2 Distance measurement from obstacle

This is done by using a pair of IR sensors that are used to compute the distance from the obstacle which is given by

Distance in mm = 10.00*((1.00/ ((0.001240875*(float) ADC value) +0.005)) – 0.42)

5. HARDWARE IMPLEMENTATION

The hardware implementation is done on a hybrid version created by combining iCreate Roomba and firebird4.

5.1 iRobot Create

iRobot Create is a robotic platform in which various robotic motions can be programmed without worrying about mechanical assembly or low level codes. The greatest feature about it is that it can be communicated and programmed through OI (open interface).Many sensors like Cliff, Bump etc. are also interfaced to this robot.it also facilitates different modes of operation like passive, full, safe as shown in fig 5.

It also facilitates inbuilt controller programming as it consist of ATMEGA 168 microcontroller, also provided with a rechargeable battery that can hold the charge till 5 hours of continuous operation by the robot. iRobot Create comes with a serial protocol called the Open Interface (OI) consisting of a set of commands that can be used to control the robot and read its sensors.

Fig 5: iRobot Create

5.2 Firebird 4

Firebird 4 is a robotic platform that provides a wide range of programmable features. It consists of a central controller ATMEGA 128 that is interfaced with bump and IR sensors so that direct controller programming will enable to read these sensors as shown in fig 6. It gives the distance of the robot from the closest obstacle by using passive IR sensors.

Fig 6: Firebird 4

5.3 Ebox 4861

Ebox is a simplified & an economical design of an embedded computer for Special Purpose Personal Computing. Similar to the modern day CPU but Ebox can be used for specific application that makes it an embedded device. It comprises of the latest x86 technology design with on-board 512MB/1GB System memory as shown in fig 7.It is compatible with Windows XP, WEPOS, Windows Embedded CE, Linux and Windows Embedded Xp.

It has an inbuilt VIA Esther processor that runs at speed of 1.2GHz. It has Display, USB, Keyboard, PS2 Mouse, LAN and Audio Interfaces. It is also provided with an optional WLAN interface card. In our case we have deployed the application used to do video streaming in Windows Embedded XP.

Fig 7: Ebox 4861

Real time implementation of vision based navigation for disaster scenarios are done by fusing iRobot and firebird 4 robot modules together as shown in fig.8.Webcam along with Ebox is mounted on iRobot, which captures the video and transmits it to the user end. Ebox consist of the application for video streaming on server deployed in Windows Embedded Xp. Firebird 4 along with IR sensors are being mounted on the front end of the iRobot as shown in fig 8.at the user end each gesture is assigned an action to perform. Whenever a match occurs, the robot's behaviour is controlled through an open interface.

Fig 8: Hardware implementation of the navigation system

6. RESULTS AND DISCUSSION

The system was tested using four hand gestures for controlling the robot. The recognition rate for the gestures using PCA and moments were 82.3%. In the case of illumination, a small variation in the lighting condition doesn't affect this recognizer.

Video streaming also was successful with a frame rate of 5 frames per second. When tested with LAN it gave a frame rate around 30-34 fps, but when tested with WLAN it was around 19-23 fps. The IR sensor could detect an obstacle within distance of 40 cm since its range is limited by its specification. The OS image that we created for the deployment of video streaming application along with other required components for Windows Embedded XP only has a size of 350 Mb compared to the usual Windows XP which has a size of 700 Mb.

6.1 Control Interface for Robot

The control over the robot is done by natural means i.e. by the help of hand gesture. The control interface shown in the fig.9 moves the robot for accordingly detected gesture. The interface also shows the distance of the robot to the closest obstacle which is detected and measured by pair of IR sensors placed over the robot. Using this calculated distance we can navigate our robot more efficiently by avoiding the obstacles

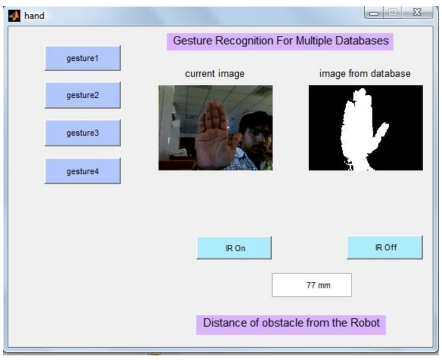

Fig.9. Gesture control interface for robot

6.2 Interface to receive the streamed video from the robot

This interface is used to receive the streamed video of the environment from the robot. Using the video and the distance from the closest obstacle, we navigate the robot in the disaster scenarios.

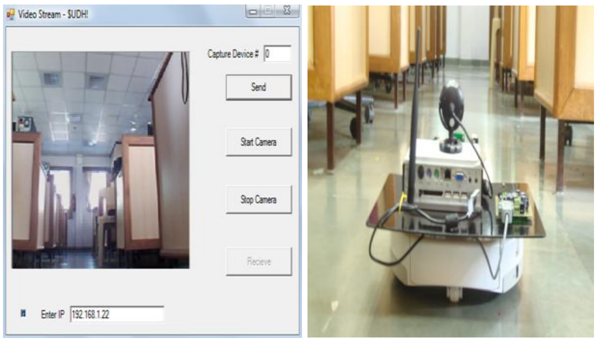

Fig 10: Video streaming from the robot to user

6.3 Performance evaluation

The table 1 below gives us the recognition rates for the gesture recognition for the control interface for the robot.

Table 1. Recognition rate for different gestures

Gestures	Recognition rate with low illumination	Recognition rate with proper illumination
Gesture1	60	80
Gesture2	70	85
Gesture3	50	78
Gesture4	65	82

Fig 11: Different gestures

7. CONCLUSION AND FUTURE WORK

In this paper we have implemented a novel approach for control and navigation for a robot in disaster scenarios. Instead of interfaces like keyboard and joystick, incorporating a natural means of controlling the movement of the robot is being performed along with establishing of video streaming from robot to the user using socket programming. The size of the operating system was also reduced to 350Mb thereby reducing the memory usage. The usage of IR sensors interfaced with firebird 4 increased the navigation efficiency as we could also get to know the distance of the closest obstacle to the robot. In future work for hand gesture tracking and recognition PCA can be replaced with SVM and camshaft to obtain better recognition rates. The IR sensors can be replaced with wide range Ultrasonic sensors to track the obstacles till larger distances.

8. REFERENCES

[1] Dardas. N.H, Petriu.E.M, "Hand gesture detection and recognition using principal component analysis", International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA), 2011 IEEE, pp. 1-6.

[2] Deng-Yuan Huang, Wu-Chih Hu , Sung-Hsiang Chang, "Vision-Based Hand Gesture Recognition Using PCA+Gabor Filters and SVM", Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP '09, pp. 1-4.

[3] Tudor Ioan Cerlinca, Stefan Gheorghe Pentiuc, Marius Cristian Cerlinca, Radu Daniel Vatavu "Hand Posture Recognition For Human Robot Interaction" ,Universitatea "Stefan cel Mare", 2006.

[4] Y. Wei,"Vision-based Human-robot Interaction and Navigation of Intelligent Service Robots", PhD Dissertation, Institute of Automation, Chinese Academic of Sciences, 2004.

[5] Priyal, S.P, Bora P.K, O. "A study on static hand gesture recognition using moments". International Conference on Signal Processing and Communications (SPCOM), pp: 1 – 5 July 2010.

[6] T. Cerlinca, S. Pentiuc, and M. Cerlinca, "Hand posture recognition for human-robot interaction," in Proceedings of the 2007 workshop on Multimodal interfaces in semantic interaction, 2007.

[7] H. K. Kim, J. D. Kim, D. G. Sim, and D. I. Oh, "A modified Zernike moment shape descriptor invariant to translation, rotation and scale for similarity-based image retrieval", in Proceedings of the IEEE International Conference on Multimedia and Expo, Vol. 1, 2000, pp. 307-310.

[8] C. H. Teh and R. T. Chin, "On image analysis by the methods of moments," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, 1988, pp. 496-513.

[9] Pai-Hsiang Hsiao, H.T. Kung and Koan-Sin Tan, "Video over TCP with Receiver-based Delay Control," in Proceedings of ACM NOSSDAV, 2001.

[10] http://www.video.ja.net/intro/#top, Introduction to Video Conferencing System.

[11] Gurmeet Singh. "Secure Video Conferencing for Web Based Security Surveillance System". M.Tech thesis, Department of Computer Science and Engineering Indian Institute of Technology, Kanpur July, 2006.

[12] Harrison.C, Hudson.S.E, "Pseudo-3D Video Conferencing with a Generic Webcam", Tenth IEEE International Symposium on Multimedia, 2008. ISM 2008.pp- 236 – 241.

[13] Monwar, A Real-Time Face Recognition Approach from Video Sequence using Skin Colour model and Eigenface Method. In: Canadian Conference on Electrical and Computer Engineering pp.2181 - 2185 (2006).

[14] Siddharth Swarup Rautaray, Anupam Agrawal, A Real Time Hand Tracking System for Interactive Applications.in International Journal of Computer Applications (0975 – 8887) Volume 18– No.6, March 2011.

[15] Yoav Freund, Robert E. Schapire, "A Short Introduction to Boosting", Journal of Japanese Society for Artificial Intelligence, 14(5):771-780, September, 1999.

[16] Freund Y, Schapire R E. "Experiments with a New Boosting Algorithm", Proceedings of the 13th Conference on Machine Learning 1996.

[17] J.L. Raheja, R. Shyam, U. Kumar, "Hand Gesture Capture and Recognition Technique for Real-Time Video Stream", Proceeding (683) Artificial Intelligence and Soft Computing – 2009.

[18] Zhengmao Zou, Premaratne, P., Monaragala, R., Bandara, N. ; Premaratne, M. , "Dynamic hand gesture recognition system using moment invariants", 5th International Conference on Information and Automation for Sustainability (ICIAFs), 2010.pp.- 108-113.