Performance Improvement in Hand Gesture Recognition Using Adaptive Kalman Filter

[1] **Amol Mahadik,** [2] **Madhuri Joshi**

[1] Computer Science &Engineering, Dr. B.A.M.University., Jawaharlal Nehru Engineering college
Aurangabad, 431006, M.S., India

[2] Computer Science &Engineering, Dr. B.A.M.University., Jawaharlal Nehru Engineering college
Aurangabad, 431006, India

Abstract - This paper suggests method to improve hand gesture recognition rate in limited environment. The system consists of two main stages which are Hand detection and tracking. In Hand detection, hand region is first detected by combining motion and skin color pixels. A region of interest (ROI) is then created in the detected hand region. In tracking stage, skin and motion pixels are scanned around top, left and right corners of the ROI to detect the moving hand in consecutive video frames. These pixels are used to actually measure the ROI position and fed into measurement update of Adaptive Kalman Filter (AKF) operation. The experimental result shows the proposed method has the robust ability to track the moving hand under real life scenarios with average 96.66% result.

Keywords — *Hand detection; Hand tracking; Hand gesture; Adaptive Kalman Filter; Region of Interest.*

1. Introduction

Nowadays framework of interactive, intelligent computing, an efficient human–computer interaction is assuming great importance in our daily lives. Gesture recognition can be termed as an approach in this direction. It is the process by which the *gestures* made by the user are *recognized* by the receiver. Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body with the intent of: conveying meaningful information or interacting with the environment. They constitute one interesting small subspace of possible human motion. A gesture may also be perceived by the environment as a compression technique for the information to be transmitted elsewhere and subsequently reconstructed by the receiver. Gesture recognition has wide-ranging applications [1] such as the following:

- developing aids for the hearing impaired;
- enabling very young children to interact with computers;
- Designing techniques for forensic identification;
- Recognizing sign language;
- Medically monitoring patients' emotional states or stress level;
- Lie detection;
- Navigating and/or manipulating in virtual environments;
- Communicating in video conferencing;
- Distance learning/tele-teaching assistance;
- Monitoring automobile drivers' alertness/drowsiness levels, etc.

Hand detection and tracking is an important component of gesture recognition system, where hands must be localized in every image sequences which directly affect the efficiency of the system [2]. In recent years various approaches to track hand gesture have been proposed. Cheng *et al.* [3], have develop a real-time hand gesture tracking using five complementary processes; motion skin and edge detection, movement justification and background subtraction. In their work, they managed to detect and track hand under complex background assuming that the hand object move in constant speed.

Isard and Blake [4] adopted parameterized B-spline curves to model hand contours, and tracked hands by tracking the deformed curves. However, since hand contours are view dependent and vary dramatically in natural hand motion, the contour-based trackers usually constrain the viewpoint and assume that hands keep several predefined shapes [5]. Zhang [6] also track hand gesture using contour tracking which is based on skin color probability and state estimation model technique. However in their work, they assume the hand shapes is smooth and the difference between two continuous frames is not obvious. Imagawa [7] extract hand regions using skin color pixels compute blobs and track the hand location using Kalman Filter. In their work, they assume that face does not move as much as hand during the tracking. Binh [8] also used Kalman filter to predict the location of the hand in an image frame based on a model of a constant acceleration where, the

Kalman filter operation tracked the hand movement from frame to frame to provide accurate starting point to search for a skin color region. However the proposed method in [6, 7] which only use skin color feature, is hard to deal with the situation where hand is occluded with other skin regions. Moreover, the Kalman filter framework used in their work is difficult to estimate the hand position when it moves in nonlinear fashion. He proposed hand tracking in cluttered background by combining color and motion cues. They extracted skin color pixels from image frames using YCbCr pixels distribution difference operation is used to eliminate the unwanted skin regions and the position of the moving hand is computed based on the centre of the gravity of the remaining binary image.

In their work, they performed difference operation after skin segmentation, which exhibits very less or no information in the remaining binary image to track hand when it is occluded other skin regions. In general, robust detection and tracking of the hand is hard to achieve for several reasons. Firstly, in natural ways, hand motion is non-rigid due to its continuously changing shape and appearance which is hard to model. Secondly, in many real-world applications, hand needs to be tracked in cluttered scenes and under uncontrolled conditions. Finally, there are cases where hand can appear occluded by other skin colored regions such as clothing, wall and face. This paper presents a system that addresses each of these difficulties. This system consists of two main stages which are initialization and tracking. During initialization, gesturer needs to make significant move using his hand, but others part of his body is allowed to move in small scale. Then, hand detection is achieved by combining the skin color and motion cues and hand ROI is created around the detection location. In tracking stage, using a model of constant velocity, we develop a state vector based on hand's position and velocity as an Adaptive Kalman Filter (AKF) estimation process. From the estimated location, we calculate the hand displacement by scanning the motion and skin pixels around right, left and top corners of the ROI and measure the actual hand position. The measurement noise is obtained from the error between the actual and estimated position, which is than fed into measurement update equation of AKF. To deal with nonlinear movement, which the hand velocity is changing, the measurement noise covariance and process noise covariance of AKF are adjusted adaptively by acceleration magnitude.

2. Existing System

There are many existing systems for hand gesture recognition. One of the existing systems for hand gesture recognition is HMM. It is a rich tool used for hand gesture recognition in diverse application domains. In this approach, a discrete HMM and a sequence of vector-quantized (VQ)-labels have been used to recognize six classes of tennis strokes. Before applying the HMM, the image sequence goes through several preprocessing steps such as low-pass filtering to reduce the noise, background subtraction to extract the moving objects, and binarization of the moving objects in order to generate blobs. The blobs roughly represent the poses of the human. The features are the amounts of object (black) pixels. These features are vector quantized, such that the image sequence becomes a sequence of VQ-labels, which are then processed by a discrete HMM.

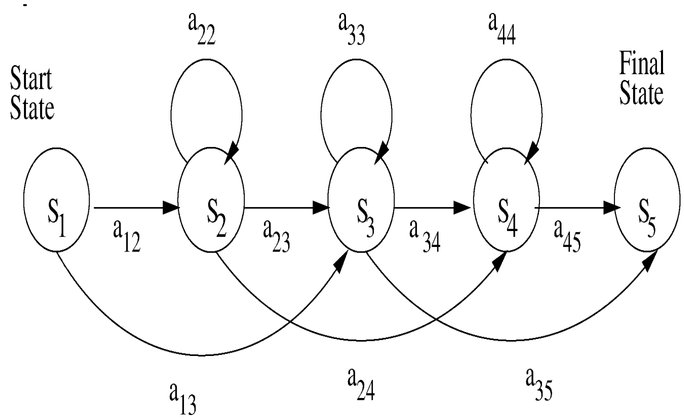

Fig 1. Five-state left-to-right HMM for gesture recognition.

A real-time HMM-based system has been designed for recognizing sentence-level American Sign Language, without explicitly modeling the fingers. Since each gesture in a sign language has an already assigned meaning, strong rules of context and grammar may be applied to make the recognition tractable. Typically, the gestures represent whole words. A five state HMM is used to recognize data strings, and is combined with statistical grammar to incorporate context during training and recognition. The Viterbi algorithm is used both with and without a strong grammar, based on the known forms of the sentences.

However, once the recognizer starts, the subject must conduct only sign languages. This is because the model cannot distinguish undefined hand motions. The subject wears distinctly colored gloves on both hands, and sits in a chair in front of the camera to aid hand tracking. An eight-component feature vector consists of the hand's 2-D coordinates, axis angle of least inertia, and the eccentricity of the bounding ellipse. Considering that all human hands have approximately the same hue and saturation but vary in their brightness, in another approach, the hands are tracked based on skin tone. The leftmost and rightmost hands are assigned to be "left" and "right," respectively. [2]

IJCSN International Journal of Computer Science and Network, Volume 3, Issue 4, August 2014
ISSN (Online) : 2277-5420 www.IJCSN.org
Impact Factor: 0.274

2.1 Disadvantages of HMM

The HMM-based approach, described above, requires extensive training sets for modeling and it is computationally expensive. It was found to be effective for practically just around 50 words and required heavily constrained artificial grammar on the structure of the sentences. Performance of HMM-based systems could be limited by the characteristics of the training dataset. [2]

3. Proposed System

This proposed system is divided into two parts
A] Hand detection and tracking
B] Command to operating system

3.1 Hand Detection & Tracking

To use hand gesture, extraction of hand region information from image frame is first step. Many methods are there to extract hand region using vision based technique like using depth camera in which object nearer to camera can be extracted that is hand region can be extracted. Skin color detection technique is one more method used for recognition of hand gesture. While using this technique, removing face region and other background object having skin color is challenging task. In this paper very simple and effective technique to extract the hand region information is used .We have used camera of resolution 320x240 pixels. For first 3 frames we have created square of 40x40 pixels that is for first three frames square will appear on preview window .

Keep the hand in such a way that hand region will come in square. Crop the square from third frame to find color information of hand .Fig. 2 show first two frames in which square of 40x40 is formed on preview window and Fig. 3 shows cropped square which contain hand region information. Fig. 4 shows block diagram of the gesture based operating system Control using AKF. As in fig. the first step is creating rectangle on preview window. Then you will have to place your palm in rectangle as shown in fig. 3. Pixels in rectangle give RGB information which is explained mathematically in latter part. After evaluating pixel information you will able to move cursor according to your palm. Then by bending palm you can perform mouse operations. If palm is bent then obviously number of pixels will be less than previous frame. If this condition satisfies then order to OS can be given to perform desired operation.

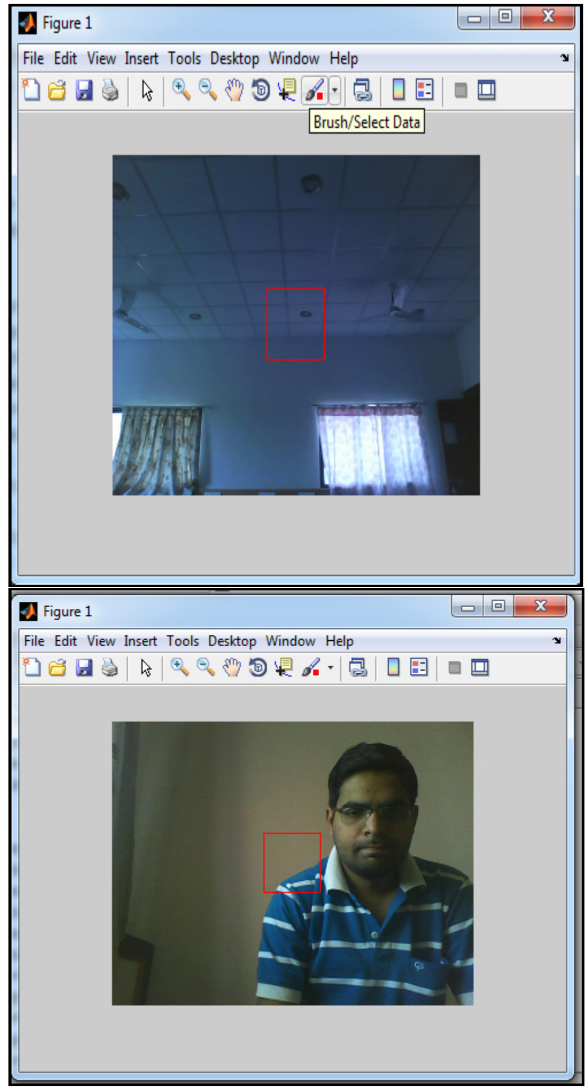

Fig 2: First two frames

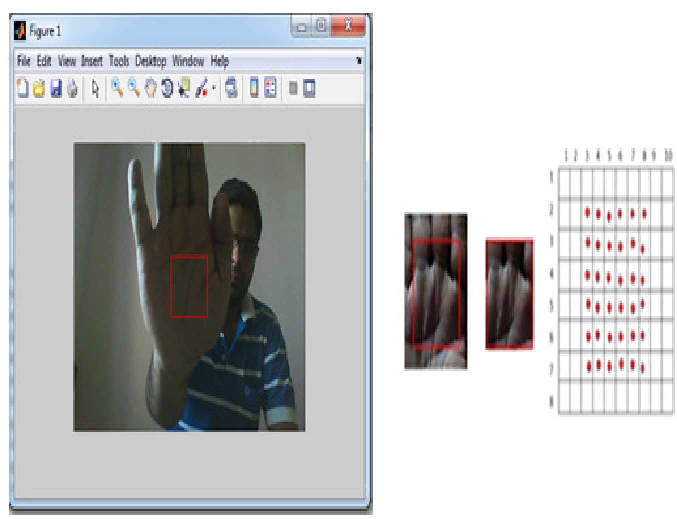

Fig 3: Third frame & feature color extraction

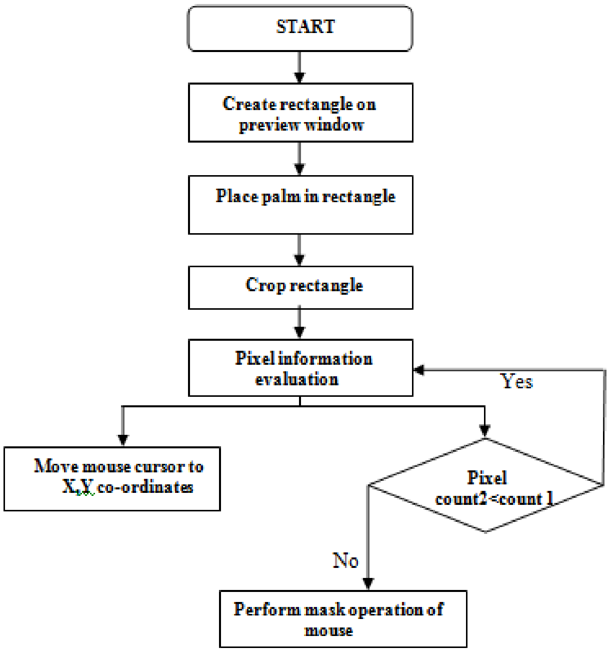

Fig 4. Block diagram of the gesture based operating system Control using AKF

Find the red, green & blue component from cropped square area .Calculate average value of red, green & blue component. Find the precision value of these three components.

Rmax = Ravg + 15 & Rmin = Ravg -15 ……………… (1)
Gmax = Gavg + 15 & Gmin = Gavg -15……………… (2)
Bmax = Bavg + 15 & Bmin = Bavg -15………………. (3)

These R, G, B values are used to filter the data for next frames. For next frames extract the pixels whose value is in between maximum and minimum limit of R, G, B component .These pixels obtained from this filtering will represent hand color region in that frame. Assign this pixel value to 1. Calculate total number of pixels whose value is one. Cursor of mouse will move as per centroid of hand .To find center of hand area following method is used center of hand is calculated by finding average value of co-ordinates of the obtained pixels.

$X=\sum$pixel position on x-coordinate. ………………. (4)
$Y=\sum$ pixel position on y-coordinate. ………………. (5)
Xavg=X/N …………………………………………(6)
Yavg=Y/N …………………………………………(7)
$\frac{Y}{N}$
N=Total number of pixel having value 1.
Average value of x and y coordinate represent position of hand on 320x240 figure window. We have to control position of cursor on screen of computer /laptop which is 1366x768.Screen window size is three times of figure window size i.e. there are 1:3 ratios. Position of mouse cursor on main screen is obtained from following equations

X=X1-[X1/L*Xavg] ………………………………… (8)
Y=Y1-[X1/L*Xavg]…………………....................... (9)
X1=size of main screen along x axis
Y1= size of main of screen along y axis
L= size of figure window along x axis
K=size of figure window along y axis

These X and Y will represent co-ordinate of cursor on main screen. For each frame number of palm region pixels is counted and this value is compared with number of palm region pixels in next frame. If number of pixels are reduced that is when we bend palm, command to perform event is generated. This event can be single click or double click or wheel scrolling. For each frame this procedure is repeated & command to operating system is generated. To perform click operation we have to bend our palm so that number of skin color pixels will get reduced and click event will be generated. Fig. 5 shows complete dotted hand region and reduced skin pixels when hand is bend is shown in Fig.6

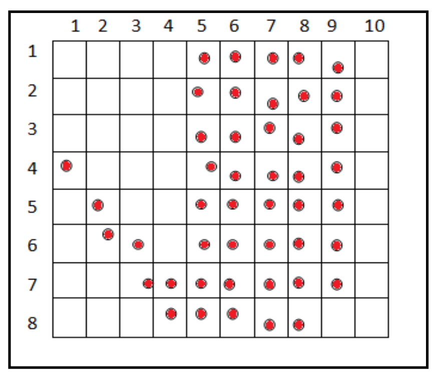

Fig 5:- Palm region pixels 4th frame

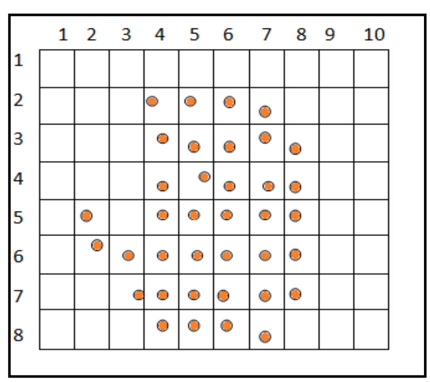

Fig 6:-Reduced skin color pixels in next frame

3.2 Command To Operating System

To move cursor to (X, Y) co-ordinate JAVA.AWT.ROBOT class is used. This class is used to generate system events to control mouse, keyboard function and test automation. AWT is Abstract Window Toolkit.AWT provide interface between java and mouse event. MATLAB imports java.awt.robot class to perform mouse operation. For mouse controlling action this class generates three input events

java.awt.event.inputevent.button1_MASK
java.awt.event.inputevent.button2_MASK
java.awt.event.inputevent.button3_MASK

Button1_Mask is used for left click, Button2_Mask is used for middle mouse wheel and Button3_Mask for right click. **For single click** of mouse following java instruction is used

robot.mouse(java.awt.event.inputevent.BUTTON1_MASK)

robot.mouse(java.awt.event.InputEvent.BUTTON1_MASK)

For double click following java instruction is used

robot.mousePress(java.awt.event.inputevent.BUTTON1_MASK)

robot.mouseRelease(java.awt.event.inputevent.BUTTON1_MASK)

robot.mousePress(java.awt.event.inputevent.BUTTON1_MASK)

robot.mouseRelease(java.awt.event.inputevent.BUTTON1_MASK)

For scrolling of mouse following java instruction used is

Mouse.mouseWheel(inputevent.BUTTON2_MASK)

Following are the experimental results. Here we have considered a parameter i.e. illumination. We have got two experimental results one with illumination and other is with less illumination. We have three actions in form of left click, right click & scroll. All the three actions are performed with same gesture i.e. by bending palm. One of disadvantage of this system is, we will have to perform each action independently. At one time we can't perform all the operation. This can be considered as future scope of this system.

Table 1: Experimental results with normal illumination.

Sr. no	Action			Success rate in %
	Left button	Right button	Scroll button	
1	N	Y	Y	66.66
2	Y	Y	Y	100
3	Y	Y	Y	100
4	Y	Y	Y	100
5	Y	Y	Y	100
6	Y	Y	Y	100
7	Y	Y	Y	100
8	Y	Y	Y	100
9	Y	Y	Y	100
10	Y	Y	Y	100
				Avg =96.66

Table 2: Experimental results with low illumination.

Sr. no	Action			Success rate in %
	Left button	Right button	Scroll button	
1	N	Y	Y	66.66
2	Y	Y	Y	100
3	Y	Y	Y	100
4	Y	Y	N	66.66
5	Y	N	Y	66.66
6	Y	Y	Y	100
7	Y	Y	N	66.66
8	Y	N	Y	100
9	Y	Y	Y	100
10	Y	Y	N	66.66

Table 3: Result comparison of two methods

Sr. No	Name of the method	Result in accuracy
1	Hidden Morkov Model	92.53%
2	Adaptive Kalman Filter	96.66%

The experimental result of HMM is according to paper "Hand Gesture Recognition Using HMM" by Byung-Woo Min, Ho-Sub Yoon, Jung Soh, Yun-Mo Yang, [9].

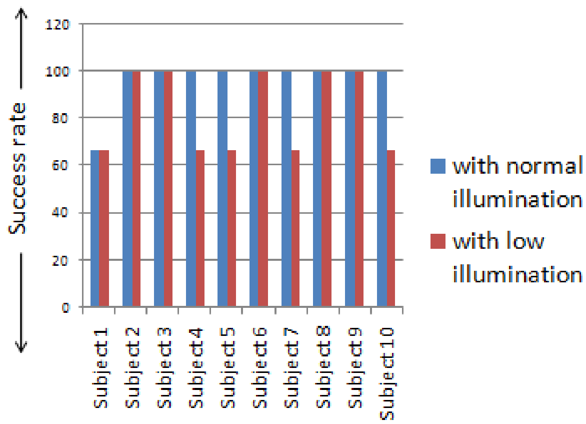

Fig 6: Graphical representation of experimental results.

4. Conclusions

Adaptive Kalman Filter (AKF) gives advantages for Hand Gesture Recognition as compared to HMM. AKF doesn't require any kind of training. It can work without sensors to track the hand. It is simple to use. It gives more accuracy than HMM. Practical accuracy might vary as per the available illumination. The proposed system can be made even more robust so that it will give same performance for all types of users irrespective of their expertise. Due to extensive training sets for modeling HMM is computationally expensive. Authors are willing to continue the experimentation further..

References

[1] Sushmita Mitra, and Tinku Acharya, "Gesture Recognition: A Survey" IEEE VOL. 37, NO. 3, MAY 2007

[2] Mohd Shahrimie Mohd Asaari, "Hand Gesture Tracking System Using Kalman Filter", 2010 IEEE, 10th International Conference on Intelligent Systems Design and Applications

[3] F.S. Chen, C.M. Fu and C.L. Huang, "Hand gesture recognition using a real-time tracking method and hidden Markov models", Image Vision Computer, IEEE v 21 n 8, pp 745- 758, March 2003.

[4] M. Isard and A. Blake, "Conditional Density Propagation for Visual Tracking," IJCV, v 29, n 1, pp 5-28, Aug. 2008.

[5] G. McAllister, S. McKenna and I. Ricketts, "Hand Tracking for Behavior Understanding," Image and Vision Computing, VOL 20, NO 12,Oct. 2009.

[6] Q.Y. Zhang, M.Y. Zhang and J.Q. Hu, "Hand Gesture Contour Tracking Based on Skin Color Probability and State Estimation Model," IEEE, VOL 4, NO 6, Dec. 2009.

[7] K.Imagawa, S.Lu and S.Igi, "Color-Based Hands Tracking System for Sign Language Recognition," Proc. International Conference on Face and Gesture Recognition, Apr. 14-16, 1998, Nara, Japan.

[8] N.D.Binh, E.Shuichi, and T.Ejima, Real-Time Hand Tracking and Gesture Recognition System," Proc. GVIP05, Dec.19-21, 2005, Cairo, Egypt.

[9] Byung-Woo Min, Ho-Sub Yoon, Jung Soh, Yun-Mo Yang, "Hand Gesture Recognition Using HMM ", IEEE International Conference on computational Cybernetics and Simulation 1997,Vol 5 pp 745- 758, March 1997.

Amol Mahadik completed his B.E. in Information Technology from B.A.M. University Aurangabad. Currently he is perusing his master's degree from same university. His research interest lies in Human computer Interaction.

Prof. Madhuri S Joshi completed her BE from College of Engineering, Pune (1985), M.Tech. (CS) (1993) from IIT, Madras and Ph.D. from SRT University, Maharashtra, India. She has published 29 research papers in various International Journals, International and National Conferences. Her areas of interest are Data Mining ,Image Processing and Pattern Recognition.