AN INTERACTIVE HAND GESTURE RECOGNITION SYSTEM ON THE BEAGLE BOARD

PNVS Gowtham
Electronics and Communications
International Institute of Information and Technology
Hyderabad ,India
gowtham.peddada@gmail.com

Abstract— **Interaction so far with many day to day objects has only been mechanical or physical .Though times have changed , we have not yet changed much in this aspect .Many simple tasks can be performed much faster by ways of signaling than by doing things physically .The aim of this paper is to provide with a real time application based on hand gesture recognition .The hardware requirements of the system are kept on the minimum , limiting only to a simple USB webcam and a PC .Other alternatives to the PC have also been presented , such as the Beagle Board , which is a mini version of a PC and very inexpensive .The approach is based on simple and fast motion detection trying to eliminate unnecessary regions of interest and a recognition algorithm based on the hand contours , their convexities and a further comparison of hand shapes based on HU moment matching which works on image contours .The paper also presents a way of improving the distorted hand contours by distinguishing between external contours and holes and explicitly filling up holes .The use of a preloaded set of contours for each gesture helps in better recognition .A simple state machine is implemented to further improve reliability .A total of 5 gestures have been implemented and tested on the PC and the Beagle Board and the performance compared .A practical application using the five gestured detected has been proposed , which utilized the Linux XLIB library and the X display to control mouse cursor actions and other display properties .**

Keywords; Gesture Recognition , HU Moments , Hand Tracking , Contours ,Xlib , Beagle Board

I. INTRODUCTION

One of the greatest gifts of almost all living things in this world is the ability to communicate , not just through words , but through actions also .Many a time , feelings are better conveyed through gestures and expressions than through any other means . Human and machine interaction by far has only been through simple means of communication like a mouse , keyboard or switches .Voice recognition and gesture recognition are two powerful and more natural means of communicating with machines as is with human beings .There are innumerable instances where the conventional keyboard and mice applications can be replaced with computer vision based gestures .One such example is the Microsoft Touch Wall which uses infrared lasers as triggers and cameras to recognize the different gestures .In many cases special gloves or markers have been used for efficient detection and tracking [3] which constraints the user .In [4] a very real time method of hand recognition entirely based on convexity defects is presented but an analysis based only on convexity defects is prone to be dependent on the smoothness of the background .[1] implements gesture recognition based on Hausdorff distance with good accuracies but is not real time especially dedicated and less powerful systems such as the Beagle Board. This paper tries attempts to solve these problems by proposing a real time gesture recognition system with no user constraints and with flexibility in background environment .

II. SYSTEM COMPONENTS

A low cost PC with any inexpensive webcam is all that is required .The PC is later substituted with the Beagle Board as an embedded alternative .The operating system used on the PC is Ubuntu and the operating system installed on the Beagle Board is the Ubuntu Natty.

III. WORK FLOW

There are five stages in the working of the system as explained below

A. INITIALIZATION

A set of recognizable postures are stored in the system memory at the startup of the system [1] .These postures correspond to the contours of different hand gestures .They are very essential in the later stages of Hand Recognition.

B. ACQUISITION

Once the system is begun, every frame from the webcam is acquired and processed .Each frame is a 3 channel RGB frame with each channel having 8 bit depth .The size of each frame is 640X480.

C. PRE PROCESSING BEFORE ANALYSIS

A series of image processing operations are undertaken on each frame captured from the webcam .These operations include motion detection , skin segmentation ,thresholding , smoothing , contour analysis , hole filling , morphological

operations such as dilation , contour cropping and contour approximations.

D. POSTURE RECOGNITION

Once the user's hand contour has been successfully retrieved, the number of fingers in the contour are counted using convexity defects and then recognition is made by comparison with the contours stored in the startup step .For this purpose of recognition a comparison technique based on HU moments is presented.

E. EXECUTION

Once the gesture has been recognized, the corresponding action is executed .Several actions based on mouse events have been presented in the later sections.

IV. GESTURE RECONITION

The steps involved in recognizing the hand gestures are as shown in the flowchart in figure 4.a

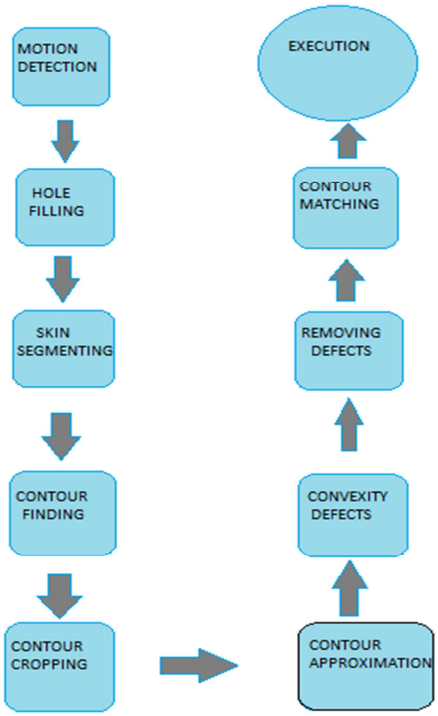

Figure 4a. Flow chart of the algorithm of the gesture recognition

A. MOTION DETECTION

Every frame from the webcam is converted into a gray scale image and a pixel wise subtraction is performed between the current frame and the previously acquired frame. The resulting output is then smoothed to remove noisy pixels .A simple Gaussian smoothing is performed on the resulting subtracted image .A threshold is now applied on the smoothed image thereby changing the image into a binary image with maximum value 255 and minimum value 0 .255 corresponds to pure white and 0 corresponds to pure black .The requirement of a fixed background is eliminated by using motion detection .The background subtraction image would automatically adopt to the new background immediately with an error of change in the background only in one frame , which is tolerable . The change in background should not be continuous in the time domain, but can be discreet at random time intervals.

A simple threshold of 10 is experimented to work well .The threshold is a simple binary threshold as follows [2]

If image(i,j) >10 ; then image(i,j)=255;
Else image(i,j)=0;

Figures 4.1.a, 4.1.b, 4.1.c, 4.1.d, 4.1e show the original frame, the gray scale image, the subtraction image, the Gaussian smoothed image and the threshold image respectively.

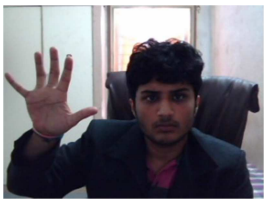

Figure 4.1a .The original frame

Figure 4.1b. The Gray Scale Image

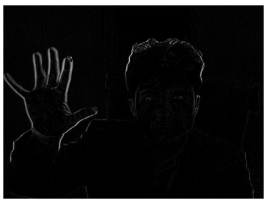

Figure 4.1c.Subtraction Image

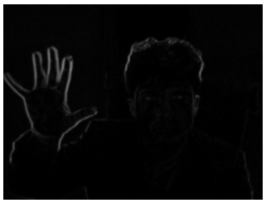

Figure 4.1d. Gaussian smoothed image

Figure 4.1e.Threshold image

B. HOLE FILLING

The output of the motion detection does not yield the full area of the object under motion .As seen in the previous step; there are many black regions or holes with in white boundaries, in the motion detection image .For this region, a hole filling operation is performed on the image. A simple Dilation of the image, reduces the number of holes with in the image, but the hand contour will still not be fully devoid of holes .The holes in the finger regions are better covered after this step but the black region at the center of the hand is uncovered .Figure 4.2a shows the Dilation output .At this stage multiple dilations are not performed as this increases the size of unnecessary regions in the image.

Figure 4.2a: A Dilated Image

C. SKIN SEGMENTATION

The output of 4.2 is again done a logical AND with the original color frame .This operation yields a color version of the image in step 4.2 .This new color motion detection image, is now converted into YCrCb color space and a simple upper and lower bound is applied on the image to color segment it .This process is not accurate, but will suffice because most of the image is eliminated in the motion detection image leaving apart only very small areas which move.

The lower limit is (0,113,67) and the upper limit is (255,173,127) .The output of skin segmentation and the combination of background subtraction and skin segmentation are shown in figure 4.3a and 4.3b

Figure 4.3a. Skin segmentation

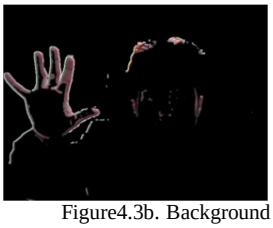

Figure4.3b. Background subtraction and segmentation combination.

D. FINDING APPROPRIATE CONTOURS

A contour analysis gives groups of pixels that have the same label due to their connectivity in the binary image .This is useful because after this process the image is considered as a set of contours and each contour is concentrated individually rather than treating an image as a collection of discreet pixels. This contour analysis also gives us information of the total number of contours, their area, their length etc. A minimum size limit of 4000 pixels is set on each contour to reduce the number of contours which have to be processed .Hence all contours smaller in area than 4000 will not be considered.

A contour analysis is performed on the skin segmented background subtraction image .After skin segmentation , moving objects not of skin color are removed , potentially leaving behind only the hand .Hence in the contour analysis , only the largest contour is selected for the next stage of processing .Once this contour has been selected ,hole filling operation is again performed on the particular contour .This is a contour specific hole filling unlike the dilation performed in previous steps .There are two types of contours namely exterior contours and holes .The contours which are grouped as holes are identified in the image and they are filled up in white .After this contour specific hole filling is performed, a simple Dilation is performed once again .Figure 4.4a, 4.4b denote the dilated motion detection image of step 4.2 and the output of contour hole filling for a sample gesture .Figure 4.4c and 4.4d show the output contours before contour hole filling and after contour hole filling .

Figure 4.4a.Dilated motion detection image.

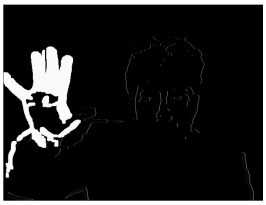

Figure4.4b. Contour Hole Filling

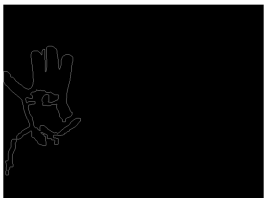

Figure 4.4c. Before Hole Filling

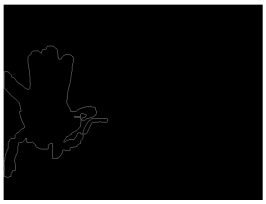

Figure 4.4d.After Hole Filling

E. CONTOUR CROPPING

The most useful information of the hand undoubtedly lies in its fingers .Hence the hand contour is cropped to the upper portion , leaving only the orientation of the fingers to analyze .Figure 4.5a and 4.5b show the upper half binary image and the upper half contour .

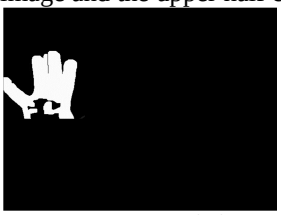

Figure 4.5a. Upper half of the binary image

Figure 4.5b. Contour of the upper half

F. CONTOUR APPROXIMATION

This output contour of step 4.5 is still uneven in shape with very small distortions as shown above. Hence a polygon approximation of the contour is done to approximate the upper part of the hand as a polygon .The polygon approximation yields a contour derived of straight line segments making it easier in finding convexities .The result of polygon approximation is shown in figure 4.6a.

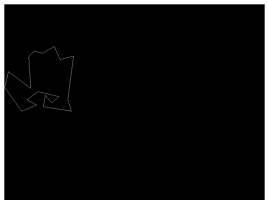

Figure 4.6a.Polygon approximation of the upper half contour

G. FINDING CONVEXITY DEFECTS

Convexity defects are valley point's .In particular, convexity defects are sequences of contour points between two consecutive contour vertices on the contour Hull. Figure 4.7a , 4.7b , 4.7c , 4.7d , 4.7e shows the convexity defects in a palm , open thumb , V shaped fingers , a fist , and in three fingers .These convexity defects are very useful in providing information about the location and state of the fingers . In the figures below, the pure white markers are the convexity defects or the valley points and the dark markers are the start and end points respectively.

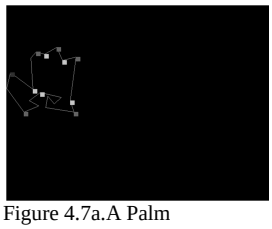

Figure 4.7a.A Palm

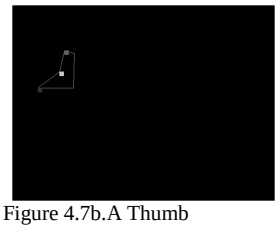

Figure 4.7b.A Thumb

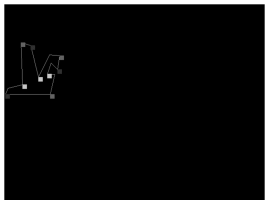

Figure 4.7c. V Shaped Fingers

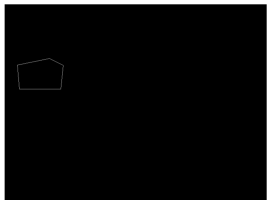

Figure4.7d. A Fist having no defects

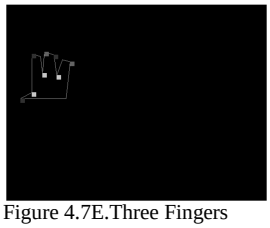

Figure 4.7E.Three Fingers

H. ELIMINATING UNWANTED CONVEXITY DEFECTS AND FINDING THE NUMBER OF FINGERS

As shown in figure 4.8a not all the convexity defects uncovered provide us useful information .Some defects are formed especially due to incomplete contours and some are formed due to distortions in the contour. Most of the defects which are on the underside of the contour are not related to the fingers .For this purpose of eliminating all unnecessary defects , all the defects are sorted in ascending order of their X – coordinates .Every defect has a start and an end point .The start and the end points correspond to the coordinates where the valley begins and ends .For a defect to qualify as a defect caused by fingers , a condition is set that the y coordinate or the height of the defect is greater than the height the heights of both the start and end points by a minimum value .This minimum value , on experimentation is set to be 20.

This condition does not cover the defects at the hand sides which have the height of the end point less than the height of the defect and height of the start point greater than the defect height .Such defects are generally the first and last defects in the sorted list of defects of the hand contour .Such defects are identified, but not removed .They are marked as they are required at later stages .Let such defects be called special defects.

A minimum bounding rectangle is calculated , which bounds the contour .Information about the coordinates of the vertices of the rectangle are also calculated .In case the first defect starts with a special defect , then the horizontal distance from the special defect to its next defect (in the updated list of defects which has some unnecessary defects removed) is measured . In case the first defect is not a special defect , then the horizontal distance between the nearest vertical side of the bounding rectangle and the defect is measured .After that , the distance between every two consecutive defects is measured .That is (defect2.xcoordinate – defect1.xcoordinate) , (defect3.xcoordinate – defect2.xcoordinate) and so on are measured until the last defect is reached .If the last defect is a not a special defect , then the distance between the last defect and the nearest vertical side of the bounding rectangle is measured .If it is a special defect , then it is ignored . Based on the measured distances, a specific range is provided to approximate the number of fingers .In this case, with a distance greater than 75, the number of fingers is approximated as 3 .With the distance greater than 50 the number of fingers approximated is 2 and a distance greater than 15 one finger is approximated .These distances are nearly found to be the same except for very small and very large hand contours. However this is just an approximation .This has to be further confirmed in later stages.

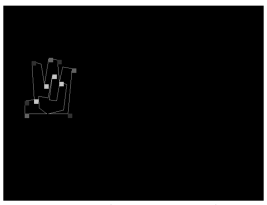

Figure 4.8a.Note the unwanted defect at the left bottom .These defects are not caused by fingers. All such defects are removed.

I. CONTOUR MATCHING

After the finger classification in step 4.8 , the classified images are further confirmed using HU moment matching as follows .A Set of contours for each of the five gestures are stored in the memory at start up . Within each set which

contains four to seven contours , which differ from each other significantly when compared with each other using HU moments .These stored contours are now compared with the hand contour based on its number of fingers .That is if two fingers were found out, it is first compared with the V shaped set of contours . If 5 fingers are found, then it is first compared with the Palm set of contours.

Hu moments are said to be invariant to scale, translation and rotation .They are defined as shown in the figure 4.9a below.

$$I_1 = \eta_{20} + \eta_{02}$$

$$I_2 = (\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2$$

$$I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$I_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$I_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].$$

Figure4.9a showing Hu Moments

Where

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{\left(1+\frac{i+j}{2}\right)}}$$

Where μ (p, q) are the central moments defined as

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

Where f(x,y) is the grayscale image. Hence f(a,b) would be the pixel (a,b) in the binary image . .

The closer the result of the match is to zero , the greater is the resemblance between the two contours .The posture is confirmed to be a recognized gesture only if at least 2 of the following conditions are satisfied.

The minimum value of the match from all the contours in the set is less than 0.2

Out of the all the contours in the set, at least three contours yield a match less than 0.3

The average of the best three matches is less than 0.35

IF at least two conditions are satisfied, then the output is confirmed. Else the output is matched with the next most probable gesture. For example , if 3fingers were counted wrong as two fingers , the output is matched with 3fingers and a thumb and selecting the best match.IE the output is matched with the templates having one extra finger and one finger less .

IN order to improve accuracy, the output is declared as a recognized gesture only after processing 5 continuous frames in which at least 3 frames give the same recognized output.

IE out of 5 frames if 2 frames are matched to a fist and 3 frames are matched to a palm , then the output would be a palm .IN cases where there is equal weightage for multiple gestures , for example 2 -palm , 2- thumb , and 1 finger , the output is declared to be the previous output .IE if the output previously was palm , then the output of this set of 5 frames would also be a palm .Figure 4.9 b,c,d,e

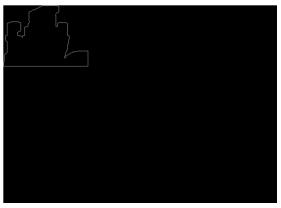

Figure 4.9b

Figure 4.9c

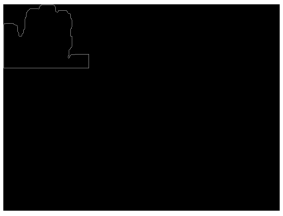

Figure 4.9c

Figure 4.9d

V. EXECUTION

Once the gesture has been detected, it can be used as a trigger to perform an action .The X11 library is a C library which enables writing of programs called X-Clients.

.For example, a detected Palm is realized as a “no action once a fist is recognized, it is realized to be a “left mouse click”. The movement of the mouse pointer is made in accordance to hand being tracked .The operating system used for this purpose is LINUX and the libraries used are the X11 libraries .Every action such as movements of the pointer , single click , double click , right click are events which are called XEvents. .Functions such as XWarpPointer , XOpenDisplay , XCloseDisplay ,XQueryPointer XSendEvent are used in performing the required action .The language used is C language and OPENCV libraries are also used for image processing.

VI. EXPERIMENTS

Experiments on a set of five people with varying skin tone have resulted in the following outcomes .The following table illustrates the accuracies of the finger detection explained in the above section.

GESTURE	CORRECT
Fist	95
Thumb	90
V shaped Fingers	88.23
Palm	85.4
Three consecutive fingers	89

The accuracies are further improved in the later stages by using HU moment comparisons and the state machine. The beagle board on running the same algorithm is measured to be 3.2 times slower than the PC.The PC is a Sony vaio with a 2.24 GHz Intel Dual Core Processor while the Beagle Board XM runs on a 1 GHz OMAP processor.

VII. CONCLUSION AND FUTURE WORK

An approach based on hand contour convexities and Hu moment matching has been proposed .A method of having different hand gesture templates at the system start up and finding the best match has been presented . The implementation on the Beagle Board shows that the system is suitable in real time even with an embedded device of lower processing power compared to the PC. Any simple USB camera can be connected with the PC or the Beagle Board with no limitations .Future work includes trying to further improve the accuracy of the work and study and compare other techniques on Hand Gesture Recognition which have been proposed so far .

ACKNOWLEDGEMENTS

This work has been done under Professor Akash Kumar and Professor Bharadwaj Veeravalli from the National University of Singapore as a part of my summer project

REFERENCES

[1] Elena Sanchez-Nielsen , Luis Anton-Canalis , Mario Hernandez-Tejera “Hand Gesture Recognition For Human-Machine Interaction “ Journal of WSCG ,Vol 12,No 1-3 , ISSN 1213-6972

[2] Shahzad Malik “Real-time Hand Tracking and Finger Tracking for Interaction “

[3] Robert Y.Wang,Jovan Popovic “Real-Time Hand-Tracking with a Color Glove”

[4] Cristina Manresa , Javier Varona , Ramon Mas and Francisco J.Perales “Real –Time Hand Tracking and Gesture Recognition For Human –Computer Interaction “

[5] Zhengo Zou , Prashan Premaratne ,Ravi Mongarala , Nalin Bandara ,Malin Premaratne “Dynamic Hand Gsture Recognition System using Moment Invariants “ 2010 IEEE ICIAfS10

[6] Prateen Chakraborty,Prashant Sarawagi ,Ankit Mehrotra ,Gaurav Agarwal ,Ratika Pradhan “Hand Gesture Recognition : A Comparitive Study” Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol 1 IMECS 2008 , 19-21 March 2008 , HONG KONG