2011 International Conference on Advancements in Information Technology
With workshop of ICBMG 2011
IPCSIT vol.20 (2011) © (2011) IACSIT Press, Singapore

An Interactive Hand Gesture Recognition System on the Beagle Board

PNVS Gowtham[+]

Electronics and Communications International Institute of Information Technology

Abstract. 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 predefined 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 utilizes 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

1. Introduction

One of the greatest gifts of almost all living things in this world is the ability to communicate through vocals and actions. 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 conventional keyboard and mice applications can be replaced with hand 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 gesture recognition based on convexity defects is presented but an analysis based only on convexity defects is dependent on smoothness of background. [1] Implements gesture recognition based on Hausdroff distance but is not real time especially when implemented on dedicated systems such as Beagle Board .This paper attempts to solve these problems by proposing a real time gesture recognition system with no user constraints and with any kind of background environment.

2. Gesture Recognition

The steps involved in recognizing hand gestures are as shown in figure 2.a

[+] Corresponding author. Tel.: +7893298258.
E-mail address: gowtham.peddada@gmail.com.

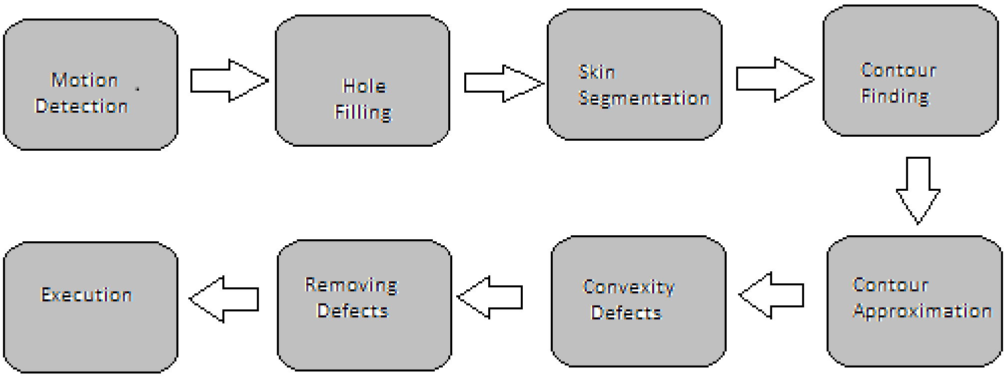

Fig. 2a: Flow chart of the algorithm of gesture recognition

2.1. Motion Detection

Every frame from webcam is converted into a gray scale image and a pixel wise subtraction is performed between the current frame and previously acquired frame. The resulting output is then smoothed using a Gaussian filter to remove noisy pixels .A threshold is now applied on smoothed image thereby changing it 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 immediately adapt to new background with an error only in the first frame after change. Change in background should not be continuous in time domain, but can be discreet at random time intervals.

A simple threshold of 10 is experimented to work well [2].
If image(i,j) >10 ; then image(i,j)=255; Else image(i,j)=0;
Figures 2.1.a, 2.1.b show original frame and threshold I mage respectively.

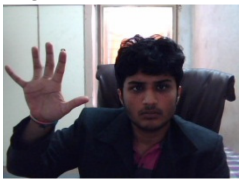


Fig.2.1a: Original Frame Fig.2.1b: Threshold Image Fig.2.2: Dilation Image Fig.2.2c: Finaloutput

2.2. Hole Filling and Skin Segmentation

Output of motion detection does not yield the full area of the object under motion .As seen in figure 2.1b, there are many black regions or holes with in white boundaries in the motion detection image .For this reason, a hole filling operation is performed .Dilation reduces number of holes, but hand contour will still not be devoid of holes. Figure 2.2a shows Dilation output .At this stage multiple dilations are not performed as this increases size of unnecessary regions in the image.

A logical AND is performed between the dilated image and the original frame. This new image is now converted into YCrCb color space and an upper and lower bound are set 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) .Combination of background subtraction and skin segmentation is shown in figure 2.2b which is the final output.

2.3. Finding appropriate Contours

A contour analysis gives groups of pixels that have same label due to their connectivity in binary image. After such an analysis the image is treated as a set of contours and not as a collection of discreet pixels. After contour analysis individual contours with a minimum size of 4000 pixels are isolated.

The largest contour is the one with highest probability of being a hand. Once this contour has been selected, hole filling operation is performed on the particular contour. There are two types of contours namely exterior contours and holes .Holes associated with the largest contour are identified in the image and are filled white. This is a contour specific hole filling unlike the dilation performed in previous steps. Figure 2.3a and 2.3b show the output contours before and after contour hole filling.

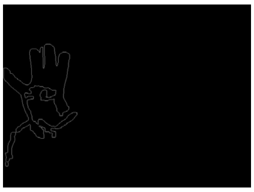

Fig.2.3a: Before Hole Filling

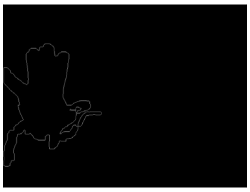

Fig.2.3b: After hole Filling

Fig.2.4a: Upper Contour

Fig.2.4b: Contour Polygon

2.4. Contour Cropping and Contour Approximation

The most useful information of the lies in its fingers .Hence the hand contour is cropped to the upper portion, leaving only the orientation of the fingers to analyze .Figure 2.4a shows the upper half contour.

This output contour is still uneven in shape with very small distortions. Hence a polygon approximation of the contour is done to approximate the upper part of the hand as a polygon .This yields a contour derived of straight line segments making it easier in finding convexities .The result of polygon approximation is shown in figure 2.4b.

2.5. Finding Convexity Defects

Convexity defects are valley points. In particular, convexity defects are sequences of contour points between two consecutive contour vertices on the contour Hull. Figure 2.5a , 2.5b , 2.5c , 2.5d , 2.5e 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.

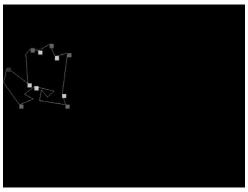

Fig. 2.5a: Palm

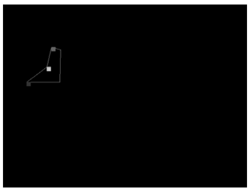

Fig. 2.5b:Thumb

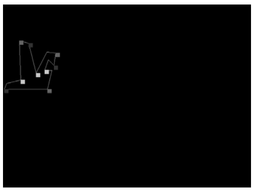

Fig. 2.5c: Fingers

Fig. 2.5d: Fist

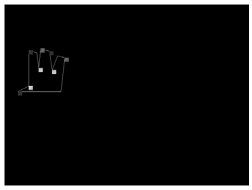

Fig.2.5e: three Fingers

2.6. Eliminating Convexity Defects

As shown in figure 2.6a not all the convexity defects provide useful information .Some defects are formed especially due to incomplete and distorted contours. Most of the defects which are on the underside of the contour are not related to the fingers. 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 of 20 units.

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 height of the defect .Such defects are generally the first and last defects in the list of defects which are sorted in increasing order of their X-coordinates. Such defects are identified, but not removed .They are marked as special defects, as they are required at later stages.

A minimum bounding rectangle is a rectangle of minimum area which bounds the contour. The rectangle and coordinates of its vertices are 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 edge 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 distance between last defect and nearest vertical side of bounding rectangle is measured .If it is a special defect , then it is ignored . Based on measured distances, a specific range is provided to approximate the number of fingers .In this case, with a distance greater than 75, number of fingers is approximated as 3 .With distance greater than 50 number of fingers approximated is 2 and a distance greater than 15 one finger is approximated .These distances are found to be nearly the same except for very small and very large hand contours. This is confirmed in later stages.

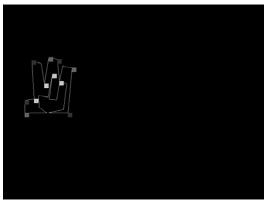

Fig. 2.6a: Note the unwanted defect at the bottom left corner .These defects are not caused by fingers.

2.7. 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 . Each set which contains four to seven contours differ from each other significantly when compared using HU moments .Depending on the number of fingers calculated, the contour polygon is compared to one set among the five sets of contours stored at startup. 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

$$h1 = \left(\eta_{(2,0)} + \eta_{(0,2)}\right)$$

$$h2 = \left(\eta_{(2,0)} - \eta_{(0,2)}\right)^2 + 4\left(\eta_{(1,1)}\right)^2$$

$$h3 = \left(\eta_{(3,0)} - \eta_{(1,2)}\right)^2 + \left(3\eta_{(2,1)} - \eta_{(0,3)}\right)^2$$

$$h4 = \left(\eta_{(3,0)} + \eta_{(1,2)}\right)^2 + \left(\eta_{(2,1)} + \eta_{(0,3)}\right)^2$$

$$h5 = \left(\eta_{(3,0)} - 3\eta_{(1,2)}\right)\left(\eta_{(3,0)} + \eta_{(1,2)}\right)\left(\left(\eta_{(3,0)} + \eta_{(1,2)}\right)^2 - 3\left(\eta_{(2,1)} + \eta_{(0,3)}\right)^2\right) + \left(3\eta_{(2,1)} - \eta_{(0,3)}\right)\left(\eta_{(2,1)} + \eta_{(0,3)}\right)\left(3\left(\eta_{(3,0)} + \eta_{(1,2)}\right)^2 - \left(\eta_{(2,1)} + \eta_{(0,3)}\right)^2\right)$$

$$h6 = \left(\eta_{(2,0)} - \eta_{(0,2)}\right)\left(\left(\eta_{(3,0)} + \eta_{(1,2)}\right)^2 - \left(\eta_{(2,1)} + \eta_{(0,3)}\right)^2\right) + \left(4\eta_{(1,1)}\right)\left(\eta_{(3,0)} + \eta_{(1,2)}\right)\left(\left(\eta_{(2,1)} + \eta_{(0,3)}\right)^2\right)$$

$$h7 = \left(3\eta_{(2,1)} - \eta_{(0,3)}\right)\left(\eta_{(2,1)} + \eta_{(0,3)}\right)\left(\left(3\eta_{(3,0)} + \eta_{(1,2)}\right)^2 - \left(\eta_{(2,1)} + \eta_{(0,3)}\right)^2\right) - \left(\eta_{(3,0)} - \eta_{(1,2)}\right)\left(\eta_{(2,1)} + \eta_{(0,3)}\right)\left(3\left(\eta_{(3,0)} + \eta_{(1,2)}\right)^2 - \left(\eta_{(2,1)} + \eta_{(0,3)}\right)^2\right)$$

Where $\eta_{(p,q)}$ are the normalized moments.

The measure of similarity is given by $S(A,B) = \sum_{i=1}^{7} \left| \frac{1}{m_i^A} - \frac{1}{m_i^B} \right|$

Where $m_i^A = \operatorname{sign}(h_i^A) \log\left|h_i^A\right|$ and $m_i^B = \operatorname{sign}(h_i^B) \log\left|h_i^B\right|$

Where h_i^A and h_i^B are the Hu moments of Contours A and B respectively.

The closer the result $S(A,B)$ 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 three 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 output is confirmed. Else output is matched with next most probable contour set. If k be the number of fingers detected in the contour polygon then next probable contour sets are the sets having k+1 and k-1 fingers. 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 .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. Figure 2.7b, c, d,e show the set of palm templates

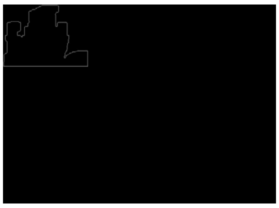

Fig. 2.7b: Template 1

Fig. 2.7c: Template2

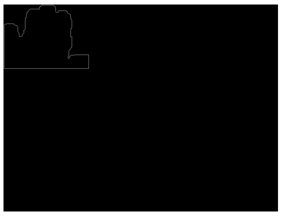

Fig. 2.7d: Template3

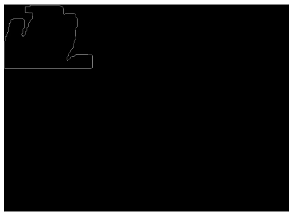

Fig. 2.7e: Template 4

3. Execution

Once gesture has been detected, it can be used as a trigger to perform an action. X11 library is a C library which enables writing of programs called X-Clients. The movement of mouse pointer is made in accordance to hand being tracked.X11 Libraries and Linux are used for this purpose. 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.

Gesture Sequence	Action
Palm	Single Click
V Shaped Fingers	Right click
Fist	No action .Only Hand Tracking
Thumb	Double Click

4. Experiments

Experiments on a set of five people with varying skin tone have resulted in the following outcomes .The following table illustrates the ac curacies of the output gesture recognition which includes finger detection and Hu Moment Matching and State Machine .

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

The beagle board on running the same algorithm is measured to be 3.2 times slower than a 2.24 GHz Intel Dual Core Processor while the Beagle Board XM runs on a 1 GHz OMAP processor.

5. 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 .

6. 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.

7. 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