

Real Time Hand Gesture Recognition for Computer Interaction

Javeria Farooq

Department of Electronic Engineering
Balochistan University of Information Technology
Engineering & Management Sciences
Quetta, Pakistan
Javeria.farooq74@gmail.com

Muhaddisa Barat Ali

Department of Electrical and Electronics Engineering
Bilkent University, TR-06800 Bilkent ,
Ankara, Turkey
Muhaddisa.ali@bilkent.edu.tr

Abstract—Hand gesture recognition is a natural and intuitive way to interact with the computer, since interactions with the computer can be increased through multidimensional use of hand gestures as compare to other input methods. The purpose of this paper is to explore three different techniques for HGR (hand gesture recognition) using finger tips detection. A new approach called “Curvature of Perimeter” is presented with its application as a virtual mouse. The system presented, uses only a webcam and algorithms which are developed using computer vision, image and the video processing toolboxes of Matlab.

Keywords-human computer interaction; hand gesture recognition; finger tips detection

I. INTRODUCTION

Hand gesture recognition is one of the most active areas of research in computer vision. It provides an easy way to interact with a machine without using any extra devices. If the user doesn't have much technical knowledge about the system then human-computer interaction (HCI) enables the user to use the system without any difficulty [1]. In some HCI systems based on hand gestures, the users have to wear a kind of glove [2],[3], but this does not make any sense for practical use. People want something more natural. This technology can be used as an aid for people with handicaps.

II. METHODS

There are several methods for Hand gesture recognition. Two of them are mentioned here.

A. Template Matching:

In the template matching approach an image or part of image is compared with some template given in a database. In other words this approach tries to find some pattern in the image [4],[5]. Figure 1 and figure 2 show the examples of template matching.

B. Finger tips detection

In this approach hand gestures can be recognized by means of finger tips detection. Many researchers have used fingertips detection for hand gesture recognition [7],[8],[9]. In this paper, the fingertip detection approach is used.

In literature, several different methods are used for finger tips detection. In this paper three algorithms are employed as follows:

- Convex -Hull
- K-Curvature
- Curvature of Perimeter

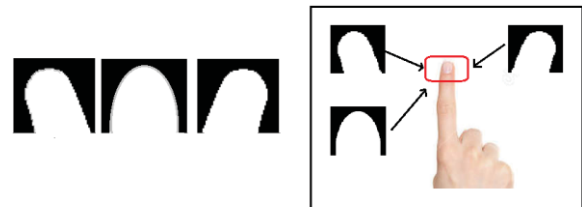


Figure 1. Template matching example “Redrawn from [4]”



Figure 2. Template matching example [6]

III. SYSTEM MODEL

The system flow diagram is shown in figure 3. Initially the system acquires images captured from a webcam. The output of the webcam is basically a video sent to the system. A system will acquire video in the form of sequences of frames. Matlab has an efficient image acquisition toolbox for this purpose. After acquiring an image, it will be converted into $L*a*b$ space for color segmentation. Then edges of a segmented hand are detected using canny edge detection algorithm and the noise is removed. From the segmented hand, fingertips can be found by any of three given approaches. Music player and virtual mouse are two applications based on these approaches.

IV. HAND GESTURE RECOGNITION

A. Image Acquisition

In order to initialize the system user have to wave his/her hand in front of webcam. The image acquisition toolbox of Matlab acquires the output of the webcam in terms of frames. The two functions “getsnaphot” and “getdata” in Matlab are used for grabbing the frame.

- **frame = getsnaphot(obj)**
- **data = getdata(obj)**

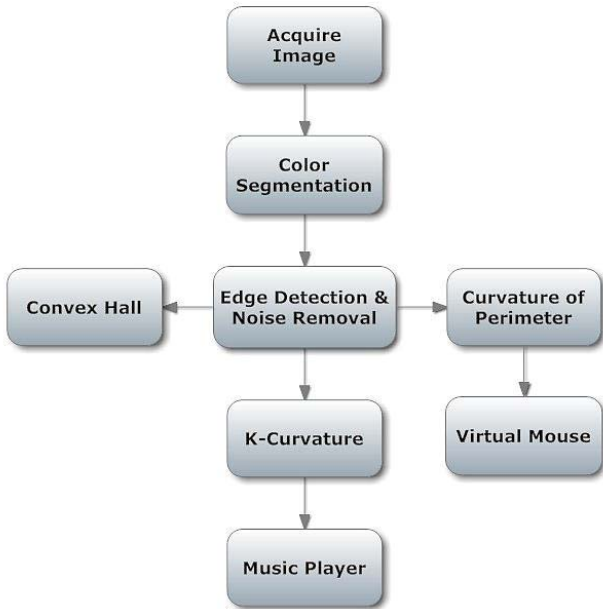


Figure 3. Hand gesture recognition using three different approaches

B. Color Segmentation

In the next step, it is needed to segment the hand from the background. Skin color extraction in real time is a very difficult task. So, for this reason RGB is converted to L^*a^*b color space. L^*a^*b color space is derived from the CIE XYZ tristimulus values. L^*a^*b space consists of luminosity ‘ L^* ’ or brightness layer, chromaticity layer ‘ a^* ’ indicating where color falls along the red-green axis, and chromaticity layer ‘ b^* ’ indicating where color falls along blue-yellow axis [10].

Converting L^*a^*b moves color on three dimension (R, G, B) onto two chrominance channels and one luminance channel. First grab the two chrominance channels; calculate the distance from a reference color to color of the given pixel. Classify using nearest neighbor rule. Finally, any pixel close enough will set as ‘1’. Hence, a binary image of segmented hand is acquired.

Skin colors can be extracted by choosing particular values for ‘ a^* ’ and ‘ b^* ’. In this paper, color A=17.3 and color B= 13.82 are chosen. It should be noted that these values may vary according to the different skin colors. Figure 4 shows the results.

C. Edge Detection and Removing Noise

The next step is to detect edges or to find contours of the segmented image. This is done by “Canny edge detection”. RGB color space is converted to L^*a^*b color space and the hand region is segmented from the input image.

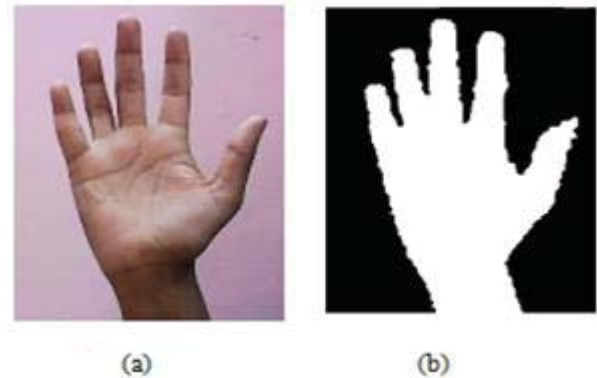


Figure 4. Color Segmentation. (a) RGB input image (b) Segmented hand in binary

Noise can be removed by considering the segmented hand or blob as a biggest region and removing the small objects. Results are shown in Figure 5.

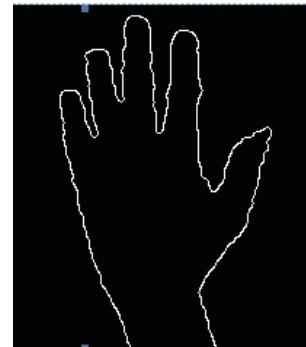


Figure 5. Edge detection using canny edge detection

D. Finger Tips Detection

1) Convex-Hull

The first technique to find a fingertip is with a convex hull algorithm. In this approach, a convex hull or a polygon around a blob or segmented hand, i.e. a smallest convex that encloses all the points of the binary image, is made.

The kinks in convex hull are used to find the finger tips. This is done by going around each point in a convex hull and calculating the angles at those points. In this paper law of cosine is used to find angles. Filter the angles to find the finger tips. Figure 6 shows the results.

a) Flaws in convex hull

The convex hull method is fast but it is not robust. For example, when making a fist the knuckles are identified as fingertips since they are the points in convex hull, likewise when moving the finger inwards, it losses the fingertips. Figure 7 shows the faults in convex hull.

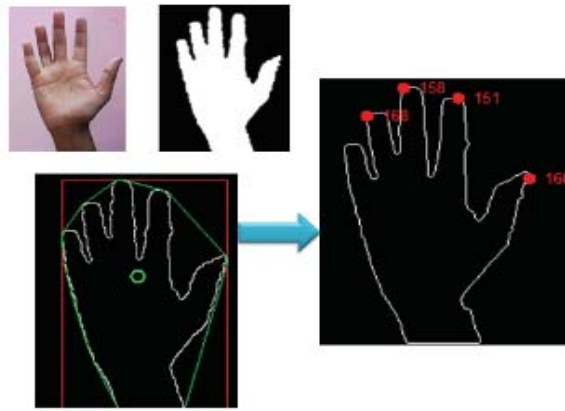


Figure 6. Convex hull method

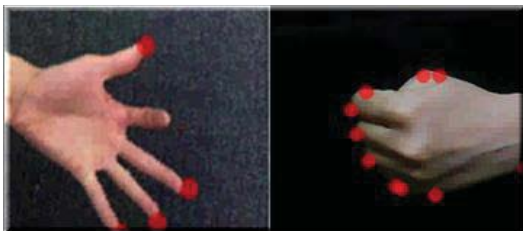


Figure 7. Live video results and flaws in convex hull

2) K-Curvature

In this approach, we represent the contour in a list of boundary points $P(i)=(x(i),y(i))$, and we compute k-curvature i.e. the angle between the two vectors $[P(i-k), P(i)]$ and $[P(i), P(i+k)]$, where k is a constant (here we set $k=35$). K-Curvature can be calculated easily using dot product. The main idea is that the points in k-Curvature close to the “0” are considered as candidates points (represent peaks and valleys). Figure 8 shows the idea behind this approach. A threshold angle $\theta_{th}=30$ is used such that the points below this angle will be considered further.

In order to find whether it is a peak or valley, the vectors are converted into 3D, lying into xy-plane and then computing their cross-product. If the sign of the z component is positive, then it is considered as peak while for a negative sign it is considered as valley.

By knowing the number of peaks and valleys, one can find the gesture [3],[11]. Figure 9 shows the live video results of k Curvature.

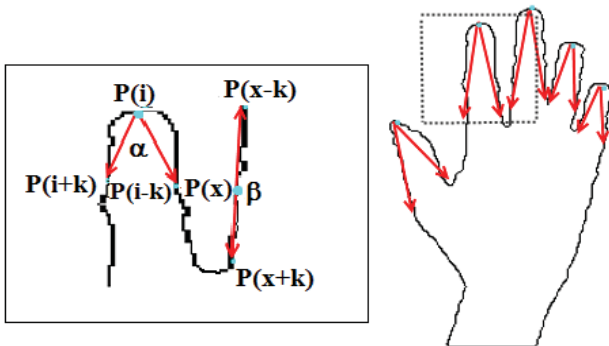


Figure 8. Idea of K-Curvature “Redrawn from [3][12]”

a) Limitaion in K-Curvature

K-curvature cannot be used for dynamic gesture recognition, and it is not robust. For K-curvature, if there is a fist then it will always identify as “1” finger.

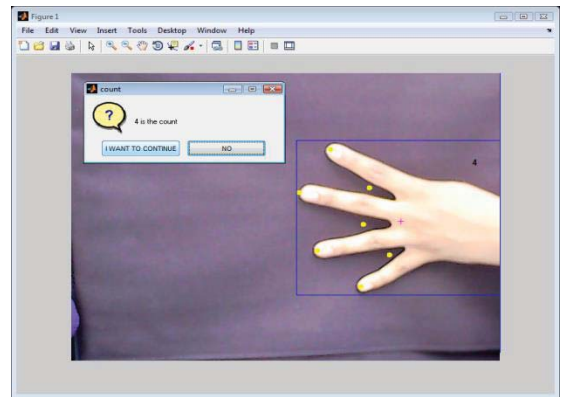


Figure 9 Live video results of K-Curvature

3) Curvature of Perimeter

In this technique, some morphological operations are used to find the finger tips. Firstly, the segmented hand is eroded using distance transform and then finding the perimeter of the regional hand. After this, the corner points of the eroded version of the segmented hand are explored. All the corner points will be candidate points. Eliminate the corner points closer to the boundary. At each point, crop a section of the perimeter of the segmented hand and calculate the eccentricity. Low eccentricities indicate shapes closer to circles similar to the tips of fingers. With certain value of eccentricity it can be said that it is closed to the fingertips. Other candidate points, such as knuckles, it is going to be a straight line. Now the candidate points at fingertips and candidate points along the finger are distinguishable.

Figure 10 shows the idea of a curvature of perimeter. This approach is pretty much robust and can be used for dynamic gesture recognition. Similarly figure 11 shows the live video results of a curvature of perimeter.

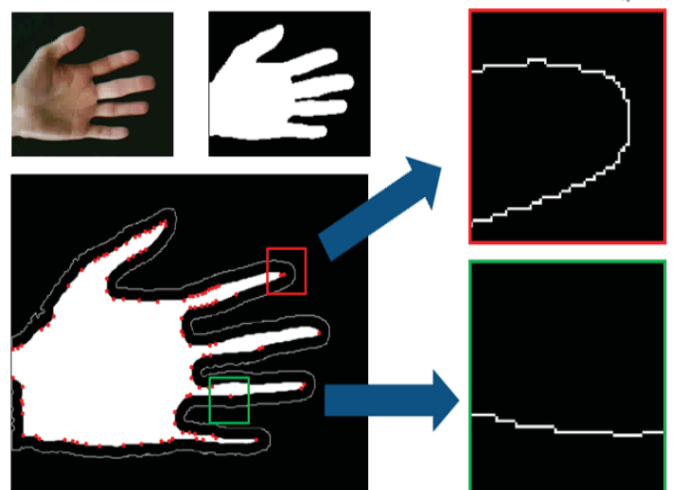


Figure 10. Curvature of perimeter idea, the red box shows the candidate point at fingertips while green box shows the candidate at knuckle.

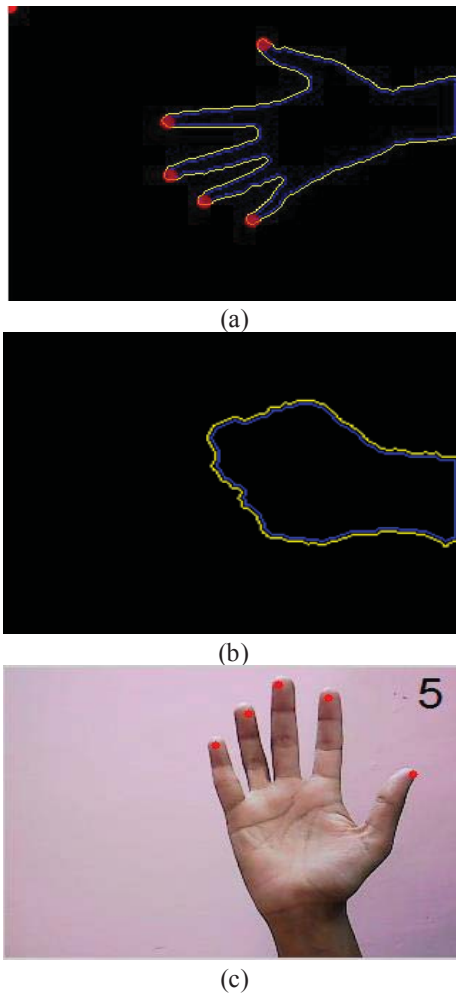


Figure 11. Live video results of Curvature of Perimeter, (a) open hand, (b) fist and (c) showing the fingertips and count.



Figure 12. Live video snapshot of Music Player

V. APPLICATIONS

A. Music Player

Music player is an application of a static gesture using K-curvature method that we implemented using Active-X protocol.

ActiveX is a set of technologies developed by Microsoft for sharing information among different applications. ActiveX is an outgrowth of two other Microsoft technologies named OLE (Object Linking and Embedding) and COM (Component Object Model) [14]. By using Active-X, it is possible to integrate windows media player in Matlab using the following command

```
h = actxcontrol('progid') [13]
```

where the 'progid' for window media player is 'WMPPlayer.OCX.7'. It is also possible to change the song by changing the path and file name. Different tasks are performed for different hand gestures such as "Open Hand" which is used to stop the music. Figure 12 shows the live video results of music player with different hand gestures for respective operation.

B. Virtual Mouse

Virtual mouse is an application of Curvature of perimeter. The ability to control the mouse pointer position, motion and clicks are not available in Matlab. To resolve this issue, java class "java.awt.Robot" which has this ability, is imported [15].

To control mouse position and motion, "Centroid Tracking" is used. The centroid of binary hand is passed to class mouse.mouseMove(x,y). Hence the mouse pointer is moved according to the centroid motion. Other tasks are performed by identifying the different gestures in the current frame. If the system does not detect any figure (fist), it will only track the hand, and when it detects five fingers it performs "click" function. Figure 13 shows the live video snapshot of virtual mouse using paint.

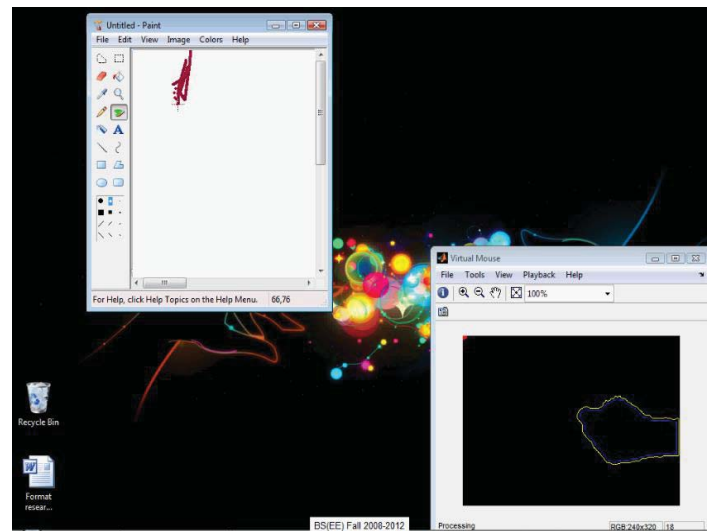


Figure 13. Live video result of virtual mouse using "Paint"

VI. SUMMARY AND FUTURE SCOPE

This paper explores three techniques for hand gesture recognition called convex hull, k-curvature, and curvature of perimeter. Experimental results show that curvature of a perimeter is the most robust approach among them. Music player and virtual mouse are two applications discussed in this paper.

The system has other applications in computer games, robot navigation, and touch less interaction with computers.

The next step is to refine the detection portion by tracking the fingertips using Kalman filtering and estimate where fingers are going. The methodology for hand segmentation from a background can be improved. Since the background was simple, it is easily segmented but for the complex background K-mean clustering can be used which is a statistical approach. Lastly, the algorithms can be optimized further by converting them into a fixed point and generating C code from it to embed them into systems like DSP or FPGA.

ACKNOWLEDGEMENT

I would like to express my sincere thanks to Miss Muhaddisa for her support and guidance throughout.

REFERENCES

- [1] Ankit Chaudhary, J. L. Raheja , Karen Das , Sonia Raheja, “ Intelligent Approaches to interact with Machines using Hand Gesture Recognition in Natural way”. International Journal of Computer Science & Engineering Survey (IJCSES) Vol.2, No.1, Feb 2011 .
- [2] K. G. Derpanis, “A Review of Vision-Based Hand Gesture,” Department of Computer Science, York University, February, 2004.
- [3] Thiago R. Trigo, Sergio Roberto M. Pellegrino, “An Analysis of Features for Hand-Gesture”. IWSSIP 2010 - 17th International Conference on Systems, Signals and Image Processing .
- [4] Jong-Min Kim; Woong-Ki Lee, “Hand Shape Recognition Using Fingertips” Fuzzy Systems and Knowledge Discovery, 2008. FSKD '08. Fifth International Conference on , vol.4, no. , pp.4448,18-20 Oct. 2008
- [5] Oka, K.; Sato, Y.; Koike, H., “Real-time fingertip tracking and gesture recognition,” Computer Graphics and Applications, IEEE, vol.22,no.6, pp. 64-71, Nov/Dec 2002.
- [6] <http://www.mathworks.nl/matlabcentral/fileexchange/30447-point-pattern-matching-algorithm-for-hand-gesture-american-sign-language-asl-recognition>
- [7] Nolker C., Ritter H., Visual Recognition of Continuous Hand Postures, IEEE Transactions on neural networks Vol 13, No. 4, July 2002, pp.983-994
- [8] Nguyen D.D., Pham T.C., Jeon J.W., Fingertip Detection with Morphology and Geometric Calculation, IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, USA, Oct 11-15, 2009, pp. 1460-1465.
- [9] Lee D. and Park Y., Vision-Based Remote Control System by Motion Detection and Open Finger Counting, IEEE Transactions on Consumer Electronics, Vol. 55, issue 4, Nov 2009, pp. 2308-2313..
- [10] MATLAB example, Color based segmentation using the L*a*b color space , Copyright 1993-2007 The MathWorks, Inc. Published with MATLAB® 7.11 <http://www.mathworks.com/products/image/examples.html?file=products/demos/shipping/images/ipexfabric.html>
- [11] Shahzad Malik ,”Real-time Hand Tracking and Finger Tracking for Interaction”, CSC2503F Project Report, December 18, 2003
- [12] Truyenque, M. A. Q., "Uma Aplicação de Visão Computacional que Utiliza Gestos da Mão para Interagir com o Computador," Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, Rio de Janeiro, Brazil, March 2005.
- [13] ActiveX Control , Mathworks Documentation Center : Retrieved from: <http://www.mathworks.com/help/matlab/ref/activexcontrol.html>
- [14] ActiveX, Webopedia, Retrieved from: <http://www.webopedia.com/TERM/A/ActiveX.html>
- [15] How can I programmatically control mouse motion and clicks with MATLAB?, Retrieved from: <http://www.mathworks.com/support/solutions/en/data/1-2X10AT/index.html>