Visual Approach to Dynamic Hand Gesture Recognition for Human Computer Interface

Jagdish Lal Raheja¹, Rahul Jain², and Pritish Mohapatra²

¹ Central Electronics Engineering Research Institute/Council of Scientific & Industrial Research (CSIR), India
Email: jagdish@ceeri.ernet.in
² Birla Institute Technology and Science, Pilani, India
Email: {rahulspsec, pritish.eee}@gmail.com

Abstract—This paper presents an approach for detection of fingertips and features of human hand and its application in the field of human computer interface. The fingertip detection algorithm consists of four steps. First, a skin filter based on the YCbCr colour space is used to obtain a binary silhouette of the image of the hand. Second, the region of interest (fingers in this case) is then separated from the rest of the binary image by using a rectangular mask of appropriate dimensions. This process is made fast by using the summed area table of the binary image for calculating the mask value at different locations of the image. Third, the filter response of circular separability filter is calculated for all points in the region of interest. Fourth, the points whose value is beyond an appropriately chosen threshold are grouped by the 8-connectivity criteria. The mean of the coordinates of these grouped points then gives the approximate location of the fingertip. The precise fingertip location is then calculated by making use of the orientation of the finger. The fingertip detection is particularly useful for human computer interface. This concept is used to implement applications like virtual touch screen, manipulation of virtual objects and virtual mouse.

Index Terms—Circular separability filter, fingertip detection, skin filter, summed area table.

I. INTRODUCTION

There has been a great emphasis lately in Human computer interface (HCI) research to create easier to use interfaces by directly employing natural communication and manipulation skills of humans. Adopting direct sensing in HCI will allow the deployment of a wide range of applications in more sophisticated computing environments such as Virtual Environments (VEs) or Augmented Reality (AR) systems. The development of these systems involves addressing challenging research problems including effective input/output techniques, interaction styles and evaluation methods. In the input domain, the direct sensing approach requires capturing and interpreting the motion of head, eye gaze, face, hand, arms or even the whole body. Among different body parts, the hand is the most effective, general-purpose interaction tool due to its dexterous functionality in communication and manipulation. Various interaction styles tend to import both modalities to allow intuitive, natural interaction.

Gesture languages made up of hand postures (i.e., static gestures) or motion patterns (i.e., dynamic gestures) have been employed to implement command and control interfaces [1–4]. Gesticulations, which are spontaneous movements of the hand and arms that accompany speech, have proved to be a very effective tool in Multimodal User Interfaces [5–9]. Object manipulation interfaces [10–12] utilize the hand for navigation, selection, and manipulation tasks in VEs.

The use of hand gestures, as an interface between human and computers or machines in general, has always been a very attractive alternative to the conventional interface devices. Information from hand gesture can either be gathered by using sensors attached to the hand, or it can be obtained from visual data. The glove based method hinders the ease and naturalness with which humans interact with the computer. This has led to an increased interest in the visual approach.

There have been many methods proposed for dynamic hand gesture recognition by fingertip detection [14–18]. But there have been several limitations in these approaches. The method proposed by Yang in [14] analyses the hand contour to select fingertip candidates, then further finds peaks in their spatial distribution and checks local variance to locate fingertips. This method is not invariant to the orientation of the hand. So are the methods using directionally variant templates to detect fingertips [15–16]. Some other methods are dependent on specialized instruments and setup, a fixed background [15–16] or the use of markers on hand. This paper proposes a novel method for recognition of motion patterns of the hand without any kind of sensor or marker being attached to the hand.

The different steps of the method, some applications built using the method and the results obtained are discussed in the subsequent sections. In the second section, the several steps involved in determining the location of the fingertip and the center of the palm are described. Discussions on the results obtained and some applications developed using this method are included in the third and fourth section. Conclusion and discussion about the possible future extensions to this work is included in the last section.
II. FINGERTIP DETECTION

The fingertip detection algorithm comprises of four stages. First, a skin filter based on YCbCr colour space is used to obtain the binary silhouette of the hand image. Second, the region of interest (fingers in this case) is then separated from the rest of the binary image by using a rectangular mask of appropriate dimensions. Third, the value of circular separability filter is calculated for all the points in the region of interest. Fourth, the points whose value is beyond an appropriately chosen threshold are grouped by the 8-connectivity criteria. The mean of the coordinates of these grouped points then gives the approximate location of fingertip. The precise location of the fingertip is then calculated by tracing along the line passing through the approximate location point and in the direction of the orientation of the finger. The orientation of the finger is determined by using a mask consisting of two concentric circles.

A. Skin Filter

The skin filter used is based on HSI and YCbCr color spaces. In the HSI colour space the skin is filtered using the chromacity (hue and saturation) values. In the YCbCr color space the (Cb-Cr) value is used for filtering skin. The skin filters are used to create a binary image with the background black and the hand region appearing white. The binary image is then smoothened using the averaging filter.

B. Region of Interest

The region of interest, i.e. the region of finger is separated from rest of the hand. This is done by using a rectangular mask of appropriate dimensions, 31x38 in our implementation, and separating the finger from rest of the hand. The mask is applied over the binary image of the hand and sum of the values of all the pixels that lie in the rectangular region is calculated. If this sum crosses the experimentally set threshold of 1100, all the pixels that lie in the rectangle region are merged with the black background. Thus separating the finger region, that is still white. This process is made fast by using the summed area table of the binary image for calculating the mask value at different locations of the image. The summed area table [13] is a type of representation in which the value of every element is equal to the sum of the values of all the pixels to the left of and top of the current pixel in the actual image. The value of the pixel \((x,y)\) in the summed area table will be,

\[
I(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')
\]

Where \(I(x,y)\) is the summed area table and \(i(x,y)\) is the original image. Using the following pair of recurrences:

\[
Rsum(x, y) = Rsum(x, y - 1) + i(x, y)
\]

\[
I(x, y) = I(x, y - 1) + Rsum(x, y)
\]

(Where \(Rsum(x,-1) = 0,\) and \(I(-1,y) = 0\)) the summed area table can be computed in one pass over the original image.

Using the summed area table, the sum of the pixel values in the rectangle made by points 1,2,3,4 i.e. the region D can be calculated as,

\[
Sum(D) = I(1) + I(4) - I(2) - I(3)
\]

So, the value of the rectangular mask over a region can be calculated by simply four look ups. This improves the speed of the computation by a factor of around 250. The center of the palm is then calculated as the mean of the centers of all the rectangular regions that satisfy the threshold condition.
C. Circular Separability Filter

The circular separability filter has the shape of a square with a concentric circle inside. In our implementation the radius of the circle is taken to be equal to 5 pixels and the bounding square had a size of 20 pixel length. The weight attached to the points inside the circle is +2 and the weight attached to the points outside the circle but inside the square is -2. When the filter response is computed for all the points of the region of interest in the binary image, the filter response for the fingertip regions are found distinctively different from that of other regions. And the candidate fingertip locations are determined by using an appropriate threshold condition.

D. Fingertip detection

The exact fingertip location is calculated in two steps. First, an approximate location for the fingertip is determined and then the exact location is calculated by further calculation using the orientation of the finger. The 8-connected points that satisfy the threshold condition for the filter response of the circular separability filter are grouped together. The groups having number of pixels more than a set threshold are selected and the centroids of the groups are taken as the approximate fingertip positions. The fingertips are then accurately detected by using the following method.

The orientation of each finger is determined using a filter with 2-concentric circular regions. The diameters of the inner and outer circular regions of the filter are equal to 10 pixels and 20 pixels respectively. The points inside the inner circle, are assigned a value of +2, points outside the inner circle but inside the outer circle, are assigned the value -2, and the points outside the outer circle but inside the bounding square are assigned the value 0.

The filter is then applied on the binary silhouette of the hand image at the previously detected approximate finger tip locations. The pixels that lie in the -2 region are grouped by the 8-connectivity criteria. Then the largest group is selected and the centroid of the group calculated. The orientation of the finger is calculated as the angle \( \theta \) made by the line joining the centroid of the largest group and the previously calculated approximate finger tip location, with the horizontal. Then we move in this direction in several steps with an incremental distance \( r \) using the below mentioned formulae till the edge of the finger is reached.

\[
R_{\text{new}} = R_{\text{old}} + r \cos(-\theta) \tag{5}
\]
\[
C_{\text{new}} = C_{\text{old}} + r \sin(-\theta) \tag{6}
\]

Where, \( R_{\text{old}} \) and \( C_{\text{old}} \) are the 2D coordinates of the previous trace point, \( R_{\text{new}} \) and \( C_{\text{new}} \) are the 2D coordinates of the current trace point and \( r \) is the incremental distance.

The values of \( R_{\text{new}} \) and \( C_{\text{new}} \) after the iterations give the exact coordinates of the fingertips.

RESULTS

The above proposed method was tested with different poses of the hand and the results were found to be exceedingly good. The system could detect fingertips almost unfailingly for any pose of the hand and the fingertip location calculated was found to be highly accurate. The latency period of the system was 0.12 seconds when an input image size of 160x120 was used. Experiments were performed on a computer with Intel Centrino processor. The system performed almost flawlessly when some props like a pen or a bottle was introduced inside the frame. Some of the results are displayed in figure 8.
The method proposed was used to develop some applications. The three applications developed, virtual touch screen, virtual object manipulator and virtual mouse are discussed in this section.

A. Virtual Touch screen

The proposed concept was used to implement a virtual touch screen. In this application, a virtual key board is overlaid on the screen. The finger is moved in front of the camera. With visual feedback from the screen the fingertip can be easily brought over the cell of the keyboard that needs to be selected. When the finger is tapped over a cell, that cell is selected and the corresponding operation is executed. The concept being, whenever the finger is tapped, the distance between the fingertip and the center of the palm is decreased below a set threshold, and this is taken as a key press. Figure 9 shows some frames from virtual touch screen application of a telephone key matrix.

Figure 9: Some frames showing the results of the virtual touch screen

B. Virtual Object Manipulator

One of most important motivations for the active research going in the field of HCI is to develop a method for controlling and manipulating objects in a virtual environment by using natural human gestures. Using the method proposed in this paper, an application involving manipulation of virtual color discs appearing on the screen by use of natural hand motion was implemented.

The algorithm used was as follows. Whenever the distance between two fingertips is less than a certain threshold, the object in the vicinity of the midpoint of the
two fingertips is selected and its center is aligned to the midpoint of the fingertips. Then the virtual object is moved along as the fingers are moved. Taking the two fingers apart deselects the virtual object and it stays in its new location on the screen. Figure 10 shows some frames from this application.

Figure 10: Some frames showing the results of the virtual object manipulator

C. VirtualMouse

In the implementation of the virtual mouse, the new coordinates of the tip location are compared with the previous coordinates, and the difference between the coordinates is appropriately scaled and added to the previous coordinates. This results in mouse motion. Presently, the system is restricted to a single click. The clicking is simulated in the same manner as used in the virtual touch screen i.e. tapping the finger at the desired location. This simulates the left click of the mouse. The position of the pointer may not remain constant due to noise and hence if the fluctuation is within 1 grid pixel, the tip is considered immobile.

CONCLUSION

We have developed novel method for fingertip detection which can be used to capture the motion of the fingers. The fingertip locations along with the center of the palm can be used to implement several applications. The speed of the system developed is around 8 frames per second. The system is very robust towards interference by other objects in the frame. The proposed method has been used to develop applications like virtual touch screen, virtual object manipulator and virtual mouse.

Future work on the system can include improvement of the hand region segmentation algorithm for more complex background. Methods for determining the location of finger joints and the location of the wrist can be explored.

REFERENCES