ISSN No. 0976-5697
Volume 7, No. 7, Nov-Dec 2016
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Design and Development of Hand Gesture Recognition System

Kavitha Sooda
Dept. of CSE
Nitte Meenakshi Institute of Technology
Bangalore – 560 064, India

Kathakali Majumder
Dept. of CSE
Nitte Meenakshi Institute of Technology
Bangalore – 560 064, India

Aishwarya Muralidharan, Ayesha Sultana, Aliya Farheen
Dept. of CSE
Nitte Meenakshi Institute of Technology
Bangalore – 560 064, India

Abstract: In this paper a vision based hand gesture recognition system is developed. Here the computer perceives the human gestures and responds in a pre-defined way. It uses the Kinect sensor built on a software technology developed internally by a subsidiary of Microsoft Games Studios owned by Microsoft. The proposed algorithm consists of a sequence of steps which are further divided into sub-steps, each of which is performed using different algorithms. The major steps include preprocessing, hand segmentation and feature extraction. In this work the segmentation of the hand from the rest of the image is carried out. Also elimination of the background noise, calculation of centroid of the hand, identifying the gesture and matching it to the stored templates is simulated. To capture the images concepts like contours, convex hull and convexity defects are applied. The paper provides a prototype on how different gestures of the hand can help interact with the computer interface easily.

Keywords: Contours; Hand Gesture; Image Processing; Kinect sensor; Preprocessing;

I. INTRODUCTION

In our everyday life, we use a number of hand gestures for communication. They are considered the most natural yet innovative means of communication. Gestures are the simplest means of conveying something meaningful. Technological advancements in any existing equipments are gauged by the parameters of customer comfort and convenience. In most of our systems the input devices are a mouse or keyboard. From past three decades we are using the same techniques for communicating with the computer. Understanding and decoding hand gestures as input allows the computers to be more accessible for the physically-impaired and makes interaction more natural in a gaming or 3D virtual reality environment. The earlier techniques of gesture recognition were done using instrumented gloves, optical markers and many more. These techniques have both advantages and disadvantages attached to them.

The instrumented gloves which were used to track the location of the hand and finger tips was a very important part of the gesture recognition process as it helped to detect the palm and segment it from the rest of the hand. The gloves contained number of sensors in it, which gave the information about hand location, orientation and finger tips. These gloves had high accuracy but were very expensive and needed wired connection. An optical marker was another important tool which works with infrared light. Working with the optical markers was a very complicated process and the system would require complex configuration.

The "air gesture" system on Samsung Galaxy S4 drains the battery when run continuously [1]. Similarly, wire-fewer approaches [2, 3] consume significant power and computational resources that limit their applicability to plugged-in devices such as Wi-Fi routers. This is due to two main reasons:

First, existing approaches need power-hungry sensing hardware. Cameras drain the battery.

Second, they require rich signal processing capabilities — computing optical flows, FFTs, frequency-time doppler profiles etc. which is computationally expensive and performing these operations while maintaining fast response times consumes power.

The growing interest in using human gestures as a means of interaction with computing device has led to the success of Xbox Kinect which can be used for the development of novel gesture-recognition approaches including those based on infrared [4], electric-field sensing [5], and more recently, wireless signals (e.g., Wi-Fi) [3]. We introduce , a novel gesture-recognition system which has the capability to enable a number of interactive applications for Internet-of-things, where sensor deployments enable smart homes and cities[6,7] and many other related applications.

In this paper, we present an efficient and effective method for hand gesture recognition.

The novelty of the proposed method is listed as follows:

- The first novelty of the proposed method is that the hand gesture recognition is done by the first detecting the hand region through background subtraction method and then the finger recognition is carried out using suitable algorithms to add to the accuracy of the gesture recognition.

- Some previous works need the users to wear data glove to acquire hand gesture data. However, the special sensors of data glove are expensive and hinder its wide application in real life. In our work, we have made use of the Kinect sensor, to capture the depth of the environment. Our approach only uses the Kinect sensing camera to capture the vision information of the hand gesture and thus does not need the help of a special tape to detect hand regions.

- The third advantage of the proposed method is that it is highly efficient and fit for real-time applications.

The rest of the paper is organized as follows. In section 2, the scope of the project is described in detail. In section 3, the system architecture with data acquisition details, hand segmentation, region growing and finger identification is described. Section 4 consists of template matching with static and dynamic hand gestures. Section 5 represents the conclusion and future work.

II. Scope of the Project

The development of our projects involves the following major objectives:

- Interacting with the computer through various gestures in order to get a useful output.
- Understanding the working of the Kinect sensor.
- Using Kinect sensor to capture the gestures.
- Interfacing of Kinect sensor with Gesture Recognition System.
- Conversion of the captured RGB image to grayscale image that provides the depth parameters of the captured image for ensuring accuracy in gesture recognition.
- Segmentation of the hand to detect the palm and capture the exact gesture.
- Enhancing the captured image by calculating a threshold value for accurate segmentation and eliminating the background details.
- Assigning a specific function to each of the detected gestures which will be displayed to the user.
- Recognition of 3-4 major gestures by the human hand.
- The gestures include (1 or 2) hand movements and static hand postures of a single hand.

Finally, the design of the application with attractive and user friendly UI using Qt software is provided.

III. System Architecture

Hand gesture recognition system has been divided into five different phases. They are data acquisition, hand segmentation and pre-processing, feature extraction and finally the recognition. Data acquisition involves capturing the image of the hand using a suitable input device (the Kinect camera) [8]. The image that is captured by the camera contains the users hand along with the background. The image, thus, needs to be segmented in order to remove the unwanted background details. The third phase of this process is the image processing which is further subdivided into a number of steps which include: noise removal, edge detection, contour detection, and normalization of the image to add to the accuracy of the detection. The features are then extracted from the segmented and pre-processed image for recognition. Finally, the image is recognized as a meaningful gesture based on the gesture analysis.

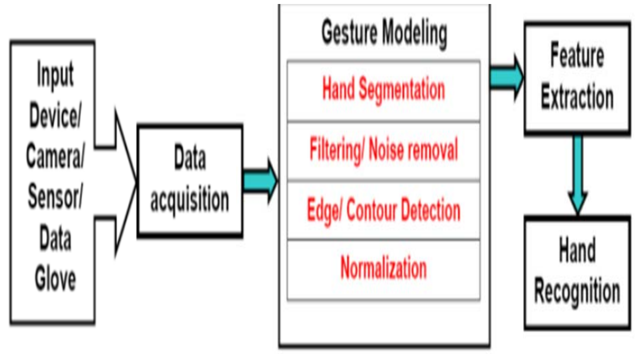

Figure 1. Generalized System Architecture for Hand Gesture Recognition

A. *Data Acquisition*

For efficient hand gesture recognition, data acquisition should be as accurate as possible. The key to ensuring this kind of accuracy is by ensuring the selection a suitable input device. Input devices are available in various types and forms. There are data gloves, markers, images from a webcam or a stereo camera and drawings. a simple web camera can be used to feed bare hand gestures to the system easily and naturally. The Microsoft Kinect 3D depth sensor is considered to be one of the best and most accurate 3D motion sensing input device in today's day and age. It is widely used for gaming purposes. The Kinect sensor consists of a laser projector and a CMOS sensor which is operational in any lighting condition.

The Kinect driver which is OpenNI is also popularly known as Sensor Kinect, and is derived from an open source. An important reason for using OpenNI is its support for middleware [9]. The NITE library that is used is the most suitable as it understands different hand movements as gesture types based on how 'hand points' change with time. The gesture information is sent to the back end and gesture matching take place. The information is displayed through User Interface. The camera is initialized and the gestures are captured. The gesture recognition take place and hand is tracked. Unwanted background details are removed and the hand is segmented. This segmented hand information is processed and matched with the gestures in the database. This process is shown in the figure.

B. *Region Based Hand Sementation*

The main aim of this process is to divide the image into meaningful structures or segments to make it easier to analyze these structures and process them. The main focus in our previous work was how to analyze and represent a certain object assuming that the pixels that represented the object was already known to us. Now, our focus is to find which particular pixels identify an object and how to recognize them. A number of segmentation methods like threshold based segmentation, region based segmentation, edge based segmentation, clustering have been used in the past.

Segmentation of an image primarily means to partition an image into specific regions. Some segmentation methods such as "Thresholding" achieve this goal by looking for the boundaries between regions but these methods have numerous disadvantages. Advantages of using this method over the other methods are:

- This method separates the regions that have the same properties that we define from the other regions in the image.
- Region growing method provides the original images but with clearer and more defined edges.
- We only need to discover a small numbers of seed points to represent the property we want. We can then grow that region.
- We can choose the multiple criteria or properties at a time.
- This method performs well with respect to noise.

C. *Thresholding*

One of the best and most widely used methods of image segmentation is Thresholding. This operation is a grey value remapping operation g which is defined by:

$$g(v) = 0 \text{ if } v < t \text{ } 1 \text{ if } v >= t \quad (1)$$

where v represents a grey value, and t is the threshold value. Thresholding maps *a* grey-valued image or a grey scale image that is captured by the Kinect camera to a binary image which is represented in terms of 0's and 1's. After the thresholding operation, the image is now segmented into two parts, identified by the pixel values 0 and 1 respectively. In order to this, a threshold value needs to be identified[10]. The methods of threshold value selection are discussed further.

D. *Histogram Analysis*

Identifying a suitable threshold value usually involves histogram analysis. Different features give rise to distinct features in a histogram. In general the histogram peaks corresponding to two features will overlap. The degree of overlap depends on peak separation and peak width. The threshold we are setting is 100 or 0 and that threshold image is the exact image which is black and white.

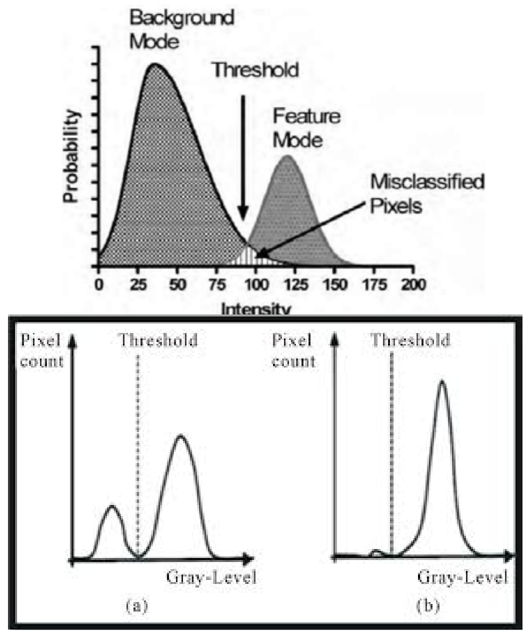

Figure 2. Histogram calculation image of thresholding

E. *Iterative Threshold Selection Algorithm*

Assume that the four corner points of the image are background pixels (part of segment 0), and set μ0 to the average grey value of these four pixels. Assume all of the other pixels are object pixels, and set μ1 to their average grey value.

- Set the threshold

$$T=1/2\ (\mu 1 + \mu 2) \quad (2)$$

- Re compute μ0and μ1, the mean of the original grey values of the two segments.
- Go to step 2 and iterate until the threshold value no longer changes (or no longer changes significantly

Some of the possible variations are:

- Pick an initial threshold value, t (say 128).
- Calculate the mean values in the histogram below (m1) and above (m2) the threshold t.
- Calculate new threshold. tnew = (m1 + m2) /2. If the threshold has stabilized (t = tnew), this is your threshold level.
- Otherwise, t become tnew and reiterate from step 2.

F. *Region Growing*

- The hand tracking method done in the previous stage provides us with a single position of the centre of the hand. This point is taken as the SEED POINT.
- The SEED POINT has a certain intensity which is considered as the SOURCE PIXEL. Considering this point as our source we use the region growing

technique to increase the region and hence obtain the entire hand.

- Once the threshold value is calculated it is compared with the seed point. Now if the intensity of the seed point is 165(say) the next step is to check its immediate neighbours.
- All the pixel values that have the same intensity or are within the threshold range is connected to form a polygon. This process continues until all its neighbours are 0.
- An error rate of a few pixels needs to be set based on trial and error method.

G. Finger Identification

The finger identification in an image is carried out everytime the input data source changes. If the object does not change from one frame to the other with probably only a slight transformation in the object, then all the properties of the current frame is mapped with the previous frame . If this is not the case, then the object is collected by an unmapped item collector.
Identifying the fingers of the hand is a simple process. The first step is to identify the thumb and the index finger, as the distance between these two fingers is the largest. Once the thumb and index fingers are identified, the little finger is the farthest from the thumb and the middle finger is the closest to the index finger. The only left over finger is the ring finger.

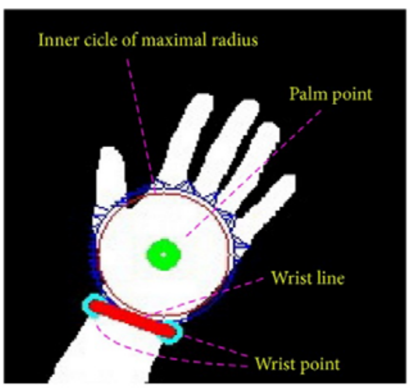

Figure 3(a). Hand palm point detection

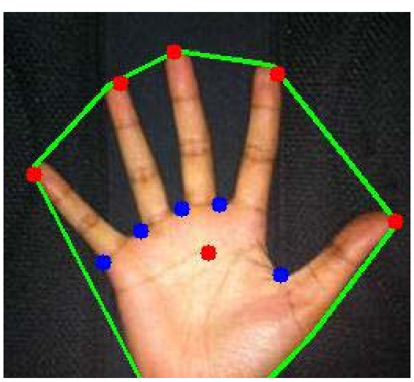

Figure 3(b). Finger border detection by polygon shape

The palm point is defined as the center point of the palm. It is found by the method of distance transform. In Figure 3a, the detection of hand palm helps in detecting the palm of the hand and each finger border. This helps to detect each finger and take the points to form a polygon and get the boundary as in Fig: 3b. The palm point is of minimal radius of inner circle and the inner circle of the maximal radius. When the palm point is found, it can draw a circle with the palm point as the center point inside the palm. The circle is called the inner circle because it is included inside the palm. The radius of the circle gradually increases until it reaches the edge of the palm. That is the radius of the circle stops to increase when black pixels are included in the circle [11]. The circle is the inner circle of the maximal radius which is drawn as the circle with the red color in figure 3b.

Fingertips are detected by checking the three-point alignment algorithm relationship. The following provides the Three Point Alignment Algorithm:

- Let C be the set of all candidate points that are both on the convex hull and the hand contour.
- For each point *Po* m C, take the two points PI and *P2* m the two opposite.
- Find the midpoint of *PI* and *P2* and calculate the distance between this midpoint and *Po*. If the distance is larger than a specific value, the three points are not aligned, i.e., far from being collinear. Then this *Po* is identified as a fingertip. Otherwise go back to Step 2 and check the next candidate point.
- After fingertips are detected, their direction vectors are easy to calculate by subtracting the position of the palm centre.

  Pc(xc, Yc) 11 = *(x - xc, y - Yc)'*

- The vectors are pointing from the palm centre to fingers. Finger names are determined according to their relative distances, which require the user to open palm and to extend fingers. The Fingertip class has a string variable to store the corresponding names.

H. Template Matching

Template matching technique is an age old technique to find areas of an image that match a template image stored in the system. This technique has two basic components:

1. **Source image (I):** The image captured by the camera which needs to be matched to the template.
2. **Template image (T):** The patch image which is compared to the source image.

This technique aims at finding the highest matching area. The threshold value has already been set to 0.7 based on the previous technique of finding the most suitable threshold value. This value is the max value. The minMaxLoc function is used to locate the highest value (or lower, depending on the type of matching method) in the R matrix. The Figure 4a,4b represents the process of template matching[12].

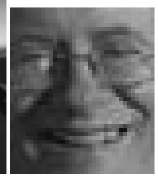

Fig. 4(a) Input Image Fig. 4(b) Testing Image

Static hand gestures are defined as orientation and position of hand in the space during an amount of time without any movement. Hand and body gestures can be amplified by either a controller that contains accelerometers and gyroscopes to sense various movements of the hand like tilting, rotation and acceleration of the hand or body part. A system can also contain a high resolution camera that is able to sense and detect gestures made by the user and perform a function accordingly for example, a wave of the hand, might terminate a program.

Our system is basically detecting static hand gestures and each gesture, if detected by the sensor correctly, will display a meaningful message in the dialogue box. The templates of the static gestures are set previously and the system matches with these pre-defined templates. All these static gestures using various sign help to perform some function and tell the user that the system is working properly. We can also change the message according to customer's requirement or understand ability. The name of the gestures should be saved in sequence. This is the prototype model which is just showing how static gestures can be detected and it can be used for vision interaction application.

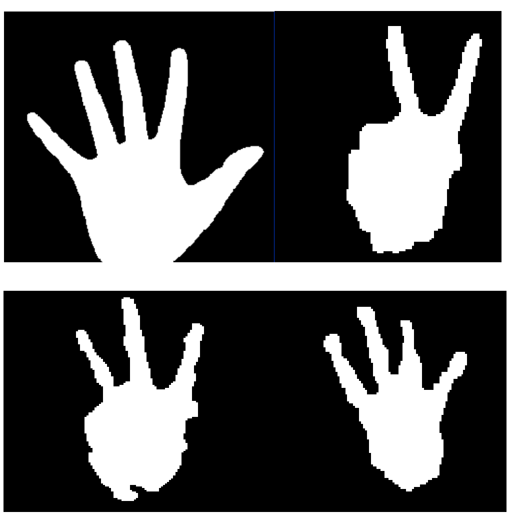

Figure 5. Static gestures to count the number of fingers

Dynamic gesture recognition is far more complicated than the traditional static gesture recognition because of the rate of movements of the body part in dynamic gestures. Gesture recognition is a process in which the hand needs to be separated from the rest of the surroundings and then the palm position is to be determined. This process becomes difficult as the position of the hand and the background keeps changing at a certain rate in dynamic movements. So we have solved all these problems and tried to show the hand in dynamic movement.

In dynamic hand gesture we have detected one hand gesture that is hand wave. There will be only movement of hand in x direction and little movement in y direction and thus the hand will be tracked. So our system is basically detecting a dynamic(moving) hand gesture when we wave our hand and it will say a message "Hello".

IV. QT SOFTWARE

QT is a cross platform application framework that is widely used for developing application software that can be run on various software and hardware platform with little or no change in the underlying codebase. It is C++, Java and QML integrated development environment which is part of the SDK for QT GUI Application development framework. It includes a visual debugger and an integrated GUI layout and forms designer.

QT designer is a tool for designing and building GUIs from QT widgets. It is possible to compose and customize the widgets or dialogs and test them using different styles and resolutions directly in the editor. Widgets and forms created with QT designer are integrated with programmed code using the QT signals and slots mechanism.

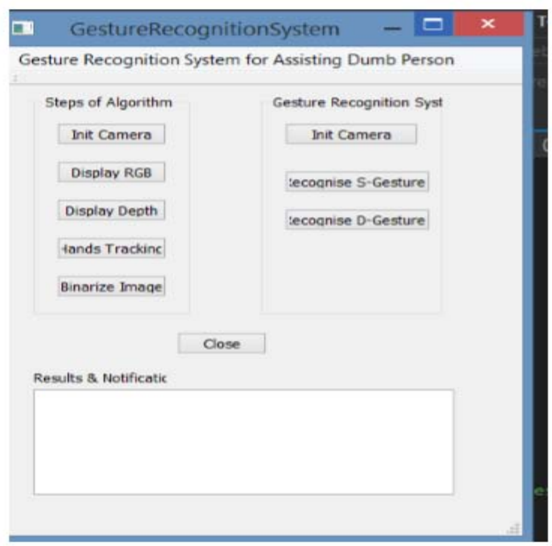

Figure 6. Overview of UI design

V. CONCLUSION AND FUTURE WORK

Hand gesture recognition system is a very vast and innovative field. Current research indicate gesture recognition plays a very important role in helping the physically challenged people to use computer systems with much more ease by facilitating nonverbal communication between human and computer. Gesture recognition has made playing video games very close to a live experience. With the increase in applications, the gesture recognition system demands lots of research in different directions. In this paper, we have tried and put together different sub-components and methodologies used for recognition of mainly hand gestures used in the past and also a brief comparison of those methods with the method that we have used in developing this system.

However, in order for the algorithm to work optimally, the following constraints have been imposed:

1. Only one hand at a time
2. The hand needs to be within a specific depth range (~70cm)
3. There should be no obstacle between the camera and the hand
4. The palm and forearm should be close to perpendicular
5. The fingers should be spread apart, and pointing upward.

A new method for hand gesture recognition is introduced in this paper. The hand region is detected from the background by the background subtraction method. Then, the palm and fingers are segmented. On the basis of the segmentation, the fingers in the hand image are discovered and recognized. well and is fit for the real-time applications. There are numerous potential improvements to the project as future work. Connection of any other software or application with our system's existing hand gestures can help the user start or end an application with just a simple movement of the hand.

VI. REFERENCES

[1] Air Gesture on Samsung S4 drains battery: http://www.1techportal.com/2013/05/maximize-your-galaxy-s4s-battery-in-five-ways/.

[2] F. Adib and D. Katabi. "Seeing Through Walls Using WiFi! In SIGCOMM", 2013.

[3] Q. Pu, S. Gupta, S. Gollakota, and S. Patel" Whole-Home Gesture Recognition Using Wireless Signals." In MOBICOM, 2013.

[4] "Infrared gesture recognition", Technical Report, Sili-con Labs. http://www.silabs.com/Support%20Documents/TechnicalDocs/AN580.pdf.

[5] GestIC: electrical near field (E-field) 3D Tracking and Gesture Controller by Microchip http://www.microchip.com/pagehandler/en-us/technology/gestic/home.html

[6] Smart Things. http://www.smartthings.com/

[7] G. Cohn, E. Stuntebeck, J. Pandey, B. Otis, G. D.Abowd, and S. N. Patel. Snupi "Sensor nodes utilizing powerline infrastructure", in Proceedings of the 12th ACM international conference on Ubiquitous computing, Ubicomp '10, ACM, 2010.

[8] http://www.iprogrammer.info/programming/hardware/2623-getting-started-withmicrosoft-kinect-sdk-html

[9] Yi Li "Hand Gesture Recognition Using Kinect" B.S Communication University of China, Master of Science Page-25- 25-28 2010.

[10] OpenNI organization. *OpenNI User- Guide,* 2010.

[11] Suryanarayan P, Subramanian A, Mandalapu D.Dynamic hand pose recognition using depth data. Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10).

[12] Real-Time Hand Gesture Recognition Using Finger Segmentation Volume 2014 (2014), Article ID 267872.

[13] "Hand Gesture Recognition Systems: A Survey"International Journal of Computer Applications *(0975 – 8887) Volume 71– No.15, May 2013.*