Hand Gesture Recognition System for Image Process (IP) Gaming

Ashwini Shivatare, Poonam wagh, Mayuri Pisal,Varsha Khedkar *Prof. Mrs. Vidya Kurtadikar*
Department of computer engineering, MMCOE , Pune-52

***Abstract*- Hand gesture recognition (HGR) provides an intelligent and natural way of human computer interaction (HCI). Its applications range from medical rehabilitation to consumer electronics control (e.g. mobile phone). In order to distinguish hand gestures, various kinds of sensing techniques are utilized to obtain signals for pattern recognition. The HGR system can be divided into three parts according to its processing Steps : hand detection, finger identification, and gesture recognition. The system has two major advantages. First, it is highly modularized, and each of the three steps is capsuled from others; second, the edge/contour detection of hand as well as gesture recognition is an add-on layer, which can be easily transplanted to other applications. In IP Gaming we are proposing a system in which without using sensors and devices, we are detecting the hand and gesture with simple web camera and performing the image processing technique in which using those gesture, we can play game on console. In Image Process Gaming, the motions are detected through a web camera. These images are then passed for the image processing. The techniques used for image processing are hand gesture detection, edge detection, thresholding, contour detection. Using OpenCV, which provides a library collection of functions for different image processing techniques, these input images can be processed and corresponding key strokes will be generated.**

I. INTRODUCTION

Hand gesture recognition provides an intelligent and natural way of human computer interaction (HCI). Its applications range from medical rehabilitation to consumer electronics control (e.g. mobile phone). In order to distinguish hand gestures, various kinds of sensing techniques are utilized to obtain signals for pattern recognition. Acceleration-base and electromyogram-based techniques are two research branches in the field of hand gesture pattern recognition. Acceleration-based (ACC-based) gesture control is usually studied as a supplementary interaction modality. It is well suited to distinguish noticeable, larger scale gestures with different hand trajectories of forearm movements. With ACC-based techniques some subtle finger or hand movement may be ignored whereas electromyogram-based (EMG-based) gesture recognition techniques use multi-channel EMG signals which contain rich information about hand gestures of various size scales. Due to some problems inherent in the EMG measurements, including the separability and reproducibility of measurement, the size of discriminable hand gesture set is still limited to 4-8 classes. In order to realize a natural and robust gesture-based HCI system, the selection of input hand gestures that are well discriminable from each other is of crucial importance. Considering the complementary features of ACC- and EMG-measurements, we believe that their combination will increase the number of discriminable hand, wrist and forearm gestures and the accuracy of the recognition system.

This paper describes In IP Gaming we are proposing a system in which without using sensors and Devices, we are detecting the hand and gesture with Simple Web camera and performing the Image Processing technique in which using those gesture, we can play game on console. In Image Process Gaming, the motions are detected through a web camera. These images are then passed for the image processing. The techniques used for image processing are hand gesture detection, edge detection, thresholding, contour detection. Using OpenCV, which provides a library collection of functions for different image processing techniques, these input images can be processed and corresponding key strokes will be generated.

II. LITERATURE SURVEY

Motion capture and depth sensing are two emerging areas of research in recent years. With he launch of Kinect in 2010, Microsoft opened doors for researchers to develop, test and optimize the algorithms for these two areas. J Shotton proposed a method to quickly and accurately predict 3D positions of the body joints without using any temporal data. Key prospect of the method is they are considering a single depth image and are using a object recognition approach. From a single input depth image, they inferred a per pixel body part distribution.

Leyvand T discussed about the Kinect technology. His work throws light on how the Identity of a person is tracked by the Kinect for XBox 360 sensor. Also a bit of information about how the changes are happening in the technology over the time is presented. With the launch of Kinect, there is a sea change in the identification and tracking techniques. The authors discussed the possible challenges over the next few years in the domain of gaming and Kinect sensor identification and tracking. Kinect identification is done by two ways: Biometric sign-in and Session tracking.

A method to track fingertips and the centers of palms using Kinect was presented by Raheja. It applied thresholding on the depth of hand regions for segmentation. Then the palm was filtered and subtracted from the hand, so that only the fingers were left in the image. Under most situations when the hand was in front of the user, the fingers should be closest to the Kinect with the shallowest depth. Therefore, by determining the minimum depth, fingertips were found. The center of the palm was determined by finding the maximum of distance within the image of the hand. When fingers were extended, the accuracy of detecting fingertips was nearly 100% accuracy, and that of palm centers was around 90%. However this method did not attempt at gesture recognition. He proposed another approach using depth data provided by Kinect to detect fingertips. First, it found hand points by thresholding on depth elata, and then generated the convex hull containing the hand by Graham Scan. Fingertips were detected by calculating the angle between candidate points. After fingertips were found, the mouse clicking motion was recognized and tested on the popular game Angry Bird; that is, it recognized only one gesture.

1. EDGE DETECTION

Edge detection is one of the most commonly used operations in image analysis, and there are probably more algorithms in the literature for enhancing and detecting edges than any other single subject. The reason for this is that edges form the outline of an object. An edge is the boundary between an object and the background, and indicates the boundary between overlapping objects. This means that if the edges in an image can be identified accurately, all of the objects can be located and basic properties such as area, perimeter, and shape can be measured. Since computer vision involves the identification and classification of objects in an image, edge detections is an essential tool.

2. CANNY

The most significant new dimension to the canny algorithm is that it tries to assemble the individual edge candidate pixels into contours. These contours are formed by applying a hysteresis threshold to the pixels. This means that there are two thresholds, an upper and a lower. If a pixel has a gradient larger than the upper threshold, then it is accepted as an edge pixel; if a pixel is below the lower threshold, it is rejected. If the pixel's gradient is between the thresholds, then it will be accepted only if it is connected to a pixel that is above the high threshold.

```
void cvCanny(
const CvArr* img,
CvArr* edges,
double lowThresh,
double highThresh,
int apertureSize = 3 );
```

The cvCanny() function expects an input image, which must be grayscale, and an output image, which must also be grayscale.

3. THRESHOULD

```
double cvThreshold(
CvArr* src,
CvArr* dst,
double threshold,
double max_value,
int threshold_type
);
```

Frequently we have done many layers of processing steps and want either to make a final decision about the pixels in an image or to categorically reject those pixels below or above some value while keeping the others. The OpenCV function cvThreshold() accomplishes these tasks. The basic idea is that an array is given, along with a threshold, and then some-thing happens to every element of the array depending on whether it is below or above the threshold. The cvThreshold() function handles only 8-bit or floating-point grayscale source images.

III. IP GAMING SYSTEM

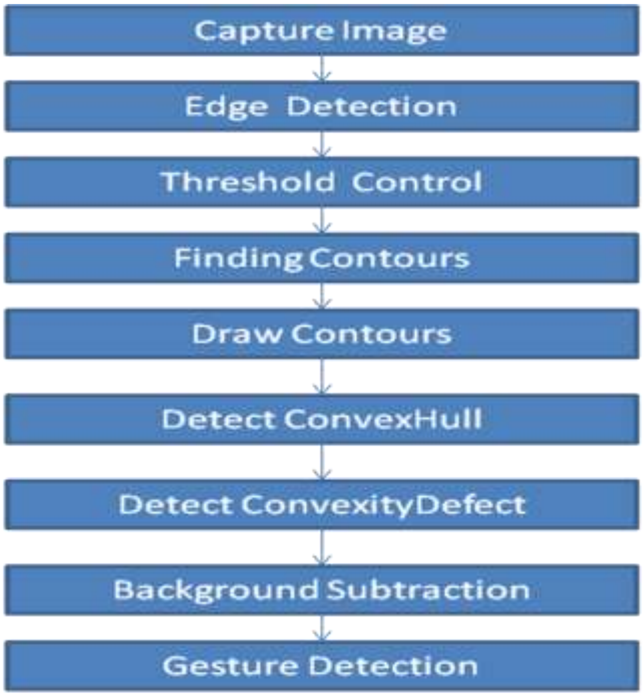


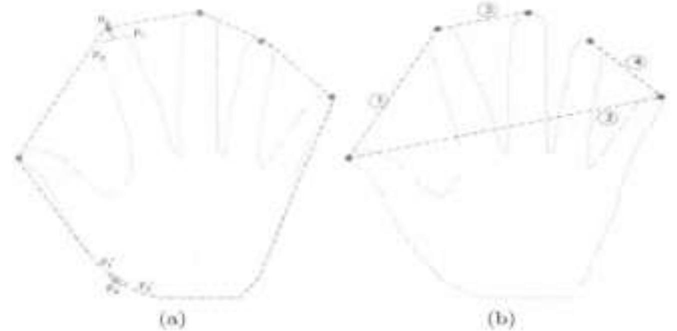
Fig 1. Sequence of System Architecture

1. Capture the images using web camera.
2. On applying Image Processing technique we will be able to detect the hands and Gesture and perform some console based operation.
   i. Edge Detection: Edges are the sharp black shadow surrounding the objects.
   ii. Threshold Control: for controlling sharpness of edges.
   iii. Finding Contours, contours are nothing but shadow areas of hand.
   iv. Set the proper beginning of the contours.
   v. Detect Convexity Defect in the picture. Defects are the points which are having thick edges.
   vi. Detect Convexity Defect ending points for the tip of our hand detection.
   vii. Draw Circles on the defects we obtained.
   viii. Save the Co-ordinates of the Defects obtained in each areas.
3. Trigger Image Capture.
4. Background Subtraction is done for clearing background.

IV. ALGORITHM FOR HAND DETECTION

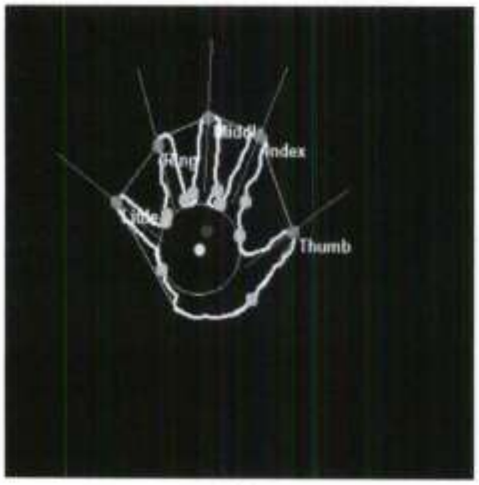
Finger Identification Algorithm

1. The first and easiest step is to identify the thumb and the index finger, since the distance between them is the largest among all neighboring fingers.
2. The little finger is identified as the farthest finger away from the thumb; meanwhile the middle finger is identified as the closest one to the index finger.
3. The remaining one is the ring finger. The process of detecting hands and identifying fingers are performed every time when the input data source changes. If the same object still exists in the current frame with some transformation compared to the previous frame, all properties of this object is mapped from the previous frame to the current frame; otherwise the disappeared object is collected by an unmapped item collector.



(a) (b)

Fingertip Detection, The distance between *Po* and the line made by *PI* and *P2* is apparently larger than that of p~ and the line made by p~ and *p;*. Thus a threshold distance could be used to differentiate fingertip points and non-fingertip points.



(a)

V. SYSTEM ARCHITECTURE

The architecture diagram consists of:

A player

System

OpenCV Library

Java Native Interface

Background Subtraction

Hand Gesture Detection

Key Stroke Generation

The given diagram has a single player playing a game through hand gestures using a web camera. The motions are detected using web camera and passed to the system for processing.

The system consists of OpenCV and Java Application, using these Hand Detection and Background subtraction is done. This processed image in the checked in which area it has occurred and its corresponding key event is called for action to be performed in the game. These steps are repeated till the player is playing the game.

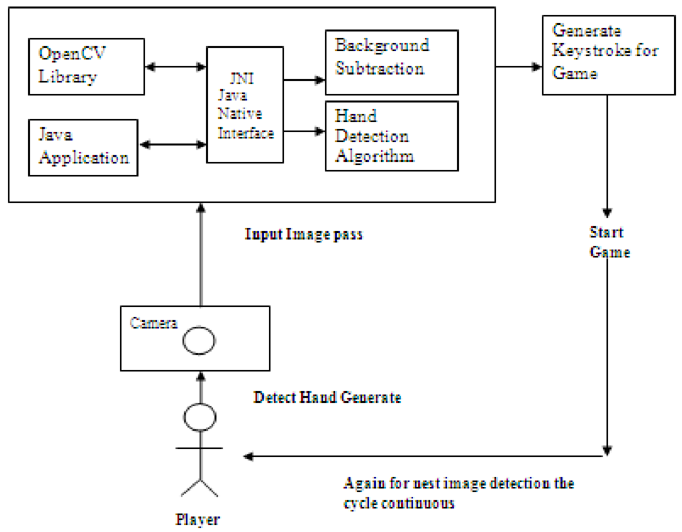


Fig 2. System Architecture

VI. CONCLUSION

In this paper, we have discussed how using camera can be used for Detection hand gestures and can be applied to any game control. We are using camera as a detecting device as well as input device for Augmented Reality System. The proposed system helps reduce the burden on experts to look into few regular activities. Instead, they can use our system for such activities. Also, the work simplifies the documentation process. The supervisor can keep track of current status of activity from his desk. Also, stepwise verification is possible as the system keeps track of each step. Through the introduction of our system, we will bring new opportunities for mechanical engineering based companies to use Augmented Reality for simplification of their complex tasks. This will add new dimensions to the conventional way of maintenance activities.

VII. REFERENCES

[1] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore,A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 1297 –1304, June 2011.

[2] T. Leyvand, C. Meekhof, Y.-C. Wei, J. Sun, and B. Guo. Kinect identity: Technology and experience. Computer, 44(4):94 –96, April 2011.

[3] S. Henderson and S. Feiner. Exploring the benefits of augmented reality documentation for maintenance and repair. Visualization and Computer Graphics, IEEE Transactions on, 17(10):1355 –1368, Oct. 2011.

[4] S. Henderson and S. Feiner. Augmented reality for maintenance and repair (armar). Technical report, AFRL-RH-WP-TR-2007-0112, United States Air Force Research Lab,, Jul 2007.

[5] Saket Warade, Jagannath Aghav, Petitpierre Claude, Sandeep Udayagiri. Real-Time Detection and Tracking with Kinect. International Conference on Computer and Information Technology (ICCIT'2012) June 16-17, 2012, Bangkok

[6] R. Owens, "Lecture 6", Computer Vision IT412, 10/29/1997. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT6/node2.html

[7] S. Price, "Edges: The Canny Edge Detector", July 4, 1996. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MARBLE/low/edges/canny.html

[8] Ehsan Nadernejad, Sara Sharifzadeh, Hamid Hassanpour. Edge Detection Techniques: Evaluations and Comparisons. Applied Mathematical Sciences, Vol. 2, 2008, no. 31, 1507 – 1520.