International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 11 | Nov -2017 www.irjet.net p-ISSN: 2395-0072

# Gesture Recognition Based Video Game Controller

**Sudarshan Pol[1], Pratik Pagade[2], Deepak Pati[3]**

[1, 2, 3] *Department of Computer Engineering, Ramrao Adik Institute of Technology, Navi Mumbai,Maharashtra, India*

---------------------------------------------------------------------***---------------------------------------------------------------------

**Abstract -** *Gesture Recognition has been used in many Fields like sign language Translation, Wireless Controllers etc. It can also however be used as a medium to control desktop Applications or Video Games or as an Alternative to the interactive Controllers Provided by VR Headsets. It Combines many Different Challenges such as Complex Backgrounds, Illumination, Object Distance. In this Paper; we Propose a System Which Detects the Hand Determines its Center and Extremities, tracks its position, analyses the variation in movements and Recognizes the Gesture. Results Pertain that after Calibration at any Fixed Distance the System Can Recognize over 16 Gestures on a Single Hand with an ~ 91% success Rate Varying only with Video Input Quality.*

***Key Words***: **Gesture Recognition, Gaming, Gesture Controller, Background Subtraction, Image Processing**

## 1.INTRODUCTION

The Development of Virtual Reality Headsets and Handheld Peripherals like HTC VIVE to play games and Interact with Virtual Environments has become quite Popular and as an alternative to the Handheld controllers' direct use of hands as an input device is an attractive method for providing natural Human Computer Interaction to any System. Since hand gestures are natural form for communication it does not negatively affect the health of the used and is less cumbersome than continuously holding a handheld controller and moving it around. This paper introduces a Gesture recognition system which recognizes 'Dynamic Gestures' each of which is performed in a complex background. Our System Does not require the Use of a 'Marker' or Gloves but the implementation doesn't expressly prohibit it either. The System uses 2D video Input from any Source of Standard Definition Quality; the process involves detecting the hand location tracking its trajectory and tracking its hand location variations Then the Obtained motion information is matched to existing gesture states and when matched correctly an appropriate call is triggered.

## 2. SYSTEM MODEL

The system works in four stages, where each previous step provides instructions for the next. The Gestures are Classified based on the distance and angle of each finger from the center of the palm.

The System requires an Inexpensive approach where even a simple webcam and a low-end processor can suffice for the system to work efficiently.

| | | | |
|---|---|---|---|
| Image Acquisition | Background Subtraction | Feature Extraction | Gesture Extraction |

The System approach consists of the following stages

1. ***Image Acquisition:*** A stream of Realtime Images captured from the Webcam.
2. ***Background Subtraction:*** Processing the input image to Separate the Foreground Object (Skin Pixels) from the background and refining the same by utilizing noise removal techniques.
3. ***Feature Extraction:*** Identifying the Outline of the Palm, calculating its center, identifying Tips of the Fingers and tracking the motion of the hand is done in this step.
4. ***Gesture Classification:*** The Position of the fingers relative to the center of the palm is calculated and matched to existing data if a match is detected the corresponding input command is called.

## 3. PROPOSED SYSTEM

The isolation and Extraction of Relevant Data is done using a variety of Algorithms the steps and Techniques are Detailed as follows:

### 3.1 Background subtraction

We use Hue Saturation Value (HSV) color space which is useful in specifying color properties numerically as well as better able to more closely align how human vision is perceived as compared to the RGB model. Hue is concerned with the key color (like red, green or yellow) of a pixel; saturation is the concentration of color in a pixel in comparison to the brightness of that color. The "intensity", "lightness" or "value" is related to the color luminance. Hue of the human skin can be used as a decision parameter. The System Requires an early input of HSV Skin Color so as to isolate it from the image. The Following

Algorithm is used to isolate of Foreground Object (Hand) using HSV color Model procedure [11].

**Algorithm1: Background Extraction**

*Input Parameters*
*Image: Input Image*
*$Thresh_1$ and $Thresh_2$: Threshold.*

*Input Parameters*
*Image: pre-detected Skin Pixel color.*

*Procedure*
- *Read the image header, head*
- *for i=1 to head.height*
- *for j=1 to head.width*
- *Read pixel Color*
- *max=max(Color.Red,Color.Green,Color.Blue)*
- *min=min(Color.Red,Color.Green,Color.Blue)*
- *∂ = mx – mn*
- *If (mx=Color.Red) then*
- *h = (Color.Green – Color.Blue)/δ*
- *Else If (mx=Color.Green) then*
- *h =2 + (Color.Blue – Color.Red)/δ*
- *Else*
- *h =4 + (Color.Red – Color.Green)/δ*
- *End if*
- *h = h * 60*
- *If (h<0) then*
- *h = h + 360*
- *End if*
- *If ($Thresh_1 \le h \le Thresh_2$) then*
- *Identify pixel as skin*
- *End if*
- *End*

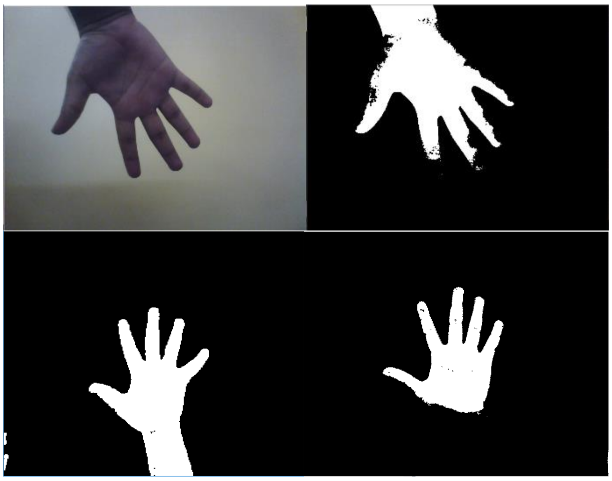

**Fig-1**: Background Subtracted Image Full Hand and Palm Isolated.

***Noise Removal:*** In Order to Obtain a proper Silhouette of the hand the isolated image is passed through Noise filters. The Following Filters are used.

***Erosion Filter**:* Erosion of an image involves a structuring element s. If there is an Image f, the erosion of f will be denoted as $f \ominus s$. The output of this function generates a new image $f_2$. Such that, $f_2 = f \ominus s$, when structuring element s fits into the image f it replaces all the pixels of co-ordinates (x, y) with ones.

That is $f_2(x, y) = 1$ if Structuring Element 's' fits Image 'f' else 0. This process is repeated for all pixel's co-ordinates (x, y). Images having structuring square elements with a small size for e.g. a 2x2 or 5x5 are minimized by removing a whole layer of pixel from inner boundary as well as outer boundary.

Erosion applied on larger structuring elements on the other hand have a larger effect, results of these erosions are quite similar to the results of multiple erosion of same image but with smaller structuring elements having same shape.

If s1 and s2 are given as two structuring elements, both of them identical in shape while s2 is twice as big as s1, then,

$$f \ominus s2 \approx (f \ominus s1) \ominus s1.$$

***Dilation Filter**:* Dilation of an image involves a structuring element s. If there is an Image f, the erosion of f will be denoted as $f \oplus s$. The output of this function generates a new image $f_2$ such that $f_2 = f \oplus s$, when structuring element s hits the image f it replaces all the pixels of co-ordinates (x, y) with ones.

That is $f_2(x, y) = 1$ if Structuring Element 's' hits Image 'f' else 0. This process is repeated for all pixel's co-ordinates (x, y). In Erosion as we saw a layer of pixel is removed from the inner and outer boundaries, In Dilation we have an opposite effect to Erosion. Instead of removing a layer it adds a layer of pixels to both inner as well as outer boundaries of the regions.

We have to keep in mind that the size and shape of the structuring element affects the results of both Erosion and Dilation. Dilation and Erosion are two operations that have opposite effects. Take $f^c$ be the complement of image f. That is image produced by changing 1 to 0 and 0 to 1 of image f. So, the dual operations can be written as,

$$f \oplus s = f^c \ominus s_{rot}$$

**1** in which case we'll consider it a "White" pixel and it will be part of the pattern

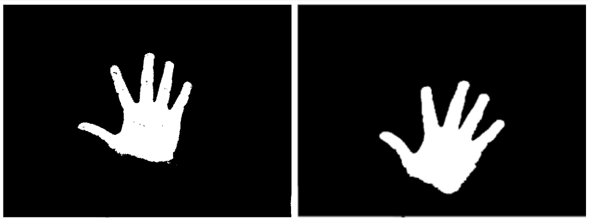

**Fig -2**: Before & After Erosion and Dilation

### 3.2 Feature Extraction

Contour Tracing is used to extract the boundary lines of a digital images. This technique is also known as border following or boundary following. This technique works as, there are two values background or foreground depending on the position of a pixel a value is assigned.

*OR*

The boundary of any given pattern, is the set of border pixels of that particular pattern. For Reference if we take a

Square matrix, there will be 2 kinds of border (or boundary) pixels: 4-boundry pixels and 8-boundary pixels. A white pixel is shares an edge with at least one black pixel it is considered a 4-border pixel. Furthermore, if a white pixel shares an edge or a vertex with again at least one black pixel is considered as an 8-border pixel. (An 8-

*0,* in which case we'll consider it a "Black" pixel and it will be part of the background

border pixel can or cannot be a 4-border pixel but a 4-border pixel can also be called as 8-border pixel.). In some application it is not enough to just identify the boundary pixel of a pattern to extract the contour. We also need an ordered sequence of the boundary pixel through which we can retrieve the general shape of the pattern.

**Algorithm2: Contour Extraction**

*Input Parameters*
*Image: Input Image*

*Input Parameters*
*Image: Background Subtracted Image*

*Procedure*

- *Read the BGsubstractedImageData*
- *Read the image header, head*
- *Display_contour = '1'(Yes)*
- *for i=1 to head.height*
- *for j=1 to head.width*
- *Read current pixel*
- *If previous_pixel.color = current_pixel.color*
- *previous_pixel.color = current_pixel.color*
- *Else if current_pixel.color = "White"*
- *Mark Pixel-1 as Border Pixel*
- *Else*
- *Mark Pixel as Border Pixel*
- *End if*
- *If display_contour = '1'*
- *Change border pixel color to 'Yellow'*
- *End if*
- *End*

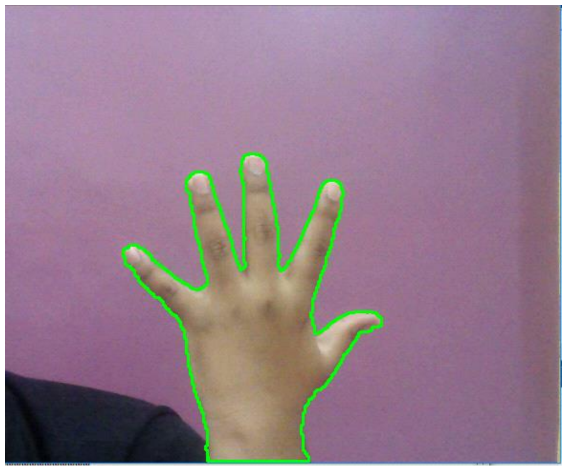

**Fig -3:** Contour Extracted

### 3.3 Gesture Classification

For the sake of simplicity, we assume that no three points are collinear, and hence are in general position. The algorithm can be easily modified to deal with points lying on a single line, also including if it should consider only vertices of the convex hull or all points that lie on the convex hull and not just the extremes. The implementation must deal with Outliers; As to when the convex hull has only 2 or less vertices, as well as with the issues of limited precision, both of computer Processing and input Parameters. [10], convexity defect is a cavity in an object (a blob), separated out from an image; Meaning the area does not belong to the given object but is located inside of its outer boundary.

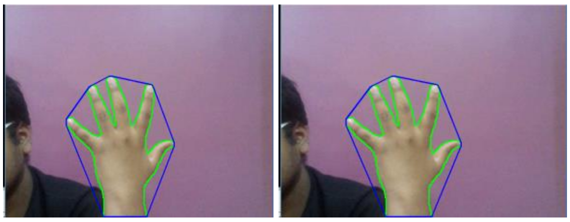

**Fig -4:** Convex Hull and Convexity Defects Obtained

Using these parameters, we can Classify certain finger positions as Particular gestures, the system can be programmed to identify basic number of fingers based on the convexity defects identified Palm tracking can be programmed to be used as a cursor input.

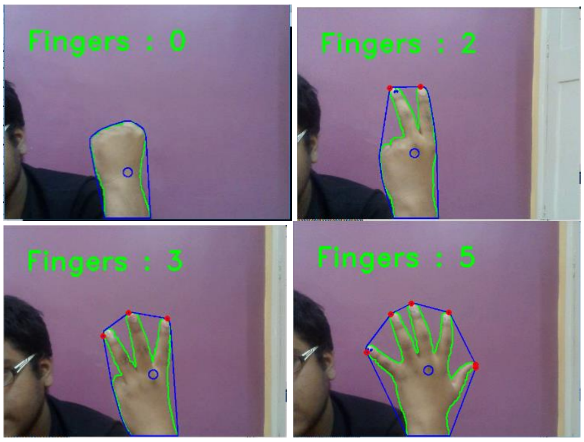


**Fig -5**: Finger Counting

## 4. RESULTS

The Proposed System was implemented using an Optimized Visual C++ and OpenCV Auxiliary Library. We were able to test multiple functions of the system with 9/10 of the Gestures being properly Classified with efficiency increasing if a higher quality video input was used. We used a basic Gameboy Advance Emulator; Visual Boy Advance to Test out our Game Control we used a Homemade ROM for our Tests.



**Fig -6**: Using Gestures as Input to play on the Visual Boy Advance

## 5. CONCLUSIONS

The System is a working prototype of a gesture based controller which can be further developed in the future to adapt to a wide range of applications, we have implemented a real-time version of the system which can be used to provide input to and Operating system and thereby allowing it to be used on multiple platforms with no special hardware other than a camera input. The system works under multiple degrees of Background Variance and Illumination with more than 91% success rate and can be used to control Any Application using programmable inputs.

## ACKNOWLEDGEMENT

A Big Thanks to our Professor and Project Guide Ms. Harsha Saxena for Helping and Guiding us throughout this Project.

## REFERENCES

[1] Yuanyuan Zhang ; Sch. of Inf. & Commun. Eng., Beijing Univ. of Posts & Telecommun., Beijing, China ; Xiaojun Jing ; Xiaoyong Wang ,An approach to improve the ZS thinning algorithm

[2] Background Subtraction Algorithm with Post Processing in Video Surveillance (1)Ms. Rucha D. Pathari, (2)Asst. Prof. Sachin M. Bojewar. International Journal of

Engineering Research & Technology (IJERT) Vol. 3 Issue 1, January - 2014 IJERT ISSN: 2278-0181

[3] Yannick Benezeth, Pierre-Marc Jodoin, Bruno Emile, H´el`ene Laurent, Christophe Rosenberger, Comparative study of background subtraction algorithms.

[4] ZhengjieWang, YuanZhao, JifenZhang, YinjingGuo, Research on Motion Detection of Video Surveillance System, 3 rd International Congress on Image and SignalProcessing, vol.1, pp. 193-197, October 2010

[5] Robust techniques for background subtraction in urban traffic video by Sen-Ching S. Cheung and Chandrika Kamath.

[6] R. E. KALMAN Research Institute for Advanced Study,2 Baltimore, Md, , A New Approach to Linear Filtering and Prediction Problems.

[7] System Science, Engineering Design and Manufacturing Informatization (ICSEM), 2010 International Conference.

[8] Chris Stauffer and W.E.L Grimson ,The Artificial Intelligence Laboratory Massachusetts Institute of Technology , Adaptive background mixture models for real-time tracking.

[9] Jae-Ho Shin, Jong-Shill Lee, Se-Kee Kil, Dong-Fan Shen, Je-Goon Ryu, EungHyuk Lee, Hong-Ki Min, Seung-Hong Hong Dept. of Electronic Eng., Inha University, Korea, Hand Region Extraction and Gesture Recognition using entropy analysis

[10] Charbel Fares, Holy Spirit University of Kaslik, Lebanon , Convex Envelope Generation Using a Mix of Gift Wrap and QuickHull Algorithms

[11] Comparative Study of Statistical Skin Detection Algorithms for Sub-Continental Human Images, M. R. Tabassum, A. U. Gias, M. M. Kamal, H. M. Muctadir, M. Ibrahim, A. K. Shakir, A. Imran, S. Islam, M. G. Rabbani1, S. M. Khaled, M. S. Islam, Z. Begum Institute of Information Technology, University of Dhaka.
https://sourceforge.net/projects/vba/ Visual Boy Project link
https://en.wikipedia.org/wiki/Gift_wrapping_algorithm.
https://en.wikipedia.org/wiki/Background_subtraction