2014 13th International Conference on Control, Automation, Robotics & Vision
Marina Bay Sands, Singapore, 10-12th December 2014 (ICARCV 2014)

SHIMPG: Simple Human Interaction with Machine using Physical Gesture

Md. Asif Ur Rahman*, Md. Saef Ullah Miah*, M. Abrar Fahad* and Debajyoti karmaker†
Computer Vision Lab
Department of Computer Science
American International University-Bangladesh
{rahman.m.asifur, md.saefullah & abrarfahad2020}@gmail.com, d.karmaker@aiub.edu

***Abstract*—with all the advances accomplished in the computer gaming world progressively emerging more equipment's into the market in order to communicate with computers. Most of these sophisticated devices are for targeted applications or special purpose and also very expensive. With this in our mind we propose a Simple Human Computer Interaction with Machine using Physical Gesture Framework which uses regularly used equipment like low resolution camera and also has a very robust uses like simple games and smart house. The main goal of this paper is to outline a mechanism of computer vision for controlling any application or hardware. For this purpose we propose a generalized framework which can be seamlessly integrated with any networked controlled application as it works as a separate engine which can be easily integrated. In addition we have prototyped a simple application using this engine, image processing and 3D model augmentation.**

***Index Terms*—Computer Vision, Image Processing, Hand gesture recognition, Skin Segmentation.**

I. INTRODUCTION

This paper aims to design and build a man-machine interface framework for robust use. As a prototype we have used a video camera to interpret some predefined gestures and generate gesture wise events to interact with the system. We have named it "Simple Human Interaction with Machine using Physical Gesture (SHIMPG)".

Currently keyboard and mouse are the primary interfaces between man and computer. In other areas where 3D information is required like computer games, robots and design other systems like smart house, smart TV or collaborative environment [1,2] uses vision and sound as these are human's main communication medium, hence a man machine interface will be more intuitive if it uses a great deal of audio and visual recognition unlike the auditory system. The promise of computer vision for human-computer interaction (HCI) is great. Vision-based interfaces would allow unencumbered, large-scale spatial motion that offers many possibilities of defining new forms of interaction [3]. The visual system is more suitable for chaotic environment. That is why we have chosen visual system for our prototype application. The amount of computation required to process gestures is much greater than that of the mouse and keyboard process; however at present days standard personal computers are fast enough to compute this.

A gesture recognition system can be used in any of the following areas:

1) Man-machine interface: using hand gesture to control the computer mouse and keyboard functions or to control smart house.
2) 3D animation: rapid and simple conversion of hand movements into 3D computer space for the purpose of computer animation.
3) Visualization: visual examination of any object in 3D space by rotating the hand in space.
4) Computer games: using the gestures to interact with computers would provide more ease of access.
5) Control of mechanical system: using the hand gesture a remake manipulation can be controlled like robotic arms.

In order to detect any gestures data about the gesture making part will have to be collected. In our case we are using hand for creating gestures. This data collection process can be done in two ways:

1) A glove with sensors attached that measures the position of fingers.
2) An optical method.

We have chosen the optical method. Since it is more practical, cost effective and less likely to be damaged through use. For simplicity of recognition part we have chosen the Convex Hull method. The next step is to process the differentiated data and integrate it with the gesture generated events.

II. SYSTEM ARCHITECTURE

We have used the background subtraction and skin segmentation for hand detection and for recognizing the gestures we have used the Convex Hull method [5,6,7]. After the recognition part we have generated gesture wise events and used them to interact with the machine, in our case we have developed the mouse functionalities and we have also implemented the whole interaction to control an augmented reality based game through the RPC[8]. The image below will provide a more clear view of the overall system.

978-1-4799-5199-4/14/$31.00 ©2014 IEEE

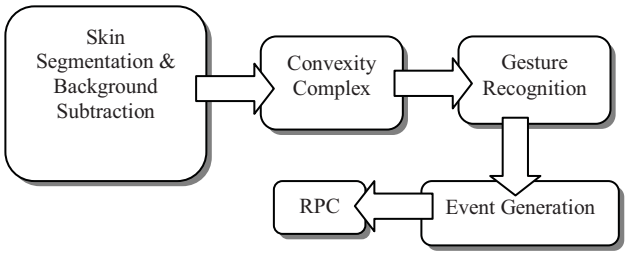

Figure-1: System Process Flow.

The system works within a few simple steps. Steps are listed below.

1) Put the hand in front of the camera.
2) Then the Hand detection and tracking system tracks and detects the hand.
3) Next Convex Hull model unit is activated to match the gestures that are predefined in the system.
4) After the gesture is recognized gesture-wise events are generated.
5) RPC's called to fulfill the events.

A. Hand Detection

1) Background Subtraction

For any human computer interaction system it is very important to capture and analyze every frame for the gesture creation skin area. To find the perfect skin area we need to differentiate the skin area from the background of each frame. Removing the background part from each frame is called background subtraction [9,10]. Without background subtraction it is impossible to make the system more smooth and performance worthy. For system performance and smoothness we need the background subtraction. In our system we have used the averaging method. We have computed the difference between the current frame and background model image that is we also computed background model image. We have also used certain Threshold to avoid ghosting problem. We have calculated weighted sum of input image I and accumulator R so that R becomes a running average of frame sequence:

$$R(x,y)=(1- \alpha)\bullet R(x,y) + \alpha \bullet I(x,y) \text{ [if } mask(x,y)!=0 \text{]}$$

Where α regulates update speed (how fast accumulator forgets about previous frames).

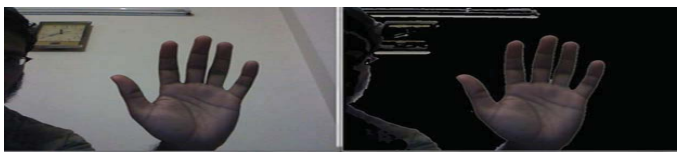

Figure-2: Background Subtraction.

for t = 1 to 30 do

a. I(x,y) <- capture image
b. Calculate running average R using following equation
   $R(x,y)=(1- \alpha)\bullet R(x,y) + \alpha \bullet I(x,y)$ [if mask(x,y)!=0]……..(i)
   //Where α regulates update speed (how fast
   //accumulator forgets about previous
   //frames).

end for

2) Skin Segmentation

Segmentation, in the area of image processing, means one tries to extract the parts of the image that satisfies pre-determined conditions. Segmentation can be used in pattern recognition, medical imaging etc., where one or more parts of an image is interesting, not the entire image. We have separated segmentation into pixel-wise segmentation then we have applied some morphological operation to reduce noise from the segmented image. Pixel-wise segmentation methods consider only one pixel at a time without regard to surrounding pixels [10,11]. We have used static thresholds segmentation method for our purpose.

3) Static thresholds

The simplest segmentation is static threshold. Its basic function is to check if the input is larger than, or in some cases smaller than, a threshold value. When working with controlled environments, this is possible and very predictable. In uncontrolled environments however, the input is often too complex for a system using static thresholds. If the input is an image, a controlled environment would be equivalent to the same place, the same objects and the same lighting conditions. If anything is changed, the risk is that the input value is changed and the threshold value is too large or too small. There was a lot of discussion in forums and some research that suggested that HSV is better for skin extraction than RGB as it is more consistent among different lighting and also different ethnicities. It turns out that flesh has distinct hue value, which allows separating the hand from the rest of the image by simple thresholding [12].

Webcams are known for low image quality (especially in low light conditions), so in order to combat color noise, we ran Gaussian smoothing before converting to HSV color space. Before comparing we had to get the data into HSV color space. We have converted the color model for the data from RGB to HSV.

As the pixel data is stored in arrays different places so first pixel in 3 channel image is stored at locations 0, 1 and 2 and the second pixel is at 3, 4, 5 and so on. The data after conversion from BGR to HSV is in that particular order. The pseudo code for the calculations was:

*Hue = (int) ((float) PIXEL_ARRAY[0] / 180 * 360)*
*Saturation = (int) ((float) PIXEL_ARRAY[1] / 255 * 100)*
*Value = (int) ((float) PIXEL_ARRAY[2] / 255 * 100)*

As we can see the H, S and V values are taken from the array in that order and divided by maximum value for that field (180 as its 360 divided by 2 and 255 because it's the maximum value for the field). After that it is multiplied by the HSV model specification values (H is 360 degrees and S/V are percentages). We used dynamic min and max values on hue and saturation to define the values on runtime and tweak then with default value on start. The method produced a mask of possible skin. As was pointed out by the research [17] the human skin hue values are near the red color range so they can be near 0 and 360 degrees on hue. That makes them harder to test. That's why we shifted the hue values by 90 degrees so the hue range is now between 70 and 140. These values are averages from couple of proposed values in other articles.

4) Morphological operation

Morphological image processing is a set of tools that are useful when working with regions. Dilation, erosion and pruning are examples on morphological operations that are used for pre- or post-processing. Gray scale images can be used for morphological image processing but the most common method is to use binary images.

5) Dilation and erosion

Dilation and erosion are two fundamental morphological operations. Dilation means to grow a region and erosion means to shrink a region. A structuring element is traversed through the image and the shape of the structuring element is what determines the effect of the dilation or the erosion. We have used 3×3 rectangular structuring element for erosion and dilation.

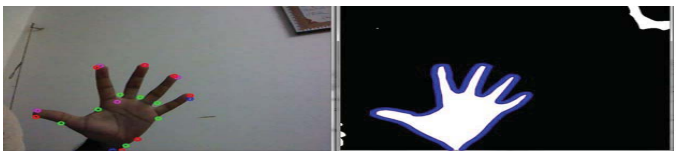

Figure-3: Skin Segmentation.

B. Gesture Recognition

We have used the convex hull method for recognizing the gestures. We have some predefined gestures set in our system, which can be recognized by the count of convex of the hand surface. We implemented the convex hull as it is a faster and accurate method to recognize any gestures made by hand and it is also a more memory efficient method.

Figure-4: Gesture Recognition.

C. Remote Procedure Call (RPC)

As our goal is to control another application from a computer to another computer or device (Mobile device like smart phone) from our application therefore we used RPC (Remote Procedure Call). It is a technique to call procedure from one program to another program, which can be located in the same computer or in the network computer. In our program there are two parts for RPC one is server and another is client. The client part will take the input from the keyboard and it will catch the key event and encode it with a predetermined integer value. Then we called the Remote Procedure with the encoded integer value as the parameter and this procedure also have a return after successful execution to release the keyboard key forcefully. On the server side the program will decode the parameter value and create a key press and release event with a certain time interval and return the corresponding key value to the client. This will enable the client computer to control the server computer. We may also use the mouse event and position by calling the remote procedure with the extra parameter of coordinate of mouse position. Here if the screen regulation of two computers is not same then the screen regulation should be adjusted to get the correct position on another computer. To control the device like smart phone (Android, iPhone etc.) or control the computer from these devices is possible using the XmlRpc.

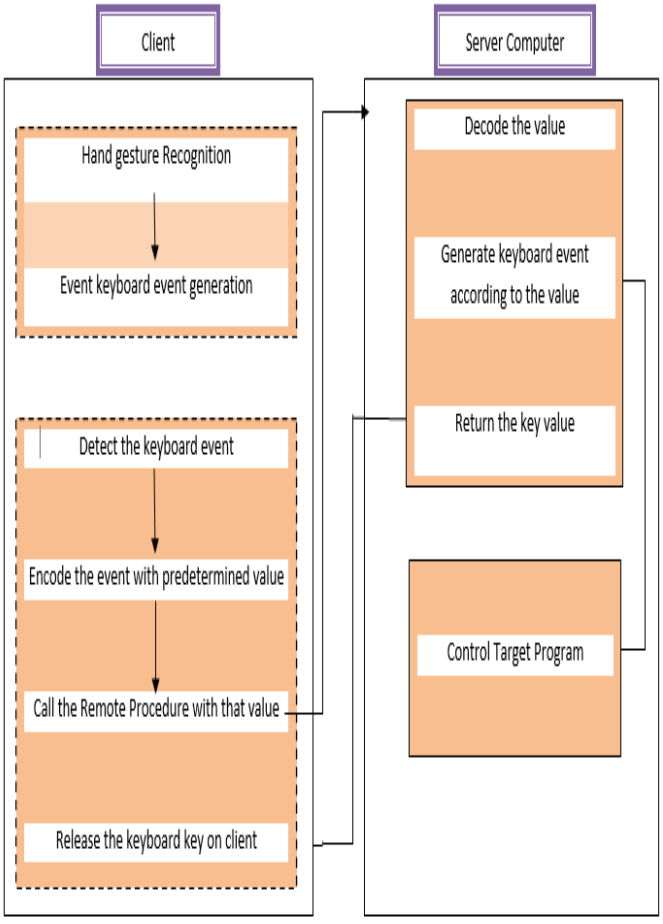

Figure-5: Overview of RPC in SHIMPG.

D. Mouse Functionalities

As one of our goals we have implemented the Mouse Functionalities for both a local machine and a remote machine using the RPC. In our system we have designed the mouse movement along with the movement of the hand and gesture one for left click and gesture two for right click. In more specific way, the cursor moves in the direction of the hand movement and if number one (1) is made with finger it will produce a left click and two (2) will produce a right click mouse event. An Illustration is provided for the gestures required for mouse functionalities.

AUGMENTED REALITY BASED GAME CONTROL

Augmented reality (AR) is a term for a live direct or an indirect view of a physical, real-world environment whose elements are augmented by computer-generated sensory input such as sound, video, graphics or GPS data. It is related to a more general concept called mediated reality, in which a view of reality is modified (possibly even diminished rather than augmented), by a computer. As a result, the technology functions by enhancing one's current perception of reality. By contrast, virtual reality replaces the real world with a simulated one. [18]

With our system we have implemented a tracking based Augmented Reality game. To develop the game we have used ARtool Kit. With the ARtool kit we have designed the game that recognize some patterns and create a 3d model according to that pattern. For the pattern recognition we have used the

Web camera and some predefined patterns. In the game we have designed some 3d models of pillars and a main character. That main character can be controlled with keyboard's arrow key, and with our system we have controlled the character with the hand movement. We have implemented it with the help of two computers. In one computer we have deployed our system and in other computer the game is deployed. In front of the SHIMPG enabled computer we have made the hand movement and on the other computer the game has been controlled with this hand movement, here the computers were connected with network and the hand movement events have been implemented via RPC. An illustration of the game face is provided below

E. Data Acquisition

1) Camera Orientation

For our system camera should be placed in such a way where the hand movement can be easily viewable by the camera, thus camera can track the hand and the hand movement. An ideal scenario can be the built in camera position of any laptop or an external web camera placed on the middle of the monitor.

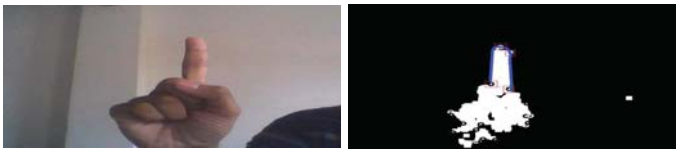

Figure-6: Gesture for Left Mouse

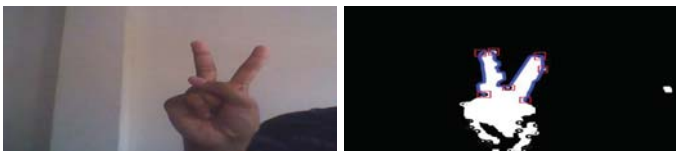

Figure-7: Gesture for Right Mouse

Pseudo code for gesture recognition from background-subtracted model is furnished bellow.

for each $I(x,y)$<- capture image do

a. I_{bgs} <- *I and R*
b. Update R using equation (i)
c. $I_{segmented}$<- Skin segment I_{bgs}using algorithms mentioned in [1]
d. Perform some morphological operation on $I_{segmented}$
e. C_{all}<- Find all contours from $I_{segmented}$
f. A_{big} <- *null* and C_{big} <- *null*
g. for each C_n <- C_{all}do
   i. A_n<- *calculate area of* C_n
   ii. if $A_n > A_{big}$ then
1. A_{big} <- A_{big} *and* C_{big}<- C_n
h. end for
i. Calculate convex hull H of C_{big}
j. D_{all} <- find convexity defects of H
k. Calculate min rectangle area region $A_{min}(C_{x,}$ $C_{y,}W,H)$of C_{big}
l. *gestureNo <- 0*
m. for each D_n<- D_{all}do
   i. if $std(abs(A_{min}(C_x) - D_{ns}(x))) > th_x$ && $std (abs(A_{min}(C_y) - D_{ns}(y))) > th_y$ && $sqrt(pow(abs(D_{ns}(x)- D_{nd}(x)), 2) + pow(abs(D_{ns}(y)- D_{nd}(y)), 2)) > A_{min}(H)/ th_h$)then
   Where $D_{ns}(x,y)$ is defects starting point ,
   $D_{nd}(x,y)$ is defects depth point,
   th_x ,th_y is standard deviation threshold for x and y,
   ii. gestureNo ++
n. Raise gesture event according to the gesture number
o. end for

end for

Figure-8: Augmented reality game.

2) Lighting and Background Condition

For any HCI based system lighting and background condition is some important factors for the smooth operation of the system. For our system the lighting condition is defined in such way like there should not be a direct source of light facing to the camera that is much bright, in a word much bright lighting should be avoided and the background of the frame should not be as like as of the color of skin.

III. Experimental Results

1) The output of our system is quite good for such a simple construction.

2) The False Acceptance Rate (FAR) is too low to count it is near about 8% to 10%.

3) The Failure to capture rate (FTC) is near about 0% that means the system does not miss any frame containing hand movement that is being made as a normal speed of regular movement.

IV. Conclusion

Now- a -days more sophisticated and complex HCI based system are available which are much costlier and complex to use in everyday life and they are not available for home users to use it on their everyday computing.

Our goal was to design and develop a cost efficient, faster and easy to use system for everyday computing. Our system is not only faster and cheap it is more easy and flexible to use. The system is fully automatic and it works in real-time. It is fairly robust to background cluster. The advantage of the system lies in the ease of its use. The users do not need to wear a glove, neither is there need for a uniform background. Experiments on a single hand database have been carried out and recognition accuracy of up to 92% has been achieved.

There is a great scope for enhancing this system for future use like - Learning agent based lighting condition adjustment, that is a learning agent that will learn the lighting condition from each operation and after sometimes it will adjust the whole condition automatically and initialize the system with the ideal condition. User customizable gesture set, which is nothing but a database of gestures that can be created and used by the user. Implementation of Hidden Markov Model to recognize motion based gestures.

Using such a system in everyday life not only reduces the interaction time it is also exciting using this system. Moreover this system can assist the handicapped persons to use computer systems in their day-to-day life.

References

[1] Borchers, J.; Ringel, M.; Tyler, J.; Fox, A., "Stanford interactive workspaces: a framework for physical and graphical user interface prototyping," Wireless Communications, IEEE , vol.9, no.6, pp.64,69, Dec.2002 doi: 10.1109/MWC.2002.1160083

[2] B., et al.EasyLiving: technologies for intelligent environments. In Handheld and Ubiquitous Computing Second In ternational Symposium HUC 2000. 2000. Bristol, UK: Berlin, Germany : Springer-Verlag, 2000.

[3] P. Maes, T.J. Darrell, B. Blumberg, and A.P. Pentland. The alive system: Wireless,full-body interaction with autonomous agents. MultSys, 5(2):105{112, March 1997.

[4] Hand Gesture Recognition using Computer Vision - Ray Lockton, Balliol College, and Oxford University.

[5] Hand Gesture Recognition for Human-Machine Interaction - Elena Sánchez-Nielsen , Luis Antón-Canalís , Mario Hernández-Tejera.

[6] Real-Time Hand Tracking and Gesture Recognition System - Nguyen Dang Binh, Enokida Shuichi, Toshiaki Ejima.

[7] Hand Gesture Recognition for Human-Computer Interaction - S. Mohamed Mansoor Roomi, R. Jyothi Priya and H. Jayalakshmi, Department of Electronics and Communication, Thiagarajar College of Engineering, Madurai, Tamil Nadu, India.

[8] Birrell, A. D.; Nelson, B. J. (1984). "Implementing remote procedure calls". ACM Transactions on Computer Systems 2: 39. doi:10.1145/2080.357392.

[9] Real-time foreground–background segmentation using codebook model - Kyungnam Kima, Thanarat H. Chalidabhongseb, David Harwooda, Larry Davisa - Computer Vision Lab, Department of Computer Science, University of Maryland, College Park, MD 20742, USA

[10] Gesture Recognition with Applications – Oleksiy Busaryev, John Doolittle, Department of Computer Science and Engineering, The Ohio State University.

[11] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Third Edition, 2008

[12] Kendon, Adam. (2004) Gesture: Visible Action as Utterance. Cambridge: Cambridge University Press. ISBN 0-521-83525-9.

[13] Dave Marshall. 1999. Remote Procedure Calls (RPC). [ONLINE] Available at: http://www.cs.cf.ac.uk/Dave/C/node33.html. [Accessed 16 June 14].

[14] Noelle, S., "Stereo augmentation of simulation results on a projection wall by combining two basic ARVIKA systems," Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. International Symposium on , vol., no., pp.271,322, 2002 .doi: 10.1109/ISMAR.2002.1115108

[15] Thiago Chaves, Lucas Figueiredo, Alana Da Gama, Cristiano de Araujo, and Veronica Teichrieb. 2012. Human Body Motion and Gestures Recognition Based on Checkpoints. In Proceedings of the 2012 14th Symposium on Virtual and Augmented Reality (SVR '12). IEEE Computer Society, Washington, DC, USA, 271-278. DOI=10.1109/SVR.2012.16