# Robust Hand Gesture Recognition Algorithm for Simple Mouse Control

Vivek Veeriah J. and Swaminathan P. L.

***Abstract*—The main aim of Human Computer Interaction (HCI) is to research and develop new and simpler ways to interact with computers and many other devices as well. Hand Gesture Recognition is one such area of active research for computer scientists. In this paper, we discuss a new method for controlling the mouse movement with a camera. Our method is unique as it does not involve Fuzzy models, Hidden Markov Models, etc. for recognition. Instead we use simpler segmentation and recognition techniques for recognition of simple hand gestures.**

***Index Terms*—Human computer interaction, hand gesture recognition.**

## I. Introduction

Human Computer Interaction (HCI) is an interesting and active area of research. Many researchers and engineers involved in this field research and develop new and simpler ways to interact with computers. These new ways are not restricted for interaction with computers alone. Although the current methods we use to interact with the computers such as keyboards, mouse, touchscreen, light pens etc are sufficient for most of our purposes, some of them are quite costly whereas the others occupy more physical space.

Several Hand gesture recognition techniques already exist and most of them are based on Hidden Markov Models, Fuzzy Logic, Neural Networks, etc [1], [2], [3]. These methods provide accurate recognition of hand gestures but the computational cost required to achieve this is pretty high. Therefore, those methods are not robust enough for real-time implementation. To overcome this, we have developed a robust method for recognizing simple hand gestures which depend purely on the simple segmentation and techniques.

## II. Literature Survey

Many methods have been developed by several researchers for controlling the mouse movement using a real time camera. Most of them are not robust enough for real time implementation and all of them use ambiguous methods for making a click event of a mouse [4].

Pandit et al. developed hardware related approach for hand gesture recognition. This requires the user to wear data gloves with markers from which hand posture could be extracted. An approach developed by Chu-Feng Lien [5] used finger tips for mouse movement and actions. Another approach from Erdem used finger tracking for mouse control and the click was performed when the hand passed over a specified region [6]. A simpler method was developed by Park. The action of clicking of mouse was done by keeping a track of the finger tips [4]. Paul et al, used still another method to click. They used the motion of the thumb (from a 'thumbs-up' position to a fist) to mark a clicking event thumb. Movement of the hand while making a special hand sign moved the mouse pointer [4], [6].

## III. System Flow

Our paper was inspired by the work done by Asanterabi Malima et al and Park [4], [7]. They developed a finger counting system to control the motion of a robot. We have adopted their algorithm for segmentation and have improved their recognition algorithm which shows that the recognition algorithm in its improved version is robust for real time implementation. The process of the gesture recognition can be divided into two separate problems 1) Segmentation of hands 2) Noise removal 3) Recognition.

### A. Hand Detection

Robust hand detection is the most difficult problem in building a hand gesture-based interaction system. There are several cues that can be used: appearance, shape, color, depth, and context. In problems like face detection, the appearance is a very good indicator [7]. Since our paper mainly focuses on gesture recognition, it is not harmful to assume that the hand is the major portion in the image. Since the hand is the major part, it would be easy to segment it by using the segmentation techniques proposed by Albiol et all [2]. This method of segmentation is more related to human perception as our eyes could easily recognize the skin tone from its background. This classical method for segmenting the skin pixels sets upper and lower bound values using which the hand was segmented. It classifies noisy objects as skin; therefore noise removal of the segmented image is absolutely necessary. The images are resized to a fixed resolution before performing the recognition process. In our case, the images were resized to 640 by 480 as that was the resolution of the camera used.

### B. Noise Removal

As mentioned in the previous section, some parts of the background would also be segmented and these inhibit the process of recognition. So to obtain a perfect recognition it is necessary to remove these unwanted noise. To get a better estimate of the hand, we need to delete noisy pixels from the image. We use an image morphology algorithm that performs image erosion and image dilation to eliminate noise [4], [6].

Manuscript received August 25, 2012; revised September 26, 2012.

Vivek Veeriah J. and Swaminathan P. L. are with the Dept. of ECE, Coimbatore Institute of Technology, Coimbatore, India (e-mail: vivekveeriah@gmail.com).

Erosion trims down the image area where the hand is not present and Dilation expands the area of the Image pixels which are not eroded.

Mathematically, Erosion can be defined as,

$$A \ominus B = \bigcap_{b \in B} A_{-b} \quad [9]$$

Mathematically, Dilation can be defined as,

$$A \oplus B = \bigcup_{b \in B} A_b \quad [10]$$

In our paper, we performed erode function with a structure of 8 x 8 square element three times and dilate function with a structure of 6 x 6 square element three times. It could be seen that much of the background noise has been removed by erosion and dilation process.

### C. Gesture Recognition

The recognition process is done only for simple hand gestures which are necessary for controlling the movement of the mouse and simple clicking events. Therefore, this application does not require complicate and sophistic Markov models and neural networks. In the above cited papers [4], [7] the recognition process was done simply using the segmented image but in our paper, the gesture recognition technique involves a different technique.

Firstly, the largest contour from the image is extracted. In this case this would definitely be that of the hand. Then the centre coordinate of the hand would be calculated. The size of the hand would then be determined by drawing a circle increasing the radius of the circle from its centre coordinate. This algorithm for finding the size of the hand would terminate when it meets the first black pixel such that the maximum radius of the circle drawn would give an approximate estimate of the size of the hand. Now in order to recognize the finger tips we use the convex hull algorithm. The convex hull algorithm is used to solve the problem of finding the biggest polygon including all vertices. Using this feature of this algorithm, we can detect finger tips on the hand. We used this algorithm to recognize if a finger is folded or not. To recognize those states, we multiplied 2 times (we got this number through multiple trials) to the hand radius value and check the distance between the center and a pixel which is in convex hull set. If the distance is longer than the radius of the hand, then a finger is spread. In addition, if two or more interesting points existed in the result, then we regarded the longest vertex as the index finger and the hand gesture is clicked when the number of the result vertex is two or more [4]. Sometimes, there would be more than one vertices produced by the convex hull algorithm and this would often occur near the corners. To eliminate this, a check has to be performed whether they are false vertices or not.

Therefore, the circle with the radius greater than twice the size of hand is taken as a threshold and the vertices returned by the convex hull algorithm would be the tip of the fingers.

## IV. Mouse Control

Using the above gesture recognition technique, we implemented a small program for performing simple mouse actions. The actions performed were left click, right click, double clicking and scrolling.

### A. Left Click and Double Click

For performing the action of left click, atleast two convex hull vertices should be above the threshold area as calculated in the previous sections. The double-clicking occurs when the thumb moves 0 to 90 degree and back two times fast [2].

### B. Right Click

If we make the hand pose left clicking for 3 seconds, then the system calls the right clicking event.

### C. Scrolling

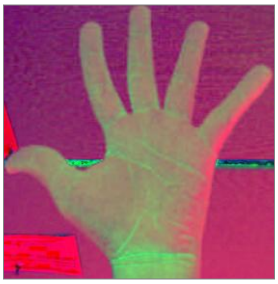
When two extreme fingers are pointed out, two convex hull vertices of large euclidean distance are recognised and this gesture is used for scrolling event.

## V. Experimental Results

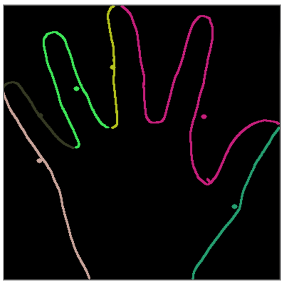
We tested all mouse tasks such that left click, right click, double-clicking and scrolling on Ubuntu.

The tested system is that Core i3, 4gb RAM, Ubuntu 12.04 LTS. Obviously the performance was lower when compared to the actual hardware mouse. Instead we tabulate the time taken for our recognition algorithm to recognize and perform the the above mouse actions and from which it could be seen that the algorithm is robust enough for real time implementation as the delay is negligible.

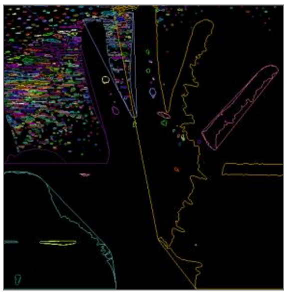
a) Skin Colour Segmentation



b) Contour extraction



c) After Convex Hull algorithm



| Results of | Reference Work[2] | This Work |
|---|---|---|
| Left Click | 1.10 | 0.97 |
| Right Click | 4.16 | 3.19 |
| Scrolling | 4.50 | 1.72 |
| Double Click | 2.60 | 2.77 |

## VI. Conclusion

In this paper, we discussed about the system which we had developed for performing simple operations of a mouse using a camera. However, this system developed using computer vision algorithms has some illumination issues. From the results, we can expect that if the vision algorithms can work in all environments then our system will work more efficiently.

This system could be useful in presentations and to reduce work space. In the future, we plan to use stereo vision techniques to obtain the depth information for more complicated hand gesture recognition.

## References

[1] A. Chaudhary *et al*., "Intelligent approaches to interact with machines using hand gesture recognition in a natural way: A survey," *International Journal of Computer Science and Engineering Survey (IJCSES)*, vol. 2, no. 1, Feb 2011.

[2] A. Albiol, L. Torres, and E. J. Delp, "Optimum color spaces for skin detection," in *Proceedings of 2001 Image Processing International Conference*, vol. 1, pp. 122-124, 2001.

[3] A. Pandit, D. Dand, S. M. Sabesan, A. Daftery, "A simple wearable hand gesture recognition device using iMEMS," *Soft Computing and Pattern Recognition, 2009. SOCPAR '09. International Conference*, vol. 4, no. 7, pp. 592-597, Dec. 2009.

[4] H. Park, "A method for controlling mouse movement using a real-time camera," 2012.

[5] C.-F. Lien, "Portable vision-based HCI - A real-time hand mouse system on handheld devices."

[6] A. Erdem, E. Yardimci, Y. Atalay, and V. Cetin, "Computer vision based mouse," *A. E. Acoustics, Speech, and Signal Proceedings. (ICASS). IEEE International Conference*. 2002.

[7] M. V. den Bergh *et al*., "Combining RGB and ToF cameras for real-time 3D hand gesture interaction."

[8] P. Kathuria and A. Yoshitaka, *Hand gesture recognition by using logical heuristics*.

[9] Filter for Unwanted Details in Image. [Online]. Available: http://en.wikipedia.org/wiki/Erosion_(morphology)

[10] Structuring Element Mathematical Morphology. [Online]. Available: http://en.wikipedia.org/wiki/Dilation_(morphology)

**Vivek Veeriah J. and Swaminathan P. L.** are undergraduate engineering students of Coimbatore Institute of Technology, India. They were introduced to the field of computer vision during an internship in Madurai and based on the internship, this human computer interaction application was developed. The authors would like to thank their staff and mentors for supporting them throughout their project.