Developing a Color Detection System for Hand Gesture-Based Control Using a Web Camera

Dr. Sophie Dubois [1], Mathieu Bernard [2] & Pauline Lefevre [3]
[1]Faculty of Sciences, University of Paris, Paris, France
[2]Department of Computer Science, University of Lyon, Lyon, France
[3]Department of Electrical Engineering, University of Grenoble, Grenoble, France

ABSTRACT

In this paper we presented an intangible hand gesture based computer mouse control system. Our approach uses a novel skin color segmentation technique to control mouse movement. The system uses morphological operations like structuring elements and blob counting. Our system can remove other skin like objects from the background. One of the key features of the system is its simplicity. We have successfully tested our system for an intangible interface between human hand and computer mouse without any complex processing. Various mouse operations like
cursor movements, right click and left click have been performed. The system has been implemented using webcam and MATLAB.

Keywords: Computer vision, Hand gesture, YCbCr, Skin color detection, blob, structuring element.

I. INTRODUCTION

Overview

Gesture is a form of nonverbal communication which involves movement of a part of the body, especially the hand usually to express an idea. In a human interaction, we make use of speech, gestures and body movements to convey information. The human-computer interaction which makes use of input devices such as keyboard or mouse for communication lacks natural communication between humans and machines. For this purpose, it is important to develop applications or devices that support intuitive communication. As computer technology continues to grow, the need for natural communication between humans and machines also increases. Although our mobile devices make use of the touch screen technology, it is not cheap enough to be implemented in desktop systems. Although the mouse is very useful for device control, it could be inconvenient to use for physically handicapped people and people who are not accustomed to use the mouse for interaction. The method proposed in this paper makes use of a webcam through which gestures provided by the user are captured, processed and the function related to that gesture is carried out. For example, a gesture “V” i.e. two fingers, could be predefined in the system to perform a click operation. The system has four phases namely, image acquisition, image pre-processing, feature extraction and gesture recognition, as described by Babu et.al [1]. Image acquisition involves capturing images frame by frame using a webcam. * Corresponding author: divinacrasta@gmail.com (Athiya Marium) Published online at http://journal.sapub.org/ajis

The captured images go through the image pre-processing process which involves color filtering, smoothing and thresholding. Feature extraction involves extracting features of the hand image such as hand contours. Gesture recognition involves recognising hand gestures with the help of extracted features.

II. LITERATURE REVIEW

There are many gesture recognition techniques developed for tracking and recognizing various gestures and these are surveyed by Madhuri and Kumar [2]. Gesture recognition using webcam is an appealing option for replacing human computer interaction using a mouse. Back in 2009, Bayazit et.al [3] implemented a GPU-based system for gesture recognition which runs in real-time. While this system demonstrated the principle, it was too bulky and unwieldy for practical use. Advancements in development of optic modules, have allowed for 3-D image sensors to emerge as a viable alternative to stereo and structured light imaging for capturing range information [4]. Due to this gesture recognition has the potential to emerge as an alternative input device in the near future. A system using pointing behaviours for a natural interface to classify the dynamic hand gesture has been proposed by Badgujar et.al. Though this system is suitable for controlling real-time computer systems, it is applicable only for the powerpoint presentations [5]. To overcome the challenges of gesture recognition using color detection and their relative position with each other, gesture recognition using contour analysis is implemented. The algorithm implemented in this paper detects the gesture based on the number of contours that are visible and thereby performs the necessary operation related to the gesture. This paper brings out an innovative idea to use the camera instead of mouse.

Many researchers in the human computer interaction and robotics fields have tried to control mouse movement using video devices. However, all of them used different methods to make a clicking event. One approach, by Erdem et al, used finger tip tracking to control the motion of the mouse. A click of the mouse button was implemented by defining a screen such that a click occurred when a user's hand passed over the region [1, 3]. Another approach was developed by Chu-Feng Lien [4]. He used only the finger-tips to control the mouse cursor and click. His clicking method was based on image density, and required the user to hold the mouse cursor on the desired spot for a short period of time. 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.

Our project was inspired by a paper of Asanterabi Malima et al. [8]. They developed a finger counting system to control behaviors of a robot. We used their algorithm to estimate the radius of hand region and other algorithms in our image segmentation part to improve our results. The segmentation is the most important part in this project. Our system used a color calibration method to segment the hand region and convex hull algorithm to find finger tip positions [7].

III. SYSTEM OVERVIEW

The cursor movement by hand gestures is done using OpenCV [6, 7] library, uses Python programming language, which provides an ease to understand code through its simplicity. Python modules and packages used here are PyAutoGUI [8, 9] and NumPy [10, 11]. The captured video is broken down into continuous image frames using functions defined in OpenCV. The image frames are processed in order to detect any valid gestures being performed by the user. The basic architecture for the system is shown below.

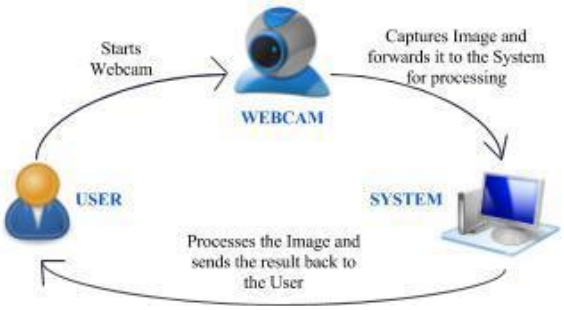

Figure 1. Basic architecture of the system

3.1 Module Description of Proposed System

Module for Finding the Number of Convexity Defects The convexity defects for the hand contour are first calculated by using the OpenCV inbuilt function "cvConvexityDefects()". After the convexity defects are obtained, we perform steps for identifying the fingertips and the fingers.

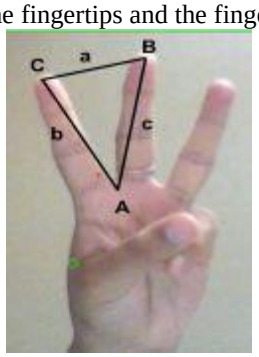

Figure 2. Finding convexity defect Convexity defects obtained, is a structure that returns four values, start point, end point, farthest point and approximate distance to farthest point, out of which three have been used.
The figure 2, denotes for one of the contours the start, the end and the far point. C represents the start point, B represents the end point and A is the farthest point. The angle made by the two fingers must be found to correctly determine if a finger is held up. This is done using the triangle formed by A, B and C. The length of each line is found using the distance formula as

Module for finding the center of the hand Before finding the gesture for one finger held up, the center of the hand should be calculated. For this, first we considered a bounding rectangle that bounded the entire hand. From this we took an approximation of the palm of the hand. Every point in the palm of the hand was put to a

pointPolygonTest of openCV, which returned the distance of the point to the nearest contour edge. A selection sort returned the point with the maximum distance. This point was considered to be the center. The red circle in Figure 3 marks the centre of the hand.

Figure 3. Finding the center of the hand 3.1.3. Module for Identifying the Gesture To find a single finger held up, here, we have taken the index finger, and a selection sort is performed to find the farthest distance from the center of the hand, given by (x,y). This farthest distance is compared against a range of distances that can be considered for the index finger. Another parameter for determining if the finger held up is the index, is the angle it makes with the center, given by (h,k). Assuming that every finger makes a different angle with the centre, we found the angle made by the finger using the formula,

$$A = tan^{-1}((k-y)/(h-x))$$

If the angle and farthest distance lie within the range specified, we have concluded that the finger held up is the index finger. The Figure 4 shows the recognition of index finger by finding the distance from the centre of the hand (h,k), represented by the green circle, and the farthest point (x,y), in the hull.

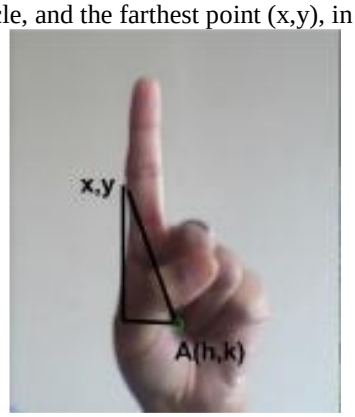

Figure 4. Identifying gesture The next gesture we worked on needed to recognise two fingers held up in the way shown in Figure 5. The same technique of finding angle between two fingers was used here. The angle to which it was compared was however different from the one we specified for two consecutive fingers held up. In addition to this, the angle made by the pinky finger with the centre was also found and compared. The distance between the two fingers was found using the distance formula used above.

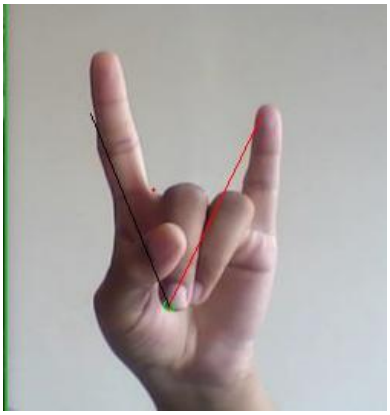

Figure 5. Identifying gesture This way any gesture can be defined and recognised. We found our approach to be simple and easy to understand and manipulate.

3.2. Algorithm for the System

The following algorithm shows the basic steps performed by the system when gestures are detected. START: Start the webcam
STEP 1: Detect the user's hand
STEP 2: Capture the image
STEP 3: Identify the specific hand gesture STEP
4: If the gesture for cursor movement is detected, go to STEP 10
STEP 5: If the gesture for single click is detected, go to STEP 11
STEP 6: If the gesture for double click is detected, go to STEP 12
STEP 7: If the gesture for drag is detected, go to STEP 13
STEP 8: If the gesture for left wave is detected, go to STEP 14
STEP 9: If the gesture for right wave is detected, go to STEP 15
STEP 10: Detect the coordinates of the mouse and perform cursor movement
STEP 11: Using coordinates from the mouse perform the selection, go to STEP 3
STEP 12: Perform selection or opening actions, go to STEP 3
STEP 13: Perform dragging action, go to STEP 3
STEP 14: Decrease the speed of the cursor
STEP 15: Increase the speed of the cursor
STEP 16: Stop

IV. EXPERIMENTAL RESULTS

The experimental setup requires a web camera in order to capture the gestures performed by the user, preferably with a plain background. The web camera should be at a fixed position to help in capturing images more efficiently.

Fig 6: Red color detection for mouse cursor moving

Fig 7:Green color detection for mouse scrolling

Fig 8: Blue color detection for mouse double click

V. CONCLUSIONS

This paper describes a system that controls computer applications with the help of hand gestures. The method proposed here successfully created a hand gesture recognition system, that is able to recognise which gesture is performed by the user and accurately perform the functionality associated with it. Presently, the webcam, microphone and mouse are an integral part of the computer system. Our product which uses only webcam would completely eliminate the mouse. Also this would lead to a new era of Human Computer Interaction (HCI) where no physical contact with the device is required.

VI. FUTURE WORK

The current system gives best results in a plain background and hence puts certain constraints on the user for successful working. The future work will include implementation of additional gestures which will enable the user to perform more functions with ease. Furthermore, background subtraction algorithm can be used for a more effective performance. The proposed system uses only the right hand to perform gestures. Hence, enhancement of the technique proposed, is possible using both hands for performing different computer operations. Experiments need to be done on a larger scale so that results can be more accurate.

VII. REFERENCES

[1] Abhik Banerjee, Abhirup Ghosh, Koustuvmoni Bharadwaj, Hemanta Saikia, “Mouse control using a web camera based on color detection,” International Journal of Computer Trends and Technology , Volume 9 number 1, pp. 15-20, March -2014.

[2] Sugreev Kaur, Rajesh Mehra, “High speed and area efficient 2D DWT processor based image compression,” Signal & Image Processing : An International Journal, Vol.1, No.2, pp. 23-31, December- 2010.

[3] Sandeep Konam, Sai Maheswara Reddy, “A novel approach towards optimization of vision based virtual mouse interface,” International Journal of Engineering Research and Technology, Vol. 3 Issue 3, pp. 1126-1131, March -2014.

[4] P. Kortum, “HCI beyond the GUI: Design for Haptic, Speech, Olfactory, and other nontraditional interfaces,” Morgan Kauf-mann Publishers, pp. 75-106, 2008.

[5] J.Weissmann and R. Salomon, “Gesture recognition for virtual reality applications using data gloves and neural networks,” International Conference on Neural Networks, Vol 3. pp. 2043-2046, 1999.

[6] T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill, “A hand gesture interface device,” CHl+GI, pp. 189-192, 1987.

[7] Yee Yong Pang, Nor Azman Ismail, Phuah Leong Siang Gilbert, “A real time vision-based hand gesture interaction,” Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation, pp. 237-242, 2010.

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

[9] Chu-Feng Lien, “Portable vision-based HCI – A real-time hand mouse system on handheld devices,” National Taiwan University, Computer Science and Information Engineering Department.

[10] Hojoon Park, “A method for controlling the mouse movement using a real time camera,” Brown University, Providence, RI, USA, Department of computer science, 2008.

[11] Abhishek Acharya, Rajesh Mehra, Vikram Singh Takher, “FPGA based non uniform illumination correction in image processing application,” IJCTA, Vol-2(2), pp. 349-358, January-2011.

[12] Bhawna Tiwari, Rajesh Mehra“Design & implementation of Reed Solomon Decoder for 802.16 network using FPGA,” IEEE International Conf.on Signal Processing, Computing & Control, pp. 1-5,

March-2012.
[13] Ann Abraham Babu, Satishkumar Varma, Rupali Nikhare, “Hand Gesture Recognition System for Human Computer Interaction Using Contour Analysis”, International Journal of Research in Engineering and Technology, eISSN: 2319-1163, pISSN: 2321-7308.
[14] K. Madhuri, L. Praveen Kumar, “Cursor Movements Controlled By Real Time Hand Gestures”, International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064.
[15] Mark Bayazit, Alex Couture-Beil, and Greg Mori, ‘Real-time motion based gesture recognition using the gpu’, in IAPR Conference on Machine Vision Applications (MVA), (2009).
[16] D. I. Ko and G. Agarwal, Gesture Recognition: Enabling Natural Interactions with Electronics. Dallas, TX, USA: Texas Instruments, Apr. 2012.
[17] Swapnil D. Badgujar, Gourab Talukdar, Omkar Gondhalekar and S. Y. Kulkarni “Hand Gesture Recognition System”, International Journal of Scientific and Research Publications, Vol. 4, Issue 2, Feb. 2014, pp. 2250-3153.