Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 22 (2016) 657 – 661

Procedia Technology

9th International Conference Interdisciplinarity in Engineering, INTER-ENG 2015, 8-9 October 2015, Tirgu-Mures, Romania

Mouse Cursor Control System Based on Hand Gesture

Horatiu-Stefan Grif[a,*], Cornel Cristian Farcas[a]

[a] "Petru Maior" University of Tirgu Mures, 1, N. Iorga, Tirgu Mures,540088, Romania

Abstract

The apparition on market of the low-cost webcams with, at least, satisfactory qualities open up new directions regarding the implementation of human computer interaction (HCI) interfaces. The paper presents a HCI interface for mouse cursor control. The purpose of the implemented solution is to control the mouse cursor by user hand gestures captured through a webcam. For improving the gesture recognition based on the fluctuation of illuminance levels the finger strips color detection was used. The results reveal the good behavior of the system in low light condition.

Peer-review under responsibility of the "Petru Maior" University of Tirgu Mures, Faculty of Engineering

Keywords: mouse control; hand gesture; human computer interaction.

1. Introduction

The evolution of the User Interface (UI) witnessed the development from text based UI based on keyboard to graphical UI based on mice. In current virtual environments applications, keyboards, mice, and joysticks are still the most popular and dominant devices. However, they are inconvenient and unnatural.[1] Research in Human Computer Interaction (HCI) primarily deals with the design, implementation, and assessment of new interfaces for improving the interaction between humans and machines so that it become natural without the use of any mechanical devices.[7] The next few recently studies may be enough to encourage the research projects in the field of HCI. Khundam introduce in [6] an interactive hand gesture system for control steering and speed for movement in virtual reality (VR). The system is based on the hand palm direction and distance from user using Oculus Rift and Leap Motion devices for VR. In [11] the authors propose a human-3DTV interaction method via the use of a virtual

* Corresponding author. Tel.: +40-265-233-112/154;
E-mail address: horatiu.grif@ing.upm.ro

Peer-review under responsibility of the "Petru Maior" University of Tirgu Mures, Faculty of Engineering
doi:10.1016/j.protcy.2016.01.137

3D interface. Based on simple gestures (captured via the Kinect sensor), the user can manipulate 3DTV fasters and more accurate than an existing product they used for performance comparison. A 3D hand tracking using two common webcams is proposed in [3]. For a better hand detection in the segmentation step the authors use an adaptive hue skin color filter combined with a template filter. In [7] the authors are focused in finding some points on the hand contour which can be used in hand gesture control application. The paper highlights the definition of the hand points of interest, the general algorithm used for their finding and some preliminary results. In [8] implemented a natural user interface based on hand detection and tracking using a webcam. Because of the low quality camera their purpose was quite complex, so that their algorithms become quite complex, combining a number of different techniques and algorithms such as: skin detector based on histograms, background subtraction and a clustering algorithm, an open hand detector and a modified particle algorithm. In [10] is proposed a method for two hands gesture detection using a Kinect sensor. Their algorithm starts with L*a*b* color skin segmentation (for finding the hands) continue, if in image are two hand, with K-means clustering (for separating the hands) and finish with the finding of the fingertips based on the hands convex hull finding through the so-called by the authors, the Graham algorithm. The practical tests used by the authors reveal the robustness and the effectiveness of their method.

2. The system architecture

The aim of the implemented system is to control the windows mouse cursor using the human hand gestures. The system (Fig. 1a) is composed by: (1) computer; (2) external webcam (Genius FaceCam 320); (3) hand pad. Following the block diagram from Fig. 1b, the software application, which runs on the computer, takes an image (a captured frame/image) of the hand pad area (where the user uses his hand as a mouse) through the external webcam and process it regarding the tracking of the user hand and recognition of the hand gesture. The position of the hand on the image will be converted in the position of the mouse cursor on the computer display. The recognized hand gesture is converted in a mouse event as: left click, right click or double click. The considered gestures corresponding to the mouse events above mentioned are presented in Fig.2b, Fig.2c and Fig. 2d. A hand gesture is considered as recognized if is detected the corresponding colored strip/strips. In Fig. 2a is presented the default gesture used for mouse pointer control. The center of the red button is considered the position of the hand.

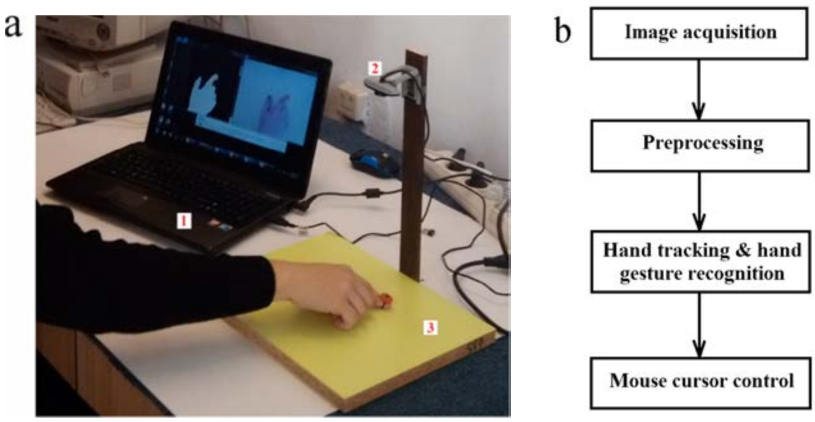

Fig. 1. (a) System overview; (b) Block diagram of the application [5].

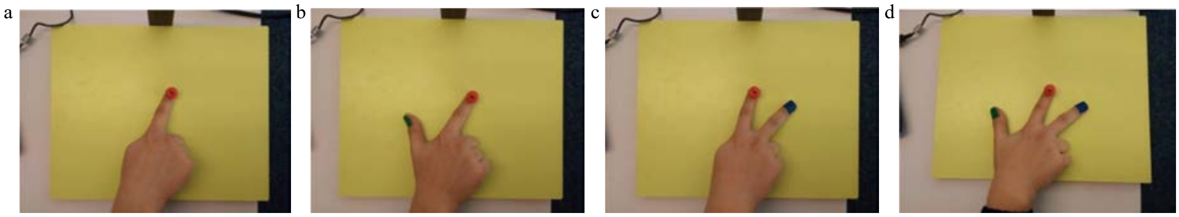

Fig. 2. (a) Mouse pointer control gesture with a red button attached on the tip of the finger; (b) Left click control gesture with a green strip attached to the left finger; (c) Right click control gesture with a blue strip attached to the right finger; (d) Double click control gesture with both green and blue strips attached to the fingers

3. The proposed procedure

In order to achieve the mouse cursor control, the authors used the following algorithm, based on color detection and gesture interpretation:

Step 1. Acquire the image of the hand placed on the hand pad;

Step 2. Remove the noise from the acquired image using a 3x3 Gaussian filter;

Step 3. Hand tracking and hand gesture recognition

- **Step 3.1.** Strips and button color segmentation;
- **Step 3.2.** Morphological erode and dilate operations;
- **Step 3.3.** Determination of how many strips are present in image (none, one or two);
- **Step 3.4.** Recognition of the performed hand gesture;

Step 4. Based on the recognized hand gesture send the corresponding commands to the operating system for mouse control;

Step 5. Jump to Step 1.

Before starting the application for every used color (strips colors, button color), the authors first determined the corresponding HSV [9] components intervals and save the limits intervals. These limits are used to achieve the color segmentation. Each pixel from the color image with its HSV color components inside of the intervals, become white pixel in the binary image otherwise become black pixel.

The morphological erode operation is used to remove the unwanted white pixels (speckles) which can appear after color segmentation. These speckles if are not removed from image, they will cause destabilization of the mouse movements and interferences in mouse commands. The erode operation affect the large regions of pixels of similar intensity. To reduce/correct the erode effect is applied a dilate operation.

A strip or the button is detected as present in the image if the numbers of the pixels that belong to it are greater than a threshold which in this paper is considered to be 100. In this manner is determined that the color detected was in fact produced by the user with the intent of giving a mouse command, or there are only interferences on the hand pad. This idea was used for helping the user have a better experience when using the application, and for stabilizing the mouse movement.

The hand gestures are recognized if the corresponding color (for single click or mouse cursor move) or the corresponding combination of colors (for double click) was detected.

4. Experimental results

The proposed procedure was implemented in C language using the OpenCV Library. Useful information about the description and the use of the functions from this library can be found in [2].

The application was tested for different illuminance levels (see Table 1.) on the hand pad area produced by an electric light system. The electric light was produced by two 36W fluorescent lamps. The different levels of illuminance for the electric light were obtained by using the automatic lighting control system (used in [4]), which control the fluorescent lamps mentioned before. The purpose of the automatic lighting control system is to maintain constant the desired illuminance level (set by the user) on the working plane (the surface of the desk where are the hand pad is situated). The illuminance on the working plane was measured using the PU150 analogical luxmeter.

Analyzing the data from Table 1 the application has a very good behavior for illuminance bigger than 22lx, the successful recognition rate is 100% for all executed hand gestures. For low illuminance level (22lx) the system has a surprising behavior, the minimum successful recognition rate is 93.33% (for left click). For very low level (7lx) of the illuminance on the hand pad plane the left click presents a smaller successful (hand gesture) recognition rate, 46.67%, than the right click which achieved 80% successful recognition rate. The difference appears due to the surface of the strips. The green strip surface is almost half of the blue strip surface. Considering this observation, increasing the green strip surface to equals the blue strip surface the system may achieve an increase successful recognition rate of the considered gestures for very low illuminance level. For a giving illuminance level the number of detected hand gestures for double click was considered the same with the minimum number of the recognized gestures for the left or right click.

The authors, also, have tested the system the during the day conditions, when the illuminance was greater than 500lx. The tests revealed a 100% successful recognition rate.

Table 1. Successful hand gesture recognition rates.

Gestures made (right click)	Gestures recognized (right click)	Successful recognition rate (right click)	Gestures made (left click)	Gestures recognized (left click)	Successful recognition rate (left click)	Illuminance level [lux]
15	0	0%	15	0	0%	0
15	12	80%	15	7	46.67%	7
15	15	100%	15	14	93.33%	22
15	15	100%	15	15	100%	54…400

5. Conclusions

The present paper continues the studies started by the first author in collaboration in [9] and continued in [5]. The application presented in [9] is very sensible to the illuminance level changes. The application presented in [5] eliminates the drawback from [9] but introduce another drawback: the recognition task is influenced by the rotation of the hand on the plane of the hand pad.

The aim of the application was reached, and the mouse cursor control was achieved using an external webcam (Genius FaceCam 320), a software that could interpret some hand gestures (based on different colors) and then turned the recognized gestures into operating system commands that controlled the mouse actions on the computer display screen.

This application has an advantage in using color detection for gesture interpretation, because it can be used in low or high intensity light (even with fluctuations of the illuminance levels on the working plane), with more precision than other applications (as seen in the experimental results, the recognition of the gestures was almost 100% from poor illuminance to high illuminance levels), and in the same time, it opens a window for further applications that intend to control any device using hand gestures or body movements.

The disadvantage of this type of applications is that the final result (the precision with which the gesture detection is made and the stability of the mouse on the working window) is influenced by the quality and fps of the chosen webcam (the frame rate at which the webcam can capture and send the images to the application).

References

[1] Bankar RT, Salankar SS. Head gesture recognition system using gesture cam. Fifth International Conference on Communication Systems and Network Technologies, Gwalior, MP, India, 2015, p.535-538.

[2] Bradski G, Kaehler A. Learning OpenCV. O'Reilly Media, Inc; 2008.

[3] Chien PY, Miao YH, Guo JI. A 3D hand tracking design for gesture control in complex environments. International Symposium on VLSI Design, Automation and Test, 2015, p. 1-4.

[4] Grif HS. PID daylight control system. Scientific Bulletin of the "Petru Maior" University of Tg. Mures, Vol. 8 (XXV), No. 1, 2011, p.12-16.

[5] Grif HS, German Z, Gligor A. Hand posture mouse. Procedia Technology, Vol. 19, 2015, p. 766–771.
[6] Khundam C. First person movement control with palm normal and hand gesture interaction in virtual reality. 12th International Joint Conference on Computer Science and Software Engineering, Prince of Songkia University, Hat Yai, Songkhia, Thailand, 2015, p. 325-330.
[7] Pradhan A, Deepak BBVL. Obtaining hand gesture parameters using image processing. International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials, 2015, p. 168-170
[8] Toni B, Darko J. A robust hand detection and tracking algorithm with application to natural user interface. Proceedings of the 35 th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2012), Opatija, Croatia, 2012, p. 1768-1774.
[9] Vincze MA, Grif HS. Hand gestures mouse cursor control. Scientific Bulletin of the "Petru Maior" University of Tg. Mures, Vol. 11 (XXVIII), No 1, 2014, p. 46-49.
[10] Wen Y, Hu C, Yu G, Wang C. A Robust Method of Detecting Hand Gestures Using Depth Sensors. IEEE International Workshop on Haptic Audio Visual Environments and Games, Munich, Germany, 2012, p. 72–77.
[11] Zhang S, Wang J, Gong Y, Zhang S. Free-hand gesture control with "touchable" virtual interface for human-3dtv interaction. 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), Lisbon, Portugal, 2015, p. 1-4.