INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 5)
Available online at: www.ijariit.com

Finger recognition and gesture based augmented keyboard

Jagannathan MJ
mjjaggu@gmail.com
SRM Institute of Science and Technology, Chennai, Tamil Nadu

Surya M
maniansurya99@gmail.com
SRM Institute of Science and Technology, Chennai, Tamil Nadu

Aqib Muhammed BT
aqibbt36@gmail.com
SRM Institute of Science and Technology, Chennai, Tamil Nadu

RJ Poovaraghavan
mjmadhavan@gmail.com
SRM Institute of Science and Technology, Chennai, Tamil Nadu

ABSTRACT

In this project, we have tried to reduce the gap between the real world and the augmented environment to produce a mixed reality system. For that purpose, we created a virtually controllable keyboard system which is created and implemented using OpenCV libraries and python3.2.1. To provide an easy immersive augmented experience which is also gesture enabled, we employ a web camera which is integrated with OpenCV libraries through a compiler. Using our system, users can control a virtual keyboard using their finger movements and finger tips. Further, users can communicate with people who are viewing the screen, the user selects an alphabet with their fingertip and can move the keyboard with the help of hand gesture. Our model can be utilized by people who are specially-abled and people who cannot talk properly or communicate as they can communicate with others using our proposed system. This paper describes the way of implementing a virtual keyboard without any additional hardware but by using the webcam available in the system. The webcam simply captures the consecutive frames and compares them to recognize it as a click if there is a difference in the contour.

Keywords— *OpenCV, Python 3.2.1, Webcam, Contours, Human-computer interaction*

1. INTRODUCTION

Our project which utilizes finger recognition is a topic which comes under two major computer science fields like augmented reality and human-computer interaction and we have developed a virtual system with the goal of interpreting human gestures via mathematical algorithms. Users can use simple finger gestures to control or interact with the keyboard without physically touching them. Many approaches have been made throughout history, using cameras and computer vision algorithms to interpret sign language. Also, the identification and recognition of posture, gait, proxemics, and human behaviors is also the subject of gesture recognition techniques.[1] Gesture recognition can be viewed as a way for computers to begin to understand human body language and signs, thus filling the void between computing machines and humans than primitive text user interfaces or even graphical user interfaces, which still limit the majority of input to keyboard and mouse and may not be very efficient at all times. Gesture recognition widely enables or equips humans to communicate with the machine and interact naturally without any mechanical devices or computing devices. Using the concept of gesture recognition, it is possible to point a finger at the computer screen or web camera so that the keypad will be pressed accordingly to form meaningful sentences or words. Depending on the type of the input data or source, the approach for interpreting a fingertip could be done in different ways. However, most of the techniques rely on key pointers represented in a 3D coordinate system. Based on the relative motion of these, the gesture can be detected with a high accuracy, depending on the quality of the input and the algorithm's approach. Currently, keyboards are static and their interactivity and usability would increase if they were made dynamic and adaptable. Various on-screen virtual keyboards are available but it is difficult to accommodate a full-sized keyboard on the screen as it creates hindrance to see the documents being typed. Virtual Keyboard has no physical appearance. Although other forms of Virtual Keyboards exist; they provide solutions using specialized devices such as 3D cameras. Due to this, a practical implementation of such keyboards is not feasible. But we can always learn to improve and adapt to whatever technology available to us. As the demand for computing machines evolves, new human-computer interfaces will be implemented to provide multiform or cross-platform interactions between users and machines. Nevertheless, the basis for most human-to-computer interactions remains the same binomial keyboard/mouse. This situation of using only a physical GUI for input may not be the most efficient way.

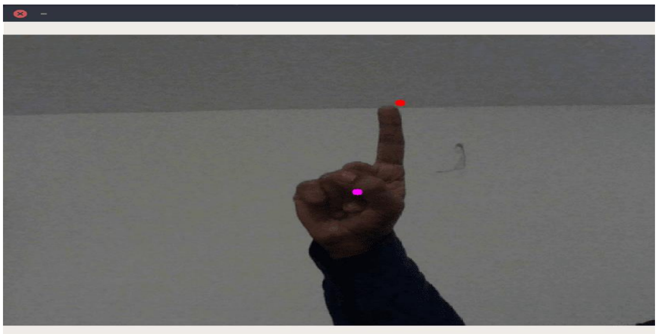

Fig. 1: Finger recognition in Open CV

2. PROBLEM STATEMENT

One of the rising problems with VR industry is there aren't many devices which are being utilized for helping disabled or especially abled people. Recent years have marked a sharp increase in the number of ways in which people interact with computers. Where the keyboard and mouse were once the primary interfaces for controlling a computer, users now utilize touch screens, infrared cameras, and accelerometers (for example, within the iPhone) to interact with their technology. In light of these changes and the proliferation of small cameras in many phones and tablets, human-computer interface researchers have investigated the possibility of implementing a keyboard style interface using a camera as a substitute for actual keyboard hardware. Broadly speaking, these researchers envision the following scenario: A camera observes the user's hands, which rest on a flat surface. The camera may observe the hands from above the surface, or at an angle. The virtual keyboard's software analyses those images in real-time to determine the sequence of keystrokes chosen by the user. These researchers envision several applications for this technology: in some countries (for example, India), users speak many different languages, which makes producing physical keyboards for many different orthographies expensive. A camera-based keyboard can easily support many languages, Smart-phone and tablet users may occasionally want to use a full-sized keyboard with their device, but are unwilling to carry a physical keyboard. Since most mobile devices are equipped with a camera, a camera-based keyboard could provide a software-based solution for this problem.

3. FINGER TIP DETECTION ALGORITHM

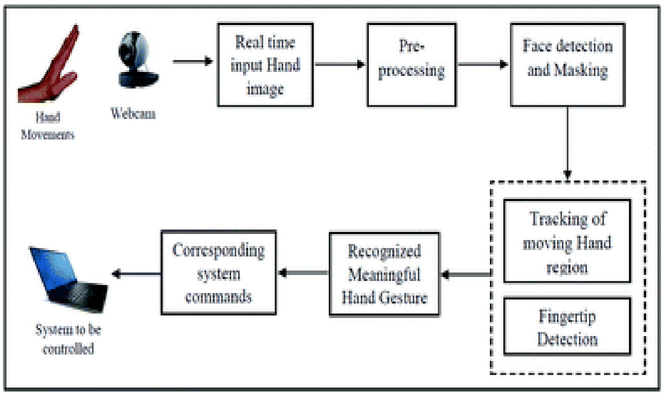

Fig. 2: Working of Gesture-based devices

In this step, we estimate the locations of the user's fingertips (in image-space) based on geometrical features of the contours and regions obtained from the last step. We represent a contour $\tilde{\gamma}$ as a sequence of positions {p1 = (x1, y1), p2 =(x2, y2)} in image-space. Given a contour, one can derive several other sequences of geometrical significance and use these to locate the fingertips. The previous processing step gives us a contour $\tilde{\gamma}$ consisting of pixel locations pi such that the Euclidean distance between pi and pi+1 is 1 for all i, and the angle in between the displacement vectors $p_{i+1} - p_i$ and $p_{i-1} - p_i$ is in $\{-\pi/2, 0, \pi/2, \pi\}$. This angular information does not give a very good idea of how the contour bends around the hand, so we consider a subsequence γ in which the points of the contour are spaced further apart. In experiments, we took γ to be every tenth point of $\tilde{\gamma}$. We then define a sequence of angles in (in $(-\pi, \pi]$) from γ as we did with $\tilde{\gamma}$, by considering the angle between displacement vectors.

4. SYSTEM ARCHITECTURE

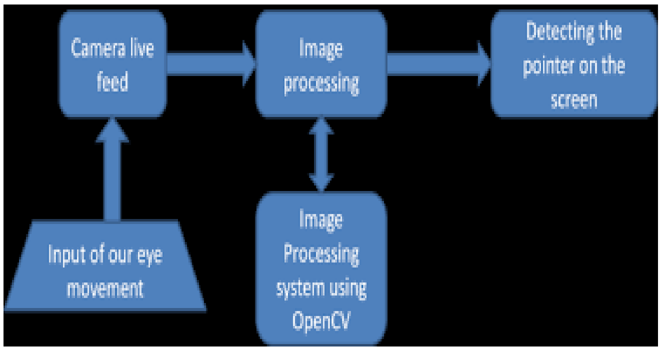

Fig. 3: The process of work flow

We describe the image analysis techniques used to convert the user's surface-touches into keystrokes. Most state-of-the-art camera-based virtual keyboard schemes, including, are implemented using the following sequence of image analysis techniques. The system's camera captures an image I for processing. By performing skin segmentation on 'I', we obtain a binary image with a region H representing the hand(s) in the scene. H is analyzed to determine the locations of the user's fingertips. The references listed above determine contours that parameterize the boundary of H. They associate fingertips with the points on the contours that optimize certain geometric quantities (for example, curvature). We consider two of these geometric approaches to finding the fingertips and compare their performance. Given the estimated fingertip positions, one must calculate which fingertips are touching the keyboard's keys. Propose a technique called shadow analysis to solve this problem. To map touch points to key presses, we assume the keys of the virtual keyboard are described as rectangles in R2. We assume the layout of the keyboard is known at compile time and the keyboard mat has control points from which we can derive a perspective correction transformation. We then apply a simple formula to convert the keyboard-space coordinates of the touch points to key presses.

5. OUR EXPERIMENT

In our experiment, we have used an HP lap-top with web cam. This implementation can be tested on any device having a front camera and image processing and OpenCV libraries installed in it. As a rule of thumb, if your laptop does not have OpenCV and python installed in it, then our system won't work or produce the desired output. Once the setting is done we display, the virtual key-board which is now ready to detect the keys. A proper click gesture is made by the user, i.e. the fingertip first moves inwards and then outwards. The two consecutive frames are captured by the web camera and only if the difference in the two frames is noticeable the gesture is recognized as a click. Once a proper click is made on any key the center of that key is marked with blue. The difference in the area of the click and that of the center is calculated to make sure that the key whose co-ordinates en-compasses the click is detected and displayed. The algorithm which we used to create this is as follows: First, we create a method which separates foreground from the background and then we setup out-file for writing video. Also, have a method that determines the distance between two points using Euclidean distance. Now, read a frame from a webcam, press a letter in the air by seeing the virtual keyboard. Webcam captures the frame and analyzes. If the difference in contour is significant, then find the contour of click gesture else if it is not very significant, then click the alphabet of the keyboard again. If a letter has been selected for 10 frames, then write the letter and clear the buffer. In our system, we have also enabled the use of swiping motion and our project recognizes swiping motion so that, it can click numbers, alphabets and other symbols in the keyboard. This is because our virtual keyboard is in the form of a long straight row enabled with clicking and swiping motion.

6. OUTPUT AND RESULTS

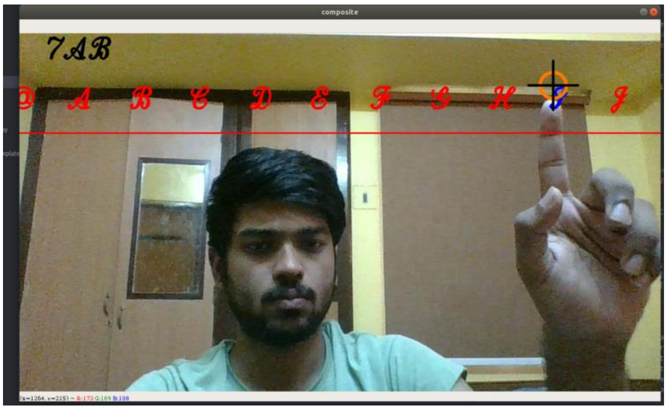

Fig. 4: Clicking a series of alphabets with a finger

Here, in the above output, the finger is placed in accordance with the letter on the keyboard. When the virtual keyboard detects the contour difference, it draws a contour of the finger and then selects and displays the letters. When the user selects a series of letters, the program analyses the coordinate's and displays the same. The user can also move the keyboard with swiping motion.

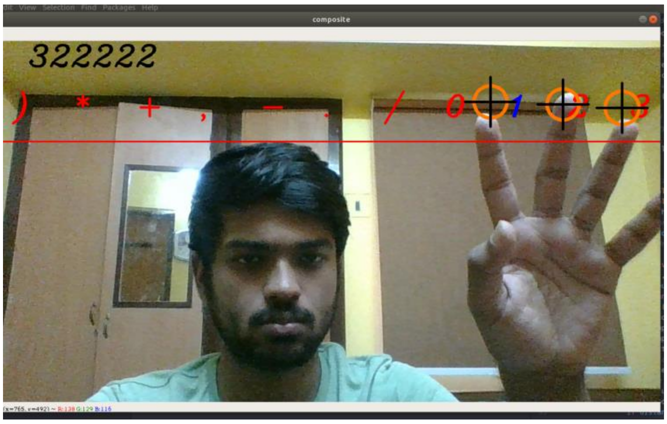

Fig. 5: Swiping motion and multiple touches enabled

7. RELATED WORK

There have been various proposed solutions for the design and implementation of the virtual keyboard, but mostly they have used an additional hardware. Proposed a way for implementing keyboard with help of a CMOS camera which has increased the complexity. Defines virtual keyboard for android mobile phones. The keyboard detects keys on the basis of manual learning or by using K-means algorithm but variance in light affects the output displayed. Another paper has used shadow analysis technique for gesture recognition but once the hand is very close to the keyboard the shadow is hidden and the camera is not able to see the shadow thus desired output is not achieved. Defines a way of implementing virtual keyboard and computes the accuracy of keystroke recognition but the finger must be held perpendicular to the keyboard for detecting a click which makes it difficult to obtain the correct results all the time. Plots a graph on the distance between the keys v/s frequency of errors using 4 different keyboard layouts with the help of Fitts's Law. A device called Tesseract has been tested for accuracy but is unable to give accurate results with camera in tilted position.

8. CONCLUSION

In this paper, we have presented a way of implementing a virtual keyboard on the screen without any use of external hardware. This drawback which can be further improved by implementing algorithms which allow the web cam in quickly capturing and comparing the consecutive frames will give a better result with less complexity. The key- board would recognize a click only if there is a significant movement of the fingertip. If the movement is quick the web cam might not be able to detect it as a click because frames will not be captured quickly and the difference in contour will not be large enough.

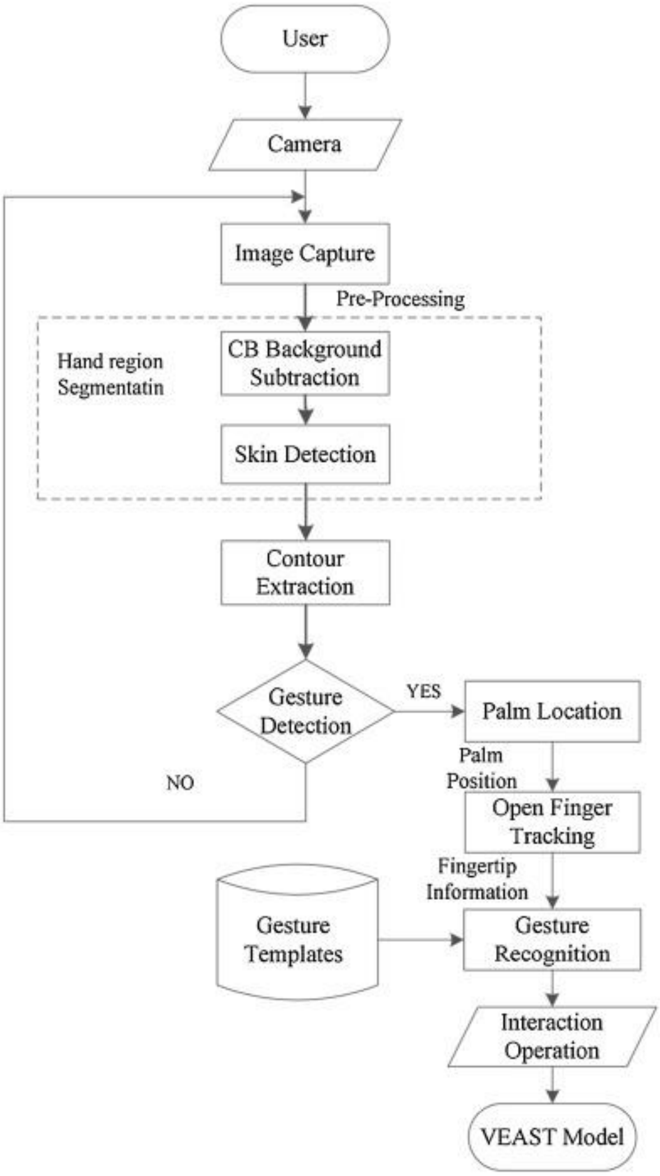

Fig. 6: Flow graph of processing

9. FUTURE SCOPE

For the future work, swipe keypads which detect the gestures in air view can also be implemented. This would improve the results when the typing speed is quick. The same key-pad can be made multi-lingual by just changing the keys on the keypad. Also, this technique can be further improvised to be used on a smart TV.

10. REFERENCES

[1] "A Novel Approach for Inbuilt Virtual In-put Devices for Human Computer Interaction" Gadiparthi Manjunath, Nune Sreenivas, Addis Abab Institute of Technology, AAU, Ethiopia.
[2] "User defined custom virtual keyboard" Rishikesh Kumar, Poonam Chaudhary.
[3] "Toward the Use of Gesture in Traditional User Interfaces" Rick Kjeldsen, John Kender.
[4] "Human-Computer Interact ion using Gesture Recognition and 3D Hand Tracking" Jakub Segen and Senthil Kumar.
[5] "Shadow Gestures: 3D Hand Pose Estimation using Single Camera" Jakub Segen and Senthil Kumar.
[6] "Virtual Keyboard Layout Optimization" Siddhart Jain, Samit Bhattacharya.
[7] "Keystroke Recognition for Virtual Key-board" Jani Mantyjarvi, Jussi Koivumaki, Petri Vuori.
[8] "Webcam Based Virtual Keyboard Using Shadow Analysis-A Review" Rupali T. Bhole, Prof. Anant S. Bhide.
[9] "Beyond the touch: a Web Camera based Virtual keyboard" Caslav Li Vada, Miro Proleta, Kresimir Romic, Hrvoje Leventic.
[10] T. Padmapriya and V. Saminadan, "Distributed Load Balancing for Multiuser Multi-class Traffic in MIMO LTE-Advanced Networks", Research Journal of Applied Sciences, Engineering and Technology (RJASET)- Maxwell Scientific Organization, ISSN:2040-7459; e-ISSN: 2040-7467, vol.12, no.8, pp:813-822, April 2016.
[11] S. V. Manikanthan and K. Baskaran "Low Cost VLSI Design Implementation of Sorting Network for ACSFD in Wireless Sensor Network", CiiT International Journal of Programmable Device Circuits and Systems, Print: ISSN 0974 – 973X & Online: ISSN 0974–9624, Issue: November 2011, PDCS112011008.

