 ISBN: 978-93-86005-02-1

Virtual Keyboard Implementation

Rajib Saha*, Soumyajit Paul, Suman Kumar Ghosh, Sucharita De and Namrata Mitra

Department of Computer Science and Engineering, RCC Institute of Information Technology, Kolkata - 700015, West Bengal, India; rajibsaha_4u@yahoo.co.in, soumyajitpal93@gmail.com, rickypanthers@gmail.com, sucharita.de.21@gmail.com, namratamitra94@gmail.com

Abstract

The Association for Computing Machinery defines Human Computer Interaction as "a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them". An important facet of Human Computer Interaction (HCI) is the securing of user satisfaction (or simply End User Computing Satisfaction). This paper aims at creating a Virtual Keyboard using a webcam and a color pointer. A Virtual Keyboard have been developed using 2D webcam instead of any other specialized hardware. This key-board will overcome problems of traditional keyboard and serve as an add-on for it to make computing very easy and efficient for general users of computer. Traditional keyboards were not reconfigurable. We cannot change the specifications of keyboard according to need of individual. We cannot use a key for some abstract or user defined purpose, those keys of traditional keyboard behaved in a specific predefined or say preprogrammed manner. In Virtual Keyboard every key will be configurable according to the need of user.

Keywords: Colour Pointer, Configurable, Human Computer Interaction,Virtual Keyboard

1. Introduction

A Virtual Keyboard is a software that allows users to give keyboard inputs to a system without using an actual keyboard. Creating a virtual human computer interaction device such as mouse or keyboard using a webcam and computer vision techniques can be an alternative way for the touch screen.

Keyboard is an input device that has been used since the beginning of computer technology. Its ability to express character with a great speed and accuracy makes this input device has consistent and stable usage. Keyboard has a long development history that consists of the port connector change, size change and some of it is the change in the input media. To realize a new revolutionary way in interaction, the main idea is by using webcam as a cheap and easily obtained input media. By using webcam as media, it is possible to interact with computer without a real physical-interaction device. Webcam media has many advantages such as the existence of many graphics processing algorithm that can help computation. The webcam is also a very common device, making the development for webcam can progress greatly, as people can easily try and continue the development easily. The main problems in webcam

**Author for correspondence*

Virtual Keyboard is how the webcam will be used and the algorithm used to transform the typing gesture to a keyboard input while still maintaining the comfort that can be found in the physical keyboard. Webcam is an applicable media for Virtual Keyboard even in its limitation with the help of fast image processing method.

2. Literature Review

2.1 Image Processing and Image Analysis

Image processing is any form of signal processing for which the input is an image, such as photographs or frames of video. The purpose of image processing is - Visualization (observe the objects that are not visible), Image sharpening and restoration (create a better image), Image retrieval (seek for the image of interest), Measurement of pattern (measures various objects in an image), Image Recognition (distinguish the objects in an image).

Image analysis is the extraction of meaningful information from images; mainly from digital images by means of digital image processing techniques. The main thing done is the color detection. At first we receive an image from the web cam. Then each pixel is retrieved from the image and extracts the Red, Green and Blue values (RGB) from each pixel. Now we can easily detect a particular color since all the colors are combinations of RGB values.

2.2 Existing Keyboard Design Approaches

A number of Virtual Keyboards have been proposed in English language. In this section, the design principles of these keyboards are discussed followed by their performance evaluations.

2.2.1 Dvorak Layout

In[11] propose an alternative arrangement for the keys (Figure 1.) which rejects the design of the QWERTY layout through an analysis of the relative frequency of used letters. The Dvorak layout is designed for the following tasks:

- To allocate more work to the right hand than the left.
- To distribute letters based upon the strength of each finger.
- To place the most frequently used letters on the home row.
- To place vowels and frequently used consonants so that they can be typed with alternate hands.

Since the Dvorak layout is well known as a QWERTY alternative, it is a logical starting point. In its physical form, the Dvorak keyboard is similar to a QWERTY keyboard. A QWERTY layout can be transformed into a Dvorak layout by renaming the keys. The Dvorak keyboard is designed to optimize two handed touch typing. The idea is that higher entry rates can be obtained if common digraphs are entered by fingers on opposing hands instead of on the same hand As well, the most common letters (e.g., E, T, A, H) are positioned along the home (viz. middle) row.

2.2.2 Fitaly Layout

The Fitaly One-Finger keyboard[1] is designed to minimize hand movement during text entry with one finger, a stylus or a pen. This keyboard is a commercial product designed to optimize text entry with a stylus. The most important feature of the layout is the presence of two space bars. The proximity of the most common letters in English (e.g., E, T, A, H) to the space bars is also immediately apparent. The keyboard's name is taken from the letter sequence along the second row of keys. The Fitaly keyboard is designed to minimize the travel from one letter to the next.

2.2.3 Opti Layout

It is one of the optimized Virtual Keyboard layouts for the English language. The keyboard layout is optimized to increase the typing speed

using trial and error method, Fitts' law[15] and character and diagram frequencies in English. Fitts'law[15] gives a function for computing the key tapping time given the length of the movement and width of the target are needed. These enable a researcher to compute a prediction for the upper bound of user performance given the keyboard layout. Trial and error is needed to generate the keyboard layouts. According to the calculations of MacKenzie and Zhang is one finger keyboard layout. The Opti layout[2] is theoretically 35% faster than QWERTY and 5% faster than Fitaly layout.

2.2.4 Lewis Keyboard

In[18] designed a Virtual Keyboard where the letters are laid alphabetically in two columns, which did not show a performance advantage, probably due its elongated shape. Lewis et al. proposed a 5 by 6 Virtual Keyboard with a strictly alphabetical sequence, which is suffered from the same problem as discovered by Norman and Fisher - the alphabetical discontinuity caused by row breaks.

2.2.5 Hooke's Layout

As discussed earlier, the goal of a good keyboard design is to minimize the statistical travel distance between characters.

The more frequent digraphs should be closer together than less frequent digraphs. In order to achieve this goal, a dynamic system technique has come up. As example, a spring connecting every pair of the 27 keys whose initial positions were randomly placed with spaces between the keys. The elasticity of the springs, when turned on, was proportional to the transitional probability between the two keys so that keys with higher transitional probability would be pulled together with greater force. In addition, there is viscous friction between the circle shaped keys and between the key surface and the table surfaces.

3. Proposed Work

3.1 Filtering, Image Subtraction, Color Segmentation and Morphological Operations

The first phase is to separate between pixels that have the potential of being hand pixels and the ones that are not. The processed frame is blurred using a 3 x 3 Gaussian blur filter. It is used to reduce image noise and to reduce the image details. Then, the image is subtracted from an initially referenced image as presented in[5]. For the purpose of enhancing the hand's detection – neglecting abnormal objects – skin segmentation is used. The image captured is coded in YUV color space due to hardware constrains. A transformation to HSV color space is performed.

3.2 Edge Detection

The edge detection phase is essential in order to find the hand's contour and then the finger tip. Hence, to extract the contour we use canny filter. Additional modifications are needed due to the fact that the contour acquired is not continuous and to make it so we use 8-point connected component analysis neighborhood system that produces set of counter-clockwise perimeter coordinates which trace the outline of the hand $(j) = \{(xj, yj)\}$ this enables complete traversal of the hand's edge used in 3. and small bumps that may have remained.

3.3 Tip Detection

In this stage the tip of the object is found using the approach presented in[8]. The finger's discrete outline is converted into a list of consecutive coordinates representing the contour of the finger. Each three j consecutive pixel coordinates $[C(j–k), C(j), C(j+k)]$ represent three vertexes of a triangle and the head angle is calculated under the assumption that

the middle coordinate is the head vertex. The angle is calculated using the law of cosines.

$$\gamma = arcos\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$$

Where γ is the angle representing the contour's peak examined as a potential peak. And a, b, c are the triangle edges calculated as distances between [C (*j*–*k*), C (*j*), C (*j*+*k*)].

3.4 Shadow Extraction

The image obtained from 1. and the captured image are subtracted from the initial reference image leaving us with only the shadow within the new obtained image (after transforming the result image to a binary image). Then, the image containing the shadow is processed as if a hand is being detected through phases 1. –3. (without skin segmentation).

3.4.1 Algorithm to Calculate the Co-Ordinate of the Keys

Algorithm 1: Calculating the co-ordinates of the keys

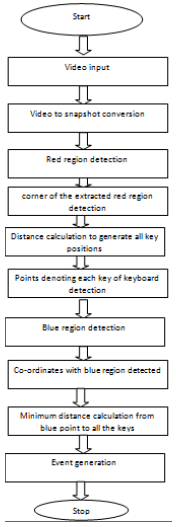
Step 1 : Take one snapshot from video input.
Step 2 : Extract the red rectangular region.
Step 3 : Obtain the co-ordinates of the four corners.
Step 4 : Choose any three adjacent corners.
Step 5 : Using any two of those corners, calculate the co-ordinates of the reference points.
Step 6 : Using those reference points, calculate the co-ordinates of the keys.

3.4.2 Algorithm to Choose the Right Key

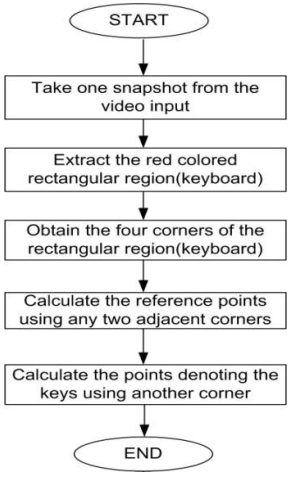
Algorithm 2: Choosing the correct key

$n = 0$
while $flag \neq 0$ **do**
  Take one snapshot from video input.
  if *Blue colored region is present in the picture* **then**
    Extract the blue region denoting touch.
    Obtain the co-ordinates of the highest point p of that region.
    for $i = 1$ **to** N **do**
      Calculate the distance d_i of p from i^a point.
      if $d_i \leq d_{i-1}$ **then**
        $Dist_{min} = d_i$
        $loc = i$
    Perform the function of the key at position loc
    $n = 0$
  else
    $n = n + 1$
  if *STOP key is chosen* ***OR*** $n = 10$ **then**
    $flag = 0$

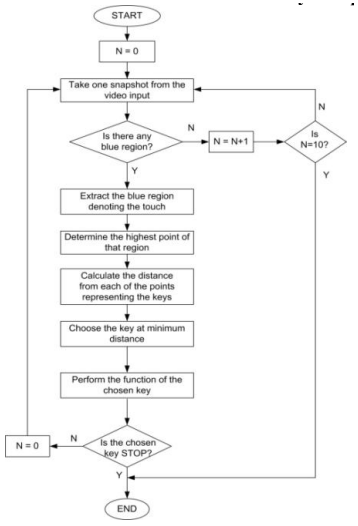
Flowchart of the System



Flowchart to calculate the co-ordinates of the keys



Flowchart to choose the correct key and perform its function



4. Implementation and Results

Figure 1. Snapshot of the Keyboard.

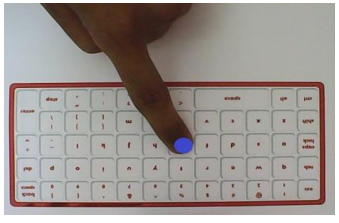


Figure 2. Snapshot of one single touch.

Figure 3. Grayscale image of the keyboard snapshot.

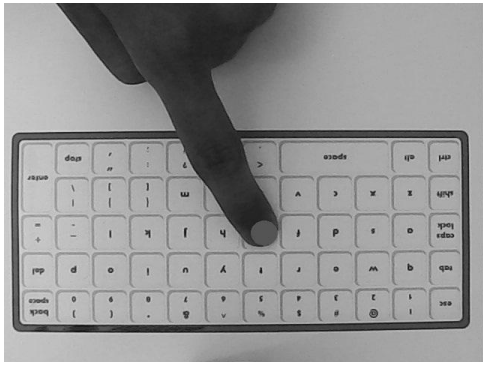


Figure 4. Grayscale image of the touch snapshot.

Figure 5. Extracted red region.

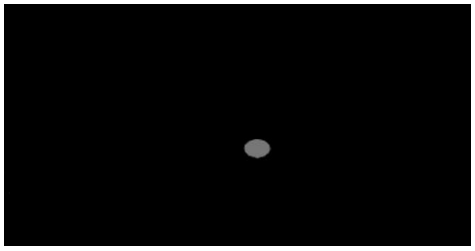


Figure 6. Extracted blue region.

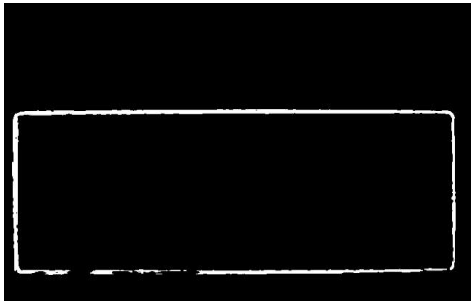


Figure 7. Black and white image of the keyboard.

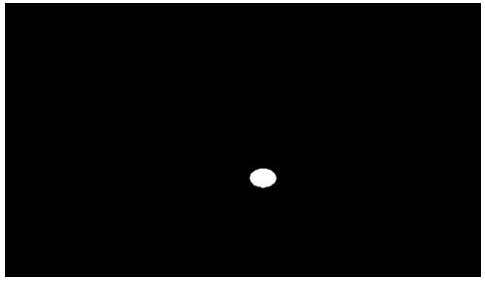


Figure 8. Black and white image of the touching region.

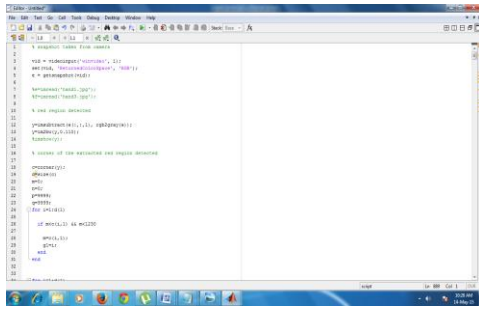


Figure 9. Running code.

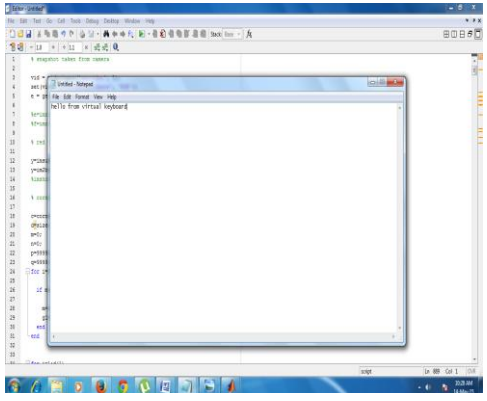


Figure 10. Sample output.

Figure 11. Snapshot of system.

5. Conclusion

The Virtual Keyboard that we propose uses only a standard web camera with no additional hardware. This paper addresses problems with current Virtual Keyboard implementations and describes a novel technique, namely shadow analysis to solve these problems. The objective of this paper is to develop a Virtual Keyboard (VK) using only a standard 2D camera without the need for additional specialized hardware.

A Virtual Keyboard for mobile devices will remove the inherent space constraints and would therefore provide for a full sized keyboard without additional hardware. In implementation, a Logitech Webcam is used and a sheet of paper with the keyboard printed on it. The only unique aspect of the keyboard is that it has four colored endpoints which are used to identify the keyboard. The implementation is based on use of image processing. We propose Shadow Analysis for detection of webcam based Virtual Keyboard (VK). We introduce detection of edge by simple way like Sobel technique. The VK presented here is only a small application of a larger idea which detects finger touches using a standard 2D camera. Finally we expect that using Image processing implementation of webcam based Virtual Keyboard would enable us to use a full sized QWERTY keyboard without the need for additional physical space or hardware. Moreover, the VK can find applications in gaming, 3d modeling etc.

6. Acknowledgment

The authors are very much grateful to the Department of Computer Science and Engineering for giving the opportunity to work on An Efficient Methods for Skew Normalization of Handwriting Image. Both the authors sincerely express his gratitude to Dr. Arup Kumar Bhaumik, Principal of RCC Institute of Information Technology College for giving constant encouragement in doing research in the field of image processing.

7. References

1. Adajania Y, Gosalia J, Kanade A, Mehta H, Shekokar N. Virtual Keyboard using Shadow Analysis. Third International Conference on Emerging Trends in Engineering and Technology; Goa. ,2010 Nov 19-20. p. 163–5.
2. Hernanto S, Suwardi IS. Webcam Virtual Keyboard. International Conference on Electrical Engineering and Informatics (ICEEI); Bandung, Indonesia. 2011 Jul 17-19. p. 1–5.
3. Kolsch M,Turk, M. Keyboards without Keyboards: A survey of Virtual Keyboards. Workshop on Sensing and Input for Media centric Systems; Santa Barbara, CA. 2002 Jun 20-21. p. 1–8.
4. Du H, Oggier T, Lustenburger F, Charbon E. A Virtual Keyboard based on true-3D optical ranging. Proc British Machine Vision Conference (BMVC); Oxford. 2005 Sep. p. 220–9.

5. Matsui N, Yamamoto Y. A new input method of computers with One CCD Camera: Virtual Keyboard. Keio University; 2000.
6. Wilson AD. PlayAnywhere: A compact interactive tabletop projection-vision system. Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology; Seattle, WA, USA. 2005. p. 83–92.
7. Segen J, Kumar S. Shadow gestures: 3D hand pose estimation using a single camera. Proceedings of Computer Vision and Pattern Recognition; 1999. p. 479–85.
8. Mantyjarvi J, Koivumaki J, Vuori P. Keystroke recognition for Virtual Keyboard. 2002. IEEE International Conference on Multimedia and Expo. 2002; 2:429–32.
9. Gupta S, Ghosh S. Sobel edge detection algorithm. International Journal of Computer Science and Management Research. 2013 Feb; 2(2):1578–83.
10. Maini R, Aggarwal H. Study and comparison of various image edge detection techniques. IJIP. 2009; 3(1):1–11.
11. Argyle E. Techniques for edge detection. Proc IEEE. 1971; 59:285–6.
12. Kalpana M, Kishorebabu G, Sujatha K. Extraction of edge detection using digital image processing techniques. International Journal of Computational Engineering Research. 2012 Sep; 2(5):1562–6.
13. Lizhen L. Discussion of digital image edge detection method. Mappingaviso; 2006. p. 40–2.
14. An M, Zhihui Z. Several edge detection operators comparation. Industry and Mining Automation; 2004. p. 54–6.
15. Zhiguo L. Image processing and analysis based on MATLAB. Beijing: Defense Industry Publication; 2007. p. 4.
16. Nadernejad E. Edge Detection techniques: Evaluations and Comparisons. Applied Mathematical Sciences. 2008; 2(31):1507–20.
17. Maini R, Aggarwal H. Study and comparison of various image edge detection techniques. IJIP. 2009; 3(1):1–11.