Virtual Keyboard Using Shadow Analysis

Y. Adajania, J. Gosalia, A. Kanade, H. Mehta, Prof. N. Shekokar
Department of Computer Engineering
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India
harsh2327@gmail.com, nshekokar@yahoo.co.in

***Abstract*— Traditional QWERTY keyboards provide a minimal but functional interface. However these keyboards are bulky and offer very little in terms of enhancements. In this age of miniaturization, where the size of laptops and desktops is becoming smaller, the traditional keyboard acts as a hindrance to further miniaturization. The Virtual Keyboard (VK), with its minimal physical form can provides a solution to this problem.**

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. 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.

Keywords- Virtual Keyboard; Shadow analysis; Image processing; shadow extraction; edge detection

I. INTRODUCTION

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. This idea is significant since such a virtual keyboard can, in the future, be developed for mobile phones. 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 our implementation, we use a Logitech Webcam 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.

II. RELATED WORK

The Virtual Keyboard has been implemented in a number of different forms, as described by Kölsch, M. and Turk, M [3]. Of these, the ones based on 3-D optical ranging [1] and CCD cameras [4] are most significant as they are based primarily on image processing.

The elaborate research done by Kölsch, M. and Turk, M. [3] highlights a variety of virtual keyboards in different forms, such as gloves, rings, hand gestures based and projection based devices.

In [1], a special 3-D camera, or two 2-D cameras are used. Additionally a pattern projector is used for projecting the keyboard. The VK designed in [4] makes use of a single CCD camera.

Even more significant is the work presented in [2], where a shadow based analysis is used to acquire depth information from a 2-D image. The device also makes use of a high-power infrared light source and a camera with an infrared filter. We use this novel technique of using finger shadows to detect finger touches in our implementation of the Virtual Keyboard.

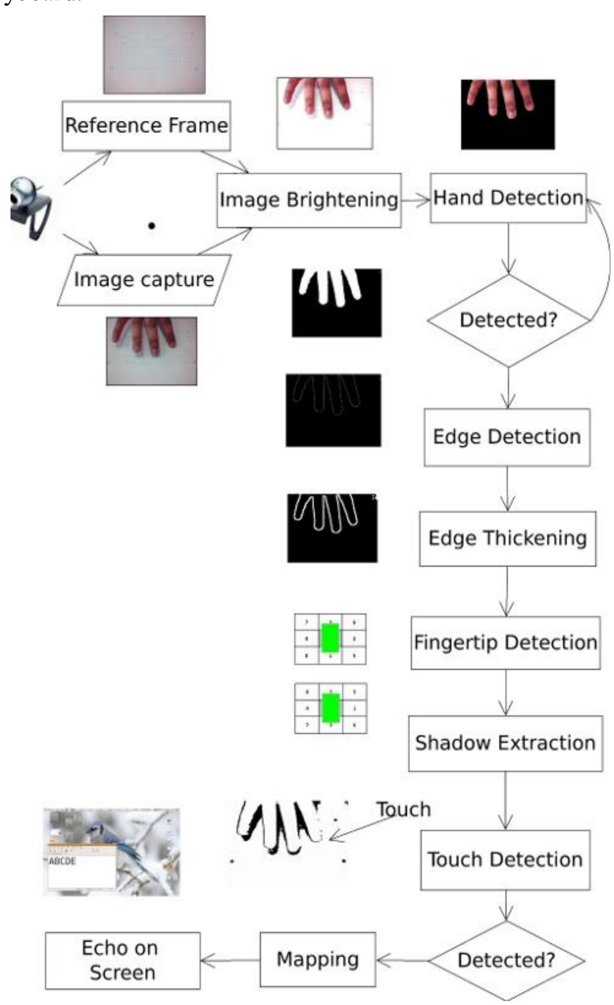

Figure 1. Flow Graph of the Virtual keyboard

978-0-7695-4246-1/10 $26.00 © 2010 IEEE
DOI 10.1109/ICETET.2010.115

The ideas presented in [2], [4] may not be practical to implement as the devices used are highly specialized. However, the ideas presented for hand, finger and fingertip detection are significant in this ever changing field of virtual human interface devices (VHID).

III. Proposed Solution

The VK implementation flow is shown in the following flow graph. Initially, the keyboard is detected using reference points. Once the keyboard is detected, the user's hand is detected using color segmentation. This process is also useful in extracting the hand's shadow. If the hand is detected, the process flow moves onto edge detection which is used to identify the finger tips. For finger tip detection, we use our own algorithm that makes use of a weighted-priority structure to traverse the detected hand edge.

1. Keyboard Detection

The endpoints of the VK are identified using color differentiation. The endpoints of the keyboard are colored blue, thus on thresholding, these points can be easily identified. The area of interest i.e. the location of the VK in the image is then defined.

2. Hand Detection

Figure 2. Original Frame

Figure 3. Hand Detected using color segmentation

Initially, a large collection of hand images was created. These images were observed for their RGB (Red Green Blue) values in the area of interest, i.e. the hand. It was noted that, in these hand regions the red component was higher than the other two components. These observations were consistent across all the images that were tested. There may be certain regions on the hand such as the fingernails and/or veins where this observation might not be true. However, this does not affect the overall result as a significant portion of the hand follows the expected pattern and the hand is appropriately detected. In order to remove the abnormalities in the hand regions, we use image enhancement techniques. Finally, the detected hand regions are thresholded to white, while the rest of the image is made black.

3. Edge Detection

For edge detection we use the Sobel technique, which is found to give better results when compared to the Canny, Prewitt and Zero-Cross. Edge detection of the hand is required in order to detect the fingertips. The edge obtained is then thickened in order to remove discontinuities. This enables a complete traversal of the hand edge which is described in the next step.

Figure 4. Hand Region thresholded to white

Figure 5. Edge detected using Sobel technique

4. Tip Detection

In this stage we intend to find all the finger tips maximum of 10 finger tips present in the image taken by the webcam. The hand edge obtained from (2) is a thin single lined edge which has lots of discontinuities. These discontinuities make it difficult to traverse along the edge of the hand. To overcome this problem, the thin single lined edge is dilated using a structuring element to give a thick edge as shown in the figure 8.On this thickened edge our algorithm is run to find the finger tips. We use the following priority structure while traversing the edge to determine the next pixel to be tested for touch.

8	2	5
4		1
7	3	6

Figure 6. Priority structure 1 for moving down the hand edge

7	3	6
4		2
8	1	5

Figure 7. Priority structure 2 for moving up the hand edge

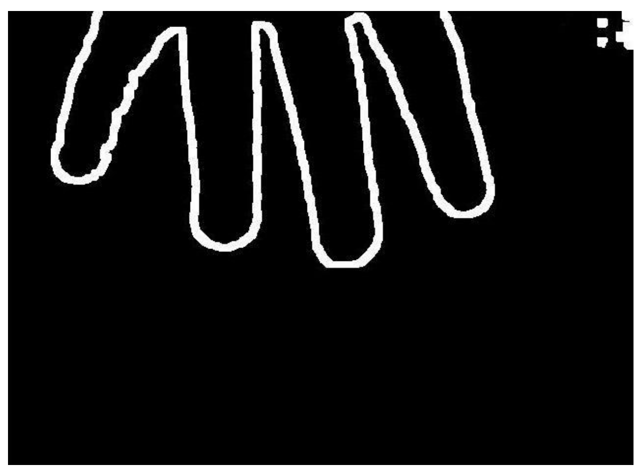

Figure 8. Thickened hand edge. Edge thickening is essential as the edge obtained using Sobel may not always be perfect and continuous. Priority structure 1 for moving down the hand edge

5. Shadow Extraction

The original image captured from the camera is dynamically brightened depending on the lightning conditions. This eliminates all background 'noise' such as shadows due to multiple light sources (the penumbra) and keyboard's characters; leaving behind only the hand and the umbra of the shadow. The hand obtained from (2) is then subtracted from this image. This gives us only the shadow of the hand.

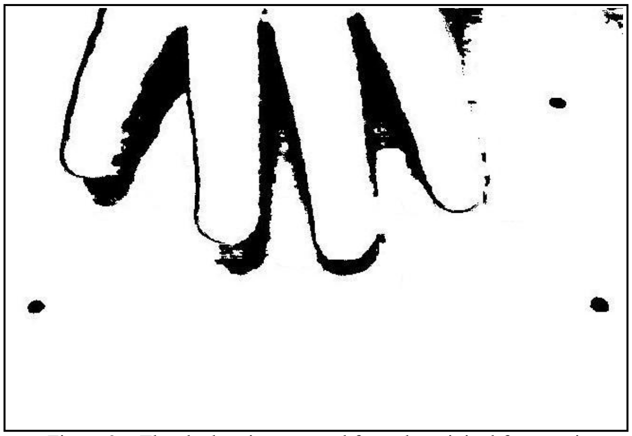

Figure 9. The shadow in extracted from the original frame using image subtraction i.e. the hand region obtained in fig.3 is subtracted from the original frame to obtain the shadow

6. *Touch Detection*

A small region around the fingertips {detected in (4)} is scanned for shadow {detected in (3)}. As shown in fig.9 the tips are white while the shadow regions are black. The ratio of the white to black pixels is determined. If the ratio of the area of non-shadow region to the area of the shadow region exceeds a particular threshold we can say that touch has occurred. The threshold is determined using an extensive set of test cases. The idea for touch detection presented here is based on [2].

7. *Mapping*

The final step involves mapping of the fingertips to actual keys. This is a simple 2D mapping based on the information available in the current frame (x, y coordinates) and relative position of the fingertip from the endpoints of the keyboard.

IV. Future Scope

The VK presented here is only a small application of a larger idea: detecting finger touches using a standard 2D camera. The true application of this idea would be in developing virtual keyboards for mobile devices, which 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.

In a country like India, which has a rich repository of languages, a Virtual Keyboard can be used to generate multilingual keyboards, by just changing the internal mapping of characters, with no change in hardware. This can make multilingual keyboards economically feasible.

Finally, such a keyboard creates the possibility of developing specialized keyboards especially for the blind, without additional expenditure.

V. Test Results

In the results obtained in various test cases, the VK was found to give an average of 3 frames per second, i.e. an average of 3 frames can be analyzed per second. It must be noted that, this rate can be further improved by reducing the resolution of the analyzed frames, with little effect on the accuracy.

It must be noted that we assume ambient lighting conditions as these are essential for image processing.

VI. Conclusion

This paper presents a practical implementation of a Virtual Keyboard and demonstrates the future of human-computer and human-mobile devices interaction lies in the creation of Virtual Human Interface Devices.

Acknowledgment

We would like to thank everyone who has helped in making this project a reality. We especially thank our guide Prof. Narendra Shekokar, for providing us the necessary feedback and backing us throughout the project. We would also like to thank our family members, without whose support and care such an effort would not be possible. Finally, we would like to thank our college, DJSCOE for providing us the necessary facilities that we needed during this project.

References

[1] H. Du, T. Oggier, F. Lustenburger and E. Charbon, “A virtual keyboard based on true-3D optical ranging,” Proc. British Machine Vision Conference (BMVC), Oxford, pp. 220-229, Sept. 2005.

[2] Andrew D. Wilson, PlayAnywhere: a compact interactive tabletop projection-vision system, Proceedings of the 18th annual ACM symposium on User interface software and technology, October 2326, 2005, Seattle, WA, USA

[3] Kölsch, M. and Turk, M. Keyboards without Keyboards: A Survey of Virtual Keyboards, Workshop on Sensing and Input for Media-centric Systems, Santa Barbara, CA, June 20-21, 2002.

[4] Matsui, N. & Yamamoto, Y. A New Input Method of Computers with One CCD Camera: Virtual Keyboard. Keio University, 2000.

[5] Thad Starner, "Keyboards Redux: Fast Mobile Text Entry," IEEE Pervasive Computing, vol. 3, no. 3, pp. 97-101, July-Sept. 2004.

[6] Segen, J., and S. Kumar, Shadow Gestures: 3D Hand Pose Estimation Using a Single Camera in Proceedings of Computer Vision and Pattern Recognition, (1999), 479-485.

[7] Mantyjarvi J., Koivumaki J., Vuori P., “Keystroke recognition for virtual keyboard”, Multimedia and Expo, 2002. IEEE International Conference on Volume 2, Issue, 2002 Page(s): 429-432 vol.2.