Proceedings of the 7th International Caribbean Conference on Devices, Circuits and Systems, Mexico, Apr. 28-30, 2008

A Fingertip Detection and Tracking System as a Virtual Mouse, a Signature Input Device and an Application Selector

Abhimanyu Sanghi[1], Himanshu Arora[1], Kshitij Gupta[1], Vipin B. Vats[2]

[1]Department of Electronics & Communication, Netaji Subhas Institute of Technology, University of Delhi, New Delhi, India
[2]Formerly with the Department of Electronics & Communication, Netaji Subhas Institute of Technology, University of Delhi, New Delhi, India
E-mail: abhimanyu.sanghi@nsitonline.in, himanshu12.arora@gmail.com, kgupta07@gmail.com, vipvats@gmail.com

***Abstract*-This paper presents a vision-based fingertip detection and tracking system on a 2D plane using a camera as an input interface to the computer. An intensity based approach is used to detect the arbitrary shaped, uniform colored 2D area on which the hand operates, and then the fingertip is effectively detected and tracked using the sampled hand contour. Grid sampling approach is used for a fast implementation. The system achieves speeds of up to 30 fps. Tip pointers such as a stylus or a pen can be used in place of the fingertip, making the device user-friendly. The system is used to implement a virtual mouse, a signature input device and an application selector.**

I. Introduction

In the field of computer vision, many prototypes of intelligent vision-based interface systems have been developed [1-6] that are more intuitive and cost-effective than the conventional interface devices such as the mouse and the keyboard. We describe such a system that detects and tracks the fingertip in a 2-D plane, and then demonstrate its application as a virtual mouse, an application selector and a signature input device.

The system is shown in Fig. 1. It comprises a panel, first used in [1], which is a 2D object of an arbitrary shape and a uniform color. The panel acts as a pad on which the finger is to be operated. A camera is focused on the panel to observe the tip pointer. This acts as the interface link between the user and the computer. Firstly, we present an overview of the system. Then, we present an analysis of the system to describe in detail, the techniques used for detecting the panel and detecting and tracking the fingertip. Finally, we present the application of the fingertip detection and tracking system as (a) a virtual mouse, (b) a signature input device and (c) an application selector. The system achieves a speed of up to 30 frames per second on a computer running on Intel Core 2 Duo processor, 1.73 GHz with 1GB RAM.

II. System Overview

The system, as shown in Fig. 1, comprises a panel, a camera focused on the panel and a computer to which the camera is connected. The motion of the fingertip is observed on the panel by the camera, and images are continually transferred to the computer. This acts as a link for interaction between the user and the computer.

The panel can be any arbitrary shaped 2D object with a uniform color. For ease of analysis, we consider a white A4 sheet that is easily available at homes and offices. The panel must not be bent as this affects panel detection. The panel is to be kept static with respect to the camera. Good lighting conditions are required for accurate panel detection. In place of a fingertip, pens or other arbitrary pointing devices of suitable thickness can be used, as long as their color is distinguishable from the color of the panel.

Fig. 1 The fingertip detection and tracking system

III. System Analysis

Firstly, we define a virtual image, which is of the same size as that of the captured image and is black in color, i.e. the value of all pixels is taken to be 0 (on a scale of 0 to 1). System analysis can be divided into three parts: Panel detection, hand detection, and fingertip detection and tracking.

A. Panel Detection

For a fast implementation with high-resolution images, we use the method of grid sampling as described in [2]. The captured image is divided into grids of block size 10 pixel x 10 pixel. Each block is referred to as a grid pixel. Hence, an image of size 640x480 pixels is transformed to an image of size 64x48 grid pixels. Each captured image is separated into its 3 component images corresponding to red, green and blue. In each of the three images, each grid pixel is analyzed. If the intensity of 95% pixels, i.e. 95 pixels out of 100 in a grid pixel, is greater than 0.85 (an empirical value on a scale of 0 to 1 for a white A4 sheet), then the corresponding grid pixel in the virtual image is set to 0.5. Hence, the panel in the captured image corresponds to a gray region in the virtual image. A number of consecutive captured images are analyzed to check whether the panel is static or not. This is done by comparing the position of the gray pixels in the consecutive virtual images. When it is ascertained that the panel is static, the virtual image is saved in a permanent location and the panel is said to be successfully detected. The actual intensity of the panel is also stored at a permanent location. A command is outputted by the system that the fingertip can be brought on the panel. An example of panel detection is shown in Fig. 2.

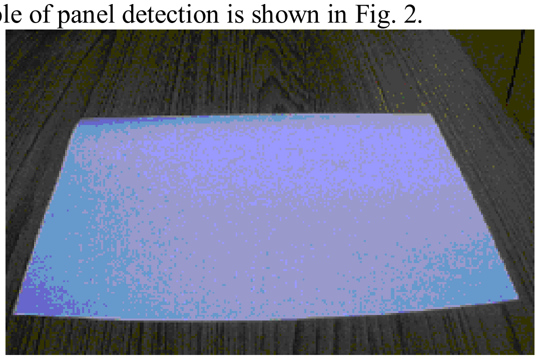

Fig. 2(a) Captured image of panel

Fig. 2(b) Corresponding virtual image of panel

B. Hand Detection

When the hand is brought on the panel, the grid pixels of the captured image are analyzed again. If 50% of all pixels of a grid pixel lying on the panel have intensity less than .65 (an empirical value on a scale of 0 to 1 for a hand, taking values between 0.65 and 0.85 as a buffer), then the corresponding grid pixel in the virtual image is set to 1. Hence the hand in the captured image corresponds to a white region in the virtual image. An example of hand detection is shown in Fig. 3.

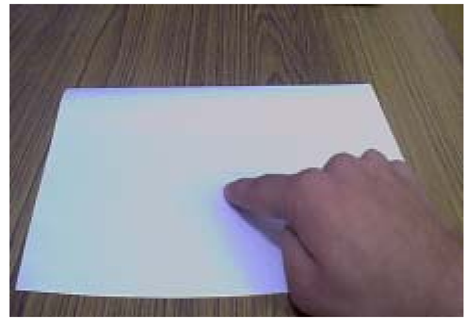

Fig. 3(a) Captured image of hand on panel

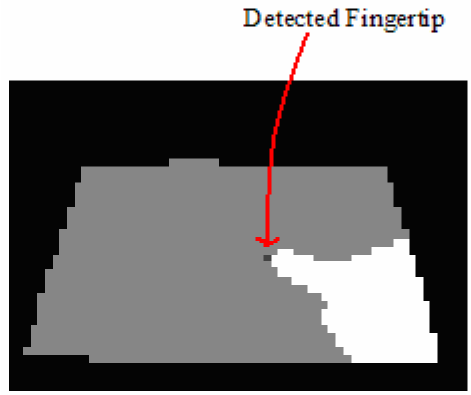

Fig. 3(b) Fingertip detected in the corresponding virtual image

C. Fingertip Detection and Tracking

For fingertip detection, a square of 7 grid pixel x 7 grid pixel with a white grid pixel at its center is considered, for every white grid pixel present on the panel. The number of white grid pixels in this square is calculated. The white grid pixel whose corresponding square has the least number of white grid pixels is the required position of fingertip. The coordinates of this tip position are calculated and returned to the system. This tip position is continuously updated for incoming images, and by this means the tracking of fingertip is achieved. The method is graphically shown in Fig. 4, where the square corresponding to the X marked grid pixel is constructed, and it consists of 12 white grid pixels. A buffer region of 3 pixels is left on the edge of the detected panel for an efficient algorithm. An example of fingertip detection is shown in Fig. 3.

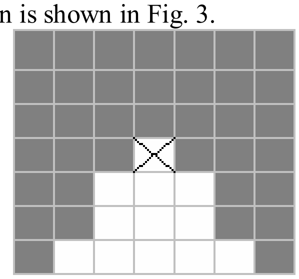

Fig. 4 A square corresponding to the X marked white grid pixel

IV. APPLICATIONS

A. The System as a Virtual Mouse

The new coordinates of the tip location are compared with the previous coordinates, and the difference between the coordinates is algebraically added to the previous coordinates. This results in mouse motion. The speed of the mouse can be controlled by using a multiplying factor M to multiply the difference between coordinates before addition to previous coordinates. Mathematically:

$$P_{x,t} = M*(C_{x,t} - C_{x,t-1}) + P_{x,t-1} \quad (1)$$

$$P_{y,t} = M*(C_{y,t} - C_{y,t-1}) + P_{y,t-1} \quad (2)$$

Where, P denotes the pointer location on screen,
C denotes the coordinates of tip,
M denotes the multiplying factor,
x & y denote the rectangular x & y axes respectively,
t denotes the present time.

Our present implementation supports single clicking mode. The clicking mode is simulated by holding the tip stationary at a location for a short duration of time, say 2 seconds. This simulates the left click of the mouse. The position of the pointer may not remain constant due to noise and hence if the fluctuation is within 1 grid pixel, the tip is considered immobile.

B. The System as a Signature Input Device

The Signature Capture ON instruction is generated by holding the tip pointer stationary at a location for a duration of time, say 2 seconds. When the application is activated, the user is asked to input his/her signature. The tip pointer coordinates obtained from successive images are joined by straight lines as shown in Fig. 5. The Signature Capture OFF instruction is generated by holding the tip pointer stationary at a location for a short duration of time after completing the signature, say 2 seconds. This application provides an efficient and economical way of obtaining digital images of signatures.

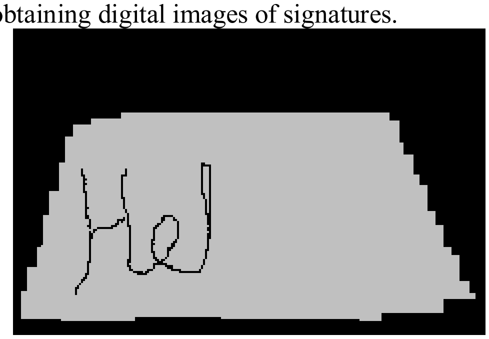

Fig. 5 Virtual image during the signature input process

C. The System as an Application Selector

In this application, the panel acts like a touch pad to initiate different commands/programs/applications. We implemented a touch-pad that provides 4 options of applications to the user. The A4 sheet is divided into blocks each specifying a particular application (as shown in Fig. 6), and the commands that trigger the application corresponding to each block are pre-stored in the memory. The paper orientation is fixed. After panel detection, the coordinates of the four vertices of the A4 sheet are stored at a memory location, and with their help, the equations of the diagonals of the quadrilateral panel are determined. Let the equations of the two diagonals be:

$$Ax + By + C = 0 \quad (3)$$

$$Dx + Ey + F = 0 \quad (4)$$

Where A, B, C, D, E and F are constants; x and y are variables of the Cartesian coordinate system.

The 4 blocks, as shown in Fig. 6, can be specified in terms of the equations of the diagonals as:

$$\text{Block 1: } Ax + By + C < 0 \text{ and } Dx + Ey + F < 0 \quad (5)$$

$$\text{Block 2: } Ax + By + C < 0 \text{ and } Dx + Ey + F > 0 \quad (6)$$

$$\text{Block 3: } Ax + By + C > 0 \text{ and } Dx + Ey + F < 0 \quad (7)$$

$$\text{Block 4: } Ax + By + C > 0 \text{ and } Dx + Ey + F > 0 \quad (8)$$

An application is selected by holding the finger stationary on the corresponding block for a short duration of time, say 2 seconds. The system determines the block in which the tip pointer is held constant with the help of (5) – (8) and activates the pre-stored commands corresponding to that block.

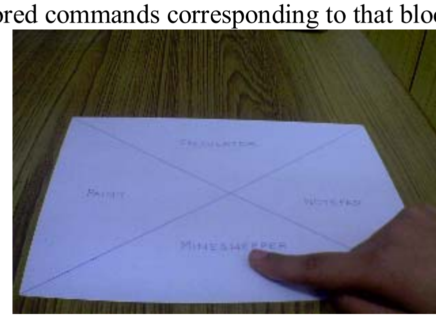

Fig. 6 The panel for application selector

V. CONCLUSION

We have developed a prototype of a vision-based fingertip detection and tracking system on a 2D plane based on the intensity of captured images. This approach combined with the method of grid sampling results in speeds up to 30 frames per second with images of resolution 640x480 pixels. This is higher than the speeds that have been achieved by some similar methods as documented in [2]. The system uses an arbitrary shaped panel of uniform color (e.g. an A4 sheet) and a web-cam, objects that are inexpensive and easily available in present day organizations. The fingertip can be substituted by a tip pointer such as a stylus or a pen. This makes the device convenient and user-friendly.

Three applications have been developed based on the fingertip detection and tracking system. The virtual mouse is an economical substitute of the hardware mouse. The signature input device eliminates the complexity involved in devices such as digital pens or touch-sensitive pads, and can be used in daily life to store digital images of signatures. This can find application in organizations such as banks. The application selector provides an inexpensive alternative to touch-sensitive pads and screens.

Future work on the system can include the integration of a character-recognition system with this system. Various extensions and improvements in the fingertip detection and tracking system can be explored.

REFERENCES

[1] Ying Wu, Ying Shan, Zhengyou Zhangy, Steven Shafer, "Visual Panel: From an ordinary paper to a wireless and mobile input device," Technical Report, MSR-TR-2000 Microsoft Research Corporation, http://www.research.microsoft.com, October 2000.

[2] Duan-Duan Yang, Lain-Wen Jin, Jun-Xun Yin, Li-Xin Zhen, Jian-Cheng Huang, "An effective robust fingertip detection method for finger writing character recognition system," Proceedings of the Fourth Int'l Conference on Machine Learning and Cybernetics, Guangzhou, pp. 4191 – 4196, August 2005.

[3] J. L. Crowley, F. Berard and J. Coutaz, "Finger tracking as an input device for augmented reality," Proceedings of Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, pp. 195-200, June 1995.

[4] F. K. H. Quek, T. Mysliwiec and M. Zhao, "Finger Mouse: A freehand pointing computer interface," Proceedings of Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, pp. 372-377, June 1995.

[5] Christian. V. H, François. Bm., "Bare-hand human computer interaction," Proceedings of the 2001 workshop on Perceptive user interfaces, Orlando, Florida, USA, pp. 1-8, Nov 2001.

[6] Oka. K, Sato. Y, Koike. H, "Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface system," Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 411 – 416, May 2002.

[7] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, 2nd ed, Prentice Hall, 2002.