

Bare Finger based Touch detection in Camera System

T.S.Kumaran.¹, J.Arul Murugan², Mr.T.Sengolrajan³

^{1,2}U.G Scholar, Department of EIE, Arunai Engineering College, Thiruvannamalai.

³ Assistant Professor, Department of EIE, Arunai Engineering College, Thiruvannamalai.

E mail: kmrntsk@gmail.com¹, arulmuruganrj@gmail.com², sengolmaha@gmail.com³.

Abstract- This paper proposed a real time application based on Interactive Projection System (IPS), which enables bare-finger touch interaction on regular planar surfaces like walls and tables. The hardware requirements of this system are kept on the minimum, limiting only to a simple USB webcam and a PC. Other alternatives to the PC have also been presented, such as the Beagle Board, which is a mini version of a PC and very inexpensive. The challenge of bare-finger touch detection is recovering the touching information just from the 2-D image captured by the camera. In this paper, the buttons are displayed on the surface or paper and is disturbed by the finger when clicking it. Hence a new breed technology proposed which is a novel is fast and robust algorithm used to detect the touch action.

Keywords: Open cv, Raspberry pi, USB camera, segmentation

I. INTRODUCTION

With an increasing demand for switching into new and virtual environment from the old physical environment where hardware has been a problem due to continuous use of it. Virtualization of keyboard to display has been carried out since decades but the precision of functionality has become a problem. In the existing system, a image processing in a separate PC and software used for image processing like MATLAB, which is not an open source. In this paper the entire module is in a designed single system on chip. Portable CPU with arm processor and software used in this paper for image processing completely is a open source. An additional image processing controller is added in beagle board xm, which increase the performance of image processing more than desktop PC's. In this paper a new breed technology proposed overcome the advantage of the button's distortions caused by the fingers to detect the touch operation on the screen. If a button is clicked by the finger, then the shape of the button will change in the camera's image plane (CIP). Instead of tracking the hand's 2-D position, which is also recognized as a challenging work in computer vision, This paper focus on detecting the deformation of the buttons to determine the touch action on the surface. The detection of a

button can be carried out using edge detection. In computer vision and image processing the edge detection technique is a mature technology that Approaches for implementing first-order and the second-order digital derivatives for the detection of edges in an image are very effective.

II. SYSTEM COMPOSITION AND PRINCIPLE

A. Block Diagram of system compoents

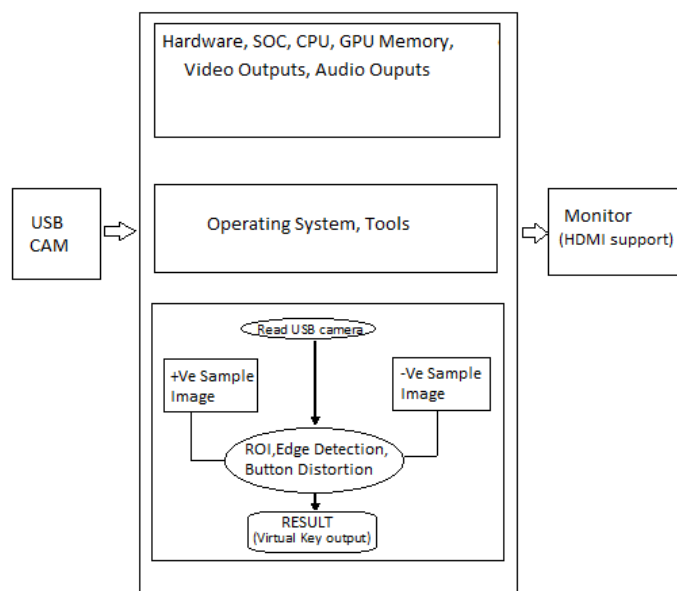


Fig.1 Block diagram of system components

The most important advantage involved in this system is to make the system independent of on-board processing environment which uses most of the battery power as well as the independency of each module makes it easier for upgrade to newer feature.

B.Design of Virtual Keyboard System

Virtual keyboard system is designed keeping in mind the maximum efficiency of algorithm and lower error rate that will

not make user feeling different response rate a physical keyboard. The design consists of Raspberry pi Board and USB Camera. The above modules are used to design a proper and highly efficient system. For implementing this technology we are using embedded Linux and open CV library.

B.1. LINUX OPERATING SYSTEM

The Linux open source operating system is a freely distributable, cross-platform operating system based on Unix which can be installed on PCs, laptops, net books, mobile and tablet devices, video game consoles, servers, supercomputers and more.

B.2. OPEN CV

Open CV is an open source computer vision library originally developed by Intel. It is free for commercial and research use under a BSD (Berkeley Software Distribution) license. The library is cross-platform, and runs on Linux, Windows and Mac OS X. It focuses mainly towards real-time image processing as it finds Intel's Integrated Performance Primitives on the system use these commercial optimized routines to accelerate itself.

III. SYSTEM HARDWARE DESIGN

The designing of hardware is the system is most important part of the system as exact configuration of components is required to simplify the algorithm and error rate. The first part of the system is camera integrated with Raspberry board which limits the camera sensing capacity only towards the focussed area. The Raspberry board consist of an ARM11 processor which has an considerable image processing speed with 512MB of RAM.

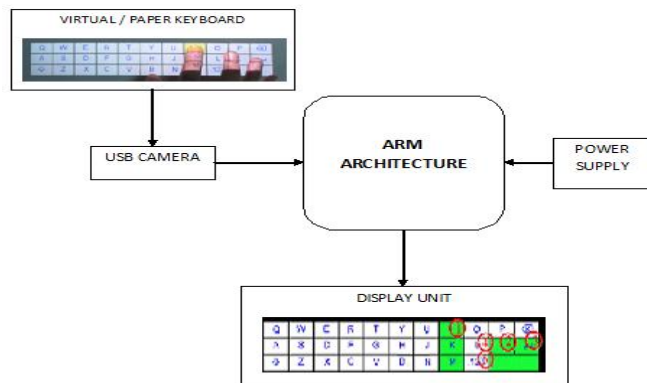


Fig.2 hardware design of system

The necessary operating system is loaded by using appropriate mass storage device. USB 2.0 port which is used for providing power supply to 5V board and power supply to generate the 3.3V

supply. It function by essentially heating' the $5-3.3V = 1.7V$, The the difference and dissipating the power difference in heat. The Raspberry pi is a series of credit card-sized single-board computers developed in the UK by the Raspberry pi Foundation with the intention of promoting the teaching of basic computer science in schools. The original Raspberry pi is based on the Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHz processor, VideoCore IV GPU and it was originally shipped with 256 megabytes of RAM, later upgraded (models B and B+) to 512 MB. The system has Secure Digital (SD) (models A and B) or MicroSD (models A+ and B+) sockets for boot media and persistent storage.



Fig.3 Raspberry pi Board (B+ Type)

IV. FLOW CHART FOR 2-D IMAGE PROCESSING

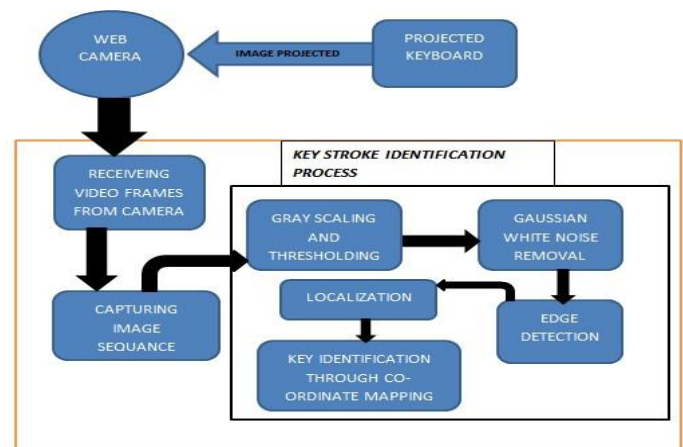


fig.4 flow chart for 2-D Image Processing

Major steps involved in the finger detection using a camera are:-

- Smoothing for noise removal,
- Color conversion,
- 1st frame is taken as background,
- Background subtraction,
- Thresholding and
- Smoothing for additional noise removal.

It deals with identifying the hand in a general video frame and distinguishing it from the background. The pixels in a frame corresponding to the hand need to be segmented and separated so as to use it for further processing. Background subtraction is used for achieving the result. Before background subtraction, smoothing function is used to remove noise. The cvSmooth function with a Gaussian Filter is used for the above said. Then color conversion of the input image from RGB to grey-scale is done.

V. CAPTURING IMAGE FRAMES FROM CAMERA

If any problem related to finger tracking or gesture recognition, then usually a few constraints regarding the background, the object, speed of motion etc. The system designed by us works on the following basic assumptions which includes the camera (webcam) is supposed to be fixed and motionless. The background must have a uniform texture. The background should contrast with the general skin tones. The only features moving in a frame should correspond to the hand i.e. no other object moves in the background and the Shadows should be kept to a minimum.

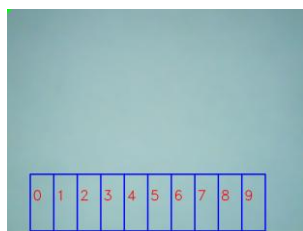


fig.5 Initial 2-D Image

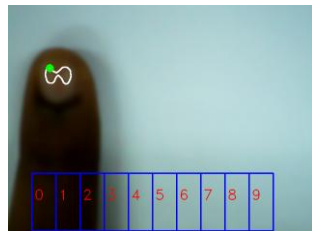


fig.6 Image capture after initial image

Initially the camera takes in the first instance of the reference image and stores it in the memory for comparison with the next image. Once the finger is placed over the region of interest the camera takes an 2D image of the finger and starts to process. The image pixel values are stored and then converted into binary set of inputs.

VI. FILTER

Filter removes away the noise level to a greater extend. Since image to be processed needs to be much precise noise distortion level of the image should be very low. Probably the most useful filter. Gaussian filtering is done by convolving each point in the input array with a *Gaussian kernel* and then summing them all to produce the output array.



Fig.7 Grayscale image

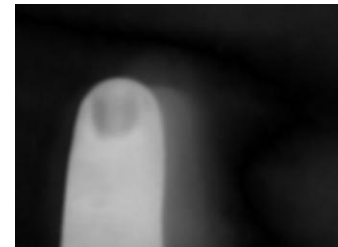


fig.8 smoothed image

The image is converted into grey scale to avoid errors, and then it gets smoothened by using the filters in order to remove the noise level. Gaussian filter is used because it removes the unwanted noise level in the image to a greater extend. And also it has higher efficiency when compared to other filters (laplacian filter, median filter). The filter applied image is much clear as it removes away the unwanted distortions present in the image.

VII. SEGMENTATION

The process of segmentation involves the isolation of the area around the finger to be detected. The area is isolated and mapped into a separate region of image to get easily processed. After the segmentation of the finger, we next move to extracting the finger region from the segmented image. We use the cvFindContour command to extract all the contours corresponding to probable hand regions. These contours trace the outline of each region. We then compute the contour having the largest area among those. As the largest object moving in the image should correspond to the finger, we conclude that the largest contour must denote the finger. Once the largest contour is found out we use curvature determination to find out the peaks and valleys in the finger.

VIII. BACKGROUND SUBTRACTION

Finally the image background is removed by comparing between the image, this gives only the finger to be detected instead of the whole image to get processed. Comparison between

the first occurrence of the image and the later occurrence of the image are then taken after segmentation. This method removes away the similar regions of the image by using the binary coded values of the image. Also by using the technique of the



fig.9 Background subtracted image

HSV- Hue Saturation Value the image colour are removed by choosing the colours of the appropriate values. Here for detecting the finger the pixels range is chosen such that it matches the colour of our finger.

IX.AREA OF CONTOUR

The region of the image gets splitted into separate regions known as area of contour which maps the image by giving the image with separate matrix form, the position of a point in the image is given with respect to the mapping of the X and Y axis values.

X.IDENTIFICATION OF THE PEAK POINT

The tip of the finger is detected by using the Peak and Valley points plotted over the image contour. Now the peak points are noted and then the highest value is determined so that the peak of the finger gets determined. This value of the X-Y reference axis is noted and mapped to the values located in the given frame developed by the user. The frame gives a set of space to draw a virtual keyboard that contains specific functional keys. And the values of the peak is taken which maps to a point on the frame, finally the corresponding key function is executed. By having the finger over the region of interest the key functions gets accessed.

The size of the keyboard is limited depending upon the camera resolution. As the camera can cover only a certain region of frame into process.

CONCLUSION

using the open cv commands into action the bare finger touch detector using camera system can be done very effectively, this

overcomes the usage of projector for virtual keyboard design. Also the usage of visual studio leads to memory error in windows and so a newer approach is implemented by using the linux platform which also removes away the cache periodically. The design of the virtual keyboard is also user dependent based on the allocation of frame space.

REFERENCE

1. F.Y. Shih, C.-F. Chuang, V. Gaddipati, "A modified regulated morphological corner detector Pattern Recognition Letters", vol. 6(7), pp. 931-937, 2005.
2. H. Bay, A. Ess, T. Tuytelaars, L. Van Gool. SURF: "Speeded Up Robust Features. Computer Vision and Image Understanding", vol. 110(3), pp. 346-359, 2008.
3. Jun Hu, Guolin Li, Xiang Xie, Zhong Lv, and Zhihua Wang, Senior Member, "Bare-fingers Touch Detection by the Button's Distortion in a Projector-Camera System" IEEE transactions on circuits and systems for video technology, vol. 24, no. 4, april 2014.
4. S. Murugappan, N. Elmqvist, and K. Ramani, "Extended multitouch: Recovering touch posture and differentiating users using a depth camera," in Proc. ACM UIST, 2012, pp. 487-496.
5. C. Holz and P. Baudisch, "Understanding touch," in Proc. ACM CHI, 2011, pp. 2501-2510.
6. J. Salvi, J. Pages, and J. Battle, "Pattern codification strategies in structured light systems," Patt. Recog., vol. 37, no. 4, pp. 827-849, 2004.
7. M. Piccardi, "Background subtraction techniques: A review," in Proc. IEEE Syst. Man Cybern., 2004, pp. 3099-3104.
8. J. Dai and R. Chung, "Making any planar surface into a touch-sensitive display by a mere projector and camera," in Proc. IEEE Workshop CVPR, 2012, pp. 35-42.
9. D. Wilson, "Using a depth camera as a touch sensor," in Proc. ACM ITS, 2010, pp. 69-72.
10. Harrison, D. Tan, and D. Morris, "Skinput: Appropriating the body as an input surface", in Proc. ACM CHI, 2010, pp. 453-462.
11. Priyanka Gaur, Shamik Tiwari, "Recognition of 2D Barcode Images Using Edge Detection and Morphological Operation" International Journal of Computer Science and Mobile Computing, Vol.3 Issue.4, April- 2014, pg. 1277-1282.
12. Maini, Raman, and Himanshu Aggarwal, "Study and comparison of various image edge detection

- techniques." *International Journal of Image Processing (IJIP)* 3.1 (2009): 1-11.
13. M.Basu, "Gaussian based edge-detection methods a survey", *IEEE Trans. Syst., Man, Cybern C*, vol. 32, pp. 252-260, August 2002.
 14. Dagao Duan, Qian Mo, Yueliang Wan and Zhongming Han, "A Detail Preserving Filter for Impulse Noise Removal", in *Proc.ICCASM*, 2010, paper V2-265.
 15. Tony F. Chan, Member, IEEE, and Luminita A. Vese, "Active Contours Without Edges", *IEEE TRANSACTIONS ON IMAGE PROCESSING*, VOL. 10, NO. 2, FEBRUARY 2001.
 16. Mahinda Pathegama & Ö Göl (2004): "Edge-end pixel extraction for edge-based image segmentation", *Transactions on Engineering, Computing and Technology*, vol. 2, pp 213–216, ISSN 1305-5313.
 17. Sajjad Ur Rahman, Zeenat Afroze, Mohammed Tareq, "Hand Gesture Recognition Techniques For Human Computer Interaction Using OpenCv", *International Journal of Scientific and Research Publications*, Volume 4, Issue 12, December 2014 ISSN 2250-3153.
 18. K. Panetta, S. Qasi, and S. Agaian, "Techniques for detection and classification of edges in color images", *SPIE Proceedings*, Vol. 6982, pp. 69820W-69820W-11, 2008.
 19. X. Wang and J. Jian-Qiu, "An edge detection algorithm based on Canny operator". *IEEE Proceedings of the seventh International Conference on Intelligent Systems Design and Applications (ISDA2007)*, 623-628, 2007.
 20. E. Posner, N. Starzicki, and E. Katz. A single camera based floating virtual keyboard with improved touch detection. In *Electrical Electronics Engineers in Israel (IEEEI)*, 2012 IEEE 27th Convention of, page 1 to 5, 2012.
 21. Y. Adajania, J. Gosalia, A. Kanade, H. Mehta, and N. Shekokar. Virtual keyboard using shadow analysis. In *Emerging Trends in Engineering and Technology (ICETET)*, 2010 3rd International Conference on, page 163 to 165, 2010.
 22. M. Hagara and J. Pucik. Fingertip detection for virtual keyboard based on camera. In *Radioelektronika (RADIOELEKTRONIKA)*, 2013 23rd International Conference, page 356 to 360, 2013.