

Research Article

Controlling of Mouse Cursor using Video Processing in FPGA

Rumaisa Iftikhar^Å, Samiya Ashgar^Å, Hira Abbasi^Å, Tahir Qadri^{Å*}, Sadia Abbasi^Å and Naeem Abbas^Å^ÅElectronic Engineering Department, Sir Syed University Of Engineering & Technology, Karachi, Pakistan.

Accepted 01 March 2014, Available online 01 April 2014, Vol.4, No.2 (April 2014)

Abstract

Digital Video processing technology is maturing rapidly with applications widely spread from everyday life to hi-tech engineering. Execution of real-time video processing requires good memory performance and high computational load; therefore its implementation is frequently confined with expensive DSP kits or large workstations. FPGA has proven to be an attractive and efficient alternative for such systems because of its flexibility and the added benefit of portability. An important parameter of video processing is ability to detect moving objects from the frame of the video. This object detection is further enhanced to gesture recognition which is gaining gigantic interest in various fields including image processing and computing comprising various factors as well as constraints nonetheless providing ease in numerous human affairs. In the presented paper, the field of gesture recognition has been narrowed to hand gesture recognition, in particular to palm of the hand used effectively to control the movement of the cursor. A simple but effectual system has been developed where image of a input hand is computed. The approach includes calculation of centroid of the hand where the system is invariant to the scale of the hand. The proposed model is implemented on Altera's DE2-115 FPGA board with input feed from HD webcam C270h capturing 1280 x 720 pixels in YCbCr format. The real life video processing results illustrates that the presented procedure has effective implementation.

Keywords: Video Processing, FPGA, Controlling of cursor, Calculation of centroid

Introduction

Video processing is gaining importance in many fields specially in surveillance, security and communication systems. Most of the systems require huge amount of information to be processed in real time. Previously, video processing required high computational work with computers and standard processors but to meet the demand of smaller size equipment with less power consumption specialized hardware equipments are evolved such as digital signal processing kits (DSP) or field programmable gate arrays (FPGA).

The applications developed using FPGA's have shown noteworthy speedups compared with software and DSP based approaches for a number of real time video processing applications. FPGA based systems have better expandability and also have higher performance. FPGAs are more suitable for applications in time-critical systems. The aim of this paper is to process the real time video using FPGA, in order to control the movement of mouse cursor by using hand gestures. As it is strongly evident that the communication between human and sensor based devices are gaining importance day by day in our society. Human gestures are interpreted by means of mathematical algorithms and there is no particular hand gesture recognition method suitable for all applications. It enables human to communicate with machines and interact in a

natural manner with the least amount of hardware involvement. This touchless interface is beneficiary in many applications like in health care environments in order to view CT scans in operation theatre. The free space hand gesture is also useful for overcoming physical handicaps. Distinguished companies such as Sony PS Vita (Play Station Vita and PlayStation Network <http://www.playstation.com/ps2>), Nintendo Wii (Wii Official Site at Nintendo <http://www.nintendo.com/wii>), Microsoft Kinect (Kinect <http://www.microsoft.com/en-us/kinectforwindows>), etc., built realistic gaming environment by including gesture recognition. Samsung (Samsung Group <http://www.samsung.com/pk>) introduced Smart TV based on similar technology.

The main aspiration of this research is to facilitate the persons to control their mouse cursor by using their hand movements. For this purpose, we interfaced the video camera with FPGA kit and processed real time video signal. Then different image processing algorithms like color space conversion and edge detection is implemented on FPGA. The next section discusses the system model.

System Model

This proposed system is mainly consists of four parts. The first part is the processing of video signal by using FPGA development kit. The second part is the detection of the object from the acquired video signal from the FPGA. The third part is the tracking of the hand and finally the last

*Corresponding author: **Tahir Qadri**

part is the control of cursor using hand gestures. The overall block diagram of the proposed system is illustrated below.

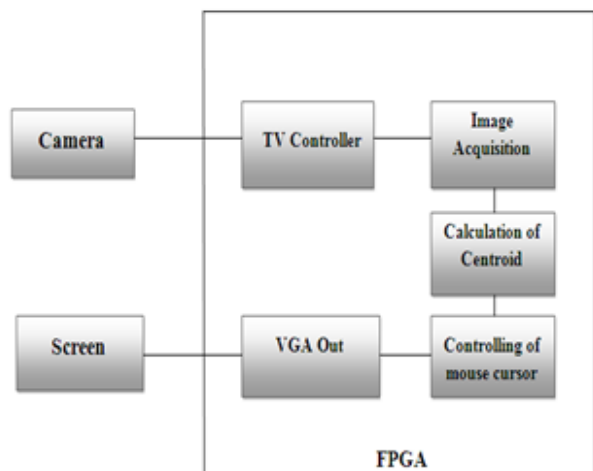


Fig. 1 Block diagram of the proposed model

The next section explains the interfacing of video camera with FPGA kit.

Interfacing of Video Camera with FPGA

The first step is to connect the camera with the FPGA development kit. The web camera used here, is the Logitech HD webcam C270h (Logitech <http://www.logitech.com/en-in/product/hd-webcam-c270h>), which can capture video up to 1280 x 720 pixels and has hi-speed USB 2.0 interface. The FPGA kit used in this work is Altera’s DE2-115 board (Altera’s DE2-115 <http://www.altera.com/education/univ/materials/boards/de-2-115/unv-de2-115-board.html>) which has I/O interfaces, memory, displays, leds and clock on board. The I/O interface of this board consist of built-in USB-Blaster for FPGA configuration Line In/Out, Microphone IN (24-bit Audio CODEC), Video Out (VGA 8-bit DAC), Video In (NTSC/PAL/Multi-format), RS232, Infrared input port, PS/2 mouse or keyboard port, Two 10/100/1000 Ethernet. Initially the web camera is fed in the composite video input port of FPGA kit. The kit also has TV decoder which converts the analog data into raw digital data after that it is fed to the FPGA kit. The 8 bit data from web camera is saved in the SDRAM of the DE2-115 FPGA board pixel by pixel occupying 16 bytes of memory. The FPGA kit has total memory of 128 MB SDRAM, 2 MB SRAM, 8 MB Flash SD memory card slot. The video data is generally in YCbCr format. Hence it is first converted into RGB format and then sent to the VGA output of the board. As soon as the bit stream is downloaded into the FPGA, the register values of the TV Decoder chip are used to configure the TV decoder via the I2C_AV_Config block, which uses the I2C protocol to communicate with the TV Decoder chip. Following the power-on sequence, the TV Decoder chip will be unstable for a time period; the Lock Detector is responsible for detecting this instability. The DE2-115 board is equipped with an analog device ADV7180 TV decoder chip. It is an integrated video decoder that detects and converts a standard analog

baseband television signals in to 4:2:2 component video data compatible with the 8-bit ITU-R BT.656 interface standard. ITU-R decoder block used to extract YcrCb 4:2:2 video signals from the data stream sent from the TV Decoder. It also generates a data valid control signal indicating the valid period of data output. Because the video signal from the TV Decoder is interlaced, we need to perform de-interlacing on the data source. We used the SDRAM Frame Buffer and a field selection multiplexer (MUX) which is controlled by the VGA controller to perform the de-interlacing operation. Internally, the VGA Controller generates data request and odd/even selection signals to the SDRAM Frame Buffer and filed selection multiplexer (MUX). The YUV422 to YUV444 block converts the selected YcrCb 4:2:2 (YUV 4:2:2) video data to the YcrCb 4:4:4 (YUV 4:4:4) video data format. Finally, the YcrCb_to_RGB block converts the YcrCb data into RGB data output. The VGA Controller block generates standard VGA synchronous signals VGA_HS and VGA_VS to enable the display on a VGA monitor.

Image Acquisition/Processing in FPGA

Image acquisition in FPGA enhances the speed of image processing whereas image acquisition systems implemented using software by the DSP chip, sometimes cannot meet the demand of high image processing speed (Chao Li *et al* 2009). The image acquisition and VGA display core is used in image processing algorithm. Here the FPGA development kit captures video from camera then precedes the video and displays its contents on the LCD or monitor. As said that an image is a set of pixels so for the generation of a video, the scanning of each and every pixel is necessary. So video is generated by scanning the pixels and each pixel represented by a value or set of values. The scanning starts from the right most pixels to the left most pixels in the first row and then comes back to the next row and then start from the right most pixels towards the end of the row and so on. Once after the scanning entire image then it again returns back to the starting point.

The interfaced video camera with FPGA shows its final result in RGB format on screen. For the processing of images or frame, the grayscale image is required which is “Y” part of YCbCr format image. When the object detection mode is started, the process begins as the acquired color image is converted to grayscale which ranges 0 to 255 values. It is required that the first frame coming from the camera as input should be subtracted from the previous frame so that the moving object can be detected. It is observed that all the moving objects are detected during subtracting of frame either in the background or foreground. The background should be static during the process as a limit. Then the image is further processed from grayscale to binary. The binary image has only two values i.e. 0 and 1. The background contains ‘1’ value and foreground contains ‘0’ value. The detected moving object consists of “0” values. The step is used to get the skin pixels. The moving detected object allocates ‘0’ binary value and background consists of ‘1’ value. Then the object, that is, the hand, is shaded black,

representing '0' in binary, whereas, the background is visible in white shade, hence representing '1' in binary. The negative aspect of this situation is that the object is represented as '0', any number multiplied by it will also give '0'. To solve this dilemma, complement of the image is obtained, background converts to black, i.e., '0' and object, hand, becomes white, i.e., '1'. Complementing the image results object detection with binary values of '1'. The complemented image is used to define the area and other properties of detected object and here it is our hand. The detected object i.e. hand is showed binary values of '1' and other than skin pixels show '0' binary values. In the next step, unnecessary noise is filtered out as the image is passed through Median filter. As there is a lot of noise and lot of things has skin color therefore filter is used to reduce the noisy effect of background and other skin pixels. To do so, we allocate the neighboring pixels with skin pixel by using the technique that if the connected components has pixels greater than 300 pixels then they are skin pixels otherwise they are considered background. Then the final image is free of noise and skin pixels are allocated each other. The steps involved in image acquisition are illustrated in figure 2 whereas the images showing their result is in figure 3.

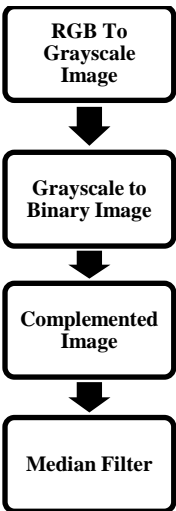
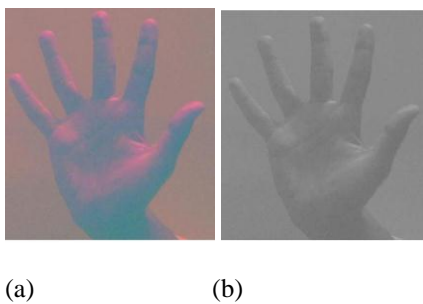


Fig. 2Steps involved in image acquisition



The next section explains the calculation of centroid.

Calculation of Centroid

Since centroid is the center of the object and for images, it is the center of the connected components or collection of

'1' values of the detected object. Resultant image is used to determine the centroid of the detected object that is our hand, in our case. The connected components having 8 neighboring pixels is considered to be a region or single area.

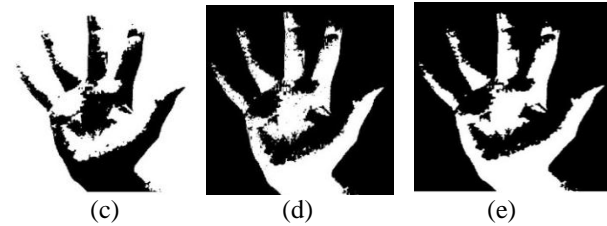


Fig. 3 Images during the image acquisition process (a) Original image (b) Grayscale image (c) Binary image (d) Complemented image (e) Filtered image

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Fig. 4 Centroid calculation of a binary image

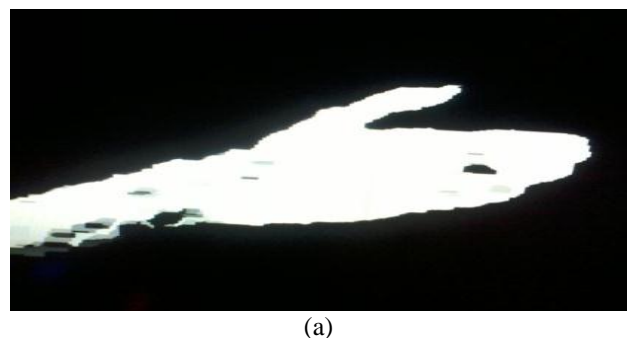
In order to calculate the centroid of an image, the following formula is used (Ma Wei-Ying et al 1999):

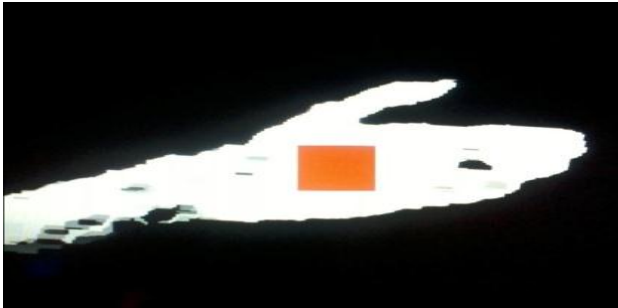
$$\text{Centroid} = C(x,y) = \left\{ \frac{U(1,0)}{U(0,0)}, \frac{U(0,1)}{U(0,0)} \right\}$$

Centroid is the average mean of x and y coordinates of the detected object. The connected components of 8 pixels have defined area and centroid. By computing the above formula, the average centroid of each defined area is obtained and saves in an array. The centroid of first frame obtained from video camera is saved in an array and this centroid is compared with the next frame. The centroid from the next frame updates the centroid of previous frames and saves in an array. The process goes on and the centroid of the detected moving object which is our hand saves in SDRAM of FPGA development kit. The next section explains the controlling of the mouse cursor.

Controlling of Mouse Cursor

The location of centroid calculated above is used to control the bounding rectangular box which is allocated to the hand. As the hand moves, the centroid updates and the red rectangular box moves to the place where the hand is.





(b)



(c)

Fig. 5 Controlling of mouse cursor

The horizontal and vertical synchronous bit of the VGA output port is also updated with reference to the x and y axis of centroid of the frame respectively.

The tracking algorithm is used in such a way that it monitors the selected object first whether it is in center or not. If the object is at the center of the screen then FPGA kit gives 00 output through its general purpose input and outputs (GP-I/Os) and if the object is at right side of the display screen then FPGA development kit gives 01 as output and if the object is at left side on the screen then FPGA development kit gives 10 output through its GP-I/Os. In this way the tracking is performed by moving the camera depending on the object movement.

Working of the System /discussion

The development of FPGA has tremendously improved applications such as image processing in previous years. While processing image data, the data is required to be stored and transmitted to different functional units in real time. Here initially, the camera is interfaced with the FPGA kit where the video format of the camera's output is converted from YCbCr format to RGB format. During further image procession, the frame is converted from RGB to grayscale and from grayscale to binary. After these conversions, image is complemented and then filtered in order that detected object's area is defined. Resultant image is used to determine the centroid of the detected object since centroid is the average mean of x and y coordinates of the detected object. The centroid of every specified area is attained and saved in array.

The centroid of first frame is compared with the following frame obtained from video camera which updates the centroid of previous frames and saves in an

array. The bounding rectangular box, assigned to hand, is controlled through the calculated centroid. With the movement of hand, box moves with respect to hand. The tracking algorithm is not dependent on the selected object to be in center for proper function. Bounding box is controlled by moving camera regarding object movement.

Conclusion and Future Work

The technology regarding video processing and image acquisition is successful. However, previous trend for image acquisition and processing required the implementation of software using PC or DSP kit. Occasionally, the software is incapable of meeting the requirement of the high speed during image processing. This article aims in simple application of controlling the mouse cursor by using hand gesture through video processing via FPGA. FPGA is an extremely useful tool for the building of algorithm and speeding up the process. In future this method can be modified to distinguish different class objects in real time video or this can be used to perform obstacle avoidance for robots or cars etc. The intended system by above mentioned approach attains a satisfying effect by experimental verification. Hand detection is performed and centroid of a hand is calculated and by using the value of centroid, cursor of mouse move according to the hand movement.

The system can be advanced by adding abilities like dragging of objects e.g. icons of desktop to one place to other. The clicking and double clicking facilities can also be implemented as future work. The whole system can act as standalone sensor for a better replacement of mouse. Further detection of objects can also be enhanced with respect to face tracking algorithms. Detection of face in front of a TV leads to switch on or off the television or also reduce or increase the brightness of a TV which can bring power saver facility. A tracking of face can also stop the playing video whenever user sees other things during the movie etc. The application can further be enhanced for face recognition or hand recognition which can be big step in security applications.

References

- Play Station Vita and PlayStation Network, PlayStation2. Available at [http:// www. playstation.com/ps2](http://www.playstation.com/ps2)
- Wii Official Site at Nintendo. Available at <http://www.nintendo.com/wii>
- Kinect for Windows. Available at <http://www.microsoft.com/en-us/kinectforwindows>
- Samsung Group. Available at <http://www.samsung.com/pk>
- HD Webcam C270h - Logitech. Description available at <http://www.logitech.com/en-in/product/hd-webcam-c270h>
- Altera's DE2-115 Development and Education Board. Available at <http://www.altera.com/education/univ/materials/boards/de2-115/unv-de2-115-board.html>
- Chao Li, Yu-lin Zhang and Zhao-na Zheng (2009), Design of Image Acquisition and Processing Based on FPGA, *International Forum on Information Technology and Applications*, pp.15-17
- Ma Wei-Ying, and Bangalore S. Manjunath (1999), Netra: A toolbox for navigating large image databases, *Multimedia systems 7.3*, pp. 184-198