AN FPGA-BASED VIRTUAL PAINTING BOARD

Ju Seok Lee [a], Jeong Beom Kim [b]
[ab] Kangwon National University, Chuncheon, Republic of Korea
Corresponding email: kimjb@kangwon.ac.kr

Abstract

In this paper, we describe the design and implementation of a virtual painting board in an FPGA board. The system has a CMOS camera as an input device and a VGA monitor as an output device. Users can draw lines, colour a picture or both and can follow it on the VGA screen. We have used VHDL and Verilog-HDL for the hardware design. The system hardware consists of CCD capture, RAW2RGB, SDRAM, Main Ctrl, M4K, and VGA controller. The system detects yellow and red colour with the camera and stores and converts the coordinate. Based on Altera DE2 board, we have used 1.3 M pixel CMOS camera, VGA monitor and coloured bands for this system. The design is developed by QuartusII11.1 software.

Keywords: Virtual Painting Board, Gesture Detection, Colour Segmentation, Fingers with Coloured Band, FPGA &Embedded System.

1.Introduction

The growth in human-computer interaction field has not only been in quality of interaction, it has also experienced different branching in its history. Instead of designing regular interfaces, the different research branches have had different focus on the concepts of multimodality rather than unimodality, intelligent adaptive interfaces rather than command/action based ones, and finally, active rather than passive interfaces. Consequently, human-computer interaction systems have been widely used for training purposes in aerospace, military, and medical fields in order to overcome the limits in real training environments (Karray et al., 2008).

Virtual painting board system is needed as a result of the increasing demand for the human-computer interaction. To draw coloured pictures on computer screen, the virtual painting board system is needed. The system can be used to draw coloured pictures on any plain surfaces using fingers with coloured bands and hand movements (Lee et al., 2010; Petouris et al., 2009; Wang et al., 2012). This system draws its motivation from the sixth sense where using gesture detection and colour segmentation techniques (Chien et al., 2002).The constant reduction both of cost and size of image sensors and the increasing complexity of FPGA circuits let us to design and implement an FPGA-based digital camera system. Furthermore, the flexibility of FPGAs gives us the possibility to integrate additional applications and image processing algorithms to the system without any cost in hardware (Ken et al., 2005). It's worth mentioning that the hardware image processing algorithms could be faster than the corresponding algorithms in C/C++.In this paper, we describe the design and implementation of a virtual painting board in an FPGA board. The system has a CMOS camera as an input device and a VGA monitor as an output device. Users can draw lines, colour a picture or both and can follow it on the VGA screen. To implement the hardware and software for virtual paint system on DE2 FPGA board, we have used VHDL and Verilog-HDL. Also, the colour segmentation and gesture detection technique are applied into embedded system of virtual paint. The system consists of CCD_capture, RAW2RGB, SDRAM, Main Ctrl, M4K, and VGA controller. The system detects a specific colour with the camera and stores and converts the coordinate. Based on Altera board (Altera De2-115), we use 1.3 M pixel CMOS camera (TRDB-DC2), VGA monitor, and coloured bands for this system. The design is developed by Quartus Ⅱ 11.1software.

This paper is organized as follows. Section 2 describes the related works for the virtual painting board. Section 3 describes the system design for the virtual painting board.Implementation result is presented in Section 4. Finally conclusion is presented in Section 5.

2. Related works

We use the five senses to perceive information. Five senses refer to the five traditionally recognized methods of perception, or sense: taste, sight, touch, smell, and hearing. However, the data and information in on-line are not fully perceived with only five senses. Sixth sense is a wearable gestural interface that augments the physical world around us with digital information and lets us use natural hand gestures to interact with that information. To activate sixth sense, “Sixth Sense Project” was processed in the MIT Media Lab. Figure 1 shows the Sixth Sense Project in the MIT Media Lab. In Figure 1, the user displaysthe information in hand and operates with camera, coloured band and beam project.

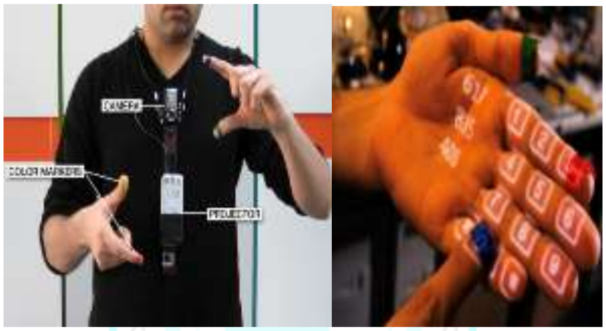


Figure 1: SixSense project

By the advancement of these technologies, the wearable devices are utilized in information technologies. Figure 2 shows “FingerReader” in the MIT Media Lab.In Figure 2, the moving fingers onto the text are a wearable device that recognizes the sentences and transfers them to a visually impaired person.This is actually a technique for visually impaired people using the sixth sense. This technique can be applied to various fields.

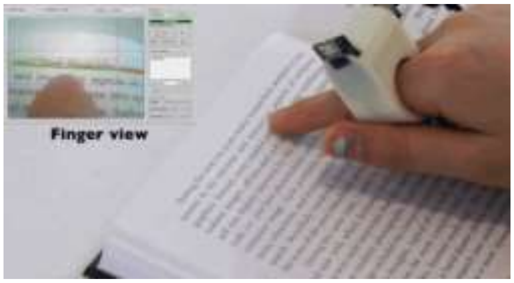


Figure 2: FingerReader(Wearable text-reading device)

3. System design

3. 1 Basic architecture

Figure 3 shows the basic architecture of system. The system consists of CMOS camera, Altera board, and VGA monitor. The CMOS camera captures the data and provides RGB information of the data to the Altera board. Start, Stop, and Reset are assigned to 3 push-buttons, respectively. Light & darkness is assigned to 16 switches, respectively.In the Altera board, the system uses a coloured band that can be inserted on a finger, and when it comes to red and yellow, it holds the cursor with the coordinates.Ifthe camera recognizes both red and yellow band on its finger, the current coordinates are represented by the cross in the VGA monitor. If the camera recognizes only a red band on its finger, you can take a point and can draw a picture with the points. The colours are selected on the palette of the left side of the screen.Depending on the movement of the fingers, the colour is painted to the screenof theVGA monitor that displays the resulting image from the Altera board. The palette provides three colours: red, green, and green.

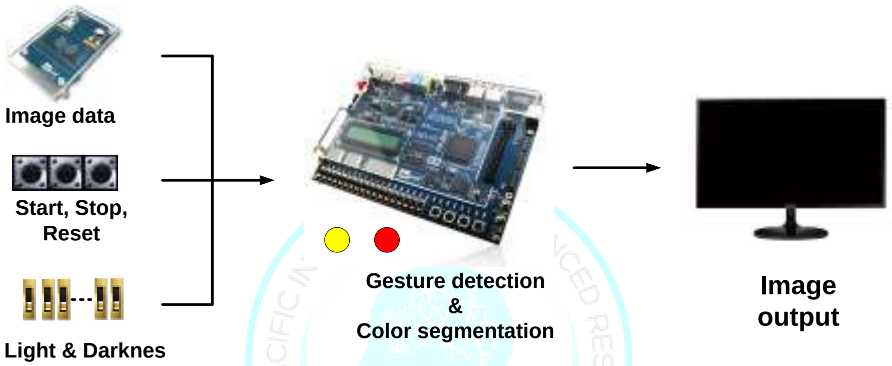


Figure 3: Basic architecture

3. 2Hardware architecture

The system hardware consists of CCD_capture, RAW2RGB, SDRAM, Main Ctrl, M4K, and VGA controller. Figure 4 shows the hardware architecture of the system.

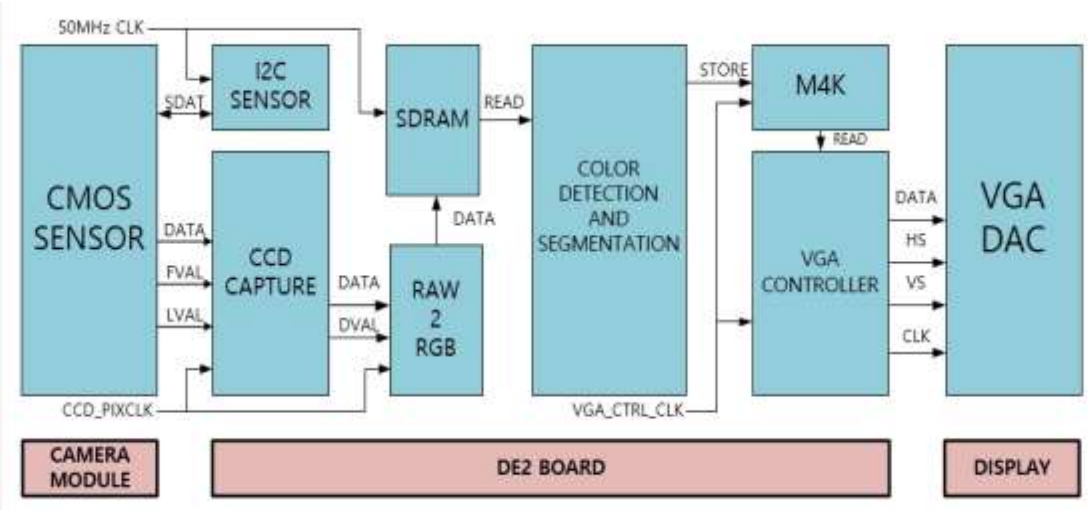


Figure 4: Hardware architecture

In CCD_Capture, the block is to produce horizontal, vertical and frame counter to insert the data from the camera.The main function of the block is to extract the data of the effective pixels from the frame validation and line validation information.The clock frequency of the block is PIXCLK.The output signal (ODVAL) indicates whether the pixel data being analyzed is valid.The coordinate X and Y axes of each valid pixel are used as inputs of RAW2RGB block.

In RAW2RGB, the block is used to convert the image format from the Bayer to RGB format and the red, green and blue values of the 10-bit. The conversion is as follows.

R <= 10bit,
G <= (G1 + G2) / 2, (G1 <= 5bit, G2 <= 5bit)
B <= 10bit

To reduce the transfer quantity of the camera, we used Bayer pattern. Figure 5 shows theBayer pattern.

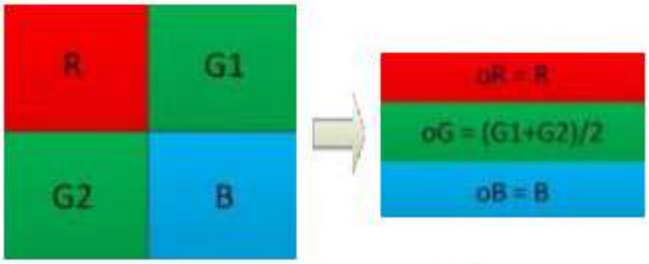


Figure 5: Bayer pattern

In SDRAM, the block is to store each pixel value for all RGB values as a frame buffer. An RGB total of 30, three colours of the 10, the SDRAM requires two lines to store the data as a pixel address.

In Main Ctrl, the block is responsible for the colour segmentation and detection. Reading the values in the SDRAM, the block calculates each value and detects different colours depending on the relative intensity of the component. Also, as a painting on the screen, the block calculates the center of coordinates and records the selected colour value toM4K.

M4K consists of RAM. The block is used to store the RGB value for each pixel address. VGA controller reads the RGB value directly from M4K.The data width in this memory is 2 bits.VGA consists of VGA controller and VGA monitor.The VGA controller is responsible for receiving the data values from the RAW2RGB and sending the RGB value to each pixel to VGA DAC.In addition, the VGA controller is used to display the colours on the left side of the screen to select different colours.

3. 3 Algorithm procedure

A colour filter was implemented to recognize yellow and red colour in the image from the camera.If a particular colour is detected through a colour filter, the system calculates the center position and stores the coordinates to M4K. The stored coordinates are immediately stored and painted on the screen.

3. 3.1Colour filtering

In this work, it was designed to recognize yellow and red colours to be implemented in order to draw a realistic picture.When both yellow and red colours are recognized, the current coordinates appear by the cross on the screen. If only the red colour is detected, the colour is painted along the coordinate. Figure 6 shows the painting algorithm flow.

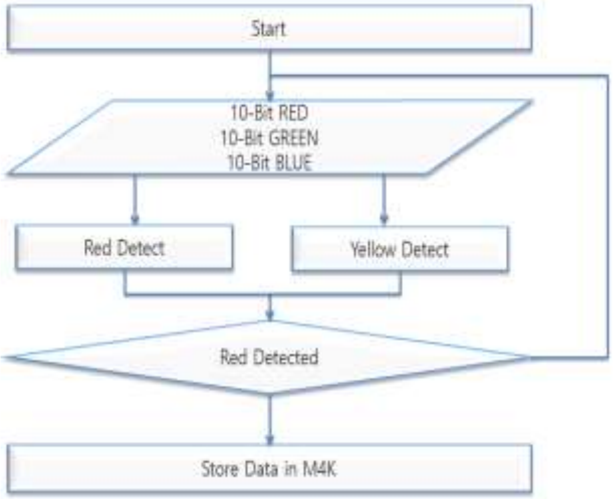


Figure 6: Painting algorithm flow

For the colour detection and segmentation, we used the critical comparison and relative intensity comparison of RGB component,because yellow and red colours have a high intensity in colour, two colours are chosen as the filtering colour. Table1 shows the colour filtering source structure. It also performed filtering under various variable conditions to match the colour bands and to reduce errors.

Table 1: Colour filtering source structure.

```
assign  RED_match=(VGA_data_iRed > 10'h100)? 1'b1:1'b0;
assign  GREEN_match=(VGA_data_iGreen > 10'h080)? 1'b1:1'b0;
assign  BLUE_match=(VGA_data_iBlue > 10'h080)? 1'b1:1'b0;

assign   y_RED_match=((VGA_data_iRed > 10'd220) && (VGA_data_iRed < 10'd225))?
1'b1:1'b0;
assign   y_GREEN_match=((VGA_data_iGreen > 10'd150) && (VGA_data_iGreen <
10'd170))? 1'b1:1'b0;
assign   y_BLUE_match=  (VGA_data_iBlue < 10'h080)? 1'b1:1'b0;

//For RED Greater Then Green detection
assign  RED_GT_GREEN_match=(VGA_data_iRed > (VGA_data_iGreen +
VGA_data_iGreen[9:1]))?1'b1:1'b0;

//For RED Greater Then Blue detection
assign  RED_GT_BLUE_match=(VGA_data_iRed > {VGA_data_iBlue[8:0],1'b0})?
1'b1:1'b0;  //Red > 2(Blue)

assign  R_detect=RED_match&RED_GT_GREEN_match
&RED_GT_BLUE_match&~GREEN_match&  ~BLUE_match;
assign  Y_detect = y_RED_match & y_GREEN_match & y_BLUE_match;
```

3. 3.2Center coordinate determination and display

To paint the colours along the hands, we compute and store the coordinates of the recognized hand. Creating the variable that counts the pixels of the detected colour, calculating the average value of X and Y coordinate, the center coordinate of X and Y are determined.Table2 shows the center coordinate source structure.

Table 2: Center coodinate source structure.

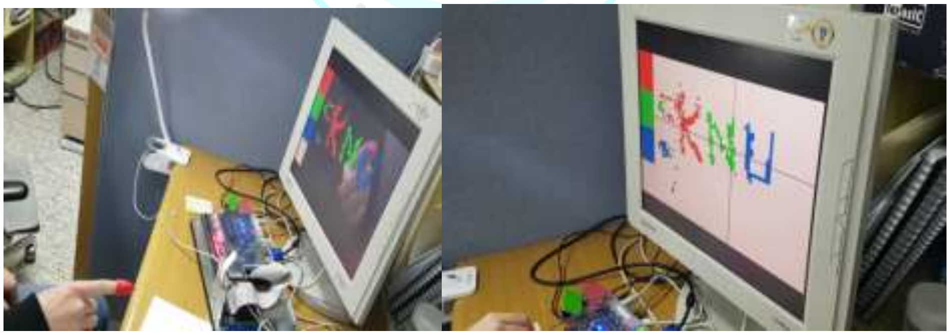
```
if (R_detect) begin
  draw <= 1'b1;
  Xaddr_sum <= Xaddr_sum + X_addr;
  Yaddr_sum <= Yaddr_sum + Y_addr;
  counter <= counter + 1;

  Xaddr_center <= Xaddr_sum/counter;
  Yaddr_center <= Yaddr_sum/counter;
```

We use M4k block to store the coordinates. Because it is used to synchronize the address to the counter, it is possible to write colour information data on the determined coordinates.When reading the data from M4K, we synchronize it to the address with H_cont and V_cont of VGA controller. Through M4k, RGB data transfers Main Ctrl block to VGA controller.

4. Implementation

This system is implemented with Altera DE2 board, 1.3 M pixel CMOS camera, and VGA monitor. The workis developed based on VHDL and Verilog HDL which are run using the Quartus11.1 software.Figure 7 shows thedemonstration that writes letters and represents coordinate.

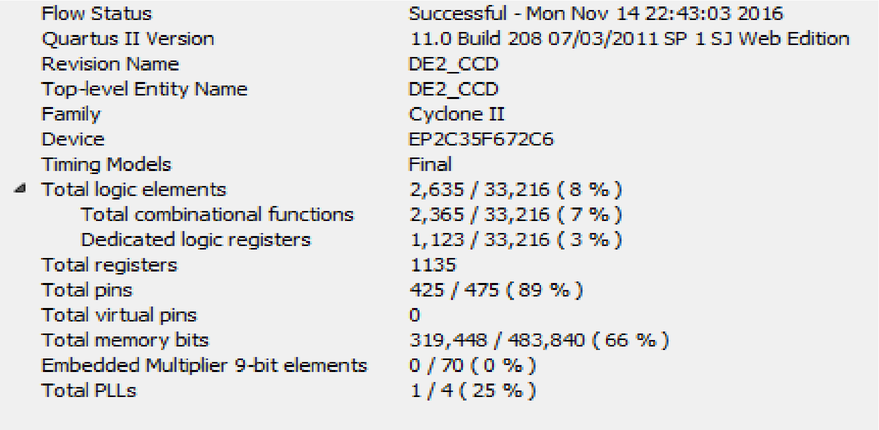


(a) (b)

Figure 7: Demonstration

When the camera recognizes only red band on its finger, it will take a spot on the screen to paint the colours.The colours are selected on the palette of the left side of the screen. Depending on the movement of the fingers, the colour is painted to the screen as shown in Figure 7 (a). The palette provides three colours: red, green, and green. For accurate positioning, the coordinates are configurable.When the camera recognizes both red and yellow band on its finger, the current coordinates are represented by the cross as shown in Figure 7 (b).

Figure 8 shows the compile results. The compile results show that the 8 % logical element,89 % pin, and 66 % memory were used.



```
Flow Status                              Successful - Mon Nov 14 22:43:03 2016
Quartus II Version                       11.0 Build 208 07/03/2011 SP 1 SJ Web Edition
Revision Name                            DE2_CCD
Top-level Entity Name                    DE2_CCD
Family                                   Cyclone II
Device                                   EP2C35F672C6
Timing Models                            Final
Total logic elements                     2,635 / 33,216 ( 8 % )
    Total combinational functions        2,365 / 33,216 ( 7 % )
    Dedicated logic registers            1,123 / 33,216 ( 3 % )
Total registers                          1135
Total pins                               425 / 475 ( 89 % )
Total virtual pins                       0
Total memory bits                        319,448 / 483,840 ( 66 % )
Embedded Multiplier 9-bit elements       0 / 70 ( 0 % )
Total PLLs                               1 / 4 ( 25 % )
```

Figure 8: Compile results

5. Conclusions

We describe the design and implementation of a virtual painting board in an FPGA board. The system has a CMOS camera as an input device and a VGA monitor as an output device. Users can draw lines, colour a picture or both and can follow it on the VGA screen. The work is developed based on VHDL and Verilog HDL which are run using the Quartus11.1 software. The system hardware consists of CCD_capture, RAW2RGB, SDRAM, Main Ctrl, M4K, and VGA controller. The system detects a yellow and red colour with the camera and stores and converts the coordinate. Based on Altera DE2 board, we use 1.3 M pixel CMOS camera, VGA monitor, and coloured band for this system. It will be much more practical if it is recognized only by the movement of the hands without colour bands used in this work. Also, it requires font adjustment and various colour supports.

References

i. Chien, S., Ma, S. Y. & Chen, L. G., 2002. Efficient moving object segmentation algorithm using background registration technique. *IEEE Transactions on Circuits and Systems for Video Technology,* 12(7), pp. 577-586.

ii. DE2-115 , n.d. *User Manual.* [Online] Available at: http://www.altera.com

iii. Karray, F., Alemzadeh, M., Saleh, J. A. & Arab, M. N., 2008. Human-Computer Interaction: Overview on State of the Art. *Proceedings of the International Journal on Smart Sensing and Intelligent Systems,* 1(1), pp. 137-159.

iv. Ken, K. K., Cho, S. H., Kim, H. J. & Lee, J. Y., 2005. *Detecting and tracking moving object using an active camera, Proceedings of the International Conference of Advanced Communication Technology.* Daejeon, South Korea, Proc. of 7th International Conference of Advanced Communication Technology.

v. Lee, G. et al., 2010. Virtual Reality Content-Based Training for Spray Painting Tasks in the Shipbuilding Industry. *ETRI Journal,* 32(5), pp. 695-703.

vi. Petouris, M., Kalantzopoulos, A. & Zigouris, E., 2009. *An FPGA-based Digital Camera System Controlled from an LCD Touch Panel,.* Iasi, Romania, Proceedings of the International Symposium on Signals, Circuits and Systems.

vii. Roth, C. H. & John, L. K., 2007. *Digital Systems Design Using VHDL.* 2nd ed. s.l.:CL Engineering.

viii. Roth, C. H., John, L. K. & Lee, B. K., 2015. *Digital Systems Design Using Verilog.* 1st ed. s.l.:CL Engineering.

ix. TRDB DC2 , n.d. *User Guide.* [Online] Available at: http://www.terasic.com

x. Wang, N. J., Chang, S. C. & Chou, P. J., 2012. *A Real-time Multi-face Detection System, Implemented on FPGA.* New Taipei City, Taiwan, Proceedings of the International Symposium on Signals Intelligent Signal Processing and Communication Systems.

Appendixes

This work was supported by the BK21 plus program through the National Research Foundation (NRF) funded by the Ministry of Education of Korea. Authors thank the IDEC (IC Design Education Center) program for its hardware and software assistance.