Journal of Scientific Research and Development 2 (13): 1-10, 2015
Available online at www.jsrad.org
ISSN 1115-7569

Real time touch less home control system with vision-based FPGA hand gesture recognition interface

Chee-Pun Ooi *, Soon-Nyean Cheong, Wooi-Haw Tan, Chin-Leei Cham

Faculty of Engineering, Multimedia University, MMU, Cyberjaya, Malaysia

Abstract: The gaining popularity of touchless applications has been facilitated by the increased adoption of Microsoft Kinect by academicians and researchers. However, the application of touchless interaction in home automation is still in a nascent stage. The primary contribution of this paper is to introduce a novel solution for a convenient touchless home control system allowing home users to operate digital appliances such as video player, lamps, fan etc. in a more natural way. A new interaction concept is introduced to improve the existing home control experience, where the user's hand, instead of remote control devices is used to control the system. The system is implemented on the Altera Cyclone II devices that consists of the image pre-processing and feature extraction states to detect and track user's hand effectively. Real time background subtraction, followed by morphological operations is employed for hand segmentation and tracking, whereas bounding box and Centre-Of-Mass based computation are used in the system for gesture sign classification. The main modules in the image processing exploit the parallelism architecture of the FPGA to achieve fast hand gesture recognition to provide several essential features: (1) Hand detecting and tracking, (2) Gesture classification (3) Real time processing and (4) Controlling home device via Ethernet. The result of this work shows good potential for providing an immersive experience for home users to use their bare hands in free-space as an interface to control home appliances connected within a home network.

Key words: Touch less interaction; Home control system; FPGA; Image processing; Hand gesture interface system

1. Introduction

Computer peripherals such as mouse, keyboards, joysticks and touch panel have been used as a human computer interface (HCI) devices in controlling hardware appliances in recent years (Chan, 2008, Deng et al, 2007, Viani et al. 2013, Alam et al. 2012). Generally, users would use the standard computer input devices to control electronic devices connected to a home media server or computer directly. However, with the advancement of technology, conventional HCI could possibly be replaced with a more natural interface that is intuitive and effective such as a hand gesture interface. Existing hand gesture interfaces available today include vision-based gesture interface, data-glove gesture interface, touch-based interface, etc. In the touch-based system, two touch pads would be used as the interface medium to reduce the switches and the buttons needed by the user and can be seen using it in automation (Chongyoon and Rantanen, 2011). Nonetheless, cognitive efforts are required to map the specific functions to the touch pads. There are several mapping requisites that need to be considered in order for the interaction with the machine to be coherent to the user. The data glove interface was introduced to provide an easier and more efficient approach to convey idea from the user's gesture to the action as seen in SudoGlove in controlling the RC car (Ballerini et al., 2011). However, using this technique, the data glove would need to be redesigned to suit different users' hand sizes and therefore cost consuming.

In order to eliminate the tedious efforts required in mapping the function keys used in touch interface and the requirement of customized data gloves to meet different users, vision-based hand gesture interface was proposed. In the vision-based touch less interface system, one or two cameras are used to track hand movements of the user and process the information in order to perform the control operations accordingly. One of the challenges in vision-based interface would be to extract user's hand from the image captured effectively from the complicated background in real time. Although research in touch less hand gesture recognition using image processing techniques has been around for many years using a standard computer, there is a need for dedicated hardware for real time image processing. This is because in a complex scene, performance of standard personal computer (PC) will be affected when objects need to be recognized increases and thus challenging the limit of real time processing ability of the PC. Besides, implementing vision-based hand gesture interface system involving image processing in real time would require rapid processing of huge amounts of data and therefore customized algorithm implemented on FPGA would

* Corresponding Author.

generally provide a better performance as compared to the implementation on standard x86 architecture as in PC (Bochem et al., 2009; Lian, 2008; Lopez, 2013). The parallel architecture offered by FPGA along with hardware circuit and logic gates implementation of the customized algorithm modules to complete the complicated computational tasks would make real time processing a reality.

The potential of touch less interaction is tremendous. One compelling reason is the integration of touch less interaction into home environment where multiple heterogeneous home appliances such as lamps, video players, fans etc. are widely used by users, and transcends conventional automation systems by providing a more convenient environment for users to interact with home devices using their bare hands. However, the application of touch less applications in home automation is still in a nascent stage.

As reported in paper (BRUSH, A et al., 2011) that convenience is the most desirable aspect in home automation. However, existing home control applications are lack of convenience. For example, NFC-enabled smart home applications, namely "Touch & Connect", "Touch & Listen" and "Touch & Watch" introduced in paper (LONGBIAO, 2012), required users to walk toward the NFC tagged devices and tap it with their NFC smartphone before performing the control operations, making the applications inconvenient.

Hence, a vision-based hand gesture recognizer that leverages on the hardware accelerator capabilities of FPGA is proposed to facilitate home control operations for various home appliances and improve the convenience in home automation. The hand gesture recognizer tracks and translates hand movements into control signals for various home appliances. Specifically, the control operations are activated by hand gestures (e.g., two active fingers up) in free space. The proposed system is distinctive compared to the existing home automation because it provides home users with a more convenient interaction experience so that they do not need to reach the physical devices prior operating it. This paper is organized as follows: Section 2 reviews on technology and related works. Next, Section 3 describes the design of hand gesture interface and control system via Altera FPGA board for effective hand tracking and controlling. Section 4 presents the results of the proposed image processing algorithm and Section 5 discusses the experimental results. Finally, Section 6 concludes the paper.

2. Related works

Different software development tools have been used in designing hand gesture base interface system which includes Matlab, C++ with OpenCV, JavaScript, etc. Various kinds of image processing algorithms were proposed to detect and track images effectively such as Haar Classifier algorithm introduced by Viola and Jones which had since been widely used (Viola and Jones, 2011). In this technique, two level approaches were proposed to achieve real time vision-based hand gesture classification. The low level approach implements the posture recognition based on Haar-like features and the AdaBoost learning algorithm while the high level approach implements the hand gesture recognition through a context-free grammar-based syntactic analysis (Qing Chen et al., 2007). Besides, standard PC and two low-cost web cameras were proposed to track 3D position and 2D orientation of the thumb and index finger of each hand. After going through the feature extraction process, the hand is recognized through the peak and valley detection (Malik, 2003). Although standard PC could possibly achieve the real time requirement of the system, the sophisticated hardware involved allocates a major challenge for portability and flexibility. Furthermore, when the amount of image processing tasks increases in a more complex scene, the ability of PC to handle the growth in real time will be affected.

Thus, to improve the performance of real time processing for the hand gesture interface, FPGA implementation that uses Hardware Description Language or NIOS soft-core embedded processor was proposed. Comparison in terms of the image processing speed between two FPGAs with the recent processors that support SIMD instructions and multi-cores have been conducted on two-dimensional filters, stereo-vision and k-means clustering is reported in (Saegusa et al., 2008).

Although the result shows that the quad cores processor can achieve real-time processing when image size is small, all the resources is used up for real-time processing. On the other hand, in the FPGA implementation, large quantity of hardware resources is still available for more sophisticated image processing works.

The high performance of FPGA is due to the highly parallel architecture and huge amount of internal memory banks, which can be retrieved in parallel (Yuan et al., 2013). Real time image processing applications which can be processed by the PC are still very limited. For practical real time applications, FPGA implementation is still a preferred choice.

A practical hand gesture interface was successfully developed by Lee et al. (Lee et al, 2011) to control R/C car. By exploiting the parallel architecture of the FPGA for image processing, the system is able to recognize user hand postures and generate corresponding signals for hardware control in real time. The system includes image preprocessing, feature extraction and gesture classification stage. In the image preprocessing stage, the human hand is located in the image frames via simple color thresholding method. Red color glove is used in the system to distinguish the human hand from the environment. The feature extraction stage extracts several features of the segmented hand including the center of mass of hand pixels, size, and aspect ratio of the bounding box that encloses all hand pixels. In the gesture classification stage, four directional signals, forward, reverse, left, and right are defined based on the distance between the topmost, bottommost, leftmost, and rightmost, of the bounding box and the center of mass for navigating an R/C car. The system is able to track user's hand and recognize simple predefined vocabulary of gestures for controlling the R/C car or other hardware appliances in real time. However, the user is required to wear red color gloves, which makes the interaction process cumbersome and not natural as the algorithm developed to recognize hand gestures is primitive.

To achieve real time processing and increase the performance of image processing, the proposed hand gesture recognition system was implemented in FPGA using background subtraction, followed by morphological operations to repair and smoothen the segmented hand image for further analysis. The process flow of the proposed system is shown in Fig. 1. The segmented hand binary image is then used in the feature extraction stage in order to provide a more natural way of interacting with hardware devices. (I.e. Use bare hands in free-space as an interface to control home appliances)

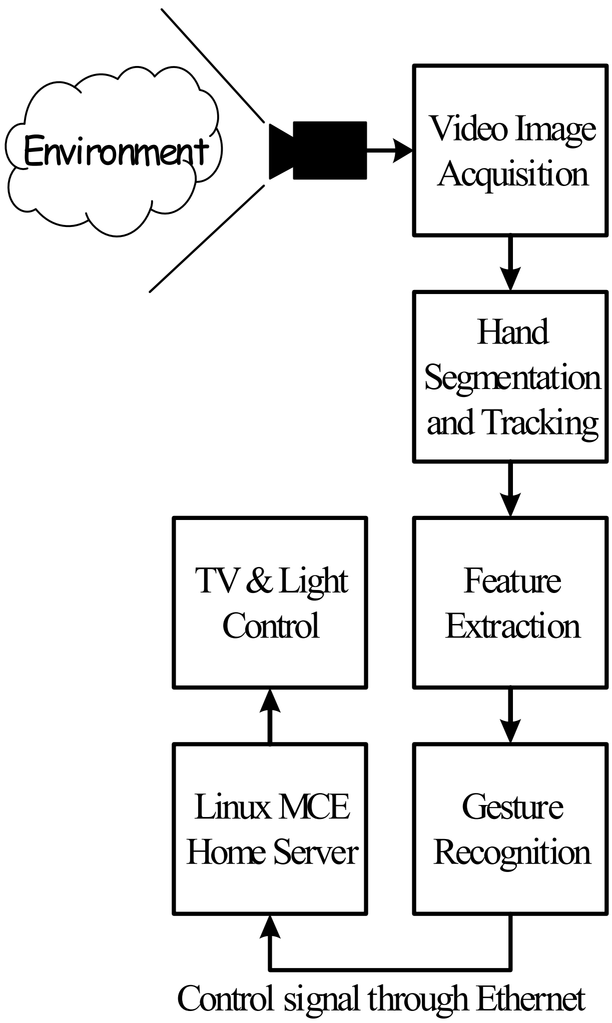

Fig. 1: Process flow of the proposed system.

3. Design and development of the hand gesture interface and control system

Altera Cyclon II Development and Education 2 FPGA board (DE2board) is employed to realize the design of the system. The hardware architecture of the system core is developed in Verilog and SOPC hardware modules and compiled via Quartus II Web-Edition 11.0 SP1. SOPC Builder tool 11.0 is utilized to facilitate the generation and the handshaking of the NIOS II CPU module, LCD interface module, SRAM interface module, DM9000A interface module, and parallel I/O (PIO) ports through the Avalon switch fabric, which is a high-bandwidth and flexible interconnect structure. The IP cores for NIOS II CPU and parallel I/O ports are available in the SOPC tool library whereas the SRAM, LCD, and DM9000A interface module are provided by Terasic Technologies Inc. The CPU is configured as NIOS II/s (standard 32-bit RISC with instruction cache, branch, hardware arithmetic, etc.) and programmed in C. The program is downloaded to the board via the Nios II Software Build Tools for Eclipse 11.0.

The 16-bit x 256K SRAM is used as the memory component for the NIOS II CPU instead of SDRAM as it has been utilized in the hardware. The CPU is Avalon Memory-Mapped Master and all other components are Avalon Memory-Mapped Slaves (an address-based read/write interface typical of master-slave connections). All the components in the system run at 50MHz provided by the board's oscillator.

Fig. 1 depicts the general process flow of the proposed system to detect and recognize hand gestures. The human hand is first identified in the image. It is then analyzed and translated into the corresponding control signals. A code based on the gestures is generated and sent via Ethernet to the LinuxMCE home server for lighting control.

A. Video Image Acquisition

The interaction between the user and the system is based on the vision based approach due to its nature of non-contact, non-intrusive interaction, simple configuration and real-time performance. A CMOS camera is integrated as part of the system to continuously capture sequence of images of the user's hand. The image is converted into 12-bit grayscale from 36-bit RGB via (1), where I is the pixel intensity value in grayscale.

$$I = 0.299R+0.587G+0.114B \quad (1)$$

Hand Segmentation and Tracking: The images captured by the camera contain visual information that is rich and noisy. A number of operations are to be carried out to correctly locate the human hand and discard any undesired information in the video images. Fig. 2 shows the process of extracting and tracking user's hand in the video image sequence and produces the corresponding binary image of the hand via background subtraction, followed by morphological operations. The process flow of morphological operation is shown Fig. 3. This segmented hand binary image is then used in the feature extraction stage.

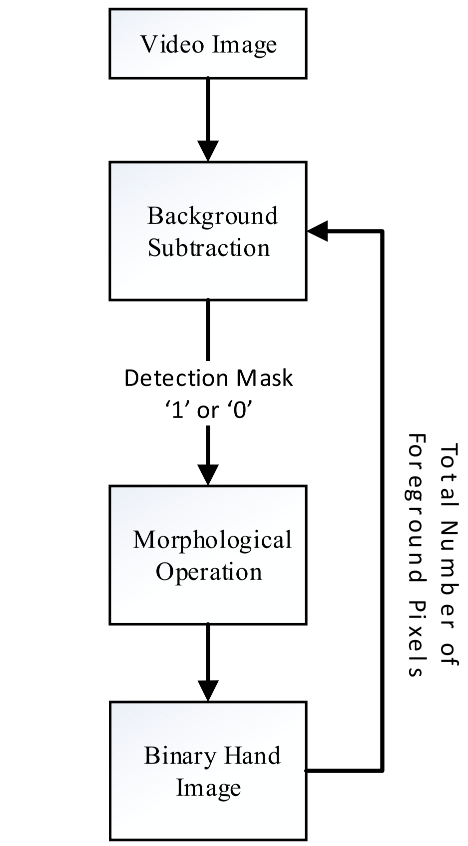

Fig. 2: Hand segmentation and tracking process

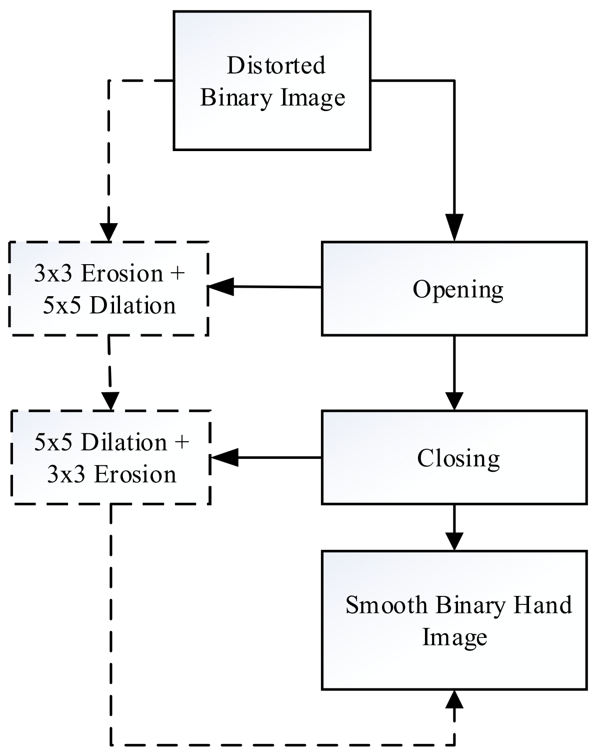

Fig. 3: Binary image morphology operations

Background subtraction is a widely used approach for detecting moving objects in videos from static cameras. The principle is that moving objects (foreground image) are separated from the rest of the stationary objects (background image) by evaluating the difference between the current frame and a reference frame, or the "background model". A pixel is classified as the foreground pixel if

$$| \mathrm{It}(x,y) - \mathrm{Bt}(x,y) | > T \qquad (2)$$

Where It(x, y) is the luminance intensity of the incoming pixel and Bt(x, y) is the luminance intensity of the background pixel at location (x, y) at time t, and T is a predefined value known as the threshold. This threshold is made global and static to all pixels. It is a calibration parameter that depends on the operating environment and a suitable threshold value is determined experimentally for optimized detection rate. The detection mask D= 1 if (2) is satisfied and D= 0 otherwise. The obtained mask is then used in the morphological operation stage. The background model chosen for the design is the selective Gaussian running average model to selectively update the background based on total number of hand pixels. Background is updated as

$$\mathrm{Bt}(x,y) = a*\mathrm{It}(x,y)+(1-a)* \mathrm{Bt\text{-}1}(x,y) \qquad (3)$$

Where a is the learning rate typically 0.05, if the total number of hand pixels is greater than a predefined value, T_p, otherwise the new background pixel is just the previous background pixel Bt-1 (x, y). This is to avoid the inclusion of the hand into the background image during interaction.

The binary hand image segmented via background subtraction may contain missing pixels (holes and gaps) and pick up a small amount of non-hand pixels (blobs) in the background. Binary image morphology operation as shown in Fig. 3 is carried out to repair and smooth the segmented hand image for analysis. Two morphological operations, opening and then followed by closing are performed on the binary image to fill holes and bridge gaps in the foreground and remove blobs in the background. Opening involves erosion followed by a dilation operation, whereas closing involves dilation followed by an erosion operation. A 3x3 and 5x5 structuring elements are used in the erosion and dilation process, respectively.

C. Feature Extractions: Features such as hand size, hand position, hand orientation, hand movements, and number of active fingers are extracted from the smoothed binary hand image.

The hand size, N is estimated by calculating the total number of foreground pixels in the image. The result is compared against the same threshold, T_p to determine the extraction of the remaining features. Feature extraction is preceded if it is larger and ceased if it is smaller.

The hand position is approximated by the center of mass (X_c, Y_c) of the foreground pixels in the image. It is calculated as:

$$(X_c, Y_c)=(\mathrm{Sum}(X_i)/N, \mathrm{Sum}(Y_i)/N) \qquad (4)$$

where (X_i, Y_i) is the i-th foreground pixel location.

For hand orientation detection, a bounding box is constructed to enclose the topmost (Y_{min}), leftmost (X_{min}), rightmost (X_{max}) and bottommost (Y_{max}) pixel in the hand region. X_{top} is also found as the foreground pixel X position at Y=Y_{min}. The difference between Y_{min} and Yc (=D_u), X_{max} and X_c (=D_r), and X_{min} and X_c (=D_l) are calculated. The hand orientation is then determined based on the largest value (D_m) among D_u, D_r and D_l. $D_m = D_u$ for up, $D_m = D_r$ for right, and $D_m = D_l$ for left direction, respectively. The process flow is shown in Fig. 4. Down direction is discarded due to the presence of the wrist at the bottom hand region.

In order to count the number of active fingers, a circle is constructed centered at (X_c, Y_c) with a radius equal to 50% of the longest pixel distance from the center of mass. The scanning is performed along the circumference for examining the total number of black-to-white or white-to-black pixel transitions, N_t. The finger count, F_n is then proportional to this output number as $F_n = N_t/2$-1. Dividing by 2 is because every pair of (0-to-1, 1-to-0) or (1-to-0, 0-to-1) transition constitutes a finger and the order is trivial. The process flow is summarized in Fig. 5.

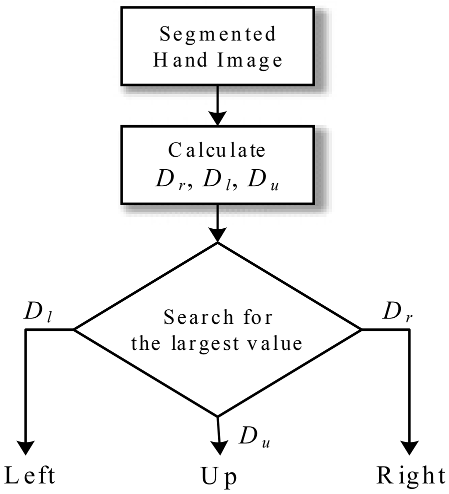

Fig. 4: Hand orientation detection flow

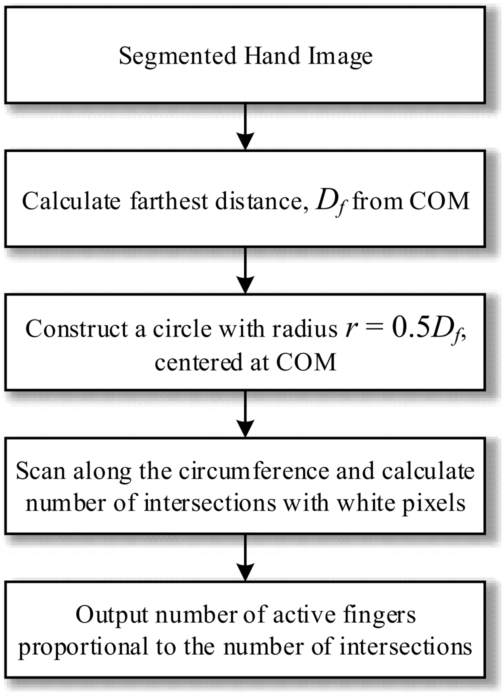

Fig. 5: Active finger counting process

Hand moving direction is detected if there is a significant change in (X_c, Y_c) over a period of time. The change is measured by evaluating the difference $d_X = X_c - X_{rp}$, $d_Y = Y_c - Y_{rp}$ from (X_c, Y_c) and a reference point (X_{rp}, Y_{rp}) in every 3 VGA frames, i.e., 3x1/60fps = 0.05s. The sign of d_X and d_Y determines the moving direction of the hand. For example, negative d_X indicates a left movement. The movements are only detected if the changes are significant, that is the requirement of $|d_X|$ and $|d_Y| > T_{mv}$ must be satisfied and 10 pixels are used for T_{mv}. In every 3 VGA frames, (X_{rp}, Y_{rp}) is updated to (X_c, Y_c) if movement occurs, otherwise it remains. The process flow is summarized in Fig. 6.

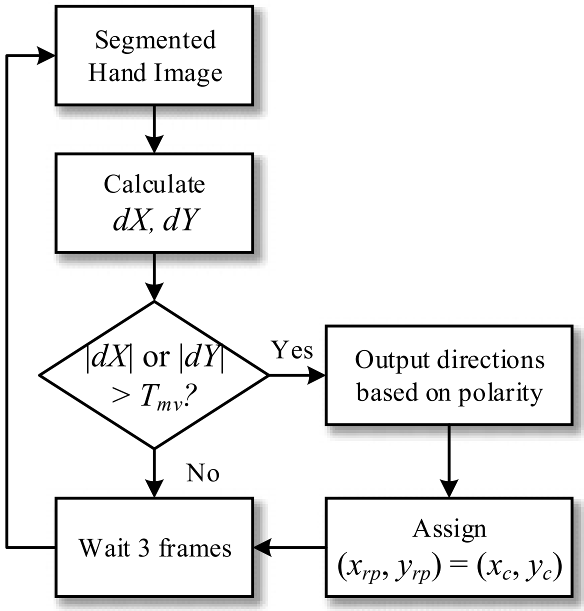

Fig. 6: Hand moving direction detection process

D. Gesture Recognition: The gestures are to be recognized for interacting with a user interface and eventually control a TV and a lamp. The user interface showing a photo of a living room with a TV and a lamp together with a hand cursor is illustrated in Fig. 7. Several vocabularies of gestures are defined to recognize hand gestures based on a rule-based technique that establishes a set of conditions on the extracted features. Hand gestures are recognized by matching the extracted information of hand palm, fingertips, and hand movements with the predefined rules as shown in Table 1. The gesture signals are then translated and sent to the LinuxMCE home server through the Ethernet, for controlling the target home appliance.

Hand cursor shown in the user interface is used to track the position of the pixel located at the topmost of the bounding box instead of the centre of mass, COM for improved sensitivity since the wrist affects the position of COM while moving.

Thus, any gestures can move the cursor around to the desired location. Normally, one fingertip pointing up or posturing zero is used to move the cursor. To control a certain appliance in the living room, the cursor is moved to where the appliance is located.

For turning on/off lights, the cursor is moved to where the lamp is located in the interface. Once the cursor is moved into the vicinity of the lamp, a rectangular highlight surrounding the lamp appears to indicate the selection of the lamp. The lamp is toggled as long as the cursor stays in the region for 2 seconds. The cursor has to re-enter the region and stays for 2 seconds for toggling to occur again.

To select the TV for interaction, the cursor is moved over the TV and hold for 2 seconds. The interface then changes to display the user's hand to indicate that the interaction with TV has been initiated. TV volume is turned down and up by showing left and right posture without hand movement detected. Left and right posture is recognized if 1 finger is detected and pointing to the left and right respectively. The requirement of 1 finger is imposed so that 2 fingers and above can be recognized in any direction for changing to the desired TV channel number. Number 1 is recognized when 1 finger is detected and pointing up. This is to avoid false recognition of left and right postures. All gestures that control the TV are recognized if and only if the TV is selected for interaction, otherwise they are ignored. To return to the user interface, hand is removed or ensure N > Tp is not satisfied for 2 seconds. The summary hand gesture for TV command is shown in Table 1.

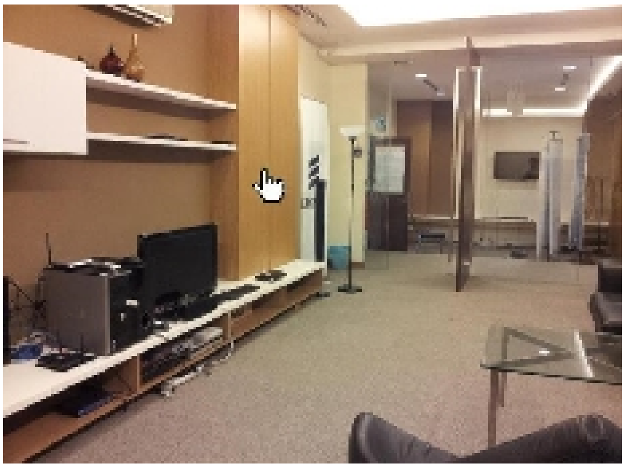

Fig. 7: Photo of living room with hand cursor and home devices.

Table 1: Hand gestures vocabulary for the tracking features

Commands	Gestures	Tracking Features
Moving Cursor	Movement	(X_{top}, Y_{min})
Selecting	Movement	(X_{top}, Y_{min})
Lights On/Off	Movement	(X_{top}, Y_{min})
TV Volumn Down	Left Posture	($D_m = D_l$) AND (F_n =1) AND (Movement = 0)
TV Volumn Up	Right Posture	($D_m = D_r$) AND (F_n =1) AND (Movement = 0)
TV Channel =1	One	($D_m = D_u$) AND (F_n =1) AND (Movement = 0)
TV Channel =2	Two	(F_n = 2) AND (Movement =0)
TV Channel =3	Three	(F_n = 3) AND (Movement =0)
TV Channel =4	Four	(F_n = 4) AND (Movement =0)
TV Channel =5	Five	(F_n = 5) AND (Movement =0)

4. Design architecture

The system level abstraction of the proposed system is shown in Fig. 8. A Terasic 5 mega pixel CMOS camera (TRDB-D5M) is mounted on the expansion port of the board to capture video images from the environment. The interfacing between the camera and the board consists of the CCD Capture module, RAW2RGB module, and I2C CCD Configure module that are provided by the Terasic Technologies Inc. camera example. CCD Capture reads out 2560x1920 pixels in Bayer pattern format consisting of three color filters Green1, Green2, Red, and Blue (G1, G2, R, B). The first row output pixels alternates between G1 and R pixels, and the second row output alternates between B and G2 pixels. The Pixels are read out whenever FRAME_VALID and LINE_VALID signals are asserted and they are synchronized to a 25 MHz pixel clock, CCD_PIXCLK generated by the PLL of the camera. Every four horizontal and vertical adjacent pixels (16 pixels) are then combined to produce a corresponding 36-bit RGB pixel by the RAW2RGB module. As a result, the resolution of one RGB frame is 640x480 pixels and running at ~4fps. This low frame rate is limited by the real-time requirement and SDRAM bandwidth. A pixel valid signal sCCD_DVAL is asserted by the RAW2RGB to indicate that a RGB pixel is available. The CCD ConFig. module contains a set of registers that allow for the configuration of pixel frequency, read out format, resolution, mirror, etc. The default values for these registers in the Terasic camera example are used.

The 36-bit RGB pixel value will be converted to 12-bit grayscale by using 2.10 fixed floating point formats via Multiplier-Adder structure. A 6-port SDRAM Controller contains six 16-bit wide, 256-bit depth FIFOs for simultaneous writing and reading at different speed at each port is used in the proposed system. The write ports are Write Port 1, 2 & 3 and read ports are Read Port 1, 2, & 3. Each port consists of a 16-bit wide data line, 1-bit write/read request, and 1-bit write/read clock. The address is incremented automatically from a starting address to starting address + 640x480 for every read at read ports and write at write ports. The starting address for a pair of Write Port x and Read Port x is at the beginning of bank x of the SDRAM. There are total of 4 banks in the SDRAM.

The SDRAM Controller is modified from 4-port to 6-port by adding one 16-bit wide, 256-bit depth FIFOs to create Write Port 3 and Read Port 3. Write Port 3 is accessed by NIOS II CPU through parallel I/O ports for writing a 640x480 living room image into the SDRAM from the SD-Card. Read Port 3 is accessed by the VGA for displaying the image.

Two write ports, Write Port 1 & 2, and two read ports, Read Port 1 & 2 of the SDRAM controller are involved in realizing the background subtraction. The write request for both write ports and read request for Read Port 1 is sCCD_DVAL, whereas the read request for Read Port 2 and Read Port 3 is the "Read" signal asserted by the VGA controller. The pixel clock of 25 MHz is used for both write ports and Read Port 1 clock frequency. The read port clock frequency for Read Port 2 and Read Port 3 is set as the VGA controller clock being 25 MHz. Thus the total and bandwidth requirement is (2 write + 1 read) x 25 MHz + 2 read x 25 MHz = 125 MHz. A PLL is used to generate 125 MHz clock frequency for the SDRAM controller and SDRAM chip. The fact that DE2 SDRAM can run up to maximum of 143 MHz becomes the limiting factor for running a higher pixel clock frequency to obtain a faster frame rate from the camera. The clock frequency for Write Port 3 is not included as it is no longer used as soon as NIOS II CPU finishes writing in the image. Summary of the ports assignment is shown in Table 2.

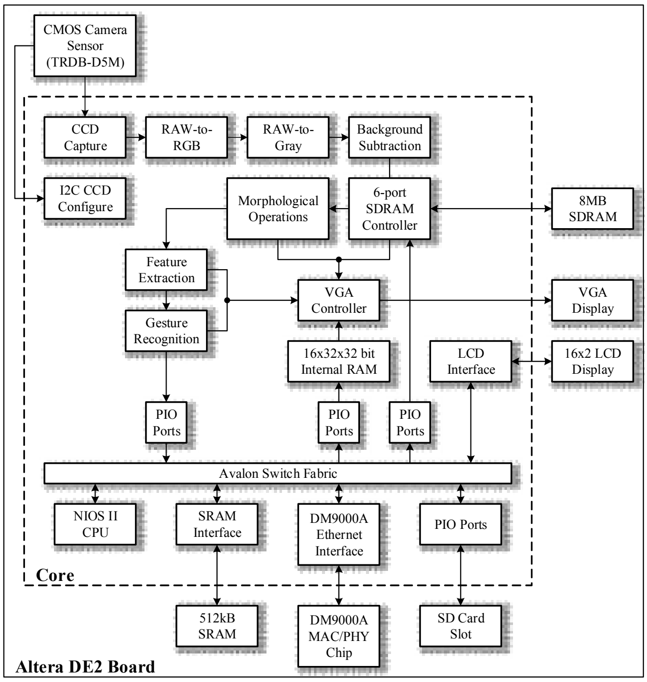

Fig. 8: System level architecture of the proposed system

5. System setup

Hardware setup of the proposed system is shown in Fig. 9. Table 3 shows the compilation report of the design by Quartus II. The design consumes 30% of the FPGA total logic elements, leaving a large amount of hardware resources for other usages and future enhancements.

Table 2: Summary of the SDRAM ports assignment

Signals	Write Port			Read Port		
	1	2	3	1	2	3
16-bit data	I_{bg_wt}	$\{fg, I_{bg\_wt}\}$	RGB_{image}	I_{bg_rd}	$\{fg_{rd}, I_{rd}\}$	RGB_{image}
Requestsignal	sCCD_DVAL	sCCD_DVAL	NIOS CPU	sCCD_DVAL	Read	Read
Starting Address	Bank1	Bank2	Bank3	Bank1	Bank2	Bank3
Size	640x480	640x480	640x480	640x480	640x480	640x480
Clock	25Mhz	25Mhz	NIOS CPU	25Mhz	25Mhz	25Mhz

Fig. 9: Hardware setup of the proposed system

Table 3: Compilation report

Items	Usage
Total logic elements	9,993/33,216 (30%)
Total combinational functions	8,910/33,216 (27%)
Dedicated logic registers	4,192/33,216 (13%)
Total registers	4208
Total pins	425/475 (89%)
Total virtual pins	0
Total memory bits	173,528/483,840(36 %)
Embedded multiplier 9-bit elements	15/70 (21%)
Total PLLs	2/4(50%)

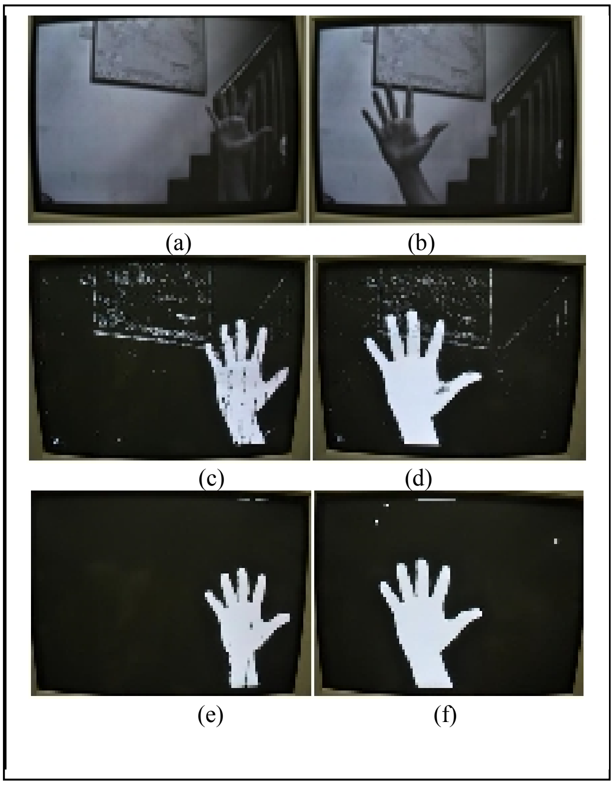

Fig. 10: (a) Hand in the dark and cluttered region (b) Hand in white homogenous region (c-d) Resulting binary hand image after background subtraction in both regions (e-f) Smooth hand image after morphological operations in both regions

6. Results

A. Result of background subtraction and morphological operation

The background image consists of a relatively cluttered region on the right (stairs) and a homogenous region (wall) for comparing the results of the segmented hand in both of this region is shown in Fig. 10.

A hand is placed on both sides of the region as shown in Fig. 10(a) and (b). The resulting segmented hand is presented in Fig. 10(c) and (d) for both regions. The white pixels are where the difference between the incoming pixel and the background differs significantly, i.e., larger than a predefined threshold. The threshold value is set to 0x10 for all the pixels in the experiment.

A significant amount of noise is noticed in the background and the hand region after subtraction. Holes are present in the hand region and island of pixels are fluctuating in the background, especially in the cluttered region, where more holes and gaps are present due to uneven distribution of the pixel intensity, such as the presence of dark areas, etc. This is undesirable for processing the binary hand image. The result is improved after going opening and closing operations as shown in Fig. 11(e) and (f) for both regions. Holes and islands are removed significantly and a relatively smooth hand image is obtained. However, still a small amount is noticed. In the homogenous region, tiny islands appeared and slits are present in the hand when it is in the cluttered region. This is due to the low threshold value used in segmenting the hand.

More noise will appear when the threshold is lowered to 0xB as due to the increased in the sensitivity in detecting the pixel's intensity change. A small amount of change will be detected even in the dark areas. A higher threshold (e.g 0x53) may be used to achieve more robustness to noise and changing illumination. However, this might distorts the hand object, making hand segmentation process difficult. Therefore, an appropriate threshold value should be adjusted to achieve the balance between the robustness and sensitivity. The threshold is set to 0x16 by experiment.

B. Result of background updating

The effect of background updating is shown Fig. 10(f). The Fig. shows the detected foreground pixels in the image when the view of the camera is changed slightly. Over time, the background is averaged out with the new incoming pixel value. This is observed when the detected foreground pixels slowly fade away as shown in Fig. 10 from (c) to (f) because the background value becomes closer to the new pixel and there are no longer any significant differences between them. Depending on the environment, few pixels can be still picked up in the steady state due to noise. They are eliminated by using a higher threshold value and by erosion process and reach the final output shown in the last image in Fig. 10(f). The update takes place as long as the total number of white pixels does not exceed a defined value for the design, (5% of 640x480=15360 is set for the design). This is to avoid the inclusion of the hand in the background while eliminating noise in the absence of the hand during interaction

C. Result of hand extraction and gesture classification

Fig. 11 presents all the results of the extraction of features. Finger counts, hand and moving direction are correctly displayed as required by Table 1. The number of active fingers is output to the seven segment display and shown in Fig. 11 from five to one. The hand orientations, up, left and right are also shown in Fig. 11(e) to (g), respectively. As observed, LEDR0 lit indicating up direction; LEDR1 lit indicating left orientation and LEDR2 lit indicating right orientation as anticipated. Fig. 11(h) shows the hand moving directions. LEDR5 and LEDR6 lit indicating left and right movements as expected.

D. User interface

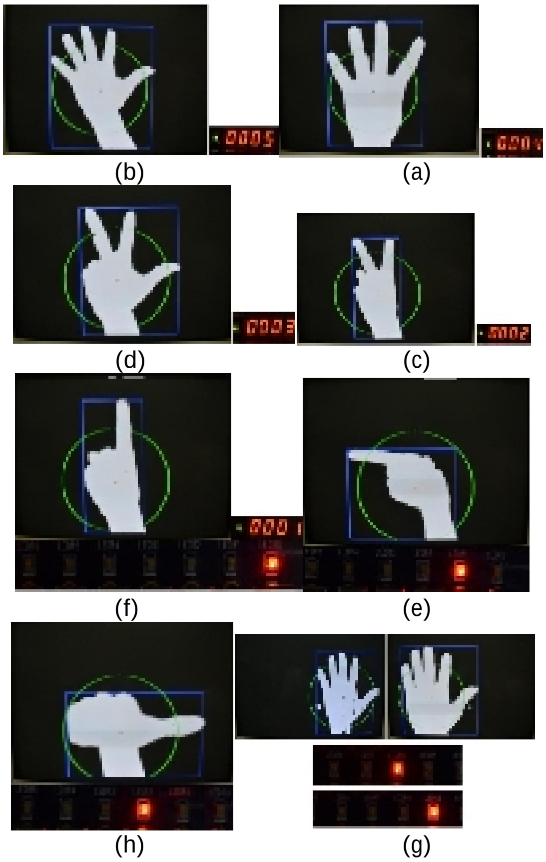

Fig. 11: (a) Five active fingers (b) Four active fingers (c) Three active fingers (d) Two active fingers (e) One active fingers and LEDR0 lit indicating up direction (f) LEDR1 lit indicating left orientation (g) LEDR2 lit indicating right orientation (h) LEDR5 and LEDR6 lit indicating left and right movements.

The interaction with the user interface is shown in Fig. 12. The hand cursor and the living room image are successfully written into RAM and SDRAM via NIOS through the SD-Card and read out by the VGA.

Fig. 12: (a) User Interface (b) Selected lamp via cursor (c) Lamp truned on (d) Selecting TV for control

The selection of appliance and the on/off lamp are working as anticipated. The blue frame is to indicate the highlight and green frame stays there to indicate the lamp is in the 'on' state as shown in Fig. 12 (b) and (c) while the TV selection is shown in Fig. 12 (d) as the cursor moves into the defined box region for the TV.

7. Discussion

The prototype system successfully detects user's hand gestures and translates it into control signals for operating home appliances. Operations supported include turning on/off lamps and fan, dimming/densifying the lights, play/pause, fast-forward/fast-backward music and video movies. Evaluation of the system has been conducted at the Multimedia University (Cyberjaya) Digital Home. Forty students were randomly chosen to evaluate the level of convenience of the proposed system in controlling home appliances. The 5-point Likert scale was implemented to gauge the responses from 1 (strong disagreement) to 5 (strong agreement). With an average of 4.5 as shown in Table 3, responses showed that users deemed the system is desirable for controlling home appliances. The responses were encouraging and reflected that the system can provide a convenient way to control home appliances.

8. Conclusion

An FPGA hand gesture recognition system for home appliance control has been developed and implemented on the Altera Cyclon II DE2 Board. The system is able to detect the movements and gestures of the user's hand and translate it into a number of controlling functions such as turning on/off lamp and fan, dimming/densifying the lights, play/pause, fast-forward/fast-backward music and video movies. Other features of the system include selective running background update and event triggering of home appliances. The algorithms for segmenting, classifying and identifying hand gestures have been developed primarily in Verilog hardware modules and they are performing successfully as anticipated. Due to the digital hardware parallelism of FPGA, the system is able to perform in real-time in spite of running at a relatively low camera frame rate of ~4fps due to the SDRAM bandwidth limitation. The real time requirement is satisfied as images are analysed at a frame rate 60fps much greater than the camera frame rate. In addition, system evaluation response from forty randomly chosen students were encouraging and reflected that the system provide a convenient way to control home appliances. Also, the Quartus II compilation report shows that the design consumes only 30% of the total logic elements of FPGA, leaving a large amount of hardware resources for other usage. However, the system requires careful calibration, in terms of the position of the camera, system parameters such as threshold values, etc., to achieve the balance between robustness and recognition accuracy. The recognition accuracy is susceptible to instability and may cause false detection due to sudden illumination changes in the environment such as shadows that affects the extracted features. Higher threshold value can be used to make the system more robust to these factors at the cost of a more distorted hand image, which in turn affects the recognition accuracy.

Table 3: Evaluation results of the proposed system

Item	No of Subjects	Mean	Standard deviation
The level of convenience of the proposed system in controlling the home appliances	40	4.50	0.54

Acknowledgment

The authors gratefully acknowledge Multimedia University for providing the financial supports as well as DE2 platform to successfully implement the prototype. The authors would also like to thank Pua Yi Xun and Wong Chin Yeow for his helps in completing the system.

References

Alam, M.R.; Reaz, M. B I; Ali, M. A M, "A Review of Smart Homes—Past, Present, and Future," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on , vol.42, no.6, pp.1190,1203, Nov. 2012

Alexander Bochem, Rainer Herpers and Kenneth B. Kent, " Acceleration of Blob Detection within Images in Hardware", University of New Brunswick, Dec 15, 2009, pp 1-37.

BRUSH, A. J. B., LEE, B., MAHAJAN, R., AGARWAL, S., SAROIU, S. & DIXON, C. 2011. Home automation in the wild: challenges and opportunities. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Vancouver, BC, Canada: ACM.

C.V. Joshi, Alwin.D.Anuse, "Use of VHDL in Rapid Prototyping" WSEAS TRANSACTIONS on CIRCUITS and SYSTEMS, Issue 5, Volume 3, July 2004. PP1085 – 1090

Chung Chongyoon and Esa Rantanen, "Gesture Interaction With In-Vehicle Audio and Climate Controls", Rochester Institute of Technology, (Online)Available: http://www.slideshare.net/chung3271/gestural-interaction-with-invehicle-audio-and-climate-control (11 May 2011).

Joe Ballerini, Tiffany Ng, Alex Garcia and Jeremy, "SudoGlove-Hardware Control Using Hand Gesture", (Online) Available: http://jeremyblum.com/2010/05/09/sudoglove / (7 May 2011).

Kofi appiah, andrew hunter, tino kluge, "An FPGA-driven Image Segmentation Algorithm" Proceedings of the 5th WSEAS Int. Conf. on SIGNAL, SPEECH and IMAGE PROCESSING, Corfu, Greece, August 17-19, 2005 (pp282-287)

Lawrence y. Deng, dong-liang lee, yi-jen liu, nick c. Tang, "Human Computer Interaction Based on Hand Gesture Ontology" Proceedings of the 11th WSEAS International Conference on COMPUTERS, Agios Nikolaos, Crete Island, Greece, July 26-28, 2007

LONGBIAO, C., GANG, P. & SHIJIAN, L. Touch-driven interaction via an NFC-enabled smartphone. Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on, 19-23 March 2012 2012. 504-506.

Lopez, S.; Vladimirova, T.; Gonzalez, C.; Resano, J.; Mozos, D.; Plaza, A., "The Promise of Reconfigurable Computing for Hyperspectral Imaging Onboard Systems: A Review and Trends," Proceedings of the IEEE , vol.101, no.3, pp.698,722, March 2013.

Marie Chan, et. al., "A review of smart homes—Present state and future challenges," computer methods and programs in biomedicine, vol. 91, pp. 55-81, 2008.

P. Thitimajshima, "FPGA-Based Real Time Digital Image Processing System", CSCC 2000, MCP 2000, MCME 2000 CD-ROM Proceedings, Athens 2000, pp. 4091-4093.

Paul Viola, Micheal Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", Accepted Conference on Computer Vision and Pattern Recognition, 2011.

Qing Chen, Nicolas D. Georganas, Emil M. Petriu (2007), "Real-time Vision-based Hand Gesture Recognition Using Haar-like Features", Instrumentation and Measurement Technology Conference.

Shahzad Malik, "Real-time Hand Tracking and Finger Tracking for Interaction", December 18, 2003, pp1-21.

Sue Han Lee, Soon Nyean, Cheong, Chee Pun, Ooi, Wei Heng, Siew "Real time FPGA implementation of hand gesture recognizer system", Proceedings of the 15th WSEAS international conference on Computers, July 2011

Takashi Saegusa, Tsutomo Maruyama and Yoshiki Yamaguchi, "How Fast is an FPGA in image processing", Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on , vol., no., pp.77,82, 8-10 Sept. 2008

Viani, F.; Robol, F.; Polo, A.; Rocca, P.; Oliveri, G.; Massa, A., "Wireless Architectures for Heterogeneous Sensing in Smart Home Applications: Concepts and Real Implementation," Proceedings of the IEEE , vol.101, no.11, pp.2381,2396, Nov. 2013

Xu Yuan; Zhou Qinghai; Gong Liwei; Zhu Mingcheng; Teng, R.K.F., "Study of a FPGA real-time multi-cameras cylindrical panorama video system with low latency and high performance," IVMSP Workshop, 2013 IEEE 11th , vol., no., pp.1,4, 10-12 June 2013.

Yu-Ping Lian, Yan Han, Mng-xu HUO, Jin-long CHEN, Yan Zhang. "Design and FPGA verification of a novel reliable real-time data transfer system", Journal of Zhejiang University SCIENCE A, vol.9 No 10, 2008