IMPLEMENTATION OF REAL-TIME SYSTEM ON FPGA BOARD FOR HUMAN'S FACE DETECTION AND TRACKING AUTHOR

MOHD NORHAFIZ HASHIM

A project report submitted in partial
Fulfillment of the requirement for the award of the
Degree of Master Electrical Engineering

Fakulti Kejuruteraan Elektrik dan Elektronik
Universiti Tun Hussein Onn Malaysia

JANUARY 2014

ABSTRACT

Face detection and tracking has been an important and an active research field because it offers many applications, especially in video surveillance, biometrics, or video coding. The goal of this project was to implement a real-time system on an FPGA board to detect and track a human face. The face detection algorithm involved color-based skin segmentation and image filtering. The face location was determined by calculating the centroid of the detected region. A software version of the algorithm was independently implemented and tested on still pictures in MATLAB. Although the transition from MATLAB to Verilog was not as smooth as expected, experimental results proved the accuracy and effectiveness of the real-time system, even under varying conditions of lights, facial poses and skin colors. All calculation of the hardware implementation was done in real time with minimal computational effort, thus suitable for power-limited applications.

ABSTRAK

Pengesanan dan penjejak wajah telah menjadi bidang penyelidikan yang penting dan aktif kerana menawarkan banyak aplikasi, terutamanya dalam bidang pengawasan video, biometrik, atau video kod. Tujuan utama projek ini adalah untuk menerapkan sistem masa sebenar pada papan FPGA untuk mengesan dan menjejak wajah manusia. Algoritma pengesan wajah yang terlibat adalah berasaskan segmentasi warna kulit dan penapisan lapisan gambar. Lokasi Wajah ditentukan dengan mengirakan kadar pusat dalam kawasan yang telah dikesan. Algoritma ini telah diimplementasikan dan diuji dengan menggunakan gambar diam dalam MATLAB. Meskipun penukaran cara tulisan algoritma daripada MATLAB ke bentuk Verilog tidak selancar seperti yang diharapkan, namun demikian hasil eksperimen telah membuktikan konsistensi dan efektif sistem masa sebenar sebenar ini. Tambahan pula dengan keadaan cahaya lampu, bentuk wajah dan warna kulit yang berbeza-beza. Semua cara kerja dan pengiraan pelaksanaan sistem papan FPGA dilakukan berdasarkan masa sebenar dengan keupayaan kompetensi minimal dan sesuai untuk aplikasi yang melibatkan kuasa terbatas.

CONTENTS

CHAPTER 3 METHODOLOGY

CHAPTER 4 RESULTS AND DISCUSSION

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Face detection and tracking is the process of determining whether or not, a face is present in an image. Unlike face recognition, which distinguishes different human faces, face detection only indicates whether or not, a face is present in an image. In addition, face tracking determines the exact location of the face. Face detection and tracking has been an active research area for a long time because it is the initial important step in many different applications, such as video surveillance, face recognition, image enhancement, video coding, and energy conservation. There are three common applications of face detection in real system such as security systems and for content-based coding. In security systems, the accuracy and reliability of face detection is vital as it must correctly detect the facial regions of authorized users from a predefined image database, which will then allow the system to perform face identification to grant them access, and vice versa. The method used to achieve the required level of reliability would normally utilize an algorithm of high complexity, which may incur higher costs and higher expense in terms of memory usage and computational time. Content-based coding is used mainly in video communications such as video conferencing over the Internet. During a video chat, the object of interest is he user's face rather than the background. A face detection application will enable localization of the face, which subsequently allows for allocation of more data bits to encode the face rather than the background. This is known as content-based coding. Thus, the decoded facial image will appear sharper at the expense of a slightly blurred background [1]. Video

communication is usually done in real time, which means that the face detection algorithm has to be efficient or it may affect video performance. Therefore, the method used for such applications should be fast, fairly reliable and simple to implement. Accuracy and precision of face detection is not as critical here as in security systems, but computation time is critical. However, it is interesting to learn how a face detection and tracking system allows power and energy to be saved. Suppose one is watching a television and working on other tasks simultaneously. The face detection system is for checking whether or not the person is looking directly at the TV. If the person is not directly looking at the TV within some time period (i.e. 15 minutes), the TV's brightness is reduced to save energy. When the person turns back to look at the TV, the TV's brightness can be increased back to original. In addition, if the person looks away for too long (i.e. More than one hour), then the TV will be automatically turned off.

Different approaches to detect and track human face, including feature-based, appearance-base and color-based have been actively researched and published in the literature. The feature-based approach detects a human face based on human facial features such as eyes and nose. Because of its complexity, this method requires lots of computing and memory resources. Although compared to other methods this one gives a higher accuracy rate, it is not suitable for power-limited devices. Hence, a color-based algorithm is more reasonable for applications that require low computational effort [2]. In general, each method has its own advantages and disadvantages.

More complex algorithm typically gives a very high accuracy rate, but also requires lots of computing resources.

1.2 Objective of The Project

The aim of this research project is to implement real-time system face detection and tracking using Verilog Hardware Description Language. The requirement of this project is to identify the accuracy and effectiveness of the hardware real-time implementation as the algorithm was able to handle varying types of input video frame. The objective of this project is to design and develop a process flow chart for face

detection and tracking project, to identify, develop and simulate algorithm for face detection and tracking, to design and develop Verilog Hardware Description Language face detection and tracking and to perform experiments and result analysis.

1.3 Problem Statement

Recently, the level of the method using in power saving, surveillance and security system in Malaysia is still in the developing state. It is found that the technologies for this file by using face detection and tracking that have been used so far are passive in nature. There are many algorithms used to detect and track the person of interest with high accuracy todays, such as Color-based skin segmentation, Principle Component Analysis (PCA), Independent Component Analysis (ICA), Kernel PCA (KPCA), Elastic Bunch Graph, 3D Morphable Model and so on. Although the face detection and tracking attract attention of many researchers to extract many useful algorithms, but many of them only focus on the platform of computer software in windows operating system. In other word, there are few to implement the face detection and tracking system in hardware design and real-time. This project is to improve the face detection system by using color-based skin segmentation to get higher accuracy result. Color-based skin segmentation is used for face detection because it can detect the desire face very fast. The algorithm has been used for the detection which achieved high detection accuracy.

1.4 Scope of Project

The scope of this project is to study color-base skin segmentation algorithm, verify and analyze the algorithm using suitable software and implement and result testing in real-time using FPGA Board. The proposed algorithm has been verified in the simulation based on MATLAB SIMULINK®. Hardware implementation in real-time using FPGA De2_70 Cyclone II board and TRDB_D5M camera.

1.5 Thesis Outline

This report is arranged and distributed into five chapters. Chapter 1 has presented a brief introduction of the project mainly about face detection and tracking, the problem statements, the objectives of the project and its scope, and the limitations identified using the proposed approach.

Chapter 2 of the dissertation includes a literature survey related to this project as per referred to previous studies and results obtained by past researchers. It also contains some important findings from past, researchers such as a review of existing face detection and tracking. Their respective advantages and disadvantages, with specific reference to face detection and tracking systems, are discussed.

Together with the literature review carried out in Chapter 2, has helped with the search for face detection and tracking system which potentially can improve performance and cost. Chapter 3 provides a methodology in how this project is conducted in sequence. It also includes the development and progress of the face detection and tracking system implementation. Details in the system are explained and description is provided in this chapter

Chapter 4 contains the results and findings of the project. The result is divided into three sections, section one is Matlab verification for selection algorithm simulation, section two is Hardware Description language (HDL) simulation result and section three is a hardware implementation result in real-time. Simulation result is analyzed and studied.

Lastly is chapter 5 where this chapter concludes the dissertation. It presents a summary of project achievements together with a discussion of their significance. Some recommended future work also presented in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Firstly, this chapter will discuss the details and explains on the theory background and literature reviews that have been done. Followed by the current techniques and algorithms used. Lastly, this chapter, followed by the previous related work that had been done on the face detection system techniques which are employed in the recent development.

2.2 Overview Face Detection

Face detection is first and most crucial step and it is necessary for an efficient face recognition system in order to distinguish the face region and the non-face region of the person of interest. However, it is difficult to detect the person of interest as it has too many variables, such as skin-color, scale, location, orientation, pose, facial expressions, illuminations, occlusions and so on [1, 2]. According to [3], there are two approaches for the face detection. Among them are feature-based approach and image based approach. The examples of the feature-based approach are the skin color, face geometry, motion analysis, and snakes and so on. For the image based approach, it has addresses face detection as a general recognition problem such as face and non-face prototype classes. Chi Zhang and Zhengyou Zhang [4] grouped the face detection approaches can be grouped in four categories. Among the categories are knowledge-based methods,

feature invariant approaches, template matching methods, and appearance based methods.

Table 2.1 Categories of the face detection approaches [2, 4]

Categories	**Explanations**	**Examples**
Knowledge-based method	Use human-coded rules to determine a face based on human knowledge.	Two symmetric eyes, middle nose, mouth underneath the nose.
Feature invariant approaches	Use facial features that are robust to pose and lighting variations or rotation.	Skin colors, edges, and shapes
Template matching methods	Use pre-stored face templates to judge the face image.	Correlation between test images and pre-selected facial templates.
Appearance-based methods	Use face models to perform the detection.	Eigenface

Nowadays, the most popular methods used for face detection is the appearance-based methods and feature invariant approaches. In this process, there are two issues to consider. Among them are the features to extract and the algorithms to apply. The examples of the feature type are Haar-like features and its variations, pixel-based features, Binarized features, generic linear features, statistics-based features, composite features and shape features [4].

2.3 Overview Face Detection Algorithm Principle

Face detection is a very active research topic in the field of computer vision and pattern recognition, Which is widely applied in the identity authentication man-machine interface, visual communication, virtual reality, management of public security files, content based retrieval and many other aspects. Therefore, it requires that the face detection system should be with strong adaptability to all environments.. According to [6], The goal of face detection is to determine whether or not there are any faces in the image and, if present, return the image location and extent of each face. The challenges associated with face detection can be attributed to the following factors:

i) Pose: The images of a face vary due to the relative camera-face pose (frontal, 45 degrees, profile, upside down), and some facial features such as an eye or the nose may become partially or wholly occluded.

ii) The presence or absence of structural components: Facial features such as beards, mustaches, and glasses may or may not be present and there is a great deal of variation among these components including shape, color, and size. Facial expression. The appearance of faces is directly affected by a person's facial expression.

iii) Occlusion: Faces may be partially occluded by other objects. In an image with a group of people, some faces may partially occlude other faces.

iv) Image orientation: Face images directly vary for different rotations about the camera's optical axis.

v) Imaging conditions. When the image is formed, factors such as lighting (spectra, source distribution and intensity) and camera characteristics (sensor response, lenses) affect the appearance of a face.

One of the most significant features of human face's surface is skin color, for the color image, skin color is the relatively concentrated; a stable region in the image. It's better to distinguish a human face from background regions of the skin color [5]. It shows that different race, age, sex with different human facial skin color seemly, but the difference mainly concentrates in brightness, if in a color space which removes brightness, the different face skin color distribution has clusteringBase on this principle, it's feasible to segmented image of the skin color. Skin color segmentation mainly refers to two aspects content color space and skin color model.

2.3.1 Skin Color Segmentation

One of the most significant features of human face's surface is skin color, for the color image, skin color is the relatively concentrated; a stable region in the image. It's better to distinguish a human face from background regions of the skin color. It shows that different race, age, sex with different human facial skin color seemly, but the difference mainly concentrates in brightness, if in a color space which removes brightness, the different face skin color distribution has clustered. Based on this principle, it's feasible to segmented image of the skin color. Skin color segmentation mainly refers to two aspects content color space and skin color model [6].

2.3.2 Color Space

Skin color has its own characteristics, which can form the different expression in the different color space. Therefore, it makes the computer has different the skin color identification ability and treatment effect in different color spaces. The main color space has RGB, CMY/CMYK, YCbCr, HIS (HSV), YIQ, YUV and so on. YCbCr space has the similar composition principle with humanity's visual perception process, which can separate brightness and chroma very well, besides, the color space is discrete, which is easy to realize clustering algorithm and other merits. This paper will describe the skin color model with YCbCr color space. Because the YCbCr space has the similar composition principle with humanity's visual perception process, which can separate brightness and chroma very well, besides, the color space is discrete, which is easy to realize clustering algorithm and other merits [5].

2.3.3 Color Model

The skin color model is one model that it needs to use the algebra (analysis) or look-up table forms to express which pixel's color belongs to the skin color, or express the similar degree between pixel's color and skin color. This paper will use skin color Gaussian model, which is not simply binary skin color, location, but by computing the pixel's probability value, constituting continual data information, then obtain a skin color Probability chart, confirms the color according to the numerical magnitude. This method overcomes the shortcomings of the geometric model, and doesn't need to consider the problem that it's difficult to extract accurately the non-skin color sample in the neural network model. The expression of skin color distribution two-dimensional Gaussian function is [5]:

$$P(Cb,Cr) = exp[-0.5(x-M)TC-1(x-M)] \text{ [5]}$$

P (Cb, Cr) = color space's ,value M = skin color's sample and see = skin color similarity model's matrix

2.4 Software Implementation

According to [7], The first and most important feature on face detection and localizing systems is the skin color. The proposed system uses a matrix that contains a collection of RGB skin values as a reference; these values were used to detect all similar pixels in the input RGB image. Each pixel in RGB image can be represented as a vector with three values; Red, Green and Blue. These values are added together in order to produce a range of colors. Since the reference matrix used contains all RGB vectors that might be a skin color, then the system computes the difference (norm) between each vector in the reference matrix and pixels in the input RGB image within a loop, the equation used to calculate the difference is:

$$norm = \sqrt{(R_i - R_f)^2 + (G_i - G_f)^2 + (B_i - B_f)^2}$$

Where:
R_i: The Red component in the input image.
R_f: The Red component in the reference matrix.
G_i: The Green component in the input image.
G_f: The Green component in the reference matrix.
B_i: The Blue component in the input image.
B_f: The Blue component in the reference matrix.

[7]

Once the difference is less than a threshold (the threshold here is the maximum allowed difference), then this pixel is set to be a skin. The results of this process are a binary image contains 1's and 0's; while the 1's represent all pixels that have similar skin color, and the 0's represents non-skin pixels. The produced binary image goes into a set of modifications as shown in figure 2.1. The first modification is removing the small areas; these small areas are mostly located far away from the actual human skin in the image. On the other hand, it may be located in the actual skin area, so the actual skin areas will contain holes, this can be fixed by filling the holes in the binary image. Another operation is to create a morphological structuring element with the specified neighborhood. The specified neighborhood represents the 1's in the skin binary image. The skin binary image would have a flat linear structure, the dots and non-uniformity in the image will disappear. This operation makes the calculation of the features more accurately.

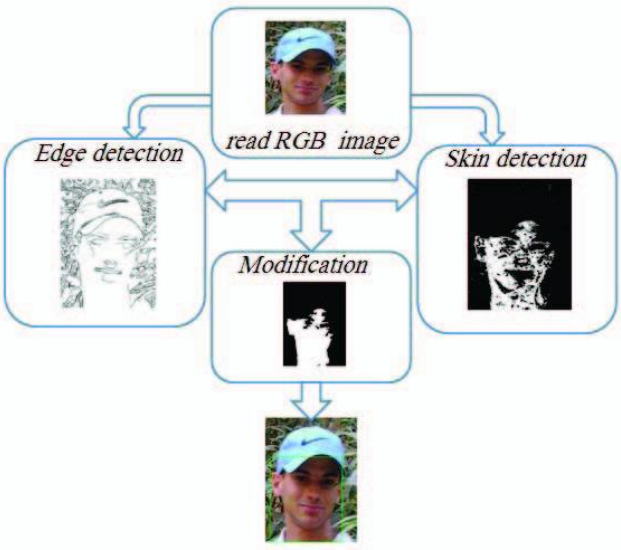

Figure 2.1 : Block diagram for overall system in Matlab [7]

2.5 Previous Hardware Implementation for Face Detection

A paper entitled "Proposed FPGA Hardware Architecture for High Frame Rate (>100 fps) Face Detection Using Feature Cascade Classifiers" [9] proposed the algorithm used for the face detection using Ada Boost method. This Ada boost method can produce fast detection implementation as compared to other methods. The paper also states that the current software version implementations can achieve about 15-25 frames per second (fps) in real time. This paper also proposed that the hardware architecture design on FPGA based on Ada Boost algorithm for face detection in high resolution images at high frame rates (>100 fps). The proposed hardware architecture is first implemented using Verilog and verified by the Modelsim® simulator and Xilinx ISE® tool for synthesis. The target platform is Xilinx XUP development board with on board Virtex-II Pro XC2VP30 FPGA. This FPGA contains 2.44MBits on-chip memory [10]. Based on the synthesis result, the FPGA can operate at a clock speed of 126.8MHz. Using the specific SRAM architecture and piped registers to provide high data access capability and multiple classifiers to parallel processing, the architecture can achieve a theoretical 143 fps detection for 640x480 images and can detecting mega-pixel images in the next stage.

In year 2004, a paper entitled “Face Detection System Based on Feature- Based Chrominance Color Information” presented a face detection system in indoor environment with non-uniform background such as the image captured at the Twins Tower or in computer lab with many computers, tables, windows, and so on, which is based on feature-based chrominance color information [11]. The algorithms used in this paper are the skin color segmentation and eyes candidate estimation. The input color face will go through both algorithms which are the eye candidate estimation and the skin color segmentation and face boundary estimation, in order to find the eye location of the person of interest. The skin color segmentation used in this paper is YCbCr color space where Y represents the luminance component whereas Cb and Cr are the blue chrominance component and red chrominance component respectively. Also, the ranges for the Cb and Cr are between 95 and 126 and between 140 and 168 respectively. For the eye candidate estimation, it was done by multilevel thresholding with 3-level priority in order to estimate the eye position of the person of interest.

A paper entitled “Hardware Implementation for Face Detection on Xilinx Virtex-II FPGA using the Reversible Component Transformation Colour Space” [12] presented Face detection is the process of locating the position where faces are present in an image. Not all proposed face detection methods are suitable for direct hardware implementation. This paper explains a method that utilises the Reversible Component Transformation (RCT) colour space and outlines its transition from a software- to hardware-based implementation. The hardware performance and efficiency of the RCT algorithm is examined using the Xilinx Virtex-II Field Programmable Gate Arrays (FPGA). Results show that there is almost negligible difference in performance after transition to hardware and its implementation on FPGA requires 255,416 NAND gates, which is only slightly more than twice the number of NAND gates of a basic video-in application. Colour segmentation using the RCT was first implemented using MATLAB and its effectiveness was investigated. MATLAB was used for the software implementation because it provides a good platform for the development and testing of software-based image processing algorithms. When the software simulation was found to be satisfactory, the algorithms in MATLAB code were then rewritten in the ANSI-C

language to ease the transition to *HandelC* (a hardware description language used to configure the FPGA). A Celoxica RC200 hardware prototyping board that contains the Xilinx Virtex-II chip. It is an advanced multipurpose hardware prototyping board with video capture/display and audio functionality. Applications are booted and run from either SmartMedia memory cards or from a personal computer (PC) via a parallel port cable. The hardware description language (HDL) used by the Celoxica design suite is known as HandelC. HDL is a programming language that is used to represent, model, test, design and simulate digital hardware [13]. It provides a convenient and compact way to a more structured design and hierarchical representation of digital systems.

A paper entitled ". A Low Cost FPGA System for High Speed Face Detection and Tracking" [14] presented, an FPGA face detection and tracking system for audiovisual communications, with a particular focus on mobile videoconferencing.
The advantages of deploying such a technology in a mobile handset are many, including face stabilization, reduced bytes, and higher quality video on practical display sizes. Most face defection methods, however, assume at least modest general purpose processing capabilities, making them inappropriate for real-time applications, especially for power-limited devices, as well as modern hardware implementations. This paper presented a method which achieves a very high detection and tracking performance and, at the same time, reduced computational complexity in real-time implementations on custom hardware or simple microprocessors. Proposed an FPGA implementation which entails very low logic and memory computational operating system (COS) and achieves extremely high processing rates at very low clock speeds.

According to research project entitled "A Hardware Design of Camera-based User's Presence Detector" [15] by Vasily G. Moshnyaga, Koji Hashimoto and Tadashi Suetsugu introduces to prototype hardware design for detecting user's presence in front of computer-based video camera. The hardware implements basic image processing techniques (filtering, color-based segmentation, thresholding) producing a signal when a human-skin color segment is detected in the image frame. Experiments show that the design allows real-time user monitoring (30fps) with 82% detection accuracy while consuming as 35 times as less power than analogous software. Digital video camera is

the only device capable of monitoring the user, which may or may not wear any badges. The user presence detector takes from the camera a color image, represented by red (R), green (G) and blue (B) image planes, and outputs a logic signal, C, based on the image content. When the user is detected, the signal C is set to one, otherwise C=0. The zero value of C enforces the voltage controller to low-down backlight voltage supply, dimming the display off. When S=1,the display operates as usual [15]. Prototype design of user presence detector in Verilog HDL and synthesized using Synopsis Design Compiler on a single FPGA (Xilinx XC3S250E) board. The board is connected to VGA CMOS image sensor, OmniVision OV7640, through parallel I/O interface. The design runs at 48MHz frequency using 3.3V external voltage and provides user presence detection at 30fps rate. Due to very small capacity of on-chip SRAM memory, input images were limited to 160x120 pixels in size and stored in the SRAM in binary form only. The total power consumption of the design was 150mW, which is 35 times less than software implementation of the user presence detector on desktop PC (Pentium4@2.53GHz). Table 2.2 shows summary of the face detection approaches.This project will use approaches number 1 and 2 because of less computational programming and sharpener image.

Table 2.2 : summary of the face detection approaches

No	Author	Research Title	Research Approaches	Summary
1	Y.H. Chan	Face Detection System Based on Feature-Based Chrominance Colour Information.	Represent a face detection system in indoor environment with non-uniform background based on feature-based chrominance color information [4].The algorithms used in this paper are the skin color segmentation and eyes candidate estimation. The skin color segmentation used in this paper is YCbCr color space where Y represents the luminance component whereas Cb and Cr are the blue chrominance component and red chrominance component [4].	It was done by multilevel thresholding with 3-level priority in order to estimate the eye position of the person of interest [4].Reduce the size of an image while preserving details from the original image. chain code is proposed to quickly locate a face region's boundary. The position of a face is further

				confirmed by eye detection algorithm. Only important features of a face are extracted applying the Modified Golden Ratio, thus serves as potential preprocess module for face recognition [4].
2	Melanie P.L.O	Hardware Implementation for Face Detection on Xilinx Virtex-II FPGA using the Reversible Component Transformation Colour Space	This paper explains a method that utilizes the Reversible Component Transformation (RCT) colour space and outlines its transition from a software- to hardware-based implementation. The hardware performance and efficiency of the RCT algorithm is examined using the Xilinx Virtex-II Field Programmable Gate Arrays (FPGA) [5]. Use MATLAB simulation then rewritten using Handel-C as a FPGA language coding technique.	Component transformation colour space is effective and can be implemented in hardware without sacrificing much of the algorithms' performance. The efficient use of NAND gates to perform face detection on the Xilinx Virtex-II FPGA suggests that it is has low hardware requirements, which makes it an attractive option for implementing hardware face detection. It is also relatively cheap to implement and is robust in design. It may be suitable for non-critical face detection purposes as well as simple video and image coding applications [5].
3	Stavros.P	A Low Cost FPGA System for High Speed Face Detection and Tracking	This focus on mobile videoconferencing. The advantages of deploying such a technology in a mobile handset	-present a method which achieves a very high detection and tracking

			are many, including face stabilization, reduced bitrates, and higher qualify video on practical display sizes. Most face defection methods, however, assume at least modest general purpose processing capabilities, making them inappropriate for real-time applications, especially for power-limited devices, as well as modest custom hardware implementations [7].	performance and, at the same time, entails a significantly reduced computational complexity. allowing real-time implementations on custom hardware or simple microprocessors. We then propose an FPGA implementation which entails very low logic and memory cost and achieves extremely high processing rates at very low clock speeds [7].
4	Vasily.G.M	A Hardware Design of Camera-based User's Presence Detector	This paper introduces a prototype hardware design for detecting user's presence in front of computer-based video camera. The hardware implements basic image processing techniques (filtering, color-based segmentation,thresholding) producing a signal when a human-skin color segment is detected in the image frame.	- Experiments show that the design allows real-time user monitoring (30fps) with 82% detection accuracy while consuming as 35 times as less power than software.

CHAPTER 3

METHODOLOGY

3.1 Introduction

There are three main processes involved in the project. The process consists of identifying an appropriate algorithm, the test algorithm with the software MATLAB, the algorithm converts into Verilog language and the last is a test run in real time on Alterra DE2 -70 FPGA board. Firstly, the algorithm is tested on a static image using MATLAB software. Several static images were analyzed using MATLAB software involving the operating system window. The purpose of this step is to obtain the appropriate coding algorithm to produce the output face image. Static background image changing with an aim to produce a suitable algorithm to be tested in real time. Input face image is adopted proportional pixel size of 2500 x 1900 and it is a .png format, and it goes into the screening process to detect lose face - wide importance. In this step, the input face image is converted into YCbCr color space conversion formula color. Followed by expansion filter, connected component labeling, projection histogram, the proportion of crop image size to 2500 x 1900. Once the algorithm has been optimized, and the algorithm will enter exchangeable into Verilog language. Finally, this algorithm is stored in the database hardware and tested in real time.

3.2 Tool Used

There are several tools used to assist in the face detection system for this project. These include the Integrate Development Environment (IDE) used for hardware or software design, and the window operating system used in completing this project. Among the IDE used are MATLAB R2011a and QuartusII 8.1. The hardware board used in this project is the Altera DE2-70 FPGA platform, TRDB-LTM touch screen and TRDB-5M camera. MATLAB is a programming environment for algorithm development, data analysis, visualization, and numerical computation. In this project, the database pictures will be trained in the MATLAB environment with MATLAB version 7.12 (R2011a). The reasons are to train the database pictures in the MATLAB environment, but not in the firmware development are to reduce the time consuming and to reduce the needs a lot of memory that used to accommodate the whole process. Also, the image format type is used in this project is .png file format and it has supported by MATLAB version 7.12 (R2011a). Altera Quartus II is a tool produced by Altera. It is used for analysis and synthesis of HDL designs, which is used to compile the hardware designs, perform timing analysis by clock setting, examine RTL diagrams of the hardware design, simulate a design's reaction to different stimuli, and configure the target device with the programmer. Now, the latest version is 11.1. However, this project will use version 8.1 as the version of 11.1 do not have the waveform simulation. The hardware description language that's used in this project is a Verilog coding. FPGA states for field-programmable gate array. It is an integrated circuit designed that composed of many fixed logic components that are called “logic block”. The logic blocks include memory elements, logic gates, Nios II processor, input/output interface, switches, LEDs and so on. Also, the FPGA contains a hierarchy of reconfigurable interconnects that allow the blocks to be connected together. This project will use the FPGA board with Cyclone II of EP2C70F896C6N.Refer to the Appendix D, the board of EP2C70F896C6N consists of 32MB SDRAM, SD memory, USB port and so on. Hence, the input images are obtained through theGPIO port on the board of the Cyclone II of EP2C70F896C6N.

3.3 Algorithm

Figure 3.1 shows general software algorithm stages for face detection of static image using MATLAB software.

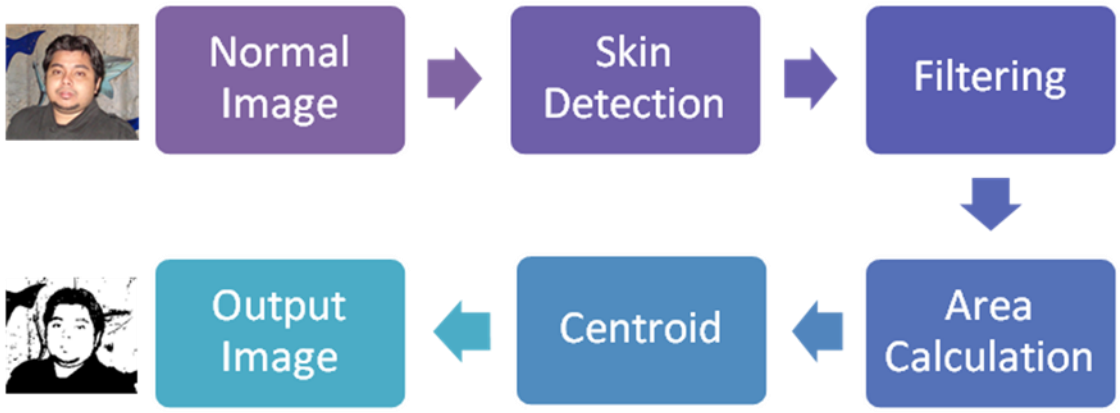

Figure 3.1: General software algorithm stage

The skin detection algorithm was derived from the method described in [1]. Color segmentation has been proved to be an effective method to detect face regions due to its Low computational requirements and ease of implementation. Compared to the featured based method, the color-based algorithm required very little training. First, the original image was converted to a different color space, namely modified YUV. Then the skin pixels were segmented based on the appropriate U range. Morphological filtering was applied to reduce false positives. Then each connected region of detecting pixels in the image was labeled. The area of each labeled region was computed and an area-based filtering was applied. Only regions with large area were considered face regions. The centroid of each face region was also computed to show its location.

3.3.1 Modified YUV Color Space

Converting the skin pixel information to the modified YUV color space would be more advantageous for human skin tones tend to fall within a certain range of chrominance values (i.e. U-V component), regardless of the skin type. The conversion equations are shown as follows [12].

$$Y = [R + 2G + B] / 4$$
$$U = R - G$$
$$V = B - G$$

These equations allowed threshold to work independently of skin color intensity. As seen in Figure 3.2, the blue channel had the least contribution to human skin color.

45 34 30	#2D221E	
60 46 40	#3C2E28	
75 57 50	#4B3932	
90 69 60	#5A453C	
105 80 70	#695046	
120 92 80	#785C50	
135 103 90	#87675A	
150 114 100	#967264	
165 126 110	#A57E6E	
180 138 120	#B48A78	
195 149 130	#C39582	
210 161 140	#D2A18C	
225 172 150	#E1AC96	
240 184 160	#F0B8A0	
255 195 170	#FFC3AA	
255 206 180	#FFCEB4	
255 218 190	#FFDABE	
255 229 200	#FFE5C8	

Figure 3.2 **:** Different skin tone sample [12].

3.3.2 Skin Detection

After skin pixels were converted to the modified YUV space, the skin pixels can be segmented based on the following experimented threshold.

$$10 < U < 74$$

$$-40 < V < 11$$

According to [14], leaving out the blue channel would have little impact on threshold and skin filtering. This also implies the insignificance of the V component in the YUV format. Therefore, the skin detection algorithm using here was based on the U component only. Applying the suggested threshold for the U component would produce a binary image with raw segmentation result, as shown in figure 3.3

Figure 3.3 : Result after threshold

3.3.3 Morphological Filtering

Realistically, there are so many other objects that have color similar to the skin color. As seen in Figure 3.4, there are lots of false positives present in the raw segmentation result. Applying morphological filtering—including erosion and hole filling would, firstly, reduce the background noise and, secondly, fill in missing pixels of the detected face regions, as illustrated in Figure 3.4. MATLAB provided built-in functions—imerode and imfill for these two operations:

```
outp = imerode(inp, strel('square', 3));
```

The command imerode erodes the input image inp using a square of size 3 as a structuring element and returns the eroded image outp. This operation removed any group of pixels that had size smaller than the structuring element's:

```
outp = imfill(inp,'holes');
```

The command imfill fills holes in the binary input image inp and produces the outputimage outp. Applying this operation allowed the missing pixels of the detected face regions to be filled in. Thus, it made each face region appear as one connected region.

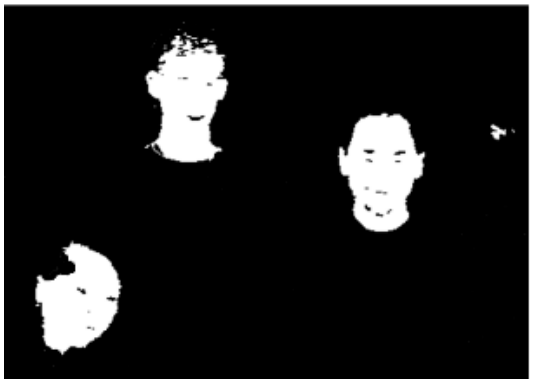

Figure 3.4 : After morphological filter

3.3.4 Component Labeling and Area Calculation

After each group of detected pixels became one connected region, connected component labeling algorithm was applied. This process labeled each connected region with a number, allowing us to distinguish between different detected regions. The built-in function bwlabel for this operation was available in MATLAB. In general, there are two main methods to label connected regions in a binary image—known as recursive and sequential algorithms:

```
[L, n] = bwlabel(inp);
```

The command bwlabel labels connected components in the input image inp and returns a matrix L of the same size as inp. L contains labels for all connected regions in inp. n contains the number of connected objects found in inp. The command regionprops can be used to extract different properties, including area and centroid, of each labeled region in the label matrix obtained from bwlabel:

```
face_region = regionprops(L,'Area');
face_area = [face_region.Area];
```

The two commands above performed two tasks , one is extract the area information of each labeled region and second, store the areas of all the labeled regions in the array face_area in the order of their labels. For instance face_area(1) = 102 would mean the area of the connected component with label “1” is 102 pixels.

3.3.5 Area-Based Filtering

Note that morphological filtering only removed some background noise, but not all. Filtering detected regions based on their areas would successfully remove all background noise and any skin region that was not likely to be a face. This was done based on the assumption that human faces are of similar size and have largest area compared to other skin regions, especially the hands. Therefore, to be considered a face

region, a connected group of skin pixels need to have an area of at least 26% of the largest area.

This number was obtained from experiments on training images. Therefore, many regions of false positives could be removed in this stage, as depicted in Figure 3.5.

```
face_idx = find(face_area > (.26)*max(face_area));
face_shown = ismember(L, face_idx);
```

These two commands performed to look for the connected regions whose areas were of 26% of the largest area and store their corresponding indices in face_idx (2) output the image face_shown that contained the connected regions found in Figure 3.5.

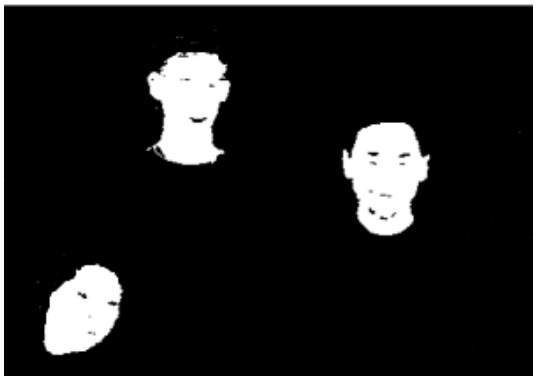

Figure 3.5 : After Area Based Filtering

3.3.6 Centroid Computation

The final stage was to determine face location. The centroid of each connected labeled face region can be calculated by averaging the sum of X coordinates and Y coordinates separately. The centroid of each face region in Figure 3.6 is denoted by the blue asterisk. Here the centroid of each connected region was extracted using regionprops.

REFERENCES

1. Oliver Jesorsky, Klaus J. Kirchberg, and Robert W. Frischholz (2001), “Face Detection Using the Hausdorff Distance”. Proc. Third International Conference on Audio- and Video-based Biometric Person Authentication, Lecture Notes in Computer Science,Vol 4, pp. 90–95.
2. M H Yang, D J Kriegman, N Ahuja (2002),”Detecting Faces in Images: A survey”, Pattern Analysis and Machine Intelligence, 24(1):34-58.
3. Jones M.J.,Rehg J.M. (1999),”Statistical color models with application to skin detection”,In:Proc.IEEE Conf.on Computer Vision and Pattern Recognition, Vol 1,pp.274-280.
4. M.J.Jones,J.M.Rehg (2002), “Statistical color models with application to skin detection”, Journal of Computer Vision, vol.46,pp:81-96 .
5. A.L.Douglas (2009),. “Interactive embedded face recognition”, Journal of Object Technology ,Vol 8,pp.23-53.
6. Z.Cha (2010), “Face Detection and Adaption.Series” ,Editors. Morgan & Claypool,Vol 2,pp:50-150.
7. Li.Y (2011), “Study on Face Detection Algorithm based on Skin Color Segmentation and AdaBoost Algorithm”, International Conference on Electronics Computer Technology,ICETT 2011.
8. D.N.Chandrappa , M.Ravishankar and D.R.RameshBabe (2010) “AutomatedDetection and Recognition of Face in a Crowded Scene” International Journal of Computer and Network Security, Vol. 2, No. 6, pp.65-70.
9. Jure Kova C, Peter Peer, and Franc Solina (2003) “Human Skin Colour Clustering for Face Detection”.IEEE Region8, Eurocon 2003,Vol 2, pp.144-148.
10. Ming Hu and Qiang Zhang , Zhiping Wang (2008) “Application of Rough Sets to Image Pre-processing for Face Detection” IEEE International Conference on Information and Automation , ICIA2008, vol 2,pp:245-248.

11. V.G.Moshnyaga, E.Morikawa (2005), “LCD Display Energy Reduction by User Monitoring”, Proc. ICCD, 2005, pp.94-97.
12. E.Hjelmas, B.K.Low (2001), “Face detection: A survey”, Computer Vision and Image Understanding, Vol.83,no.3, pp.236-274.
13. P.Viola and M.Jones (2004), “Robust Real-time Face Detection”, Int. Journal of Computer Vision, Vol 57(2),pp.137-154,2004.
14. D.L. de Ipina, P.R.S. Mendonca, A.Hopper (2002), “TRIP: A Low-Cost Vision Based Location System for Ubiquitous Computing”, Personal and Ubiquitous Comp. Journal,Vol 6(3), pp.206-219.
15. B.D. Zarit, B.J. Super, and F.K. Quek (1999), “Comparison of five color models in skin pixel classification,” ICCV’99 Int.Workshop on recognition, analysis and tracking of faces and gestures in Real-Time systems, pp. 58–63.
16. M.Soriono, S.Huovinen, and M.Laaksonen (2000), “Skin detection in video under changing illumination conditions,” Proc. 15th int Conference on Pattern Recognition,vol. 1, pp. 839–842 .
17. R.Schumer, and K.Barner (1998), “A color-based classifier for region identification in video,” Visual Communications and Image Processing 1998, SPIE, vol. 33, pp.189–200.
18. Sanjay Kr. Singh, D. S. Chauhan, Mayank Vatsa, Richa Singh, “A Robust Skin Color Based Face Detection Algorithm ” Tamkang Journal of Science and Engineering, Vol. 6, No. 4, pp. 227-234, 2003.
19. Ming.H.Y (2002), “Detecting Faces in Images: A Survey”, IEEE Transaction on pattern analysis and machine intelligence,Vol 24,pp:34-58.
20. A.M. Aibinu (2012), “Performance Analysis of ANN based YCbCr Skin Detection Algorithm”, International Symposium on Robotics and Intelligent Sensors 2012, IRIS2012,Vol 41,pp:1183-1189.
21. S. Paschalakis and M. Bober (2003) “A low cost FPGA system for high speed face detection and tracking.” Proceedings of 2003 IEEE International Conference on Field-Programmable Technology (ICFPT),Vol 17, pp: 214-221.

22. D. Nguyen, and D. Halupka (2006), “Real-Time Face Detection and Lip Feature Extraction Using Field-Programmable Gate Arrays”, IEEE trans. Systems, Man, and Cybernetics, vol. 36, no. 4, pp: 50-65.
23. K.K. Sung and T. Poggio (1998), “Example-Based Learning for View-Based Human Face Detection,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 39-51.
24. G. Yang and T.S. Huang (1997), “Human Face Detection in Complex Background”, Pattern Recognition, Vol, 27, no.9, pp. 712-735, 1997 .
25. Hung.C.L (2007), “Proposed FPGA Hardware Architecture for High Frame Rate (>100fps) Face Detection Using Feature Cascade Classifiers”, Theory , application and system conference, IEEE BTAS 2007 IEEE ,Vol 1,pp:25-63.
26. Virtex II Pro FPGA Family Data Sheets, pp50-60.
27. Chan.Y.H (2004) ,”Face Detection System Based on Feature-Based Chrominance Colour Information”, IEEE conference on Computer graphics,visualization and imaging 2004,CGIV 2004 ,pp:45
28. Melanie.P.(2006), “Hardware Implementation for Face Detection on Xilinx Virtex-II FPGA using the Reversible Component Transformation Colour Space”, Electronic Design, Test and Applications, 2006. DELTA Workshop, pp:46.
29. Zainalabedin .N (1998), “VHDL: Analysis and Modeling of Digital Systems”, McGraw-Hill, 2nd edition.
30. Vasily G(2008), “A Hardware Design of Camera-based User’s Presence Detector “,Systems, Man and Cybernetics, IEEE International conference, SMC 2010, Vol 7,pp:429-432.
31. Theocharides, T.; Link, G. (2004); Vijaykrishnan, N.; Irwin, M.J.; Wolf, W., (2004) "Embedded hardware face detection", Proceedings. 17th International Conference on VLSI Design,Vol 17, pp. 133 – 138.
32. Ramdas, T.; Li-minn Ang; Egan, G.(2004), "FPGA implementation of an integer MIPS processor in Handel-C and its application to human face detection", TENCON 2004, IEEE Region 10 Conference Volume A, Vol. 1, pp. 36 – 39.

33. K.C.YowandR.Cipolla (1997) ,“Feature-BasedHumanFace Detection”, Image and Vision Computing, vol. 15, no. 9, pp. 713-735.

34. Hakam.S (2013) ,”Human Face Detection Using Skin” ,International conference on modeling optimization and computing, Vol 38, pp:2748-2753.