REAL-TIME FACE DETECTION AND TRACKING USING SOFTWARE AND HARDWARE DESIGN

CHEN KOK KEAT

UNIVERSITI SAINS MALAYSIA

2018

REAL-TIME FACE DETECTION AND TRACKING USING SOFTWARE AND HARDWARE DESIGN

by

CHEN KOK KEAT

Thesis submitted in fulfilment of the requirements

for the degree of

Bachelor Engineering (Electronic Engineering)

JUNE 2018

ACKNOWLEDGEMENT

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my supervisor, Dr. Mohd Ilyas Sobirin Mohd Sazali for his invaluable advice, guidance and his enormous patience throughout the development of the project. His immense knowledge, profound experience and professional expertise in Field Programmable Gate Array (FPGA) has enabled me to complete this project successfully. Without his guidance and relentless help, this thesis would not have been possible. I could not have imagined having a better supervisor in the development of this project.

Deepest thanks and appreciation to my parents, family, and others for their cooperation, encouragement, constructive suggestion and full of support for the report completion, from the beginning till the end. Also thanks to all my friends and everyone, that have been contributed by supporting my work and help me during the final year project progress until it is fully completed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
AI	Artificial Intelligence
ASIC	Application-Specific Integrated Circuit
CLB	Configurable Logic Blocks
CPU	Central Processing Unit
DSP	Digital Signal Processing
FPGA	Field Programmable Gate Array
HDL	Hardware Description Language
HSI	Hue, Saturation, Intensity
HSMC	High Speed Mezzanine Card
IoT	Internet of Things
PSNR	Peak Signal-to-Noise Ratio
RGB	Red, Green and Blue
RTL	Register Transfer Logic
VGA	Video Graphics Array

REAL-TIME FACE DETECTION AND TRACKING USING SOFTWARE AND HARDWARE DESIGN

ABSTRAK

Projek ini menyampaikan reka bentuk dan pelaksanaan sistem pemprosesan video iaitu algoritma pengesanan muka manusia dalam FPGA. Pengesanan dan penjejakan muka telah menjadi bidang penyelidikan yang aktif disebabkan terdapat banyak aplikasi yang penting, contohnya, pengawasan video, biometrik, pengekodan video dan lain-lain lagi. Matlamat projek ini adalah untuk melaksanakan satu sistem video dengan masa sebenar dalam FPGA bagi pengesanan muka manusia. Projek algoritma pengesanan muka ini adalah berdasarkan segmentasi warna kulit manusia dan penapisan imej/video. Lokasi muka adalah ditentukan dengan kiraan centroid kawasan muka yang dikesan. Satu algoritma berversi perisian telah dilaksanakan secara bebas dan diuji dalam MATLAB/Simulink. Walaupun peralihan daripada pengekodan berversi perisian kepada kod berbahasa Verilog HDL adalah tidak betul-betul sama, keputusan daripada eksperimen membuktikan fungsi, ketepatan, dan keberkesanan sistem video masa sebenar ini. Bagi pelaksanaan FPGA, keseluruhan algoritma pengesanan muka boleh dilaksanakan pada peranti Cyclone IV FPGA dengan menggunakan Altera DE2-115. Video atau imej input daripada satu kamera iaitu Terasic Capacitive Multi-Touch LCD with Camera Module (MTLC) manakala sistem output dipaparkan pada satu monitor VGA. Kesemua pengiraan dalan pelaksanaan perkakasan FPGA adalah dilakukan dalam masa sebenar dengan usaha pengkomputeran minimum, maka sesuai untuk penggunaan kuasa terhad.

REAL-TIME FACE DETECTION AND TRACKING USING SOFTWARE AND HARDWARE DESIGN

ABSTRACT

This project presents the design and the implementation of face detection algorithm for video processing applications on FPGAs. Face detection and tracking has been an active research topic because it offers many important applications, especially in video surveillance, biometrics, video coding and so on. The goal of this project was to implement a real-time system on an FPGA to detect and track human faces. The face detection algorithm involved colour-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 in MATLAB Simulink. Although the transition from software version to HDL (Verilog) was not exactly the same between both approaches, experimental results proved the functionality, accuracy and effectiveness of the real-time system, even under varying conditions of lights, facial poses and skin colours. For FPGA implementation the entire face detection algorithm was implemented on Cyclone IV FPGA device using Altera DE2-115. The input video or image came from a Terasic Capacitive Multi-Touch LCD with Camera Module (MTLC) and the outputs were displayed on a VGA monitor. All calculations of the hardware implementation were done in real time with minimal computational effort, thus suitable for power-limited applications.

CHAPTER 1

INTRODUCTION

1.1 Research Background

Nowadays, many embedded systems are showing growing demand for high processing power. As a result, the electronic industry begins to follow the trend that the systems are being developed to use Field Programmable Gate Array (FPGA) with the purpose of offloading the processing functions which is traditionally done by the Central Processing Unit (CPU) or Application-Specific Integrated Circuit (ASIC). One of the reasons is the speed advantage of direct hardware execution on the FPGA. It can be estimated routinely 10x to 100x the speed of that to the equivalent software algorithm [1]. Hence, this has attracted the interest of the supercomputing community as well as Digital Signal Processing (DSP) system developers. FPGA offers significant advantages over microprocessors and DSP for high performance (taking advantage of hardware parallelism which offers much higher throughput than DSPs) [2].

On the other hand, video processing has always been one of the popular research topics because it offers many useful applications, for example in video surveillance, biometric, medical, etc. In addition, the Industry 4.0 which represents the fourth industrial revolution in manufacturing and industry is the current industrial transformation with automation, data exchanges, cloud, cyber-physical systems, robots, big data, artificial intelligence (AI), Internet of Things (IoT) and semi-autonomous industrial techniques to

realize smart industry and manufacturing goals in the intersection of people, new technologies and innovation [3]. Examples of advanced technologies which integrate video processing applications include self-driving cars, smart drones, virtual/augmented reality and so on. All these smart systems have one similar key requirement which is that processing the image or video data in real time implementation.

Face detection and tracking has also been an active research area for a long time because it is the initial but also essential step in many different applications, such as video surveillance, face recognition, image enhancement, and energy conservation. While one of the common applications of face detection in daily life system such as security system, the accuracy and reliability of face detection is important 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. The method used to achieve the required level of reliability would normally utilize an algorithm of high complexity, which may need higher costs and higher expense in terms of memory usage and computational time [4].

Furthermore, video communication is usually done in real time, which means that the face detection algorithm has to be efficient or it may affect the 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 example is that while 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 a certain period (e.g. 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 the original. In addition, if the person looks away for too long (e.g. more than 30 minutes), then the TV will be automatically turned off.

However, the need for video processing especially in real time requires high throughput rate [5] because video processing applications often require a large amount of data transfers between the input and output of a system. This large amount of data needs to be stored in memory, transferred to processing blocks, and sent to the display unit. The processing of large data in high speed makes these types of applications often being implemented in dedicated hardware such as FPGA that is capable of parallel processing.

There are basically two general alternatives for video processing development. Firstly, the software approach, which applies execution of software instructions in an embedded processor system. The advantages are the high computational capability and the availability of many development libraries optimized for efficient execution. Data flow is limited by processor bus (16-bit, 32-bit, etc.) and the processing speed of a microprocessor or effectively the instruction execution rate (instructions per second) depends upon the architecture whether the program memory and data memory share the same bus or a separate bus for each of them - the Von-Neumann and Harvard architecture [6].

On the other hand, the hardware approach allows acceleration of the algorithms by outperforming the processor capability by using custom designed implementation of specific algorithms on the programmable logic provided by the FPGA devices. However, the development process usually involves a big amount of effort and time for producing an optimized design comparing to the software approach. Designers still have to think at

the logic level rather than at the algorithmic level even with libraries of common functions like adders and multipliers [7].

There are many different approaches to detect and track human face, including feature-based, appearance-base and colour-based. The feature-based approach detects a human face based on human facial features such as eyes and nose. Due to its complexity, this method requires lots of computing and memory resources. Comparing to other methods, it gives higher accuracy rate, but it is not suitable for power-limited devices. Hence, colour-based algorithm is more reasonable for applications that require low computational effort [8]. 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 and longer development time.

In summary, this project will focus studying on implementing low-level video processing based on application of face detection and tracking on FPGA device by using both software and hardware design approach. By utilising the advantage of vast available functions and libraries for computational functionality, the software approach serves as the fast prototyping, minimise the development effort, and testing on the algorithms to be identified before starting on developing the hardware design. Therefore, the HDL development can be simplified and accelerated, while at the same time its advantage of fast processing speed can be utilised during the real-time hardware implementation.

1.2 Problem Statement

Video surveillance has been widely used to monitor security sensitive areas such as banks, department stores, airports, crowded public places, etc. This emerging trend in video surveillance applications makes low-level video processing like detection and tracking of objects plays an important role in the analysis of the data for higher level of

security assessment. However, the problem arises considering the increasing complexity of the video processing algorithm requirements in terms of processing performance, power and cost efficiency.

Video processing applications usually involve processing of large amount of data and if done serially it becomes extremely difficult to achieve real time implementation as it produces delay in carrying out all the operations [9]. The complexity also shoots up drastically. Furthermore, the general-purpose processors or the DSP boards are designed to address a broad user group and has a fixed hardware structure. It means that the transistor memory, connections and peripherals are constant. All the operations are predefined and user just uses them in a sequential way.

On the other hand, in the FPGA logic cells and the interconnections between them can be determined by the user. And the operations are not predefined as in the case of general processors thus making it possible to process the data in parallel using Hardware Description Language (HDL). Therefore, obviously for this project FPGA device is chosen to be used as the solution for providing the real-time video processing speed with minimum resource usage requirements.

In most cases, image and video processing system developers and designers are high level software practitioners and they rarely know one of the available HDLs (VHDL, Verilog) [10]. Usually, these algorithms are first developed using high-level programming languages (C, C++) and later implemented on the FPGAs. The last step requires that designers use one of the HDLs, translating and adapting the high-level algorithms. In addition, only synthesizable constructs must be used in this phase, i.e. designers use Register Transfer Logic (RTL) subsets of HDLs, ignoring most high level powerful constructs.

Although the face detection and tracking has attracted the attention of many researchers to study on many useful algorithms, most of them only focused on the development platform of computer software in windows operating system. In other words, there are fewer that truly implement the face detection and tracking system in hardware design and achieving real-time requirement at the same time.

In terms of contribution, this project is able to implement a human face detection and tracking video processing algorithms which may serve as the key processing components in the smart surveillance development. Colour-based skin segmentation is used for face detection because it can detect the face object very fast and simple to be implemented. The algorithm has been used for the detection which achieved high detection accuracy. The software design is used for fast algorithm prototyping purpose while the hardware design is used for the real-time implementation.

1.3 Objectives of Research

The objectives of the project are:

1. To identify, develop and simulate a low-level video processing algorithm, which is human face detection and tracking, using software design.
2. To develop a low-level video processing algorithm, which is human face detection and tracking, using HDL, then implement the design on the FPGA platform.
3. To perform experiments and results analysis from two different approaches in terms of accuracy and performance in real-time implementation of the algorithms

1.4 Scope of Research

The scope of research is to study low-level video processing which is colour-based face detection and tracking algorithm in this project. The proposed algorithms are

investigated, simulated and verified using software design based in MATLAB Simulink environment. This step is considered that the video system is designed in software level.

Next, in terms of hardware design process, the algorithms are designed using HDL and implemented in real-time using FPGA Board. The video processing system will collect live video data input acquired from a digital sensor camera, and the outputs will be displayed as on-screen display real time generation on a VGA monitor.

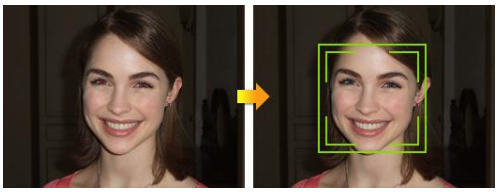
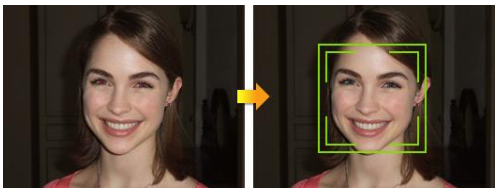
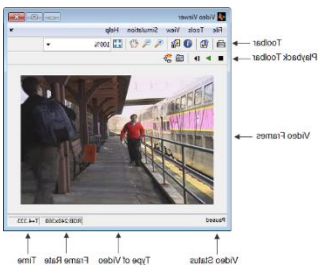
As shown in Table 1.1, it shows and compares the general scope and requirements to be fulfilled in order to apply either software or hardware design approach.

1.4.1 Limitation of research

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 presence of a face. Therefore, the project is limited to low-level video processing which is to detect and track a face object, but not recognise the face.

The project implements skin colour-based face detection and tracking, meaning one of the weaknesses is that it would produce false positive which is object with similar colours to skin colours. This problem is usually solved by adding feature-based detection function into the algorithms. Because of its complexity, this method requires longer development time. Thus, accuracy rate and precision of face detection is not as critical here as in this video processing systems, but computation time is critical. Overall, the main focus of this project is to implement a real-time video processing system in FPGA hardware so that to make use of its parallel processing capability.

Table 1.1: Comparison of the research scope of both software and hardware design

	Software Design	Hardware Design
Input	 • Laptop Webcam – MacBook Pro 720p FaceTime HD webcam	 • Terasic Capacitive Multi-Touch LCD with Camera Module (MTLC) - 5-Megapixel
Processing Algorithms	 • Face detection and tracking • MATLAB language • Simulink	 • Face detection and tracking • Hardware Description Language • Quartus Prime
Processing Unit	 • CPU - 2.5 GHz Intel "Core i5" processor	 • FPGA – Altera DE2-115 Board
Output	 • Simulink Video Viewer block	 • VGA Monitor

1.5 Thesis Outline

Overall, this thesis is arranged and distributed into five chapters which are Chapter 1: Introduction, Chapter 2: Literature Review, Chapter 3: Methodology, Chapter 4, Results and Discussion, and Chapter 5: Conclusion.

Chapter 1 describes a brief introduction and the general ideas about the project including the research background, problem statement, objectives, scope and limitations of this project.

Chapter 2 presents the literature reviews from related previous studies with their important findings and results obtained by other researchers. Background studies, previous works related to this project, fundamental theories and useful knowledge of image/video processing techniques and algorithms are discussed in this chapter.

Chapter 3 presents the methodology used in the implementation workflow of this project as well as the algorithms developed through either software or hardware design. It includes the development of the face detection and tracking algorithm processing procedures for each approach. Details in both system designs are explained and the descriptions are provided in this chapter.

Chapter 4 contains the results and findings of the project. The result is divided into two sections, section one is MATLAB verification for selection algorithm simulation, and section two is a hardware implementation results in real-time. The results obtained is then further compared, analysed and discussed.

Lastly, Chapter 5 concludes the thesis. It presents a summary of project achievements together with a discussion of their significance. Some recommended future works are also presented in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Firstly, this chapter will discuss and explain the details based on previous related work that had been done on the face detection system techniques which have been employed in the recent development. This is followed by the background studies which are the basic knowledge about the image/video processing algorithms.

2.2 Previous Work

About the related works done, the reviews are arranged into three topics to be focused namely, (1) FPGA in video processing system which highlights the advantages of using FPGA; (2) Consideration of software and hardware level design by exploring each approach's advantages and the possibility of extracting the benefits from both software and hardware design to be used in this project development workflow; and (3) face detection and tracking which will be mainly reviewed from algorithms aspect and FPGA implementation aspect.

2.2.1 FPGA in video processing system

Kehtarnavaz and Gamadia highlighted that real time image and video processing system involves processing vast amounts of image data in a timely manner for extracting useful information, which could mean anything from obtaining an enhanced image to intelligent scene analysis [11]. Digital images and video signals are essentially

multidimensional signals and are thus quite data intensive, requiring a significant amount of computation and memory resources for their processing. One of the solutions is that by using FPGA which can provide advantages of high data throughput rate, low power energy consumption, minimum time-to-market and cost, rapid prototyping of complex algorithms and simplify debugging and verification.

2.2.2 Consideration of software and hardware level design

Generally, the design effort and time needed for FPGA hardware design is usually higher. Therefore, image/video processing system usually modelled using high level language before a design is implemented in HDL code. Han has introduced and evaluated the use of model-based design to tackle the increasing complexity in the design of embedded system, where the level of complexity may consume up to 50% to 70% of the development time just for verification of the design [12].

Hai, Pun, and Haw, demonstrated the usage of Simulink as design environment on developing simple video extraction like edge detection. With the use of HDL Coder, the tool greatly helps the designer to provide visualization of rich graphic display on the hardware design and the high level of abstraction allows designer to focus on algorithms instead of coding language. As a result, the project development time was tremendously reduced [13].

One of the recent studies from Alareqi, Mezouari, and Hlou also proved in their research that optimization is possible in high level design of video processing system, which in their project is an edge detection filter, applied on FPGA platform [14]. The optimization process used mainly a graphical user interface that mixes MATLAB, Simulink and Xilinx System Generator, XSG. The results obtained showed that the achievement of an optimized real-time edge detection system using hardware co-

simulation with higher performance size, speed and image quality in terms of peak signal-to-noise ratio (PSNR) compared with the Simulink edge detection filters (Prewitt, Roberts, Sobel).

Apart from that, many other studies for example from Cerezuela-Mora, Calvo-Gallego, and Sánchez-Solano presented a work on hardware/software co-design approach of video processing applications on FPGA platform. It demonstrated that many different video processing tasks can be programmed using software or accelerated by hardware filters for specific functions implemented in reconfigurable logic device [15].

2.2.3 Face detection and tracking

From the face detection and tracking algorithm aspects, face detection is often the first and most crucial step because it is essential for an efficient face recognition system in order to distinguish the face region and the non-face region of analysed image frames. 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 [7]. According to Cha and Zhengyou [16], 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: Common types of method in face detection approaches [16]

Types of method	**Explanations**	**Examples**
Appearance-based method	Use face modes to perform the detection	Eigenface
Feature invariant approach	Use facial features that are robust to pose and lightning variations or rotation	Skin colours, edges, and shapes

Knowledge-based method	Use human-coded rules to determine a face based on human knowledge	Two symmetric eyes, middle nose, mouth underneath the nose
Template matching method	Use pre-stored face templates to judge the face image	Correlation between test images and pre-selected facial templates

Chan presented a face detection system in in-door environment with non-uniform background such as the video captured in computer lab with many computers, tables, windows, and so on, which is based on feature-based chrominance colour information [17]. The input colour face will combine a few algorithms to increase accuracy rate which are the eye candidate estimation, the skin colour segmentation and face boundary estimation, in order to find the eye location of the person of interest. The skin colour segmentation used in this paper is YCbCr colour space where Y represents the luminance component whereas Cb and Cr are the blue chrominance component and red chrominance component respectively. Also, the range for the Cb and Cr is 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 to estimate the eye position of the person of interest.

Following are the studies focusing hardware implementation of face detection. Melanie presented a method that utilises the Reversible Component Transformation (RCT) colour space and outlines its transition from a software- to hardware-based implementation [18]. The hardware performance and efficiency of the RCT algorithm is examined using the Xilinx Virtex-II. 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 13 language to ease the transition to HandelC (a hardware description language used to configure the FPGA).

Vasily 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, colour-based segmentation, thresholding) producing a signal when a human-skin colour segment is detected in the image frame [19]. Experiments show that the design allows real-time user monitoring (30fps) with 82% detection accuracy while consuming as much as 35 times less power than analogous software.

In summary, from all the studies reviewed, here are some of the key findings that were found to be the interest and motivation but also useful elements in development of this project:

- High data throughput rate using FPGA in video processing
- Software level design for modelling and identifying video processing algorithms
- Hardware level design to achieve and implement real-time video processing speed (high frame rate)
- Face detection and tracking using colour-based skin segmentation (relatively simple and suitable for hardware implementation)

2.3 Background Studies

This section will discuss the basic principles and theoretical knowledge regarding digital video processing. It will help understanding various basic image/video processing techniques and algorithms that are related to this project.

2.3.1 Digital Image/Video

A digital image is commonly defined as a two-dimensional function, *f (x, y)*, where the x and y are spatial coordinates, and the amplitude of function, *f (x, y)* at any location of an image is called the intensity of the image at that point. The continuous image, *f (x, y)* can be converted into digital image *f [x, y]* by sampling the two-dimensional function in both coordinates and amplitude. Quantization function will be applied in sampling where digitizing the amplitude values (coordinate values). A sampled image of *f (x, y)* will result in digital image that has M rows and N columns where the coordinates *(x, y)* become discrete quantities. A digital image with M x N can be represented in a matrix form as the following:

$$f = \begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,N) \\ f(2,1) & f(2,2) & \cdots & f(2,N) \\ \vdots & \vdots & & \vdots \\ f(M,1) & f(M,2) & \cdots & f(M,N) \end{bmatrix}$$

Colour image includes colour information for each pixel. It could be known as a combination of individual images of different colour channels. The mainly used colour system in computer displays is RGB (Red, Green, Blue) space. Other colour image representation systems are HSI (Hue, Saturation, Intensity) and YCbCr or YUV.

Greyscale image refers to monochrome image. The only colour of each pixel is shades of grey. In fact, a grey colour is one in which the red, green and blue components all have equal intensity in RGB space. Hence, it is only necessary to specify a single

intensity value for each pixel, as opposed to represent each pixel with three intensities in full colour images.

For each colour channel of RGB image and greyscale image pixel, the intensity is within a given range between a minimum and maximum value. Often, every pixel intensity is stored using an 8-bit integer giving 256 possible different grades from 0 and 255. The black is 0 and the white is 255, respectively.

Binary image, or black and white image, is a kind of digital image that has only two possible intensity values for every pixel, 1 as white and 0 for black. The object is labelled with foreground colour while the rest of the image is with the background colour.

Video stream is a series of successive images. Every video frame consists of active pixels and blanking as in Figure 2.1. At the end of each line, there is a portion of waveform called horizontal blanking interval. The horizontal sync signal ’hsync’ indicates start of the next line. Starting from the top, all the active lines on the display area are scanned in this way. Once the entire active video frame is scanned, there is another portion of waveform called vertical blanking interval. The vertical sync signal ’vsync’ indicates start of the new video frame. The time slot of blanking could be used to process video stream.

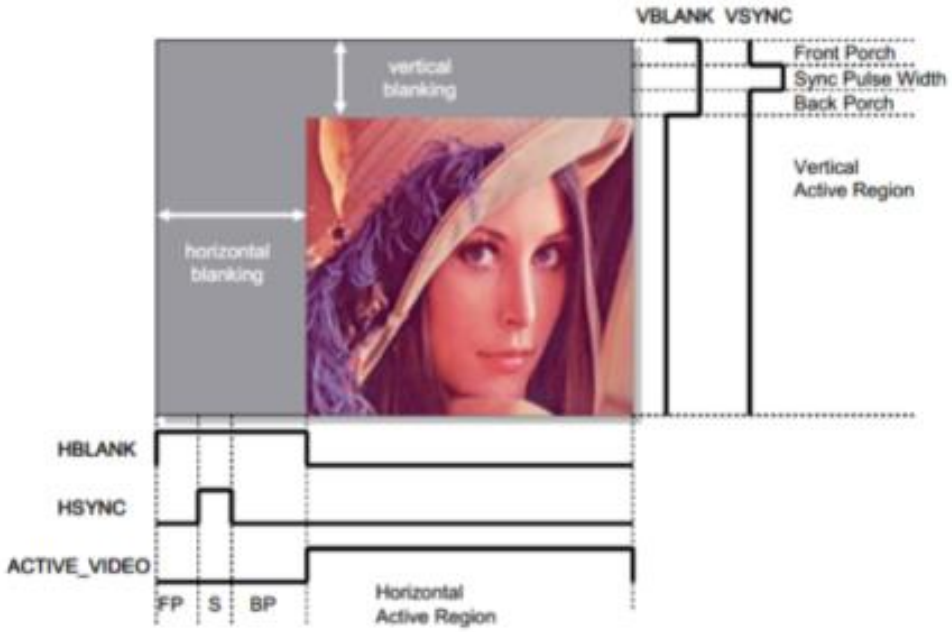


Figure 2.1: Video Signal Data Stream Timing [20]

Video data is divided into discrete images called frames. A frame is a bitmap image, transferred over the transport interface from top to bottom line by line, with each image line being transmitted from left to right. Therefore, the transmission of a frame starts with top left pixel and ends with bottom right pixel. The rate at which the video frames are transferred is called a frame rate [21].

2.3.2 Skin Colour Segmentation

One of the most significant features of human face's surface is skin colour, for the colour image, skin colour is the relatively concentrated; a stable region in the image. It is better to distinguish a human face from background regions of the skin colour. It shows that different race, age, sex with different human facial skin colour seemly, but the difference mainly concentrates in brightness, if in a colour space which removes brightness, the different face skin colour distribution has clustered. Based on this principle, it's feasible to segment an image of the skin colour [22].

2.3.3 Colour Space

Skin colour segmentation can be used based on different colour spaces. The main colour spaces include RGB, CMY/CMYK, YCbCr, HIS/HSV, YIQ, YUV and so on. YCbCr colour space has a composition which is quite similar to the process of human visual perception mechanism that can separate brightness and chroma very well. Besides, the colour space is discrete, which is easy to realize clustering algorithm and other merits [23]. It is easier compared to others. This project uses YCbCr colour space to input skin colour of image environment. In the beginning, RGB image is converted to a luminance colour space; it is often called YCbCr colour space. It is derived from the YUV colour space. Y stands for brightness. Cb and Cr are obtained by changing U and V. Cb stands for red chrominance, and Cr stands for blue chrominance. Cb and Cr are often called colour. It was found that Y has little effect on the distribution of the sample in the YCbCr colour space. But sample data are concentrated in the area of Cb-Cr.

YCbCr colour space has the following advantages:

- Its principle is similar to the process of human visual perception.
- Space format of YCbCr colour space is widely used in the television display area. It is also used in video compression coding such as MPEG, JPEG.
- Its space format separates brightness component from the colour components.
- Its space format's calculation process and representation of spatial coordinates are easier than others.
- Formula is as follows:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

2.4 Summary

In summary, the motivation of this project is expanded from the related work from previous researchers mentioned earlier. This project will be carried out with software design as the modelling of face detection and tracking algorithms, and then translation to HDL code so that finally can be implemented into FPGA. The video processing functional results are evaluated in real-time.

CHAPTER 3

METHODOLOGY

3.1 Introduction

There are three main processes involved in this project. As shown in Figure 3.1, the processes consist of; 1) identifying an appropriate algorithm, testing and simulation of the identified algorithm with the MATLAB/Simulink software; 2) the designing and translating of the identified software algorithms into HDL; 3) implementation of the video processing system in real time on hardware platform which is an Alterra DE2 -115 FPGA board.

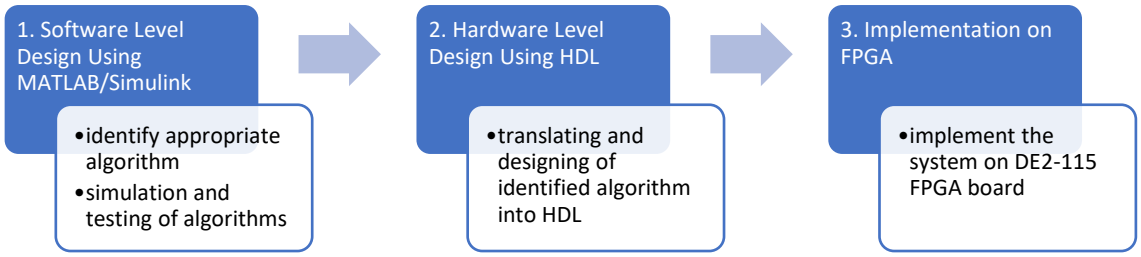


Figure 3.1: Overall project development process steps

3.2 Project Development Tools

There are two main tools used to develop the face detection system for this project. These include MATLAB R2017b and Quartus Prime 17.0. The hardware board used in this project is the Altera DE2-115 FPGA platform with an attached Capacitive Multi-Touch LCD with Camera Module (MTLC). MATLAB is a programming environment for software algorithm development, data analysis, visualization, and numerical

computation. Altera Quartus Prime 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. The hardware description language used in this project is Verilog coding.

3.2.1 FPGA (DE2-115)

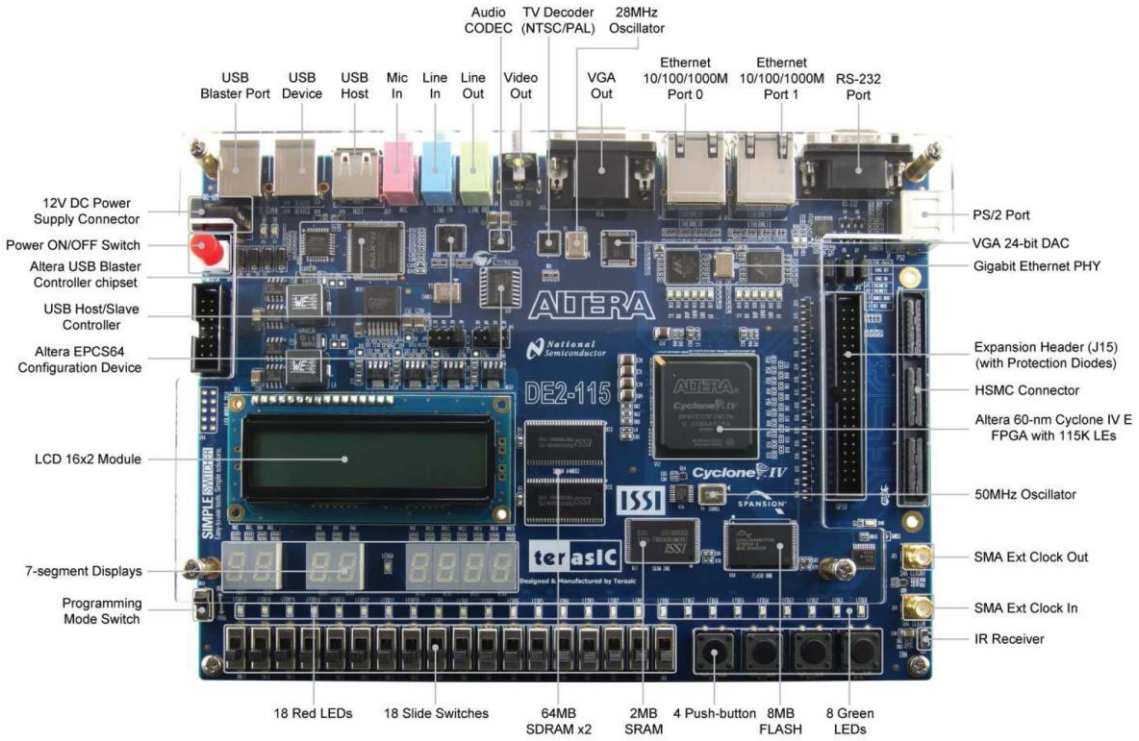


Figure 3.2: Overview of Altera DE2-115 FPGA Development and Education Board [24]

What exactly is a FPGA? Field Programmable Gate Arrays (FPGAs) are semiconductor devices that are based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects. FPGAs can be reprogrammed to desired application or functionality requirements after manufacturing. This feature distinguishes FPGAs from Application Specific Integrated Circuits (ASICs), which are custom manufactured for specific design tasks.

The device chosen to be used for this project is Altera DE2-115 FPGA Development and Education Board. Figure 3.2 shows the interface specifications supported by the device. The DE2 series has consistently been used for general purpose and educational development boards by distinguishing itself with an abundance of interfaces to accommodate various application needs. The DE2-115 features a Cyclone IV series device. The DE2-115 offers an optimal balance of low cost, low power and a rich supply of logic, memory and DSP capabilities.

Figure 3.3: Capacitive Multi-Touch LCD with Camera Module (MTLC) [25]

Figure 3.3 shows the Terasic Capacitive Multi-Touch LCD with Camera Module (MTLC) which is equipped with a 7” LCD screen, and a 5-Megapixel digital image sensor module, G-sensor and light sensor. All of these sensors connect to the FPGA device via the High Speed Mezzanine Card (HSMC) connector and they can be controlled and used directly by the FPGA device as shown in Figure 3.4. The camera will be used for capturing live video input for the hardware implementation.

Figure 3.4: MTLC connected to FPGA Development Boards DE2-115via HSMC [25]

3.2.2 Matlab / Simulink

Matlab/Simulink is a high-level language for scientific and technical computing introduced by Mathworks, Inc. Matlab/Simulink takes matrix as basic data element and makes tremendous matrix operation optimization. Therefore, Matlab is perfect for video/image processing since video/image is naturally matrix.

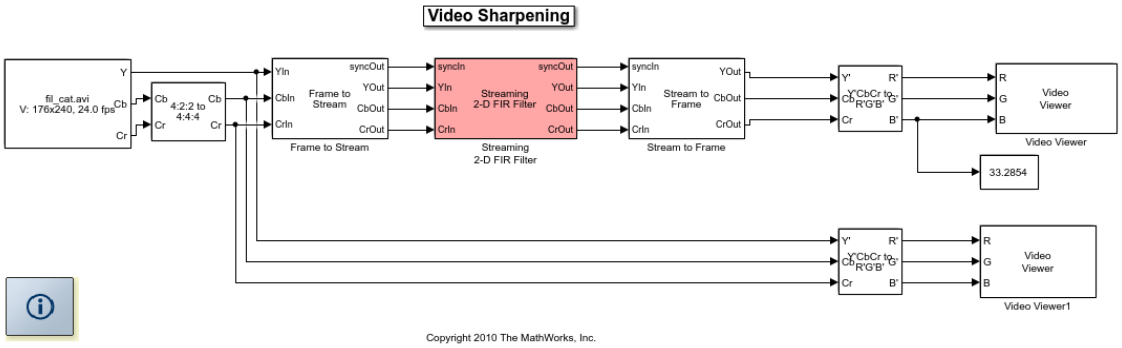


Figure 3.5: Example of Simulink Model - Video Sharpening [26]

The process shown in Figure 3.5 demonstrates a simple system that sharpens an RGB video input using Simulink tools. Simulink is a block diagram environment for multi-domain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modelling and simulating dynamic systems. It is integrated with MATLAB, enabling

designers to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis. The latest version of the software will be used for this project which is Matlab 9.3 version with release name of R2017b.

3.3.3 Quartus Prime

Altera Quartus Prime is programmable logic device design software produced by Intel Altera. Quartus Prime enables analysis and synthesis of HDL designs, which enables the developer to compile their designs, perform timing analysis, examine register-transfer level (RTL) diagrams, simulate design to different stimuli, and configure the target device with the programmer. Quartus includes an implementation of VHDL and Verilog for hardware description, visual editing of logic circuits, and vector waveform simulation. The latest version of the software is used for this project which is Quartus Prime v17.0 version released in May 2017.

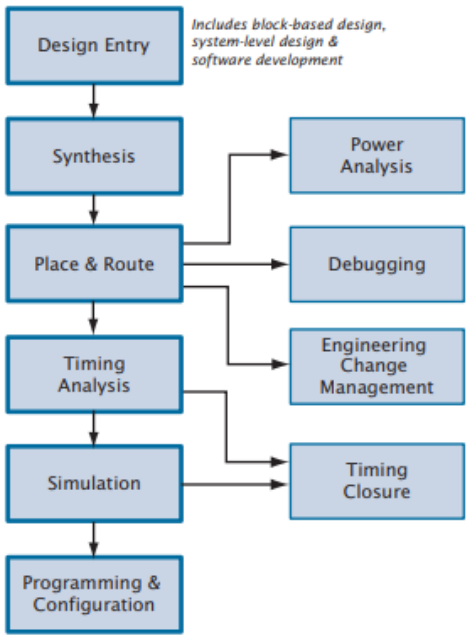


Figure 3.6 General FPGA design flow using Quartus Prime [27]

Figure 3.6 shows the general FPGA design flow using Quartus Prime software. There are different techniques for design entry namely schematic based, Hardware