Department of Electrical and Computer Engineering

Part IV Research Projects

Interim Report

[Multi-processor System on Chip for Real-Time Face Detection]

Prepared by Jack Wong

Project 109

Project partner Lin Yang

Supervised by Dr Morteza Biglari-Abhari

Co-supervised by Dr Nitish Patel

Declaration of Originality

This report is my own unaided work and was not copied from anyone or anywhere nor written in collaboration with any other person.

Jack Wong

20th May 2017

Multi-Processor System on Chip for Real-time Face Detection

Jack Wong
Department of Electrical and Computer Engineering, University of Auckland, New Zealand
UPI: lwon229
AUID: 5863577

***Abstract*—The aim of this project is to apply hardware/software co-design technique to design a real-time face detection on an embedded system. In this paper, different low computational cost face detection algorithms would be discussed and show the advantage of using FPGA board in terms of efficiency. Based on these algorithms, we propose steps of implementing the face detection system. In this paper, we have included a plan for each stages of development.**

I. Introduction

Face recognition technology is an ongoing research topic since last decade. It has been used in many applications such as biometric analysis and security system. Face detection is the first step of face recognition system since the user face must be located first before it is recognised. There are several methods used for face detection algorithm such as part-based method, feature based method and fractal based method[1] but these methods are not good for implementation on hardware because of huge complexity [2]. We would be focusing on low computational face detection algorithm.

This research paper is presented in the following sections. Section II explains past researches that has been done on face detection algorithm. Section III explains the hardware and software design approach and discuss how to leverage FPGA board to achieve real time face detection system. Section IV describes the plan for the project and how to distribute the workload. Conclusion are drawn in Section V.

II. Related Work

A. RCT Colour based segmentation [2]

RCT colour based segmentation is a face region detection algorithm with a very low computational requirement. Various researches shows human facial region have a unique colour distribution which is very different from the background objects. This algorithm determines the pixel in the image that are in the range of chrominance value that corresponds to the chrominance level of human skin. It uses multiple colour space such as HSV colour space (Hue, Saturation, Value), RGB colour space and YCbCr colour space. These colour space are used to filter the human skin colour range. This algorithm assumes the face region is always in the top part of the image so the lower part of the image is masked and ignored for faster computational time. According to the research paper [2], the research has proven the result is quite accurate and reliable. However, this algorithm has a slight drawback. Since this algorithm detect human face using human skin colour, it won't work if the feeding image is greyscale or the colour of the face is not natural (i.e. paint on the face).

B. Partially Occluded Face Detection (POFD) [1] [3]

POFD is a fusion of feature-based method and part-based method. This algorithm could detect partially occluded face. This algorithm is divided into 2 parts:

- Locate possible face component regions.
- Conclude partially occluded face for annotation.

1) Face skin detection: Face skin detection can be achieved by the RCT Colour based segmentation algorithm. It detects the face region based on the skin colour.

2) Face annotation: The underlying assumption of this part of the algorithm is that nose is present in the probe-image and few other face components might be missing. First it searches possible nose features in the image within the face bounding area. Then it creates arrays for all possible major face components base on the colour and the location of the face component. The next step is to create a threshold bound array of possible face component. Then the algorithm can annotate a partially occluded detected face based on the threshold bound array.

III. Design Approach

A. Hardware

To meet the time requirement of a real-time face detection system, we decided to implement our algorithm on a FPGA board. FPGA contains an array of programmable logic blocks which can be used for different configurations. In this project, we are using Intel DE1-SOC board to implement out real-time face detection system.

1) Reason of using FPGA: FPGA are capable of handling parallel processing which is good at handling multiple task simultaneously. Since our project goal is to implement a real-time face detection system, the processing time should

be as low as possible. Therefore, we are using hard core processor instead of soft core processor. Hard core processor can achieve much faster processing speeds since they are optimized and not limited by fabric speed whereas soft core processors are limited by the speed of the fabric. [4] [5]

To leverage the advantage of a FPGA board, multi-processors on FPGA would be used to implement the face detection system. Using the algorithm described in section II a, multiple filtering steps are taken to optimize the result. These filtering processes could be processed by multiple processors on FPGA simultaneously which would reduce the time of processing each frame.

2) Hardware design flow:

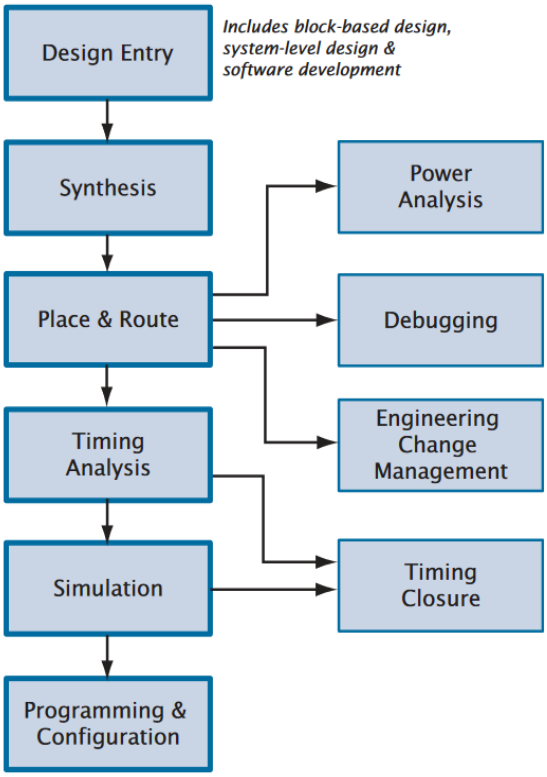

Figure 1: Quartus Design Flow [6]

Figure 1 shows the design process of using Quartus II to design the hardware component.

B. Software

We are using OpenCV (Open Source Computer Vision Library) to implement the algorithm discussed in Section II. OpenCV contains algorithms that could be used to detect colour in a picture or video. In this project, we are planning to use the image processing module in OpenCV to detect colour value in a frame.

To implement the face detection system, there are 4 steps involved.

***1) Step 1:* Video Input.**

Feed in the video input to be analysed. The video cannot be grayscale otherwise the algorithm would fail.

***2) Step 2:* HSV Segmentation.**

Write a C code that uses OpenCV image processing module to determine the HSV value of a frame. Regarding to the face detection, H and S value provide the information regarding the skin. According to the research paper [2], they have concluded the rules to detect the face:

240> H > 19 => Not Skin region

Using this rule, the algorithm can determine which pixel of the frame belongs to the facial region.

Figure 2: After HSV Segmentation [2]

***3) Step 3:* YCbCr Filtration.**

This step can be done concurrently with step 2. Using the OpenCV image processing module to determine the Cb and Cr value of each pixel. According to the research paper [2], they found out the rules to determine if the image portion belongs to the skin region.

128 > Cb> 102 => Skin
160 > Cr > 125 => Skin

If the constraint is not satisfied, the image portion does not belong to the skin region.

Figure 3: After YCbCr segmentation [2]

***4) Step 4:* RGB filtration.**

This filtration is used to remove the unnecessary pixels from the video frame. We can use the OpenCV image processing module to determine the RGB value of each pixel in the frame and filter out the pixels that does not belong to skin. According to the research paper [2], the correct threshold required in the RGB space are:

0.836G + 44 > B > 0.836G - 16 => Skin
0.780G + 42 > B > 0.790G - 67 => Skin

If the RGB value of the pixel are not within the threshold, then it does not belong to the facial region.

Figure 4: After RGB filtration [2]

The result after the filtering steps should be able to detect the user's face region. It is a very basic algorithm which can only detect the face region but it does not detect the location of major face components. It might be a problem if other body parts are shown in the video feed. The other possible limitation of this algorithm is the ability to detect face if the user is wearing glasses.

IV. Planning

In the first stage of our project, we consider different face detection algorithms and compare the difference between each algorithm in term of efficiency and complexity. To understand how to use the FPGA, we have learnt how to run a C program using the hard processor in the FPGA board. We have also considered OpenCV library and understand how we can use OpenCV to implement our system.

In stage 2, we will start the implementation process. I will do the algorithm implementation as described in Section IV b. We would apply agile development principles during this step and we would do multiple iterations of planning, analysis, design, coding, testing during stage 2. By the end of stage 2, we would finish a prototype of the face detection system.

In stage 3, we will do an evaluation of our prototype based on the requirements. We would analysis the result and determine if it meets the project requirements. Depending on the time, we might consider face recognition algorithm and possibly improving the prototype.

V. Conclusion

We have discussed the project requirements of a real-time face detection system. We have considered different low computational cost face detection algorithm and have discussed the advantage and limitation of these algorithms.

In Section III, we have discussed the capability of a FPGA board why using the hard processor is much faster than the soft processor. We have also discussed the hardware design flow using Quartus II software. For software component, we have discussed each steps of our proposed algorithm and show the limitation of the algorithm.

In Section IV, a general plan for the project has been drawn. We would apply agile development methodology to develop our face detection prototype. Depending on the progress, we might consider other face recognition algorithm and improve our prototype based on it.

Acknowledgment

We would like to thank our supervisor Dr Moterza Biglari-Abhari for his help in this project.

REFERENCES

[1] P. Shanmugavadivu and A. Kumar, "Rapid face detection and annotation with loosely face geometry," in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, Dec 2016, pp. 594– 597.

[2] D. N. Arya, S. K. L. V., R. Reddy, S. S, and S. K, "A face detection system implemented on fpga based on rct colour segmentation," in *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, Nov 2016, pp. 1–5.

[3] S. V. Dharan, M. Khalil-Hani, and N. Shaikh-Husin, "Hardware acceleration of a face detection system on fpga," in *2015 IEEE Student Conference on Research and Development (SCOReD)*, Dec 2015, pp. 283–288.

[4] A. Mohanty, N. Suda, M. Kim, S. Vrudhula, J. s. Seo, and Y. Cao, "Highperformance face detection with cpu-fpga acceleration," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2016, pp. 117–120.

[5] J. M. Saul, "Hardware/software codesign for fpga-based systems," in *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences. 1999. HICSS-32. Abstracts and CD-ROM of Full Papers*, vol. Track3, Jan 1999, pp. 10 pp.–.

[6] *Altera.com, "Quartus design flow". [Online]. Available: http://quartushelp.altera.com/15.0/master.htm*