Multi-Processor System on Chip Real-Time Face Detection

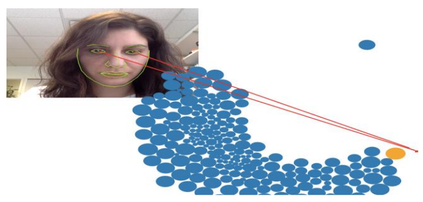
Project supervisor: Dr Morteza Biglari-Abhari | Presented by: Lin Yang and Jack Wong | The University of Auckland

Introduction

Face detection is a computer technology being used in lots of applications that identifies human faces. It is the fundamental application for developing numerous advanced computer vision applications and multi-media applications. The main objective of the face detection system is to reduce the processing time of detecting a human face in an image while maintaining accuracy. Therefore, Hardware-Software co-design approach has been taken to reduce processing time of an image and maximise the accuracy rate of face detection.

Applications

- Gaze tracker
- Facial Recognition
- Motion Capture



Aim

The aim of our work is to study a robust real-time face detecting method. The project was divided into two parts. The first one is to implement face image processing algorithm on the ARM platform based on Open Computer Vision Library (OpenCV). The second implementation was hardware acceleration, developed using VHDL on FPGA to enhance the performance of the face detection system.

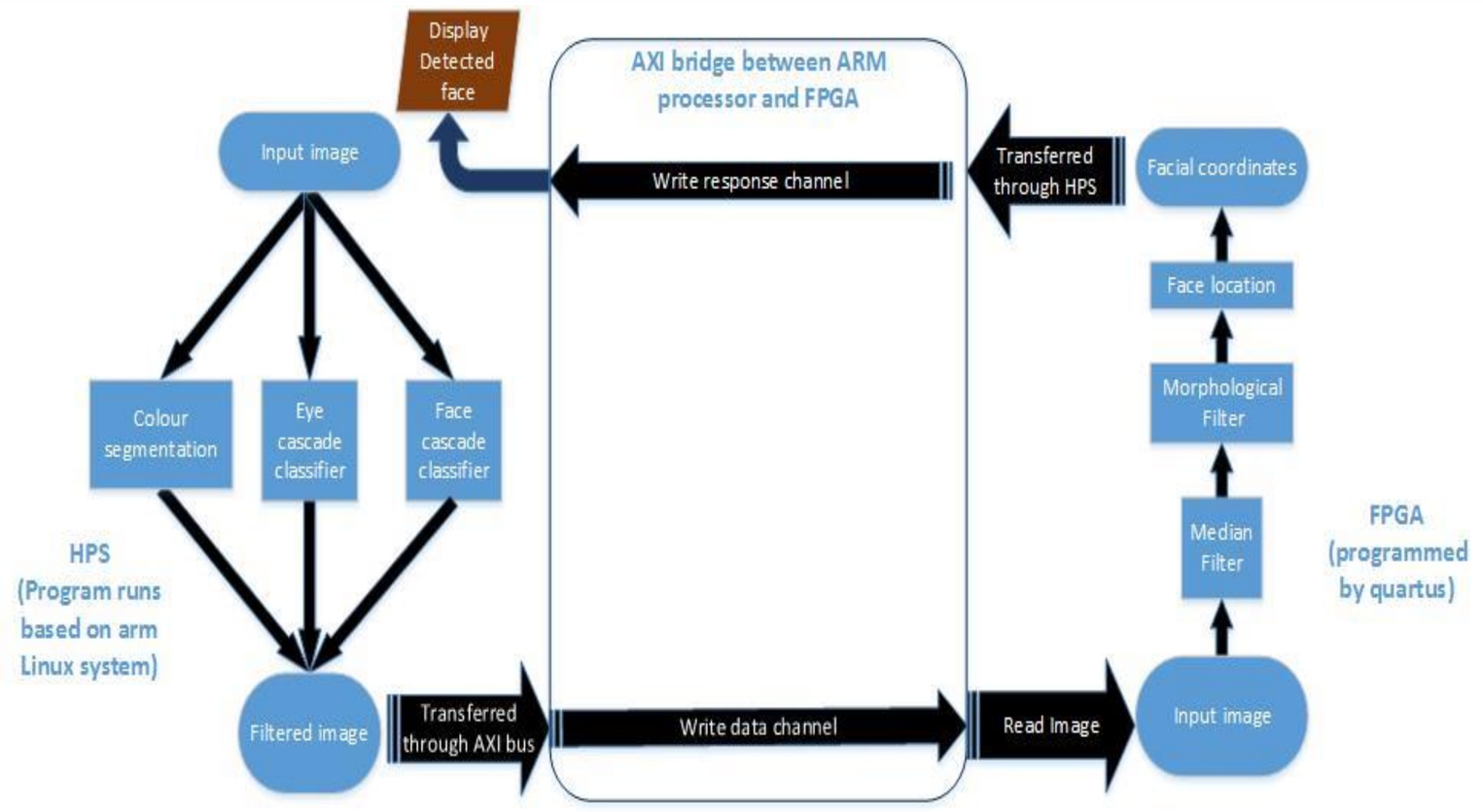
Theory

Cascade Classifier
Each classifier is trained from many positive and negative images. After the classifier is trained, it can be applied to a region of interest in an input image. It can easily extract the facial features from an image by using a search window through the image to check every location using the cascade classifier. The classifier outputs a “1” if the region is likely to show the features and “0” otherwise.

Skin Colour Segmentation
Human skin tone has a specific colour range in different colour spaces. Mostly the colour range of human skin tone is very different from the background colour. Based on this fact, we can determine the human skin region by finding the colour value of each pixels within an image. We have use HSV, YCrCb, RGB colour spaces to filter the input image in order to find the face region in an image.

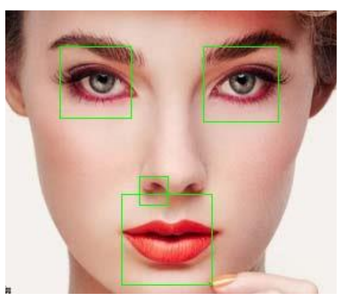
HSV colour filter:
240 > H > 19 => Not Skin Region
YCrCb colour filter:
128 > Cb > 102 => Skin
160 > Cr > 125 => Skin
RGB colour filter:
0.84G + 44 > B > 0.84G – 14 => Skin
0.78G + 42 > B > 0.79G – 67 => Skin

System overview



System Implementation

Step 1



Cascade Classifier

Use Haar’s cascade classifier to extract facial features such as eyes

Step 2

Skin Color Segmentation

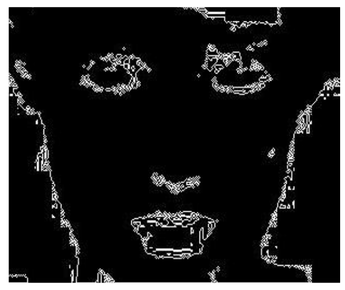
Use 3 color spaces (i.e. HSV, YCrCb, RGB) filters to remove pixels that are not in human skin tone.

Step 3

Median Filter

Use median filter to filter out salt & paper noise. This is a pre-processing step.

Step 4

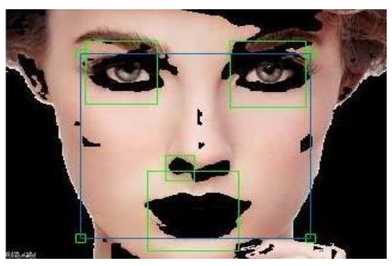


Morphological Filter

Use morphological filter to extract the face structure. The facial contour easily shows face features.

Results

Input Face Image



Output Image

	Skin Color Segmentation	Eye Cascade Classifier	Face Cascade Classifier	total	Detected face
320*240 image	0.028	0.032	0.048	0.108	Y
640*480 image	0.14	0.13	0.25	0.52	Y
1280*720 image	0.47	0.23	0.31	1.01	Y

- The complexity of our system is O(n^2). The higher the resolution of the input image, the longer the processing time.
- By combining Skin Color Segmentation and cascade classifier algorithm, the processing time increase by 200% but the accuracy rate improves by double.
- The bottleneck of the system is in the median filter hence it is implemented using hardware component.
- The ARM platform that was used in this project is a DE1-SoC development board with Cyclone V FPGA from Altera. Due to the limit of SDRAM on DE1-SOC.The test set of photos were resized to 320*240, allowing a fair comparison between Software and Hardware.
- Compared with simulation on ARM processor, FPGA processing time is tens of times faster.

Future work

- Further investigation should be focusing on the interaction between the Cortex A9 cores that we didn’t finish due to limited time.
- The accuracy of face features’ detection still needs to be improved.
- Future development can consider new applications, such as face recognition.

Conclusion

In this paper, we have presented an embedded implementation of real-time face detection algorithm, where the efficiency as well as accuracy could be improved. The main bottleneck in ARM platform was identified by using execution profiling and then moving some parts of the algorithm into hardware on FPGA. A SW-HW co-design approach was taken into consideration to reduce the processing time of a face detection algorithm while maintaining a high level of accuracy. This implementation meets the basic targeted-marketing requirements of real-time face detection.