Advances in Intelligent Systems Research, volume 130
6th International Conference on Mechatronics, Computer and Education Informationization (MCEI 2016)

# Implementation of Face Detection System Based on ZYNQ FPGA

Jing Feng[1, a], Busheng Zheng[1, b*] and Hao Xiao[1, c]
[1]Nanjing university of Aeronautics and Astronautics, Nanjing 210006, China
[a]fengjing214@126.com, [b]zhengbusheng@nuaa.edu.cn, [c]xiaohao@nuaa.edu.cn
*The corresponding author

**Keywords:** Face detection; Zynq; Adaboost; SoC

**Abstract.** Adaboost algorithm based on Haar-like feature and integral image is made up of cascaded classifier, and the computation is very large. This paper uses the hardware and software co-design for implementing the face detection system on Zynq-7000.And realizes the functions of video acquisition, processing, real-time detection and display, and designs the hardware acceleration module of the integral image. At the same time, it improves the detection speed of the system by using dual-core operation, and the detection speed of the system is obviously improved compared with the pure software solution. This design is verified on ZedBoard. It can work with high resolution stream for real-time face detection.

## Introduction

In recent years, the issue of information security has received more and more attention. The traditional identification technology can’t meet the requirements of modern Society. As one of the methods of biometric recognition, face detection is a key step in human face detection. But its speed has been relatively slow until P. Viola and Jones proposed the cascade algorithm AdaBoost in year 2001 [1]. AdaBoost is the turning point of face detection technology, which makes the detection speed and detection rate improve at the same time.

Most of the traditional face detection systems are based on PC platform or DSP processor, but these two schemes are relatively poor in portability and cost. T.Theocharides proposed hardware architecture based on array-based algorithms [2], but the architecture requires a lot of hardware resources. In 2008, Shi Yuehua and others proposed a software and hardware co-design, using the pipeline structure to further improve the detection rate [3].

This design is based on ZYNQ platform of ARM + FPGA architecture with hardware and software co-design to achieve the face detection system [4]. We design a parallel hardware architecture based on the proposed method to accelerate the calculation of integral image. And the image preprocessing module is implementation with hardware; In addition, using the multi-core structure of ARM to further improve the speed of face detection.

## AdaBoost Algorithm

This design uses the face detection cascade algorithm based on AdaBoost leaner using Haar-like features and integral image. The algorithm uses a training set to train a series of optimal weak classifiers. Then the strong classifier is obtained by combining the optimal weak classifier. Finally, the strong classifier is cascaded to obtain the face detector [5]. The Haar-like feature is a set of rectangular patterns. The result of a feature is the sum of the pixels under the white rectangle minus the sum of the pixels underthe black rectangle. Fig. 1 (a) shows several typical Haar-like features.

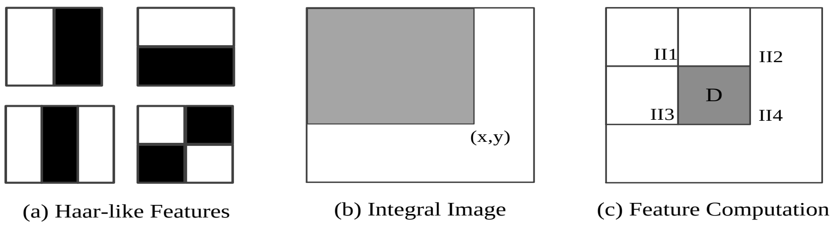
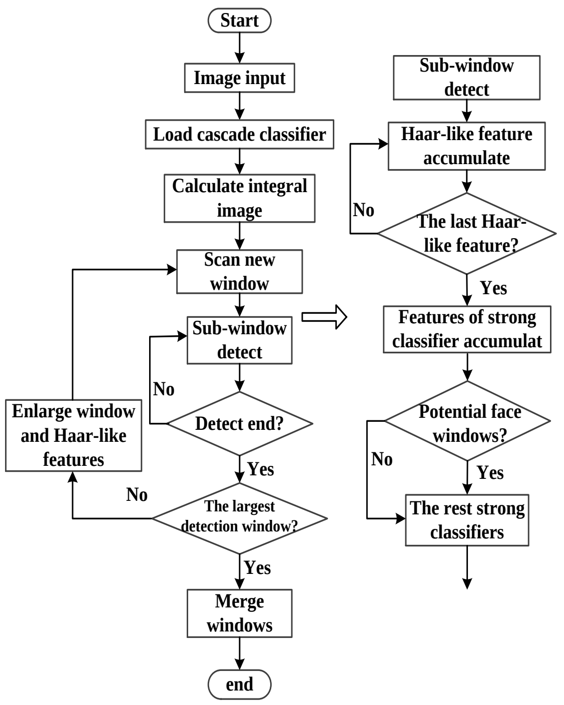


Figure 1. Haar-like feature and integral image

Integral image is Viola and Jones proposed in 2001 and applied to eigenvalue calculation [1].The integral image $II(x, y)$ at location $(x, y)$ is contains the sum of the pixels in the region determined by an origin and the $(x, y)$ point, as shown in Fig. 1 (b):

$$II(x, y) = \sum_{x'<x} \sum_{y'<y} p(x', y') \tag{1}$$

Where $p(x', y')$ is a pixel value in the original image, using the integral image can quickly calculate the features, that is $Sum = II1 + II4 - II2 + II3$, as shown in Fig. 1 (c), where $IIi$ is the integral image value of some point. We can get the Haar-like feature in the constant time with integral image, and improve the speed of face detection.


Start
Image input
Load cascade classifier
Calculate integral image
Scan new window
Sub-window detect
No
Detect end?
Yes
Enlarge window and Haar-like features
No
The largest detection window?
Yes
Merge windows
end
Sub-window detect
Haar-like feature accumulate
No
The last Haar-like feature?
Yes
Features of strong classifier accumulat
Potential face windows?
No
Yes
The rest strong classifiers

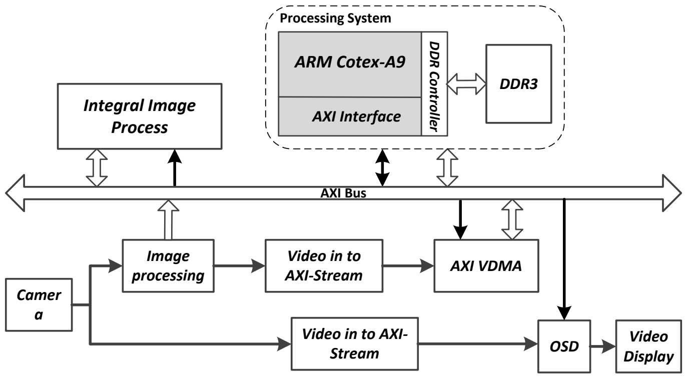


Figure 2. AdaBoost detection process

Figure 3. AdaBoost detection system

The core idea of AdaBoost cascade classifier is to classify the classifiers to complete the face detection where the order is the simple classifier first and then complexed [6]. The whole algorithm detection process is shown in Fig. 2. First, the trained cascade classifier is loaded, inputting the image, and the integral image of the original image is calculated. Next, each window of NxN (N is the detection window length or width) is scanned. For each window, each classifier is traversed in order, and the each feature in the classifier are calculated and compared with the feature thresholds $\theta$ to select the influencing factors. In each strong classifier, the sum of all the feature influencing factors represents the similarity of the window and the human face, and finally the similarity is compared with the strong classifier threshold $\beta$. If the similarity is greater than $\beta$, the window is regarded as a face window, and the next strong classifier test can be continued. The window through all strong classifiers is regarded the human face window. In order to adapt to different sizes of the face, after each test, enlarging the detection window by proportion and repeat detection process until the size of detection window closes to the picture size. Finally, merging the face windows, that is, the face windows with the same size and position are needed to merge into one face window and output [7].

**Hardware and Software Co-design of System Architecture**

In this paper, hardware and software co-design process are based on SoC system, build the entire system in a single chip, and divided into hardware platforms and software implementation. The experimental platform is the Zynq chip. In addition, we also need the CMOS camera and the VGA interface. The division scheme of the hardware and software is: the hardware part is in charge of the video signal acquisition, image preprocessing, integral image calculation and VGA display driver module and other functions; the main detection process is implemented in software.

**Hardware Platform.** The basis of the hardware part is video channel implementation based on ZYNQ. As shown in Fig. 3, the video processing and display platform.

Video data from the camera divides into two channels, one is the original video path, the other sends data to the video preprocessing module, and then sends data to the DDR through VDMA, after the processor detecting and then reading the results from the DDR with VDMA. At last, the two video channels is merged by V-OSD (Video On Screen Display) module and output to the VGA screen. The integral image is the basis of calculating Haar-like features, and the computational complexity is large. Image preprocessing includes gray scale and scale scaling of the original image. The hardware accelerator mainly deals with these two parts to improve the detection speed.

**Software Section.** AdaBoost algorithm detection process mainly complete in the software part (ARM Cortex-A9). Because of the development platform contains dual-core Cortex-A9 processor, using the dual-core structure, we can speed up the face detection process. The advantage of multi-core processing is that it can perform multiple tasks at the same time, so the face detection process can be divided into two parts in processors, each processor to complete half of the task detection, theoretically it can double the detection speed. The memory address of each kernel is allocated properly to prevent errors. At the same time, the dual-core communicate through the OCM to ensure that the synchronization of two processors during running the detection process [8].

The detection process is running in two processors, which are the master-core and sub-core. While detecting, the master core start the sub-core to run, when the master core complete its own task and also detect the task of sub-core are completed. It will merge the results from two cores. As a result, the rectangle information of the face position is output.

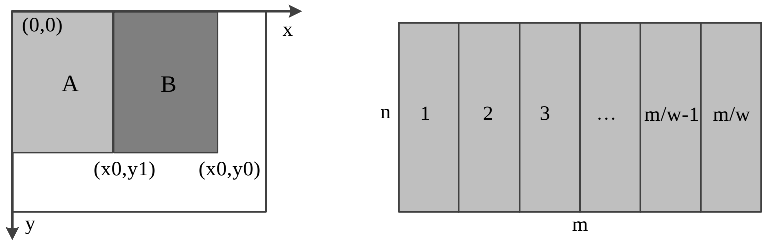
## Dedicated Hardware Acceleration

The hardware acceleration module is mainly used to calculate the integral image quickly and improve the calculation speed of Haar-like features. The definition of the integral image is shown in the Eq.1. It can also be obtained by the Eq.2 .The first term on the right part of Eq.2 is the value of the integral image at the point $(x_0, y_1)$, so we can obtain Eq.3. Hence, when the integral image of region A is known, if we want to compute the integral image values in region B, we can regard B as an independent region and compute its integral image; then, we add each row of integral values with the rightmost value at the same row in the integral image of A (Fig. 4(a)) [9]. Based on this architecture, we can compute the integral image values with blocks of original image in hardware design, and use pipelining and parallel structure to speed up the whole system.

$$II(x,y)=\sum_{x'<x_1}\sum_{y'<y_0} p(x',y')+\sum_{x_1\le x'<x_1}\sum_{y'<y_0} p(x',y') \qquad (2)$$

$$II(x,y)=II(x_1,y_0)+\sum_{x_1\le x'<x_1}\sum_{y'<y_0} p(x',y') \qquad (3)$$

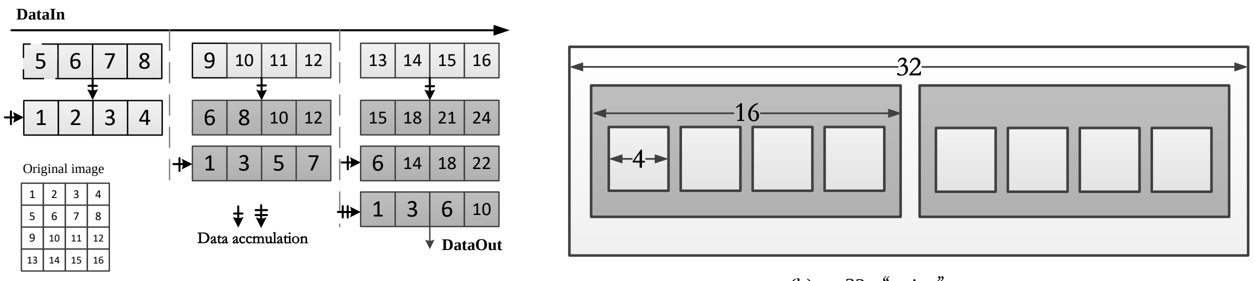
We use a 4x4 image to illustrate this pipeline calculation unit. As shown in Fig. 5 (a), we can complete the accumulation that interval $2^{n-1}-1$ at the cycle $n$, and get the accumulation of $2^n$ pixels, $n$ is calculation cycle numbers.



(a) Original image division (b) "strips"

Figure 4. "Strips" mechanism for integral image

The premise of the pipeline structure of the image is a whole line of pixel data input at the same time. For image of large size, we could not access a whole row simultaneously as the input or output data width of memory that stores image pixels is limited. To solve this problem, we refer to the analysis of Eq. (3) and Fig. 4 (a), divide the image into serval "strips" (Fig. 4 (b)). We first compute the integral image of strip 1. Then we use the property mentioned above to compute the integral image values in strip 2 as an independent image, and add each row of strip 2 with the rightmost value at the same row in the integral image of strip 1 to obtain the integral image value of region strip 2. Similarly, we can compute the integral image values of the rest strips. These strips are compute in a pipeline way. For an $m \times n$ image, if the width of each strip is $w$ (Usually m could be exactly divided by $w$), there would be $m/w$ strips, each strip requires $\log_2 w$ steps to compute the integral image [10].



(a) Pipeline process structure (b) w=32 "strips" structure

Figure 5. Calculation structure of integral image

In experiments, we find that when $w$=32 , the architecture achieves the highest power efficiency and area efficiency. Hence, the structure of "strips" are shown in Fig. 5.(b).

## Verification

In this paper, we implement the face detection system on ZedBoard development board based on ZYNQ-7000. Design uses software and hardware co-design, software needs to be modified because of the hardware accelerator, which not only deal with the detection work, but also responsible for hardware control, dual-core start and run, detection window enlarge and the result merging. After the system starts, the camera can capture the images in real time, and the result can be seen on the screen. As shown in Fig. 6, the original image captured by the camera and the face detection result.

The ZedBoard-Zynq-7000 resources consumed by the whole system are shown in Table 1. The system uses 22 classifiers, including 2135 Haar-like features. The images of $640 \times 480$ captured by the camera transit into DDR through the VDMA, the software detect and transit the result to the VGA display through the VDMA in the real-time. The SoC solution for face detection which uses an integral image accelerator as well as some pipeline and parallel technologies, face detection is greatly speed up.

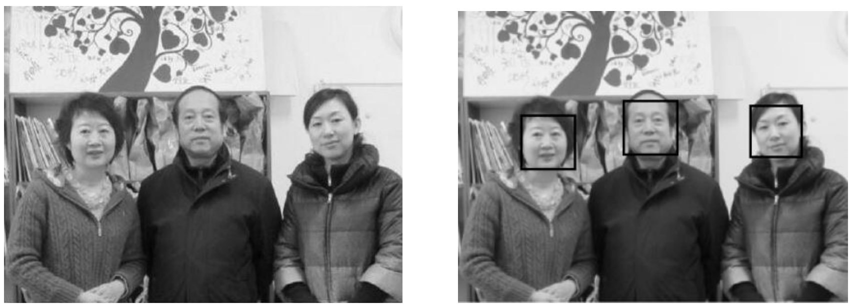


Figure 6. Original image and the detection result image

Table 1 Hardware Resources

| Resource | Used | Available | Utilization |
|---|---|---|---|
| FF | 8411 | 106400 | 7.91% |
| LUT | 6953 | 53200 | 13.07% |
| Memory LUT | 1073 | 17400 | 6.17% |
| I/O | 39 | 200 | 19.50% |
| BRAM | 119 | 140 | 85.00% |
| DSP48 | 4 | 220 | 1.82% |

## Conclusion

In this brief, we design and implement the face detection system using SW/HW co-design based on ZedBoard. On the Zedboard platform, the pipeline parallel structure of integral image calculation and image preprocessing implement on FPGA, software part use multi-core to speed up the detection.

## References

[1] P Viola, M Jones. Rapid Object Detection using a Boosted Cascade of Simple Features [C]. IEEE Conference on Computer Vision and Pattern Recognition, 2001, 1:511.

[2] T Theocharides, N Vijaykrishnan, M J Irwin. A Parallel Architecture for Hardware Face Detection [J]. ISVLSI, 2006, 2006:452-453.

[3] YH Shi, F Zhao, Z Zhang. AdaBoost algorithm of face detection system software and hardware design of SoC [J]. Journal of information technology, 2008, 09: 151-153+156. (In Chinese)

[4] Xilinx. ZYNQ-7000 All Programmable SoC Technical Reference Manual [EB/OL]. https://www.xilinx.com/.

[5] SL Wang, ZYNQ based real-time face detection technology research [D]. Nanjing University of science and technology, 2014. (In Chinese)

[6] N Zhao Face detection based on AdaBoost algorithm [D]. Beijing University,2005.(In Chinese)

[7] ZL Xia, The face recognition system based on ZYNQ research [D]. Dalian maritime university, 2015.(In Chinese)

[8] Adam P. Taylor, Both give full play to the Zynq SoC advantage [EB/OL].(2015-04-16) [2016-09-20] http: //xilinx.eetrend.com/article/8540.(In Chinese)

[9] P Ouyang, S Yin, Y Zhang, *et al*. A Fast Integral Image Computing Hardware Architecture with High Power and Area Efficiency [J]. Circuits & Systems II Express Briefs IEEE Transactions on, 2015, 62(1):75-79.

[10] P. Bellens, K. Palaniappan, *et al*. Parallel implementation of the integral histogram[C]. Advanced Concepts for Intelligent Vision Systems. Berlin, Germany: Springer-Verlag, 2011, pp. 586–598.