

A High Speed Reconfigurable Face Detection Architecture Based on AdaBoost Cascade Algorithm

Weina ZHOU^{†,††a)}, Lin DAI[†], Yao ZOU[†], Xiaoyang ZENG[†], *Nonmembers*, and Jun HAN^{†b)}, *Member*

SUMMARY Face detection has been an independent technology playing an important role in more and more fields, which makes it necessary and urgent to have its architecture reconfigurable to meet different demands on detection capabilities. This paper proposed a face detection architecture, which could be adjusted by the user according to the background, the sensor resolution, the detection accuracy and speed in different situations. This user adjustable mode makes the reconfiguration simple and efficient, and is especially suitable for portable mobile terminals whose working condition often changes frequently. In addition, this architecture could work as an accelerator to constitute a larger and more powerful system integrated with other functional modules. Experimental results show that the reconfiguration of the architecture is very reasonable in face detection and synthesized report also indicates its advantage on little consumption of area and power.
key words: face detection, reconfigurable, user adjustable, AdaBoost

1. Introduction

Face detection has developed rapidly in recent years, and it is now not only the first and critical step of face recognition, but also becomes an independent branch in video communication [1], digital camera [2], monitoring and surveillance [3], human computer interfaces [4], intelligent robots [5] and other fields. With increasing application situations, face detection technology has been confronted with a new challenge: how to meet the diverse demands on detection accuracy and speed in different situations. Since a detection method efficiently applied to one application might not be a good choice for other applications.

A traditional method to solve this problem is to modify the algorithm according to the new requirements, and redesign the overall system, mostly realizing it by high frequency devices such as personal computer or DSP. But this method leads to big development effort and long time to response to the change in situations, which is inconvenient for mobile devices needing to configure its detection capability rapidly as well as with low cost.

The problem caused the attention of many researchers, and their works are summarized as follows. Paper [6] put forward a systolic array architecture which makes the architecture highly scalable for different kinds of target detection

situations and different image resolutions. However, its reconfiguration is only available in the design period. Once the architecture is determined, its performance can no longer be adjusted either. To make the adjustment of the face detection capability easier, some researchers utilized logic programmable hardware like FPGA (Field Programmable Gate Array) and other similar platforms to realize the reconfiguration capability. Papers [7], [8] are typical examples. In paper [7], the critical part of the face detection algorithm is realized by a configurable logical module whose circuit architecture can be reconfigured according to the requirements conveniently. But its configuration is based on a Xilinx ML310 develop board, which still consumes much hardware and power. The face detection architecture proposed in paper [8] was realized on a DMV (Digital Machine Vision) platform. Like the design described in [7], its hardware logic could be adjusted by software coding. However, it also has the disadvantage of a waste of hardware and power resources.

In this paper, we proposed an ASIC (Application Specific Integrated Circuit) architecture based on AdaBoost cascade algorithm to detect faces. It not only has the advantage of little consumption of hardware and power, but also can be easily reconfigured. Since the reconfiguration of the design is realized by rewriting five registers in the initial state of the operation by the user, the architecture is regarded as user adjustable.

This paper is organized as follows. After analyzing the AdaBoost cascade algorithm in detail in Sect. 2, Sect. 3 will point out four reconfigurable factors which have great influence on the detection capabilities. In Sect. 4, the reconfigurable face detection architecture and the reconfiguration procedure will be described in detail. And at last, the experimental results in Sect. 5 show that the architecture could easily be reconfigured by user according to different demands and has a relative less consumption of area and power.

2. AdaBoost-Based Face Detection Algorithm

The AdaBoost-based face detection algorithm, proposed by Viola and Jones in 2001, can detect faces in gray images with complex background. Compared with other algorithms like skin-based detection theory and statistical method, it is a breakthrough on speed as well as accuracy, and is one of the most effective detection algorithms until now. Since it has been proposed, it quickly turned into a main research direction in real time face detection, and most of algorithms

Manuscript received April 22, 2011.

Manuscript revised September 6, 2011.

[†]The authors are with State Key Laboratory of ASIC and System, Fudan University, Shanghai, 201203 China.

^{††}The author is with Shanghai Maritime University, Shanghai, 200135, China.

a) E-mail: wnzhou@shmtu.edu.cn

b) E-mail: junhan@fudan.edu.cn (correspondence author)

DOI: 10.1587/transinf.E95.D.383

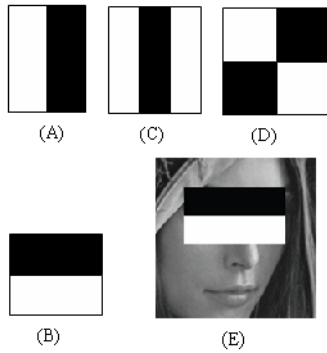


Fig. 1 Three kinds of haar-like features.

afterwards are all put up based on it.

This face detection algorithm mainly contains two parts: classifier training and pattern recognition. The training utilizes a simple modification of AdaBoost to select critical features from a large feature set, which can detect faces efficiently. These critical features, called weak classifiers, are all very simple with little computation cost, which is beneficial for the large amount of region to be processed in pattern recognition. And to achieve accurate classification, adequate trained weak features are adopted to linearly constitute a strong classifier, and then combined in a cascade structure to detect faces from the background.

The most popular weak classifiers are the “haar-like” features, which are fixed-size images containing a few black and white rectangles. They can detect edge or line characteristics in an image. Compared to pixels, the haar-like features have two merits. Firstly they usually could contain more information, and secondly, the feature-based system could operate much faster than pixel-based system. There are three kinds of haar-like features used in AdaBoost, which are described in Fig. 1. Figure 1 (A) and (B) are two-rectangle features. Figure 1 (C) shows a three-rectangle feature, and Fig. 1 (D) is a four-rectangle feature. The feature value is the difference between the sum of the pixels lying in the black rectangles and the white rectangles. The two-rectangle feature in Fig. 1 (E) shows the difference between the eye and the cheek region.

The strong classifiers made up of several haar-like features are constructed in a cascade structure. Each strong classifier is a stage in the detection process. Features will be allocated different values called weight according to the comparison results between their values and a predetermined threshold. And the sum of all the weights of a stage will constitute the stage sum to compare with another predetermined stage threshold. If the value is larger than the predetermined threshold, the region is regarded as a candidate region and will be processed by a sequence of classifiers afterwards. Or the region will no further be processed. In the architecture, the first few stages are always simple and constituted by relative small number of weak classifiers. And stages afterwards are generally slightly more complex than the last one. Processed by such a structure, the computa-

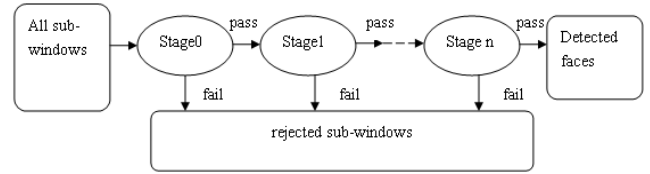


Fig. 2 Cascade structure of classifier stages.

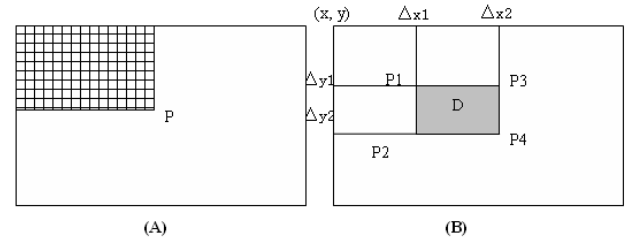


Fig. 3 (A) the integral value of p contains the sum of the pixels in the grid region from the origin. (B) the sum of D is determined by $P1$, $P2$, $P3$, $P4$. $D = P4 + P1 - P2 - P3$.

tion can be concentrated on object regions rather than the background regions, which greatly accelerates the detection procedure. The cascade structure is shown in Fig. 2.

When all the features in a search window are computed in a cascade of stages, we could then judge a search region as the face or non-face region. However, to detect faces larger than the search window, we should also enlarge the feature or scale down the image by a certain factor to repeat the cascade detection procedure. In addition, because it is usually unknown how large the face region is, the operation will repeat until the feature and the image are same in size. The two methods are always regarded as having the similar detection accuracy, and could be selected according to different applications.

To speed up the sum computation of pixels in a rectangle, another image representation called “integral image” (II) was proposed. With II, sum of pixels in a rectangle could be obtained by using one addition and two subtractions. II is a digital matrix, simply a transformation of the original input image. The value at each location of II represents the sum of the pixel values in the region between the origin and that location which is shown in Fig. 3 (A). So, as shown in Fig. 3 (B), the sum of pixel values in rectangle D could simply be gained by two additions and two subtractions of the four corner points of the rectangle.

To compensate the light variations caused by the environment, squared integral image (II^2) (each location holds the sum of the squared pixel values between the origin and that location) is also calculated for each input image. The variance (Var) and the standard deviation (σ) can be quickly computed by II^2 for lighting correction. The standard deviation will be multiplied with the original feature threshold (θ_0) given in the training set to obtain the compensated threshold (θ). By doing so, we can dynamically take care of any lighting variations occurring during the detection and improve the overall accuracy of the algorithm. In fact, this

operation needs to be done only once in every search window, and all subsequent features that search window evaluated can use this standard deviation value repeatedly.

$$Var = \sum_{i=0}^{M*N} p_i^2 / M * N - (\bar{E})^2 \quad (1)$$

$$\bar{E} = \sum_{i=0}^{M*N} p_i / M * N \quad (2)$$

$$\theta = \theta_0 * \sigma \quad (3)$$

Formula (1) shows the computation of variance. M, N are the width and height of the image respectively, P_i is the pixel value in position i , \bar{E} is the mean of all the pixel values in the search window. And as $Var = \sigma^2$, the threshold is adjusted by σ as shown in formula (3).

The flow chart of the face detection algorithm is shown in Fig. 4. Scaling image technique is adopted in this paper to detect faces larger than the search window. Because the needed integral and square integral data will become more disperse if we enlarge the feature, it will be more difficult to buffer them and accelerate the data reading. In addition, the integral and square integral data of big size images are also big, occupying much space to restore. Scaling the image with fixed-size search window will avoid above two disadvantages. In our architecture, the search window is set as 24×24 .

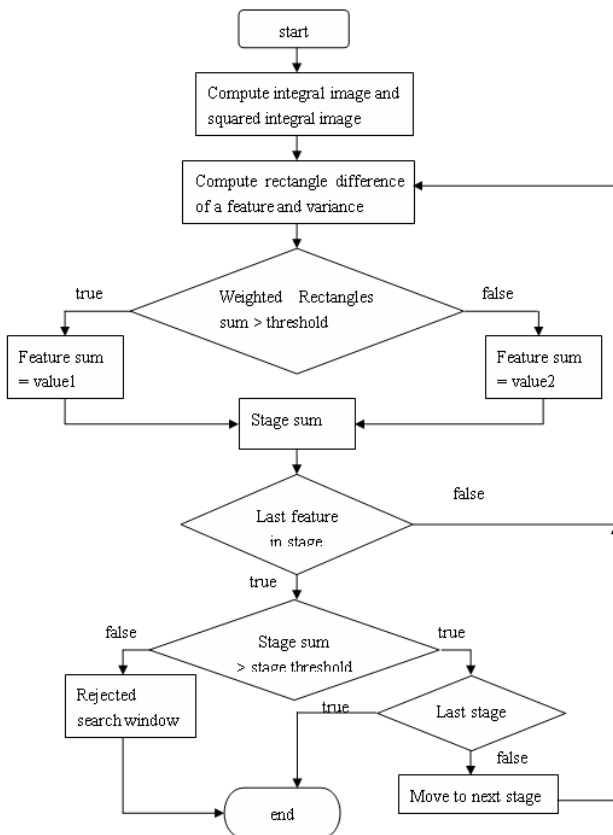


Fig. 4 The flow chart of face detection algorithm.

3. The Reconfiguration of Face Detection

After analyzing the algorithm in detail, we can find out that four critical factors of the AdaBoost algorithm can affect the detection capability, including detection accuracy, speed and size of the input images etc.. In this paragraph, we will describe the four factors in detail.

3.1 The Stage Number in Classification

As explained in Sect. 2, more classifiers are needed in latter stages in the cascade structure to locate the face position accurately. But windows they should detect even become less. In other words, as the face detection goes on, the classification efficiency declines gradually, and more classifiers are needed to exclude much less non-face regions in latter stages.

The number of the classifiers in a stage and the number of the sub-windows needed to detect in the first scale of a 320×240 image are shown in Table 1. The number of classifiers of a stage is determined after training, and will not be changed in detection. The number of sub-windows left for further detection is obtained by the face detection algorithm mention before, which use a scan step of 2 in horizontal direction and 1 in vertical direction.

From the table, one can see that, the last several stages which are always constituted by hundreds of classifiers can exclude few non-face regions.

To keep enough classifiers is indispensable in high accuracy detection. However, when the users prefer to pay more attention to find potential target position, and don't care to accept some non-face region as face, like taking picture by a camera, omitting several latter stages is feasible for accelerating classification procedure according to the requirement. In fact, the method is already adopted by some researchers when the hardware resource is limited [9]. So, the stage number is a reconfigurable parameter to be adjusted.

Table 1 Number of haar classifier and search windows in each stage.

stage	Number of classifier	Number of sub-window	stage	Number of classifier	Number of sub-window
0	1	31565	10	37	80
1	2	7967	11	53	50
2	3	4058	12	57	29
3	9	2451	13	81	24
4	14	1250	14	107	13
5	23	713	15	88	10
6	19	434	16	88	6
7	28	258	17	301	0
8	42	176	18	579	0
9	64	102	total	1596	

3.2 The Step of Scale Operation

To scale image to a smaller size is necessary to detect faces of different sizes in an image. It is determined by two factors, the step and the number of scale operation. They will affect the detection accuracy and speed.

In Viola and Jones's detection algorithm, the step of scale operation is set as 1.25. Considering that the kind of face sizes appearing in one scene is usually few, we could properly enlarge the step to accelerate the detection. Although this may result in a little decline of detection accuracy, it is well worth the decrease of computation amount sometimes. To choose a suitable step in different situations, experiments on the relation between the step of the scale operation and the detection accuracy were carried out and analyzed in Sect. 5.

3.3 The Number of Scale Operation

Besides the step of scale operation, the number of scale operation is another factor of determining the biggest size of the face which could be detected. In theory, the detected image has to be scaled down until it is as small as the search window to detect faces as big as the input image. But when the image is big, the computation amount of scale will be very large. In fact, the size of the face usually is much smaller than the input image, and it is unnecessary to scale the image until it is smaller than the search window. If the number of scale operation could be adjusted by user according to specific situation, the computation amount will decrease greatly with no decline in detection accuracy. So the scale number is another adjustable parameter to improve the speed.

3.4 The Size of the Input Image

Reconfiguration of the image size is very necessary to apply the face detection in different situations. For the resolutions of sensors are always different in different application, input images of different sizes will be encountered in life. Using same sensor or making the images the same size on purpose will decrease the detection accuracy or the speed. For example, if we compress the input image to a smaller size, it is impossible to detect faces which are as big as the search window in the original image. However, to enlarge input image will carry with much more extra computation amount. So reconfiguration of the image size is very important.

4. The Reconfigurable Face Detection Architecture

As explained in Sect. 3, there are four critical factors that will greatly affect the system's capability. In this section, we will describe how to reconfigure the system by adjusting these factors and make it operate in different modes.

The overview of the reconfigurable architecture proposed for real-time face detection is shown in Fig. 5.

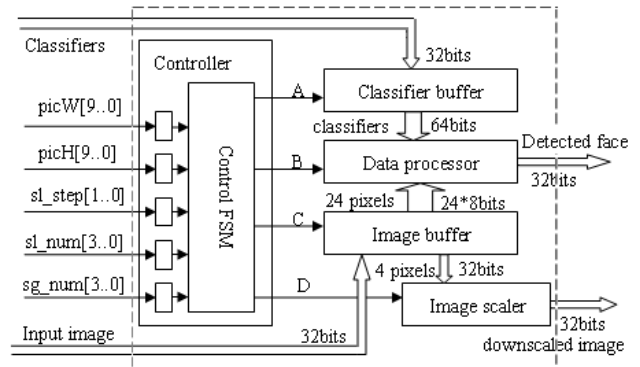


Fig. 5 The overview of the reconfigurable face detection architecture.

It consists of five modules: image buffer, image scaler, data processor, classifier buffer and controller. The classifier buffer and image buffer reserve the classifiers and image pixels respectively, and provide them to the data processor to implement face detection. Image scaler is responsible for the generation of downscaled images. Both of the detected faces and the downscaled images will be output to the external memory. The operation of the whole system is controlled by the signals "A", "B", "C", "D" sending from the controller, which not only coordinate the work between every modules, but also play a role of reconfiguring the system. And "picW", "picH", "sl_step", "sl_num", "sg_num" are just the input signals that determine the reconfiguration capability. They will be kept in the registers of the controller in the initial state of the operation, making the controller send different control signals to other four modules.

In following subsection, we will firstly introduce the architectures of classifier buffer, image buffer, data processor, image scaler in the order of system operation. And then we will introduce the controller as well as the reconfiguration capability.

4.1 The Classifier Buffer

The classifier buffer reads the trained haar features at the beginning of the detection, and reserves and provides them to the data processor. The number of haar features kept in it is determined by the stage number set by the user.

In this design, the information of every haar feature includes the feature type, position, size, threshold and weight. Since the search window is set as 24×24 , the offset of the position, the width and height of the feature can be represented by 5-bit data each. And 3 bits are needed to represent 8 different types of haar feature if reversing the black and white region will generate features of two different types. In addition, the weight and the threshold also need 16 and 24 bits data to represent respectively. So 63 bits in total are needed at least to keep all the information of a feature. In fact, as 32-bit wide bus is used in the design, each feature needs two cycles to be obtained. To promote reading efficiency, two 32-bit data of a feature are integrated before being stored into the buffer.

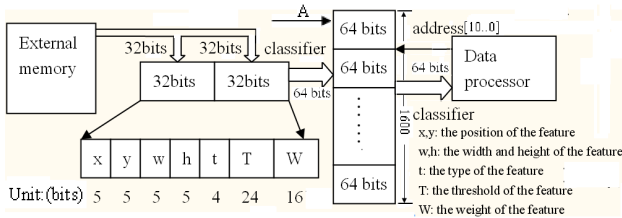


Fig. 6 The architecture of classifier buffer.

Considering that the design can reach a detection accuracy of 95% with 1596 features, the classifier buffer is made up of a single-port RAM with 64-bit wide and 1600 words. The architecture of the buffer and the bit allocation of feature are shown in Fig. 6. “A” is the control signal from controller.

4.2 The Image Buffer

The image buffer is another important accelerating unit to transfer image data between the data processor and the outside memory. It immediately begins after the finish of classifiers reading. And the data it will keep is also determined by “picW”, “picH” signal kept in the registers, which could be reconfigured by the user in the initial state of the system.

The biggest size of the image to be processed is 1024×1024 in this architecture. To save hardware cost, a 28 K byte dual-port memory is used to reserve a part of image pixels instead of all, which accounts for only 2.7% memory space of 1024×1024 images.

Considering that the dual-port memory could be read and written at the same time, it is adopted in this design to make data refreshing synchronize with the classification. In that case every pixels would be read only once from the outside RAM, which could reduce the consumption of power greatly. To further accelerate the speed, the memory is constituted by seven 32-bits wide and 1024 words memories. 1024 is the largest possible height of the image, and the actual height is determined by the “picH” signal. Thus 28 columns of the image could be reserved in the buffer. Six of the memories are used to provide the data processor with 24 pixels at the same time, and the left one works as an alternative to ensure the continuous refreshing of data. The architecture of the image buffer is shown in Fig. 7. “C” is the control signal transmitted by controller.

4.3 The Data Processor

The data processor is the critical module to perform the classification of face detection. It can generate Π and Π^2 , compute variance, extract features, do classification and output the face position and size at last. It begins as soon as the image buffer prepared the data of the first window, which is controlled by the “C” signal of the controller. And the information of reconfigurable stage number, which will affect the classification, is transferred by the signal “B”. The block diagram of the processor is shown in Fig. 8.

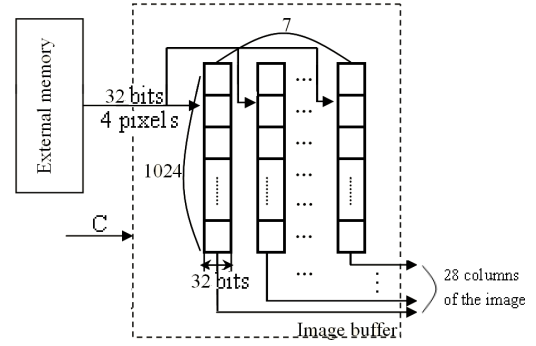


Fig. 7 The architecture of image buffer.

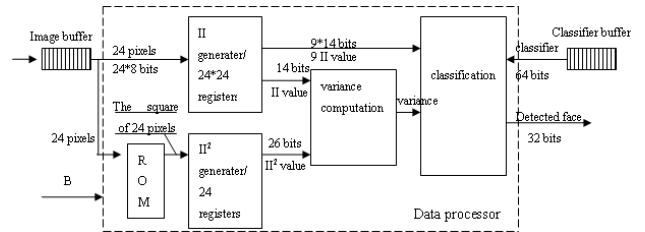


Fig. 8 The block diagram of the data processor.

24 pixels are read simultaneously from the image buffer, which just constitute a line of a search window. The Π and Π^2 will then be calculated line by line in parallel, and stored in registers to make the simultaneous access of any data possible. The variance is computed after the Π and Π^2 of a window are prepared, and its value will be used in classification. Two most important operations in this procedure are the classification and the Π and Π^2 calculation. The Π and Π^2 are calculated considering the hardware consumption and the speed of the whole architecture. And the classification is implemented by 4-stage pipeline architecture according to the characteristics of classification process.

The Π computation can be divided into two steps. The first step aims to get the Π values of each line. And in the second step, the values of a line will add to the last line of the Π window continuously, whose initial data is all zeros. The procedure is shown in Fig. 9. A 3×3 window is used to illustrate the procedure in a brief way.

From Fig. 9, one can find out that it is very convenient to refresh the Π with such a structure. As there are only differences of one line between the adjacent search windows, the refresh could be completed by subtracting the moving out data from the Π .

Π^2 are computed in parallel with the Π with the same method. A ROM is used as a look up table to get the square values of 24 pixels in a line instead of 24 multipliers, which greatly reduce the hardware and power consumption.

In classification, the first stage of cascade architecture is separated from the left ones since it is usually the most efficient step to exclude the background from further process and should be executed in each search window. In this design, there is only one two-rectangle feature in the first stage. Figure 10(A) is its flow chart. The left stages clas-

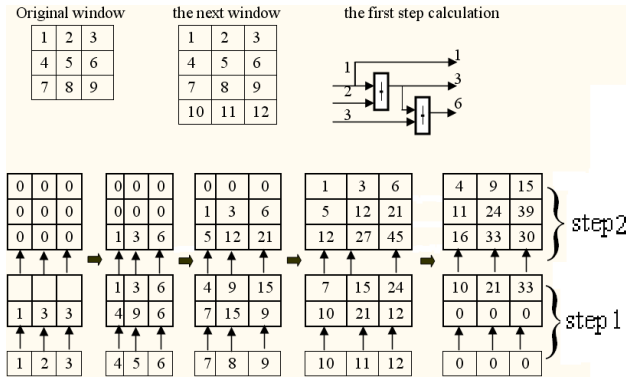


Fig. 9 The generation of integral image.

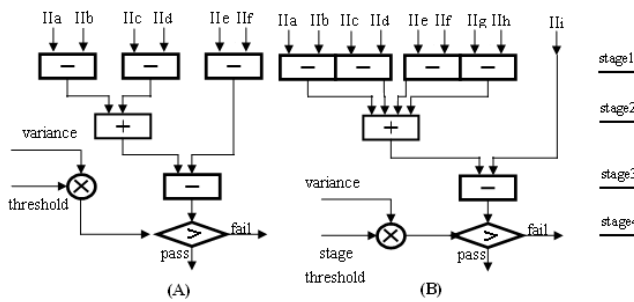


Fig. 10 (A) the flow chart of the first stage classification (B). The flow chart of the 2nd-19th stage classification.

sification shown in Fig. 10(B) only performs on left search windows which passed the first stage. And at the end of each stage, the search window will be judged to determine whether to continue for further process or end the operation to begin the classification of next search window. The candidate windows that pass all the stages will be considered as the face region. Considering of the maximum rectangle corners of haar-like feature, 9 input ports are set in classification and if a feature has only 6 or 8 input data, the left inputs will be set to 0.

As shown in Fig. 10, the left stage classification is implemented by a 4-stage pipeline architecture. It is even more efficient on processing capability than 3-classifier parallel architecture which is usually used by other researchers.

For comparison, the two architectures use the same AdaBoost algorithm just mentioned before, so the number of the classifiers in a stage and the windows left for further detection are exactly the same. The number of classifiers is fixed after training. Because the search windows should be evaluated after every stage, and 4 clock cycles are needed to implement a classifier, both stage1 and stage2 will consume 4 cycles with 3-classifier parallel architecture. And the 9 classifiers in stage 3 will consume 12 (3×4) cycles. However, with pipeline architecture, one more cycle is needed to implement the classification of the second classifier, so the two classifiers in stage1 need 5 ($4 + 1$) cycles, and cycle numbers the left stages needing is the same as the classifier number. For comparison, the total cycle of a stage is also

Table 2 The time comparison between parallel and pipeline architecture.

Stage	Number of classifiers	The windows left for further detection	Cycle numbers in a stage (3-classifier parallel)	Total cycle numbers in a stage(parallel)
1	2	7967	4	31868
2	3	4058	4	16232
3	9	2451	12	29412
			Total cycle number of stage1-3(parallel)	77512
Stage	Number of classifiers	The windows left for further detection	Cycle numbers in a stage (4-stage pipeline)	Total cycle numbers in a stage(pipeline)
1	2	7967	5	39835
2	3	4058	3	12174
3	9	2451	9	22095
			Total cycle number of stage1-3(pipeline)	74068

computed by multiplying the left window number with the cycle number of a stage, which implies the total processing time of all the windows in a stage.

As shown in Table 2, only after 3 stages, the total cycle number of the parallel architecture is bigger than that of the pipeline. Apparently, the difference will be larger when more stages are implemented. Moreover, compared to 3 times increase of the hardware consumption in 3-parallel architecture, the pipeline structure has no increase in circuit area. The experiment results also reflect that, parallel architecture is not always efficient in promoting speed. Due to the different effect of parallel execution between the early and subsequent stages in the cascade structure of face detection, speed does not become P times faster when P modules work simultaneously.

It is worth noting that, if more classifiers are performed in parallel, detection speed may be higher than the proposed pipeline architecture for sure, but that is not a good choice since they sacrifice much more hardware resources.

4.4 Image Scaler

The image scaler generates smaller scaled image to detect faces which are larger than the search window. The number and step of the scale operation can be configured by the controller according to the "sl_num" and "sl_step", which will be explained later.

The scale algorithm used here is nearest neighbor interpolation, in which a pixel value of the new image is set to the value of the nearest pixel in the original image. Although it is not as good as bilinear interpolation in some sense [10], it is adopted for its low computation load. And instead of obtaining all the downscaled images from the original input image, we generate the downscaled images with the image just generated before it. This technique reduces the difficulty in reading data from increasingly larger memory space, and accelerates the data access.

The architecture of image scaler is shown in Fig. 11. The scaler mainly consists of a single-port RAM of 32-

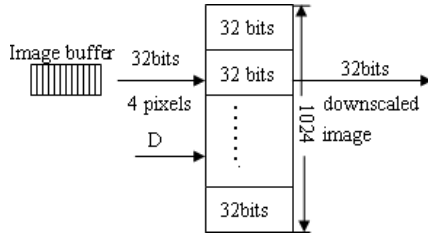


Fig. 11 The architecture of image scaler.

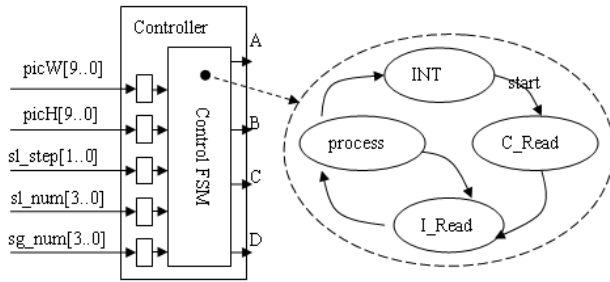


Fig. 12 The controller and its state machine.

bit wide 1024 words. Since reading data from the outside memory is time consuming, image scaler reads data from the image buffer, and refreshes them synchronized with the refreshing of image buffer. What's more, each cell of the memory reserves 4 pixels of the image to further promote the reading efficiency. And when the RAM is full, the data will be output to the outside memory at the interval of image reading.

4.5 The Reconfiguration and the Controller

The controller controls the whole circuit and realizes the reconfiguration capability. In this subsection, we will introduce the controller and explain how to fulfill the reconfiguration.

As shown in Fig. 12, there are four states of the controller. In the initial state (INT), the information of the five registers of the controller will be refreshed by 5 input signals: "picW", "picH", "sl_step", "sl_num", "sg_num". They are just the reconfiguration information which will control the system to run in different mode.

Signals "picW" and "picH" determine the size of the input image. As both of them are set 10bits in our design, the biggest size of the image the system can process is 1024×1024 . The value of "sl_step" determines the step of the scale operation. Since using a step bigger than 2 or smaller than 1.25 will result in a large accuracy decline or big computation amount in detection, 2-bit data is used to represent 4 kinds of step in the design. They are 1.25, 1.5, 1.75, 2. "sl_num" could configure the number of scale operation. 3 bits are allocated to represent 8 kinds of setting. If it is "0", the image will be scaled down until it is smaller than the search window. And the other values represent seven different numbers of the scale operation respectively. "sg_num" could configure the stage number of the classifi-

Table 3 Represent of sl_num and sg_num in different values.

sl_num	Number of scale operation	sg_num	Number of stages
0	unlimited	0	12
1	within 10	1	13
2	within 11	2	14
3	within 12	3	15
4	within 13	4	16
5	within 14	5	17
6	within 15	6	18
7	within 16	7	19

cation, which could vary from 1–19. In considering that, detection using classifiers fewer than 12 stages has an unacceptable FAR(False Accept Rate) for common use, they are not included in the adjustable range. Thus the "sg_num" also needs 3 bits to represent the stages varying from 12–19. The values of the "sl_num" and "sg_num" and their corresponding number of scale and stage are shown in Table 3.

So the system can process images of any sizes smaller than 1024×1024 , and be operated in 256 ($4 \times 8 \times 8$) modes to adapt to different applications, which is very flexible and useful.

After refreshing the registers in the initial state, the system will begin the C_Read state to read the classifiers from the classifier buffer. At this state, the classifier buffer will receive the control signal "A". When the reading of classifiers is finished, the I_Read state will begin, in which the image data will be read into the image buffer controlled by the signal "C". Once the data of the first search window is read by the image buffer, the system starts the "process" state. Signal "B", "C", "D" will then all be effective to make the data processor, image buffer, and image scaler work simultaneously. The process and I_Read state will alternatively appear in processing image with different scales.

5. Experiment and Analysis

To test the architecture's reconfiguration, experiments were made to test the effect of stage number, the step and number of scale operation on the detection accuracy and speed. And we also compared the design results with the results of software processing in 5.3 to show the effectiveness of our architecture. The face database we used in the test is selected from the CMU+IMT database including 87 pictures of different sizes.

5.1 Experiment 1: The Effect of Stage Number

We test the detection capability of the last several stages to choose suitable number of classifiers we really need in specific situations. We recorded the variation of the detection accuracy and speed with the stage numbers varying from 12 to 19. And their serial numbers are marked as 1–18. The detection accuracy can be represented by two indexes in the experiment: FAR (False Accepted Rate) and DR (Detection Rate). They are defined by formula (4) and formula (5) at

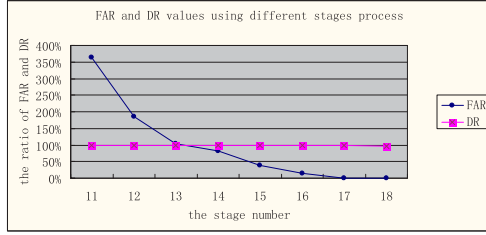


Fig. 13 FAR and DR values using different stages process.

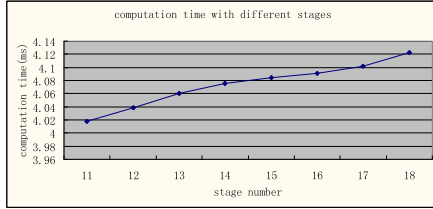


Fig. 14 Computation time with different stages.

Table 4 The accuracy and speed with different scale step.

The step of scaling	Detection accuracy	Process time(ms)
1.25	95%	6.16
1.5	94%	4.12
1.75	90%	3.40
2	84%	3.15

below.

$$FAR = \frac{\text{error detected faces}}{\text{total faces}} \times 100\% \quad (4)$$

$$DR = \frac{\text{correct detected faces}}{\text{total faces}} \times 100\% \quad (5)$$

Figure 13 shows the variation of the FAR and DR with the increase of stages. DR changes little in this procedure, for real faces are always included in the candidate regions, but the FAR decreases. However, the declining trend of FAR becomes smoother and smoother, which shows its decrease on exclusion ability. Figure 14 shows the increase trend of computation time with the same variation of stages.

From above two figures, we can find out that, because the FAR and computation time have different characteristics in changing, it is necessary to choose a suitable stage to get an acceptable DR and FAR with best computation time according to the requirements.

5.2 Experiment2: The Effects of Scaling on the Detection Accuracy and Speed

We recorded the average detection accuracy and computation time of the images with 4 different steps of scale operation in Table 4. And the images are all scaled down until they are as small as the search window.

With the bigger of scale step, the detection accuracy and the process time are both declining. Although it is contrary to select a step with best detection accuracy and least

Table 5 The speeds of hardware and software implementation.

Image size	Software	Hardware	Image size	Software	Hardware
160*120	270ms	2.88 ms	640*480	2424ms	40 ms
176*144	372ms	4.12 ms	800*600	4852ms	62.5 ms
320*240	952ms	12.50 ms	1024*768	5205ms	83.3 ms
352*288	1045ms	13.33 ms	1024*1024	14184ms	125 ms

Table 6 The synthesized result and comparison with other 3 processors.

	Presented work	Christos[6]	Hanai[11]	Chih-Rung chen[12]
Technology(nm)	65	65	90	90
area (mm ²)	1.2	88million register≈37	0.89	0.64
Detection Speed	80 fps (320*240)	133fps (320*240)	8fps (320*240)	390fps (160*120)
Clock frequency(MHz)	100	800	54	167
accuracy	95%	95%	81%	81.57%
Power(mw/fps)	1.7	2.45	0.47	0.36
reconfiguration	User adjustable	Developer adjustable	no	no

process time, choosing a relative suitable step of scaling in special situation is possible.

In addition, the number of the scale operation is largely dependent on the size of the biggest face in real situation. So we should find out the moment when all the face regions are smaller than the search window. If the size of the biggest face region can be estimated before the detection, it will not be difficult to determine the number of the scale operation then.

5.3 Experiment 3: Comparison of the Design Results with the Results of Software Processing

To compare with the software, we also implement the same face detection algorithm on DSP DM642, which run at a 600 MHz clock frequency. While the detection accuracy is almost the same, the speed of hardware implementation almost reaches 100 times as to the speed realized by software.

The speeds of processing images of 8 different sizes are shown in Table 5.

5.4 Hardware Evaluation

The architecture is synthesized using Synopsys design compiler targeting TSMC 65 nm CMOS LP library with 5-metal-layer to obtain relevant metrics such as area, operating frequency and power consumption. The results indicate that the proposed architecture occupies 1.2 mm². It can run at 100 MHz clock frequency with a 1.2 V supply, and the power consumption is 1.7 mW/fps. The results are shown in Table 6, and the synthesized results of other three representational architectures are also listed out for comparison.

From the above table, we can conclude that the architecture not only has an outstanding capability on reconfig-

uration, but also performs well on detection capability with little area and power consumption.

6. Conclusion

This paper proposed a reconfigurable architecture of face detection. Compared with other architectures, it can be reconfigured by user conveniently. By reconfiguring, the architecture will meet the different demands on detection performances according to special situations. What's more, the synthesis results demonstrate its high speed and small area, which is encouraging for further search on it.

Acknowledgements

This paper is supported by Shanghai Leading Academic Discipline Project (S30602), National Science Supporting Plan (2009BAG18B04), and Science & Technology Program of Shanghai Maritime University (20090131/20110028).

References

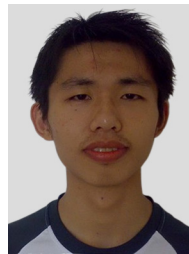
- [1] J.-Q. Wang, H.-D. Ma, and A.-L. Ming, "Fast head-shoulder detection on mobile phones," IEEE International Conf. on Consumer Electron., pp.205–206, Jan. 2011.
- [2] E. Hjelm, "Face detection: A survey," Computer Vision and Image Understanding, vol.83, pp.236–274, 2001.
- [3] Z. Guo, H. Liu, Q. Wang, and J. Yang, "A fast algorithm of face detection for driver monitoring," International Conf. on Intelligent Systems Design and Applications, vol.2, pp.267–271, Oct. 2006.
- [4] M. Yang and N. Ahuja, "Face detection and gesture recognition for human-computer interaction," International Series in Video Computing, vol.1, pp.2–3, 2001.
- [5] W. Yun, D.H. Kim, and H. Yoon, "Fast group verification system for intelligent robot service," IEEE Trans. Consum. Electron., vol.53, no.4, pp.1731–1735, Nov. 2007.
- [6] C. Kyrkou and T. Theodoridis, "A flexible parallel hardware architecture for AdaBoost-based real-time object detection," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.19, no.6, pp.1034–1047, June 2011.
- [7] V. Kianzad, S. Saha, J. Schlessman, G. Aggarwal, S.S. Bhattacharyya, W. Wolf, and R. Chellappa, "An architectural level design methodology for embedded face detection," International Conference on Hardware/Software Codesign and System Synthesis, Sept. 2005.
- [8] V. Mariatos, K.D. Adaos, and G.P. Alexiou, "Design and implementation of a reconfigurable, embedded real-time face detection system," RSP 2007. 18th IEEE/IFIP International Workshop on Rapid System Prototyping, 2007.
- [9] W. Yu, B. Xiong, and C. Chareonsak, "FPGA implementation of AdaBoost algorithm for detection of face biometrics," 2004 IEEE International Workshop on Biomedical Circuits and Systems, 2004.
- [10] M. Hiromoto, H. Sugano, and R. Miyamoto, "Partially parallel architecture for AdaBoost-based detection with haar-like features," IEEE Trans. Circuits Syst. Video Technol., vol.19, no.1, pp.41–52, 2009.
- [11] Y. Hanai, Y. Hori, J. Nishimura, and T. Kuroda, "A versatile recognition processor employing Haar-like feature and cascaded classifier," IEEE International Conf. Solid-State Circuits, vol.52, pp.148–149, Feb. 2009.
- [12] C.-R. Chen, W.-S. Wong, and C.-T. Chiu, "A 0.64 mm² real-time cascade face detection design based on reduced two-field extraction," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.99, pp.1–12, Sept. 2010.



Weina Zhou received the B.S. and M.S. degrees in electronic information engineering of Shanghai Maritime University, China, in 2004 and 2006, respectively. And since 2006, she has been a lecturer in Shanghai Maritime University. At present, she does doctoral research with the State Key Laboratory of ASIC & System, Microelectronics Department, Fudan University, Shanghai, China. Her research interests include face recognition algorithms and ASIC design.



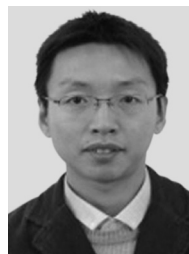
Lin Dai received the B.S. in applied mathematics and Ph.D. degrees in microelectronics from the Southeast University, China, in 2002 and 2009, respectively. Dr. Dai currently does a post-doctoral research with the State Key Laboratory of ASIC & System, Microelectronics Department, Fudan University, Shanghai, China. His research interests include multimedia and face recognition algorithms.



Yao Zou received the B.S. and M.S. degrees in school of optical-electrical and computer engineering from the University of Shanghai for Science and Technology (USST), Shanghai, China, in 2007 and 2010, respectively. He is currently working as a Research Assistant in State Key Lab of ASIC & System, Fudan University, Shanghai, China. His research interests include VLSI and face recognition.



Xiaoyang Zeng received the B.S. degree from Xiangtan University in 1996 and the Ph.D. degree from Changchun Institute of Optics and Fine Mechanics, Chinese Academy of Sciences in 2001. From 2001 to 2003, he worked as a post-doctor researcher at the State-Key Lab of ASIC and System, Fudan University, P.R. China. Then he joined the faculty of Department of Micro-electronics at Fudan University as an associate professor. His research interests include information security chip design, VLSI signal processing, and communication systems. Prof. Zeng is the Chair of Design-Contest of ASP-DAC 2004 and 2005, also the TPC member of several international conferences such as ASICON 2005 and A-SSCC 2006, etc.



Jun Han received the B.S. degree from Xidian University, Shaanxi, China, in 2000, and the Ph.D. in microelectronics from Fudan University, Shanghai, China, in 2006. He joined Fudan University as a faculty man in July 2006 and has been an assistant professor in the state key lab of ASIC & system. He is working on ASIC design of security chip, high performance security processor.