International Journal of Scientific & Engineering Research, Volume 4, Issue 10, October-2013 1244
ISSN 2229-5518

Real Time Face Detection Based on FPGA using Adaboost Algorithm

P. Vijaya, P. Sreenivasulu

Abstract— This paper presents an paper for face detection based system on adaboost and histogram equalization and it is implemented using haar features.We describe the image scaling, integral image generation, pipelined processing as well as classifier, and parallel processing multiple classifiers to accelerate the processing speed of the face detection system. The face detection system generates an integral image window to perform a Haar feature classification during one clock cycle. And then it performs classification operations in parallel using Haar classifiers to detect a face in the image sequence. The classifiers in the beginning of the cascade are simpler and consist of smaller numbers of features. However, as one proceeds in the cascade, the classifiers become more complex. A region is reported as detection only if it passes all the classifier stages in the cascade. If it is rejected at any stage, it is discarded and not processed further. If all stages are passed the face of a candidate is concluded to be recognized face. Its performance has been measured and compared with an equivalent software implementation.

Index terms- Adaboost algorithm, Haar features, Histogram equalization, Integral image.

◆

1 INTRODUCTION

Computer vision is one of the foremost fields which have experienced increasing number of applications in the recent years in various directional domains like biomedical imaging, surveillance systems, interactive systems like gesture, recognition, gaming etc. Detection of human faces is one of the key elements in the applications of computer vision in the above mentioned domains. Face detection is based on identifying and locating a human face in image regardless of size, position, and condition.

Numerous approaches have been proposed for face detection in images. Simple features such as color, motion, and texture are used for the face detection in early researches. However, these methods break down easily because of the complexity of the real world. Face detection proposed by Viola and Jones [1] is most popular among the face detection approaches based on statistic methods. This face detection is a variant of the AdaBoost algorithm [2] which achieves rapid and robust face detection. The proposed face detection framework based on the AdaBoost learning algorithm using Haar features with varying illumination is considered on of the most difficult tasks for face detection. Variation caused by illumination is highly non linear and makes task extremely complex. Well known contrast enhancement algorithm, histogram equalization is applied for compensating the illumination conditions.

Over past two decades, the problem of face detection has attracted substantial attention and witnessed an impressive growth in basic and applied research, product development and application. The purpose of this paper is to implement and thereby recreate the face detection algorithm presented by Viola-Jones with a refinement of histogram equalization technique. This algorithm should be capable of functioning in a unconstrained environment meaning that it should detect all visible faces in any conceivable image. In order to guarantee optimum performance of the developed algorithm the vast majority of images used for training, evaluation and testing are either found on the internet or taken from private collections.

Facial feature detection methods generally model two types of information. The first is local texture around a given feature, for example the pixel values in a small region around an eye. The second is the geometric configuration of a given set of facial features, e.g. eyes, nose, mouth etc.

- ***P. Vijaya,*** *PG Scholar, Department of Electronics and Communication Engineering, PBR Visvodaya Institute of Technology & Sciences, Kavali, Andhra Pradesh, India; vijayaece12@gmail.com.*
- ***P. Sreenivasulu****, Professor, Department of Electronics and Communication Engineering,PBR Visvodaya Institute of Technology.Vardhan.sreenivasulu@gmail.com.*

ISSN 2229-5518

2 COMPOSITION REQUIRED FOR FACE DETECTION

2.1 Integral Image:

For the incoming pixel where the coordinate is (x, y), the image line buffer controller performs operations such as in "(1)", where n is the image window row size, p(x, y) is the incoming pixel value, and L(x, y) represents each pixel in the image line buffer.

$$L(x,y-k) = L(x,y-(k-1)), \text{ where } 1 \le k \le n-2 \qquad (1)$$

$$L(x,y-k) = p(x,y), \text{ where } k = 0$$

The image window buffer stores pixel values moving from the image line buffer and its controller generates control signals for moving and storing the pixel values. Since pixels of an image window buffer are stored in registers, it is possible to access all pixels in the image window buffer simultaneously to generate the integral image window.

$$I(i-k,j) = I(i-(k-1),j), \text{ where } 1 \le k \le m-1 \qquad (2)$$

$$I(i,j-l) = L(x,y-(l-1)), \text{ where } 1 \le l \le n-1$$

$$I(i-k,j-l) = p(i,j) = p(x,y), \text{ where } k = l = 0,$$

$$\text{when } k+l = n-1,\ 1 \le k \le n-1,\ 0 \le l \le n-2,\ m = 2n,$$

$$I(i-k,j-l) = I(i-(k-1),j-l) + I(i-(k-1),j-(l+1))$$

For the incoming pixel with coordinate (x, y), the image window buffer controller performs operation as in "(2)" where n and m are the row and column size of the image window buffer, respectively. p(i, j) is the incoming pixel value in the image window buffer; p(x, y) is the incoming pixel value; I(i, j) represents each of the pixels in the image window buffer; and L(x, y) represents each of the pixels in the image line buffer.

$$II(s-u,t-v) = \qquad (3)$$
$$II(s-u,t-v) + I(i-k,j-l) - I(i-(2n-1),j-l)$$
$$\text{where } 0 \le u \le n-1, 0 \le v \le n-1, n-1 \le k \le 2n-2,$$
$$0 \le l \le n-1$$

Since pixels of an integral image window buffer are stored in registers, it is possible to access all integral pixels in the integral image window buffer simultaneously to perform the Haar feature classification. For incoming pixel with coordinate (i, j), the integral image window buffer controller performs operation as in "(3)"

"Fig.1" shows all of the actions in the proposed architecture to generate the integral image. For every image from the frame grabber module, the integral image window buffer is calculated to perform the feature classification using the integral image.

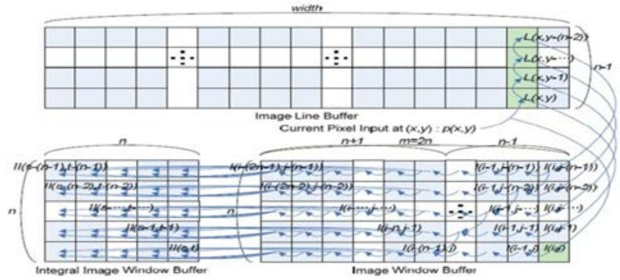

Fig.1 Architecture for generating integral image window

2.2 Haar Features:

A Haar classifier consists of two or three rectangles and their weight values, feature threshold value, and left and right values. Each rectangle presents four points using the coordinates (x, y) of most left and up point, width w, and height h as shown in "Fig.2".

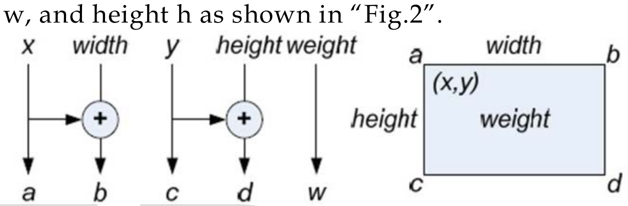

Fig.2 Rectangle calculation of Haar feature classifier.

The integral pixel value of each rectangle can be calculated using these points from the integral image window buffer as shown in "Fig.3".

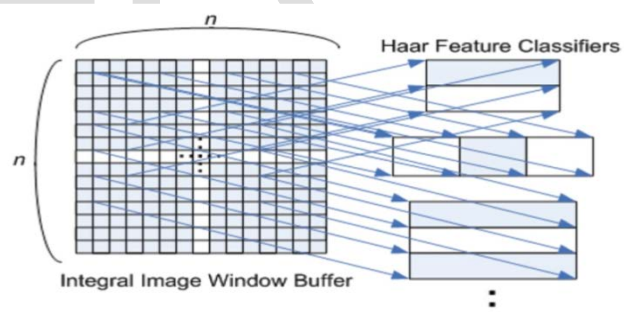

Fig.3 Simultaneous access to integral image

2.3 Haar Features Calculation:

Haar features are composed of either two or three rectangles. Face candidates are scanned and searched for Haar features of the current stage. The weight and size of each feature and the features themselves are generated using a machine learning algorithm from AdaBoost [6]. Each Haar feature has a value that is calculated by taking the area of each rectangle, multiplying each by their respective weights, and then summing the results. Several Haar feature compose a stage. A stage comparator sums the entire Haar feature

ISSN 2229-5518

resulting in a stage and compares this summation with a stage threshold. The threshold is a constant obtained from the AdaBoost algorithm [7]. The face detection algorithm eliminates face candidates quickly using a cascade of stages. The cascade eliminates candidates by making stricter requirements in each stage with later stages being much more difficult for a candidate to pass. Candidates exit the cascade if they pass all stages or fail any stage. A face is detected if a candidate passes all stages. This process is shown in Fig. 4.

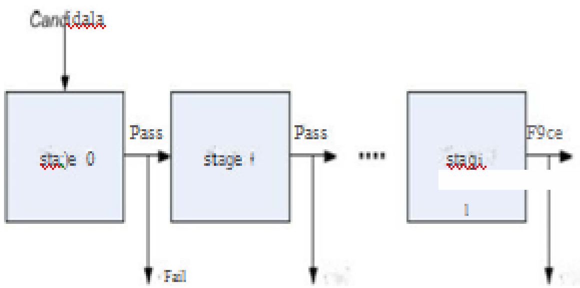

Fig. 4 Process

2.4 Adaboost Algorithm :

AdaBoost, short for Adaptive Boosting, is a machine learning algorithm, formulated by Yoav Freund and Robert Schapire[8]. The adaboost algorithm is based on the idea that a strong classifier can be created by linearly combining a number of weak classifiers. A weak classifier consists of a feature (j), a threshold (0), and a polarity (P) indicates the direction of the inequality:

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{if } pf(x) < p\theta \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

In the boosting algorithm T hypotheses are constructed each using a single feature. The final hypothesis is a weighted linear combination of the T hypotheses where the weights are inversely proportional to the training errors. Each iteration t, it will train a best weak classifier which can minimizes the training errors. After T iteration, we can obtain a strong classifier which is the linear combination of the T best weak classifiers multiplied by the weight values.

3 HARDWARE ARCHITECTURE OF THE PROPOSED FACE DETECTION

We proposed architecture for a real-time face detection system. "Fig. 5" shows the overview of the proposed architecture for face detection. It consists of five modules: variant pose, illumination condition, Facial Expression, Occulsion, Uncontrolled Background, display. Face Detection systems are not only detected faces on uniform environment. In reality, Peoples are always located on complex background with different texture and object. These 'thing' are the major factors to affect the performance of face detection system

4 IMPLEMENTATIONAND EXPERIMENTAL RESULTS

4.1 Implementation

The implementation uses the database which consists of 2429 face images and 4547 non face images. Training with yale database took 50 hours approximately and the algorithm accuracy is found as 93%.In our case we use image of variable sizes and with minimum rectangle of size 19x 19 to obtain best feature extraction. The implementation consists of 5 feature rectangle

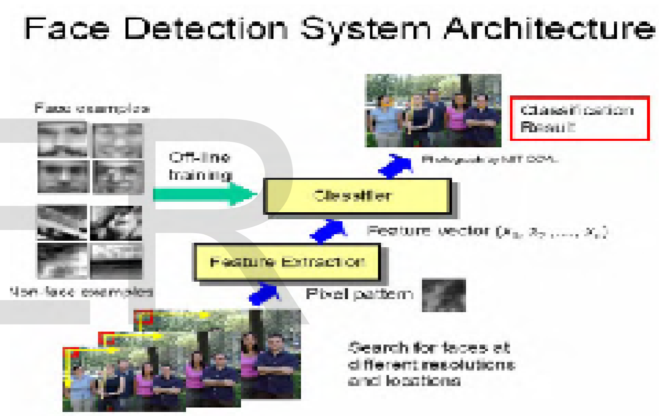

Fig.5 Block diagram of proposed face detection system using adaboost.

associated with the haar features. For every image we have calculated 51000 haar features using above said rectangle. We used the 14x 18 matrix classifier ranging from 0 to 6183 and used 6976 weights ranging from 0.000095 to 0.00023 which is distributed with range of 200% to recognize the face in the image

4.2 Experimental Result:

A high frame processing rate and low latency are important for many applications that must provide quick decisions based on events in the scene. We measure the performance of the proposed architecture for the face detection system. Face detection system when it is applied to a camera, which produces images consisting of 640×480 pixels at 60 frames per second.

Figure 10. Experimental result of face detection system

5 CONCLUSION

We present face detection based on the AdaBoost algorithm using Haar features. In our architecture, the scaling image technique is used instead of the scaling sub-window, and the integral image window is generated instead of the integral image contains whole image during one clock cycle. The Haar classifier is designed using a pipelined scheme, and the triple classifier which three single classifiers processed in parallel is adopted to accelerate the processing speed of the face detection system. Also we discussed the optimization of the proposed architecture which can be scalable for configurable devices with variable resources.

Finally, the proposed architecture is implemented on a Modelsim Altera 6.3 and its performance is measured and compared with an equivalent hardware implementation. We show about 35 times increase of system performance over the equivalent software implementation. We plan to implement more classifiers to improve our design.

REFERENCES

1. [1] Paul Viola and Michael J. Jones,"Robust real-time face detection," In International Journal of Computer Vision,pp. 137-154,2004.
2.
3. [2]Yoav Freund and Robert E. Schapire, "A decision- theoretic generalization of on-line learning and an application to boosting," in Journal of Computer and System Sciences,pp. 119-I39,1997
4. [3] Konstantinos G. Derpanis, "integral image-based representations," Department of Computer Science and Engineering,York University,
5. .
6. [4] JungukCho.Shahnam Mirzaei,Jason Oberg and Ryan Kastner, " FPGA-Based Face Detection System Using Haar Classifiers," FPGA'09,February22-24,2009.
7.
8. [5] Shian-Ru AJex.Ke, " Face Detection based on AdaBoost," Department of Electrical Engineering, University of Washington,U.S.A.
9.
10. [6] Mar Eramian "Histogram Equalization using Neighborhood Metrics," Proceedings of the Second Canadian Conference on Computer and Robot Vision (CRV'05)
11.
12. [7] Georghiades, A.:Yale Face Database, Center for computational Vision and Control at Yale University, http://cvc.yale.edu /projects/yalefaces /ya