www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 7 (July 2013), Page No. 2258-2261

Design and VLSI Implementation of High-Performance Face-Detection Engine using MCT

P.Mani Mohan[1], M. Janardhan Raju[2],
K.S Deveswari[3] M.E, B.Saroja[4].M.E(Ph.D.,),G.Sailaja[5]M.Tech
Siddharth Institute of Engineering and Technolog,[1,2,3],Siddartha Institute of Science and Technology[4,5]
Email_id: manimohan.7@gmail.com

Abstract--In this paper, we proposes a work of fiction hardware architecture of face-detection engine for mobile applications. We used MCT(Modified Census Transform) and Adaboost learning technique as basic algorithms of face-detection authenthication in mobile applications.

We have designed, implemented and verified the hardware architecture of face-detection engine for high-performance face detection and real-time processing. The face-detection chip is developed by verifying and implementing through FPGA. The developed ASIC chip has advantage in real-time processing, low power consumption, high performance and low cost which can be used as an authentication for the mobile applications. So we expect this chip can be easily used in mobile applications.

I. INTRODUCTION

Face-detection systems carry out a major role in biometric authentication, which uses features of the face, iris, fingerprint, retina, etc. These systems are usually used in places requiring high security, such as government agencies, bank, and research institutes; it is also applied to two- or three-dimensional face detection in areas such as artificial intelligence and robots, access control systems, cutting-edge digital cameras and advanced vehicle systems. Recently, the face-detection technology is being adopted in the mobile phone applications because of the pros in easy installation, low-cost, non-contacting method. Most of the existing face-detection engines in digital camera or mobile phone have been run by software. However, the tendency is that the technology is being developed to be run by hardware for improving the processing speed. These days, the technology is to combine hardware technique of face-detection and software technique of emotion, feeling, physiognomy and fortune recognition.

Face detection performance is known to be highly influenced by variations in illumination. Especially in mobile environment, the illumination condition is dependent on the surroundings (indoor and outdoor), time, and light reflection, etc. The proposed face-detection method is designed to detect in the variable illumination conditions through the MCT techniques, which can reduce the effects of illumination by extracting the structural information of objects. The proposed face-detection engine also renders high performance face detection rate by extracting highly reliable and optimized learning data through the Adaboost learning algorithm.

II. FEATURE GENERATION

The features used in this work are defined as structure kernels of size 3 × 3 which summarize the local spatial image structure. Within the kernel structure information is coded as binary information {0, 1} and the resulting binary patterns can represent oriented edges, line segments, junctions, ridges, saddle points, etc. Fig.1 shows some examples of this structure kernels. On a local 3 × 3 lattice there exist 29 = 512 such kernels. Actually there are only 29 − 1 reasonable kernels out of the 29 possible because the kernel with all elements 0 and that with all 1 convey the same information (all pixels are equal) and so one is redundant and thus is excluded. At each location only the best matching.

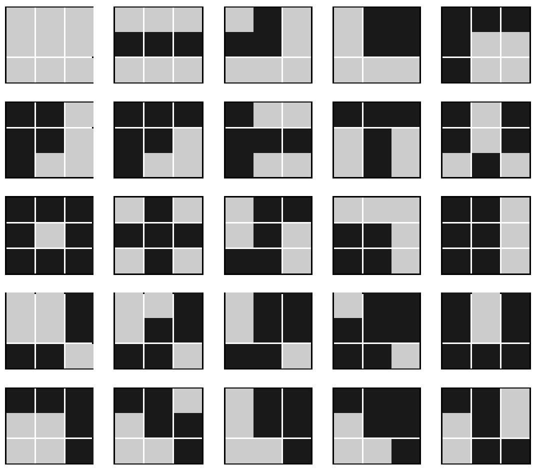


Figure 1: A randomly chosen subset of 25 out of $2^9 - 1$ possible Local Structure Kernels in a 3 × 3 neighborhood.

III. Basic Algorithms

A. *MCT (Modified Census Transform):*

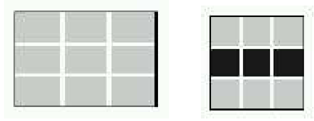
MCT presents the structural information of the window with the binary pattern {0, 1} moving the 3×3 window in an image small enough to assume that lightness value is almost constant, though the value is actually variable. This pattern contains information on the edges, contours, intersections, etc. MCT can be defined with the equation below:

$$\Gamma(X) = \bigotimes_{Y \in N'} \zeta(\bar{I}(X), I(Y)) \qquad (1)$$

Here X represents a pixel in the image, the 3×3 window of which X is the center is W(X); N' is a set of pixels in W(X) and Y represents nine pixels each in the window. In addition, $I(X)$ is the average value of pixels in the window, and $I(Y)$ is the brightness value of each pixel in the window. As a comparison function, $\zeta()$ becomes 1 in the case of $I(X) < I(Y)$, other cases are 0. As a set operator, ⊗ connects binary patterns of function, and then nine binary patterns are connected through operations. As a result, a total of 511 structures can be produced, as theoretically not all nine-pixel values can be 1. Thus connected binary patterns are transformed into binary numbers, which are the values of the pixels in MCT-transformed images.

With this transform we are able to determine all of the 511 structure kernels defined on a on a 3 × 3 neighborhood. To illustrate the modified transform consider a region of an image with the following pixel intensities in the first column. The structure kernels assigned by the original transform and the modified transform are displayed in column 2 and 3. The transforms are given for the center pixel of the image patch.

$\mathbf{I}(\mathrm{x})$ $\mathcal{C}(\mathrm{x})$ $\Gamma(\mathrm{x})$



1	1	1
5	5	5
1	1	1

From this example we can see that the original census transform will not capture the local image structure correctly in some cases while the modified transform assigns the right kernel.

The Census Transform (CT) is a non-parametric local trans-form which was first proposed by Zabih and Woodfill [10]. It is defined as an ordered set of comparisons of pixel in-tensities in a local neighborhood representing which pixels have lesser intensity than the center. In general the size of the local neighborhood is not restricted, but in this work we always assume a 3 × 3 surrounding as motivated in the last section. Let N (**x**) define a local spatial neighborhood of the pixel at **x** so that $\mathbf{x} \in_3$ N (**x**). The CT then gener-ates a bit string representing which pixels in N (**x**) have an intensity lower than **I**(**x**). With the assumption that pixel intensities are always zero or positive the formal definition of the process is like follows: Let a comparison function $\zeta(\mathbf{I}(\mathbf{x}), \mathbf{I}(\mathbf{x}'))$ be 1 if $\mathbf{I}(\mathbf{x}) < \mathbf{I}(\mathbf{x}')$ and let ⊗ denote the concatenation operation, the census transform at **x** is de-fined as

B. *Adaboost learning Algorithm*

Adaboost learning algorithm is created high-reliable learning data as an early stage for face-detection using faces

$$\mathcal{C}(\mathrm{x}) = \bigotimes_{\mathrm{y} \in \mathcal{N}} \zeta(\mathbf{I}(\mathrm{x}),\ \mathbf{I}(\mathrm{y})).$$

data. Viola and Jones [1] have proposed fast, high-performance face-detection algorithm. It is composed of cascade structure with 38 phases, using extracted features to effectively distinguish face and non-face areas through the Adaboost learning algorithm proposed by Freund and Schapire [2]. Furthermore, Froba and Ernst[3] have introduced MCT-transformed images and a face detector consisting of a cascade structure with 4 phases using the Adaboost learning algorithm.

The conventional AdaBoost procedure can be easily interpreted as a greedy feature selection process. Consider the general problem of boosting, in which a large set of classification functions are combined using a weighted majority vote. The challenge is to associate a large weight with each good classification function and a smaller weight with poor functions. AdaBoost is an aggressive mechanism for selecting a small set of good classification functions which nevertheless have significant variety. Drawing an analogy between weak classifiers and features, AdaBoost is an effective procedure for searching out a small

number of good “features” which nevertheless have significant variety. One practical method for completing this analogy is to restrict the weak learner to the set of classification functions each of which depend on a single feature. In support of this goal, the weak learning algorithm is designed to select the single rectangle feature which best separates the positive and negative examples (this
is similar to the approach of Tieu and Viola (2000) in
the domain of image database retrieval). For each feature, the weak learner determines the optimal threshold classification function, such that the minimum number of examples are misclassified. A weak classifier ($h(x, f, p, \theta)$) thus consists of a feature (f), a threshold (θ) and a polarity (p) indicating the direction of the inequality:

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

Here x is a 24 × 24 pixel sub-window of an image.
In practice no single feature can perform the classification task with low error. Features which are selected early in the process yield error rates between 0.1 and 0.3. Features selected in later rounds, as the task becomes more difficult, yield error rates between 0.4 and 0.5. Table 1 shows the learning algorithm.

This paper consists of a face detector with a single-layer structure, using only the fourth phase of the cascade structure proposed by Froba and Ernst.

IV. PROPOSED HARDWARE STRUCTURE

Regarding proposed hardware structure as shown in Figure 1, it is composed of color conversion module to convert to gray image from color image, noise reduction module to reduce image noise, image scaler module to detect various size of the face, MCT transform module to transform image for robustness various illumination, CD (Candidate detector) / CM (Confidence mapper) to detect candidate for final face-detection, position resizer module to resize face candidate areas detected on the scaled-down images as their corresponding points of original image size, data grouper module to group the duplicate areas determined to be the same face prior to determining the final face detection areas, overlay processor to play in displaying an output by marking square in relation to the final face-detection area on the color-based original image from the camera or to output to transfer the information of area and size in face-detection area to embedded system through host interface

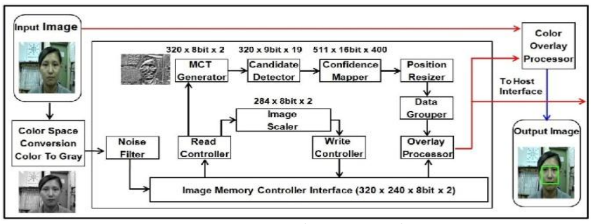


Fig 1. Block diagram of proposed face-detection engine

V. EXPERIMENTAL RESULT

The developed face detection system has verified superb performance of 99.76 % detection-rate in various illumination environments using Yale face database [4] and BioID face database [5] as shown in Table 1, Table 2, and Figure 2. And also we had verified superb performance in real-time hardware though FPGA and ASIC as shown in figure 3. First, the system was implemented using Virtex5 LX330 Board [6] with QVGA(320x240) class camera, LCD display. The developed face-detection FPGA system can be processed at a maximum speed of 149 frames per second in real-time and detects at a maximum of 32 faces simultaneously. Additionally, we developed ASIC Chip [7] of 1.2 cm x 1.2 cm BGA type through 0.18um, 1-Poly / 6-Metal CMOS Logic process. Consequently, we verified developed face-detection engine at real-time by 30 frames per second within 13.5 MHz clock frequency with 226 mW power consumption

(a) Result in various illumination conditions

(b) Results from Yale and BioID Face DB Fig 2. Face detection Results

(a) FPGA development environment (b) Developed chip and system

Fig 3. FPGA verification environment and ASIC chip system

VI. CONCLUSION

This paper has verified a process that overcomes low detection rates caused by variations in illuminations thanks to the MCT techniques. The proposed face-detection hardware structure that can detect faces with high reliability in real-time was developed with optimized learning data through the Adaboost algorithm. Consequently, the developed face-detection engine has strength in various illumination conditions and has ability to detect 32 various sizes of faces simultaneously.

The developed FPGA module can detect faces with high reliability in real-time. This technology can be applied to human face-detection logic for cutting-edge digital cameras or recently developed smart phones. Finally, face-detection chip was developed after verifying and implementing through FPGA and it has advantage in real-time, low power consumption and low cost. So we expect this chip can be easily used in mobile applications.

REFERENCE

[1] *Paul Viola and Michael J. Jones, "Robust real-time face detection" In International Journal of Computer Vision, pp. 137-154, 2004.*

[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. 119-139, 1997.*

[3] *Bernhard Froba and Andreas Ernst, "Face detection with the Modified Census Transform", IEEE International Conference. On Automatic Face and Gesture Recognition, pp. 91-96, Seoul, Korea, May. 2004.*

[4] *Georghiades, A. : Yale Face Database, Center for computational Vision and Control at Yale University, http: //cvc.yale.edu /projects /yalefaces /yalefa*

[5] *The BioID face database: [http : //www.bioid.com /downloads /facedb /facedatabase.html]*

[6] *Dongil Han, Hyunjong Cho, Jongho Choi, Jaeil Cho, "Design and Implementation of real-time high performance face detection engine", SP(Signal Processing), 47th Book, 3 Issue, The Institute of Electronics Engineers of Korea Publication(IEEK), March, 2010.*

[7] *Seung-Min Choi, Jiho Chang, Jaeil Cho, Dae-Hwan Hwang, "Implementation of Robust Real-time Face Detection ASIC", IEEK Conference 33th Book, 1 Issue, 467-470, Jeju, Korea, June, 2010*