Face Recognition System using Curvelet Transform and Haar classifier

S.Rajeshwari[1],J.Divya [2],R.Karthika [3], P.Tamilselvan[4]
UG Scholar123, Rajiv Gandhi College of Engineering and Techology,Puducherry123

Assistant Professor, Rajiv Gandhi College of Engineering and Technology, Pondicherry4

Abstract: This paper presents a hardware architecture for face detection based system on using Haar features and curvelet transform. We describe the hardware design techniques including 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. Also, we discuss the optimization of the proposed architecture which can be scalable for configurable devices with variable resources. The proposed architecture for face detection has been designed using Verilog HDL.Matlab and simulated using Modelsim software

IndexTerms - **AdaBoost, architecture, face detection, FPGA, Haar classifier, image processing, real-time.**

I. INTRODUCTION

Face detection in image sequence has been an active research area in the computer vision field in recent years due to its potential applications such as monitoring and surveillance [1], human computer interfaces [2], smart rooms [3], intelligent robots [4], and biomedical image analysis [5]. Face detection is based on identifying and locating a human face in images 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 [6] is most popular among the face detection approaches based on statistic methods. This face detection is a variant of the AdaBoost algorithm [7] which achieves rapid and Face detection in image sequence has been an active research area in the computer vision field in recent years due to its potential applications such as monitoring and surveillance [1], human computer interfaces [2], smart rooms [3], intelligent robots [4], and biomedical image analysis [5]. Face detection is based on identifying and locating a human face in images 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 [6] is most popular among the face detection approaches based on statistic methods. This face detection is a variant of the AdaBoost algorithm [7] which achieves rapid and Face detection in image sequence has been an active research area in the computer vision field in recent years due to its potential applications such as monitoring and surveillance [1], human computer interfaces [2], smart rooms [3], intelligent robots [4], and biomedical image analysis [5]. Face detection is based on identifying and locating a human face in images 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 [6] is most popular among the face detection approaches based on statistic methods. This face detection is a variant of the AdaBoost algorithm [7] which achieves rapid and robust face detection. They proposed a face detection framework based on the AdaBoost learning algorithm using Haar features. However, the face detection requires considerable computation power because many Haar feature classifiers check all pixels in the images. Although real-time face detection is possible using high performance computers, the resources of the system tend to be monopolized by face detection. Therefore, this constitutes a bottleneck to the application of face detection in real time

II. HAAR FACE DETECTION ALGORITHM

The face detection algorithm proposed by Viola and Jones is used as the basis of our design. The face detection algorithm looks for specific Haar features of a human face. When one of these features is found, the algorithm allows the face candidate to pass to the next stage of detection. A face candidate is a rectangular section of the original image called a sub-window. Generally these sub-windows have a fixed size (typically 24×24 pixels). This sub-window is often scaled in order to obtain a variety of different size faces. The algorithm scans the entire image with this window and denotes each respective section a face candidate [6]. The algorithm uses an integral image in order to process Haar features of a face candidate in constant time. It uses a cascade of stages which is used to eliminate non-face candidates quickly. Each stage consists of many different Haar features. Each feature is classified by a Haar feature classifier. The Haar feature classifiers generate an output which can then be provided to the stage comparator. The stage comparator sums the outputs of the Haar feature classifiers and compares this value with a stage threshold to determine if the stage should be passed. If all stages are passed the face candidate is concluded to be a face. These terms will be discussed in more detail in the following sections.

2.1 Integral Image

The integral image is defined as the summation of the pixel values of the original image. The value at any location (x, y) of the integral image is the sum of the image's pixels above and to the left of location (x, y). Figure 1 illustrates the integral image generation.

2.2 Haar Features

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][7]. The weights are constants generated by the learning algorithm. There are a variety of forms of features as seen below in Figure 2. Each Haar feature has a value that is calculated by taking the area of each by their respective weights, and then summing the results. The area of each rectangle is easily found using the integral image. The coordinate of the any corner of a rectangle can be used to get the sum of all the pixels above and to the left of that location using the integral image. By using each corner of a rectangle, the area can be computed quickly as denoted by Figure 3. Since L_1 is subtracted off twice it must be added back on to get the correct area of the rectangle. The area of the rectangle R, denoted as the rectangle integral, can be computed as follows using the locations of the integral image: $L_4-L_3-L_2+L_1$

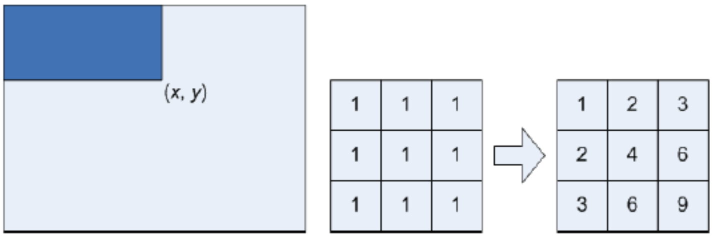

Figure 1. Integral image generation. The shaded region represents the sum of the pixels up to position (*x*, *y*) of the image. It shows a 3×3 image and its integral image representation.

Figure 2. Examples of Haar features. Areas of white and black regions are multiplied by their respective weights and then summed in order to get the Haar feature value.

2.3 Haar Feature Classifier

A Haar feature classifier uses the rectangle integral to calculate the value of a feature. The Haar feature classifier multiplies the weight of each rectangle by its area and the results are added together. Several Haar feature classifiers compose a stage. A stage comparator sums all the Haar feature classifier results in a stage and compares this summation with a stage threshold. The threshold is also a constant obtained from the AdaBoost algorithm. Each stage does not have a set number of Haar features. Depending on the parameters of the training data individual stages can have a varying number of Haar features. For example, Viola and Jones' data set used 2 features in the first stage and 10 in the second. All together they used a total of 38 stages and 6060 features [6]. Our data set is based on the OpenCV data set which used 22 stages and 2135 features in total [16][17].

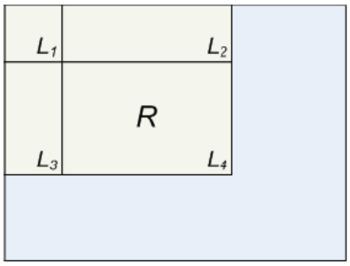

Figure 3. Calculating the area of a rectangle *R* is done using the corner of the rectangle: L_4-L_3-L_2+L_1.

2.4 Cascade

The Viola and Jones 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.

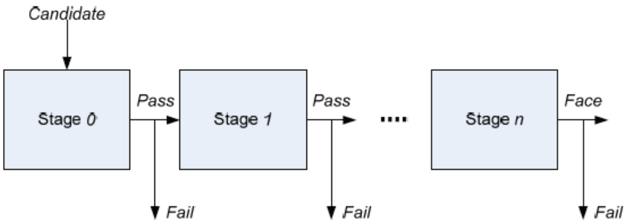

Figure 4. Cascade of stages. Candidate must pass all stages in the cascade to be concluded as a face.

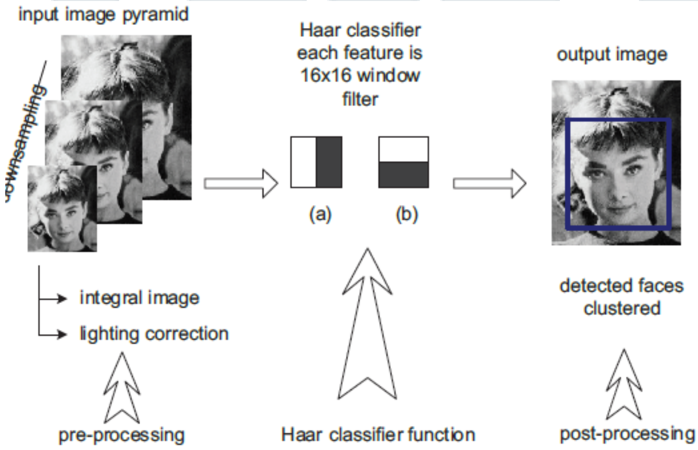

Figure. 5. Haar classifier face detection algorithm data flow.

As illustrated in Fig. 5, after we have trained the Haar classifiers with our own training images [18], we could detect faces in three steps. The first step is *pre-processing.* It is used to calculate the integral images and lighting corrections from the scaled (scale factor is 1.2) original image. The second step is *Haar classifier function.* The main purpose during this step is to scan every image pixel with the trained-Haar classifiers to determine if it is a face pixel or not.

The third step is *post-processing.* It clusters the adjacent As illustrated in Fig. 2, after we have trained the Haar classifiers with our own training images [18], we could detect faces in three steps. The first step is *pre-processing.* It is used to calculate the integral images and lighting corrections from the scaled (scale factor is 1.2) original image. The second step is *Haar classifier function.* The main purpose during this step is to scan every image pixel with the trained-Haar classifiers to determine if it is a face pixel or not. The third step is *post-processing.* It clusters the adjacent detected faces to one or several rectangles to represent faces according to distance factor of 5.

III. CURVELET ALGORITHM

Curvelet is a multidirectional and multiscale transform which represents edges and other singularities along the curves better

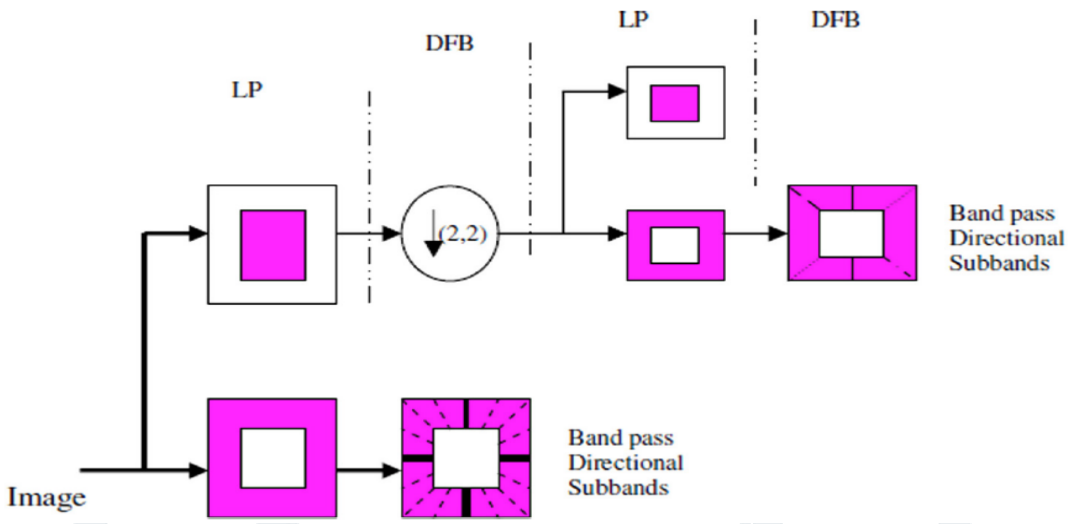

Figure.6 Decomposition scheme of CNT.

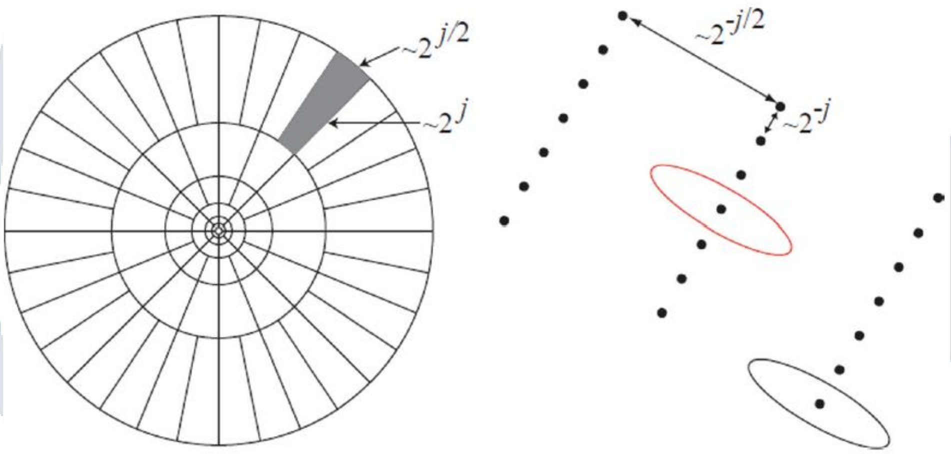

Figure.7 Curvelet tiling of space and frequency.

The proposed framework creates a novel feature space by extracting features in temporal domain using curvelet transform (CLT) in order to improve rate of face recognition under different challenges. The computational complexity of the discrete contourlet transform is Oðn2Þ for ðn _ nÞ images and for CLT the computational complexity is Oðn2lognÞ. So the total computational complexity for our algorithm is (Oðn2Þ +Oðn2lognÞ), i.e. Oðn2lognÞ because 1 is negligible w.r.t. logn where n = 128. Schematic overview of the proposed framework is illustrated in Fig. 6.

In the proposed face recognition method, initially as the preprocessing step the input image is passed through the Gaussian smoothing filter for noise removal. Histogram equalization is used to remove illumination differences in the image and the preprocessed image is now ready for subband calculation. Next the proposed entropy based subband selection procedure has been
applied to select the most important subbands, which effectively reduces the dimension of the feature vector. Finally, using a classifier the face image is recognized, which is close to the image present in the image database.

3.1 Subband Calculation

Coefficients of CNT and CLT subbands are used as extracted features. In contourlet domain, each preprocessed image has been decomposed using CNT up to 4th level. It has been observed from Fig. 5 that the image is decomposed into 21 directions at the first level, 22 at the second level, 23 at the third level and 24 at the 4th level. Thus we obtain 30 i.e. (21 þ 22 þ 23 þ 24) directional subbands for each training image. In the experiments, for each face image we choose four levels to achieve maximum recognition rate. In CNT, PKVA filter (Phoong et al., 1995) is used as LP and DFB. Similarly to obtain the CLT based features each preprocessed face image is decomposed into 4 scale and angle 8 and coefficients of 24 detail subbands are obtained as shown in Fig. 6.

3.2 Feature Extraction

In the work features are extracted from temporal domain based on the statistical characteristics of the coefficients. The idea has been adopted from the spatial domain face recognition methods where statistical features are often used (El-Khamy et al., 2001; Das et al., 2017). In spatial domain, statistical analysis reveals the relationships between the gray levels of pixels in an image. The spatial distribution of gray values are used to compute the local descriptors at each point in the image and image statistics are calculated from the distributions of the local descriptors. The statistical methods can be classified into first order (one pixel), second order (pair of pixels) and higher order (three or more pixels) statistics. The first order statistics estimate properties (e.g. average and variance) of individual pixel values by keeping aside the spatial interaction between the pixels. The second and higher order statistics estimate relationship between two or more pixel gray level values occurring at specific locations with respect to each other, like co-occurrence or co-variance matrix. These different statistical measures are used in a wide range of scientific and social research, including: biostatistics, computational sociology, network biology, social science and in social research etc. Here we use statistical measures applied on the coefficients to study the effect of the transform domain features in face recognition. Statistical measures are employed on each 24 directional subbands of CNT and 20 detail subbands of CLT, using following equations to obtain the feature set F1. Feature set F1 consists of statistical parameters mean (l), standard deviation (r), energy (e), skewness (s) and kurtosis (k).

IV. EXPERIMENTS / RESULTS

A high frame processing rate and low latency are important for many applications that must provide quick decisions based on events in the scene [19]. We measure the performance of the proposed architecture for the face detection system. Table 4 shows the performance of the implemented face detection system when it is applied to a camera, which produces images consisting of 320×240 pixels at 60 frames per second. The system performance depends on the number of faces in the images. The single classifier face detection system is capable of processing the images at speeds of an average of 15.14 fps. The triple classifier face detection system is capable of processing the images at speeds of an average of 26.51 fps. The triple classifier face detection system has the performance improvement of 1.75 times than the single classifier one. The performance of the implemented face detection system when it is applied to a camera, which produces images consisting of 640×480 pixels at 60 frames per second. The single classifier face detection system is capable of processing the images at speeds of an average of 4.35 fps. The triple classifier face detection system is capable of processing the images at speeds of an average of 6.96 fps. The triple classifier face detection system has the performance improvement of 1.6 times than the single classifier one. This is due to the concurrent operations of the three single classifiers in parallel. Although the usage of the system resource increases, the system performance increases dramatically. The performance of the software program is determined by measuring the computation time required for performing face detection on the PC; in this case a Intel Pentium I3 with 4 GB RAM.

Figure 8. Experimental result of face detection system.

V. RESULTS AND DISCUSSION

We present a hardware architecture for face detection based on the AdaBoost algorithm using Haar features and curvelet transform. 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. When the proposed face detection system is used in a system which requires face detection, only a small percentage of the system resources are allocated for face detection. The remainder of the resources can be assigned to preprocessing stage or to high level tasks such as recognition and reasoning. We have demonstrated that this face detection, combined with other technologies, can produce effective and powerful applications. The proposed method may be extended as follows:

(i) An optimization framework may be developed for the selection of subbands instead of threshold, which provides best recognition accuracy under the different constraint of face recognition.

(ii) The importance of other statistical measures in the temporal domain may be studied to improve the rate of face recognition.

(iii) For real time application compressive sensing based method can be employed for obtaining reduced search space.

REFERENCES

[1] Z. Guo, H. Liu, Q. Wang, and J. Yang, "A Fast Algorithm of Face Detection for Driver Monitoring," In Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications, vol.2, pp.267 - 271, 2006.

[2] M. Yang, N. Ahuja, "Face Detection and Gesture Recognition for Human-Computer Interaction," The International Series in Video Computing , vol.1, Springer, 2001.

[3] Z. Zhang, G. Potamianos, M. Liu, T. Huang, "Robust Multi-View Multi-Camera Face Detection inside Smart Rooms Using Spatio-Temporal Dynamic Programming," International Conference on Automatic Face and Gesture Recognition, pp.407-412, 2006.

[4] W. Yun; D. Kim; H. Yoon, "Fast Group Verification System for Intelligent Robot Service," IEEE Transactions on Consumer Electronics, vol.53, no.4, pp.1731-1735, Nov. 2007.

[5] V. Ayala-Ramirez, R. E. Sanchez-Yanez and F. J. Montecillo-Puente "On the Application of Robotic Vision Methods to Biomedical Image Analysis," IFMBE Proceedings of Latin American Congress on Biomedical Engineering, pp.1160-1162, 2007.

[6] P. Viola and M. Jones, "Robust real-time object detection," International Journal of Computer Vision, 57(2), 137-154, 2004.

[7] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generaliztion of On-Line Learning and an Application to Boosting," Journal of Computer and System Sciences, no. 55, pp. 119-139, 1997. Ali, A. 2001.Macroeconomic variables as common pervasive risk factors and the empirical content of the Arbitrage Pricing Theory. Journal of Empirical finance, 5(3): 221–240.

[8] T. Theocharides, N. Vijaykrishnam, and M. J. Irwin, "A parallel architecture for hardware face detection," In Proceedings of IEEE Computer Society Annual Symposium Emerging VLSI Technologies and Architectures, pp. 452- 453, 2006.

[9] R. McCready "Real-time face detection on a configurable hardware system," In Proceedings of the Roadmap to Reconfigurable Computing, International Workshop on Field-Programmable Logic and Applications, pp.157-162, 2000.

[10] M. S. Sadri, N. Shams, M. Rahmaty, I. Hosseini, R. Changiz, S. Mortazavian, S. Kheradmand, and R. Jafari, "An FPGA Based Fast Face Detector," In Global Signal Processing Expo and Conference, 2004.

[11] Y. Wei, X. Bing, and C. Chareonsak, "FPGA implementation of AdaBoost algorithm for detection of face biometrics," In Proceedings of IEEE International Workshop Biomedical Circuits and Systems, page S1, 2004.

[12] M. Yang, Y. Wu, J. Crenshaw, B. Augustine, and R. Mareachen, "Face detection for automatic exposure control in handheld camera," In Proceedings of IEEE international Conference on Computer Vision System, pp.17, 206.

[13] V. Nair, P. Laprise, and J. Clark, "An FPGA-based people detection system," EURASIP Journal of Applied Signal Processing, 2005(7), pp. 1047-1061, 2005.

[14] C. Gao and S. Lu, "Novel FPGA based Haar classifier face detection algorithm acceleration," In Proceedings of International Conference on Field Programmable Logic and Applications, 2008.

[15] M. Hiromoto, K. Nakahara, H. Sugano, "A specialized processor suitable for AdaBoost-based detection with Haarlike features," In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp.1-8, 2007.

[16] G. Bradski and A. Kaehler, "Learning OpenCV: Computer Vision with the OpenCV Library," O'Reilly Media, Inc., 2008.

[17] Open Couter Vision Library, , Oct. 2008. DOI=http://sourceforge.net/projects/opencvlibray/