

Real Time Face recognition Technique based on skin pixels

Vikas V. Mankar¹, Chandrakant N.Bhoyar²

Priyadarshini College of Engineering, Nagpur Maharashtra (India)

Abstract

This paper presents face detection system using Haar features. We describe 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. We show about increase of system performance over the equivalent software implementation

Index Terms—Haar Classifier, Image processing, real-time

1.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, human computer interfaces, smart rooms, intelligent robots, and biomedical image analysis[1]. 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. This face detection is a variant of the AdaBoost algorithm [1][2] which achieves rapid and robust face detection. The proposed face detection framework is 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

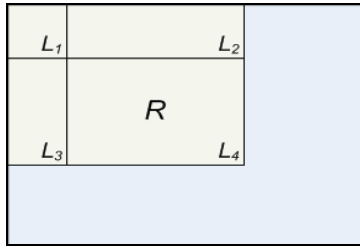
application of face detection in real time. All manuscripts must be in English. These guidelines include complete descriptions of the fonts, spacing, and related information for producing your proceedings manuscripts.

2. Face Detection Algorithm

In this paper we used Haar classifier algorithm for face detection. 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. 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.

2.1 Haar Feature Classifier

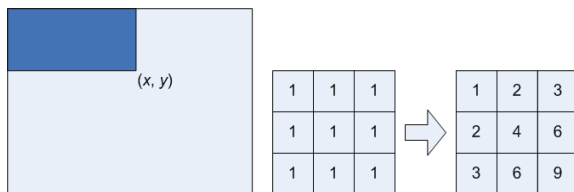
A Haar feature classifier uses the rectangle integral to calculate the value of a feature. The Haar feature classifier multiplies the weight of each rectangle by its area and the results are added together. Several Haar feature classifiers compose a stage. A stage comparator sums all the Haar feature classifier results in a stage and compares this summation with a stage threshold. The threshold is also a constant obtained from the AdaBoost algorithm.[3] 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.



“Figure 1. Calculating the area of a rectangle R is done using the corner of the rectangle: $L4-L3-L2+L1$ ”

2.2 Haar Features

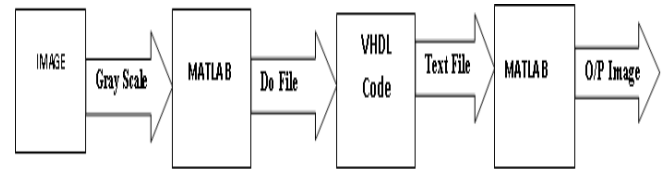
Haar features are composed of either two or three rectangles. Face candidates are scanned and searched for Haar features of the current stage. The weight and size of each feature and the features themselves are generated using a machine learning algorithm from AdaBoost. The weights are constants generated 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 rectangle, multiplying each by their respective weights, and then summing the results.



“Figure 2. 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”.

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 1. Since $L1$ 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: $L4-L3-L2+L1$

3. Schematic of proposed face detection technique

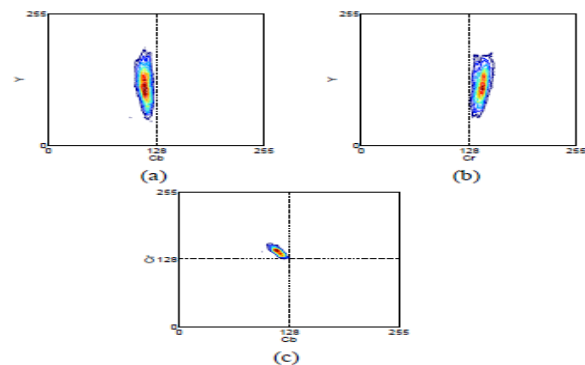


“Figure 3. Block Diagram of proposed algorithm”

Figure 3 shows the block diagram of the proposed algorithm. Once the pixels of interest were identified Projections of the histogram of a representative set of skin pixels in the Y Cb Cr color space,[1]unsupervised segmentation was used to separate these Pixels into smaller regions which were homogeneous in color. This is important because the skin detection will produce non-homogeneous regions often containing more than a single object. The unsupervised segmentation usually further partitioned the skin detected areas into smaller homogeneous regions making necessary the use of a region merging stage to extract the faces. The next subsections will describe in detail the changes made in these basic blocks which improve the overall performance of the whole system.

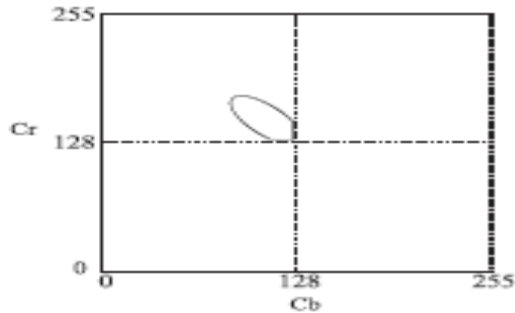
3.1. Skin detection

The first step, skin detection, is used to segment regions of the image which potentially correspond to face regions based on pixel color. Under normal illumination conditions skin colors fall into a small region of the color space and it is possible to use this information to classify each pixel of the image as skin-like or non skin-like. The projections of the 3-D histogram of a representative set of manually extracted skin pixels are plotted in the Figure 4.



“Figure4. Projections of the histogram of a representative set of skin pixels in the YCbCr color space”

Figures 4.a and 4.b show that the luminance is uncorrelated respect to the CbCr components and Fig. 4.c shows that CbCr are highly correlated and define a small cluster on the CbCr plane. Then, a pixel will be labeled as skin-like if its chrominance vector falls into the region plotted in the Figure 6 and its luminance value is within the interval 45.



“Figure5. Region bounding the skin-like colors on the CbCr color space”

These values were chosen empirically to reduce the miss detection rate since it is impossible to reduce the false alarm rate produced by skin-like colored background objects. The negative effect of this false alarm should be solved with additional processing. The most important advantage of this simple algorithm is that the skin detection can be implemented with a LUT what makes the algorithm extremely fast?

3.2. Unsupervised segmentation

Once the pixels of interest are identified, unsupervised segmentation is used to separate these pixels into smaller regions which are homogeneous in color. To that end, once the skin-like pixels are detected, a 2D histogram in the Cb-Cr color space is constructed [5]. Then, this histogram is treated as a gray-scale image. The histogram is previously smoothed with a linear filter to avoid over-segmentation.

3.3. Region merging and face extraction

The unsupervised segmentation described in the previous section can split the face regions into smaller homogeneous regions. Therefore, we must incorporate a way to merge regions into the system. To that end, we have modeled the behavior of a manually segmented set of face regions. The model takes into account parameters regarding shape, size, position and texture of the face regions. Once we get the regions of the unsupervised segmentation, we search each pair-wise merging of regions to find the merged region which best fits the face model.

[6]Then, if the new region fits the model better than the original regions, they are merged. The process is repeated until we cannot find any merging which fits the model better than the original regions. At this point, the merging of any two regions will only reduce the quality of the match to the face model. Each node represents a region of the image with the internal nodes representing regions which result from merging.

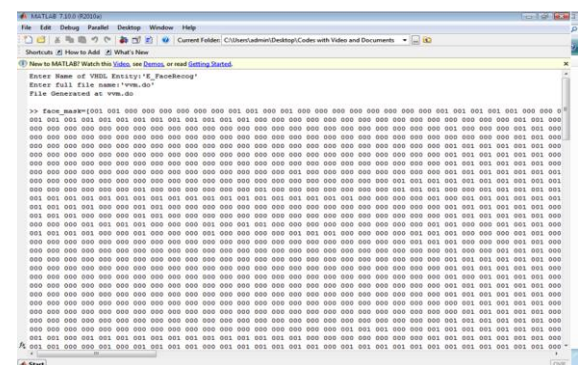
4. EXPERIMENTS / RESULTS

A high frame processing rate and low latency are important for many applications that must provide quick decisions based on events in the scene. We measure the performance of the proposed architecture for the face detection system.



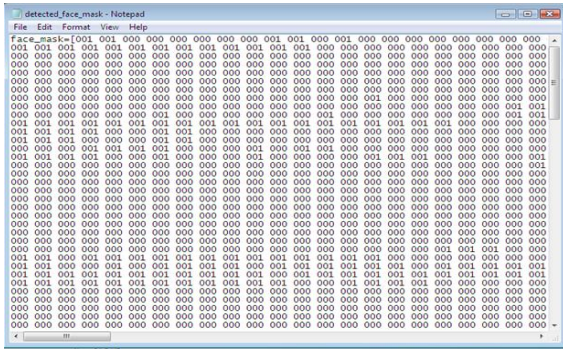
“Figure6. Input Image “

The above image is considered as an input image for detecting the faces at output.



“Figure 7. Detected Face Mask”

The above figure shows the detected face mask of the input image shown in figure 7 for the image size of 128 *128



“Figure8. Input for VHDL”

The detected face mask is processed for detecting the faces from an image on the basis of above face mask to generate the output.



Figure 9. Face detected output (128*128)

4.1 Output for same input image with size 64*64



“Figure 10. Output image (64*64)”

From the above figure comparing it with previous one it is observed that the system produces the better result for image having more number of pixel values.

4.2 Some more results



“Figure 11(a) input image”



“Figure 11(b) output image”

The figure 11(b) shows the face extracted image by the proposed face detection system for the image shown in figure 11(a)



Figure 12(a) Input image



Figure 12(b) Output image

5. Conclusion

The proposed algorithm has been checked & result is shown in Figure 9. It can be seen how the algorithm is able to detect a variety of different faces in spite of the difficulty associated to different illumination conditions and different face poses. 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. Thus the above face detection system provides the better face detection for the image having more number of pixel values.

6. References

- [1] Junguk Cho, Shahnam Mirzaei, Jason Oberg, Ryan Kastner "FPGA-Based Face Detection System Using Haar Classifiers" *FPGA'09*, February 22-24, 2009, Monterey, California, USA. ACM 978-1-60558-410-2/09/02...
- [2] 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.
- [3] M. Yang, N. Ahuja, "Face Detection and Gesture Recognition for Human-Computer Interaction," The International Series in Video Computing, vol.1, Springer, 2001.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] P. Viola and M. Jones, "Robust real-time object detection," International Journal of Computer Vision, 57(2), 137-154, 2004. Generalization of On-Line Learning and an Application to Boosting," Journal of Computer and System Sciences, no. 55, pp. 119-139, 1997