

Multi-view Face Detection and Recognition using Haar-like Features

Zhaomin Zhu, Takashi Morimoto, Hidekazu Adachi, Osamu Kiriyama,

Tetsushi Koide and Hans Juergen Mattausch

Research center for nano-devices and systems, Hiroshima University

E-mail: zzm@sxsys.hiroshima-u.ac.jp

1. Introduction

There are a number of techniques that can successfully detect frontal upright faces in a wide variety of images [1]. Some systems can explicitly address non-upright face detection [3]. This paper describes progress toward a system which can detect and recognize faces regardless of pose reliably and in real-time based on Haar-like features. Haar-like features are introduced by Viola et al. [2] and improved by Lienhart et al. The detection technique is based on the idea of the wavelet template that defines the shape of an object in terms of a subset of the wavelet coefficients of the image. We have found that the simple try-all-poses system in fact yields a slightly superior receiver operating characteristics (ROC) curve, though is slower. This approach is selected because of its computational efficiency and simplicity.

2. Face Detection Framework

The input image is scanned across location and scale using a scaling factor of 1.1. At each location an independent decision is made regarding the presence of a face. This leads to a very large number of classifier evaluations; approximately 50,000 in a 320x240 image. Following the AdaBoost algorithm [4] a set of weak binary classifiers is learned from a training set. Each classifier is a simple function made up of rectangular sums followed by a threshold. In each round of boosting one feature is selected, that with the lowest weighted error. The feature is assigned a weight in the final classifier using the confidence rated AdaBoost procedure. In subsequent rounds incorrectly labeled examples are given a higher weight while correctly labeled examples are given a lower weight. In order to reduce the false positive rate while preserving efficiency, classification is divided into a cascade of classifiers. An input window is passed from one classifier in the cascade to the next as long as each classifier classifies the window as a face. The threshold of each classifier is set to yield a high detection rate. Early classifiers have fewer features while later ones have more so that easy non-face regions are quickly discarded. Each classifier in the cascade is trained on a negative set consisting of the false positives of the previous stages. This allows later stages to focus on the harder examples.

In order to train a full cascade to achieve very low false positive rates, a large number of examples are required. After 5 stages the false positive rate is often well below 1%. The image features (see Fig. 1) are called *Rectangle Features* and are reminiscent of Haar basis functions [5]. Each rectangle feature is binary threshold function constructed from a threshold, and a *rectangle filter* which is a linear function of the image.

The value of a *two-rectangle filter* is the difference between the sums of the pixels within two rectangular regions. The regions have the same size and shape and are horizontally or

vertically adjacent. A *three-rectangle filter* computes the sum within two outside rectangles subtracted from twice the sum in a center rectangle. Finally a *four-rectangle filter* computes the difference between diagonal pairs of rectangles. Given that the base resolution of the classifier is 24 by 24 pixels, the exhaustive set of rectangle filters is quite large, over 100,000, which is roughly $O(24^4)$ (i.e. the number of possible locations times the number of possible sizes). The actual number is smaller since filters must fit within the classification window. Computation of rectangle filters can be accelerated using an intermediate image representation called the integral image [2]. Using this representation any rectangle filter, at any scale or location, can be evaluated in constant time. The form of the final classifier returned by AdaBoost is a perceptron - a thresholded linear combination of features.

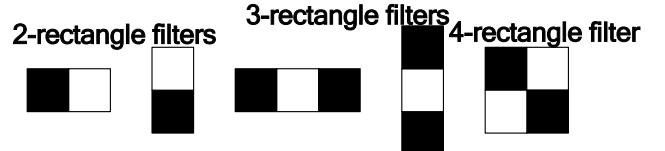


Figure 1: Haar-like features used for face detection.

An input window is evaluated on the first classifier of the cascade and if that classifier returns false then computation on that window ends and the detector returns false. If the classifier returns true then the window is passed to the next classifier in the cascade. The next classifier evaluates the window in the same way. If the window passes through every classifier with all returning true then the detector returns true for that window. The more a window looks like a face, the more classifiers are evaluated on it and the longer it takes to classify that window. Since most windows in an image do not look like faces, most are quickly discarded as non-faces. The overall algorithm for the detector is given in Figure 2.

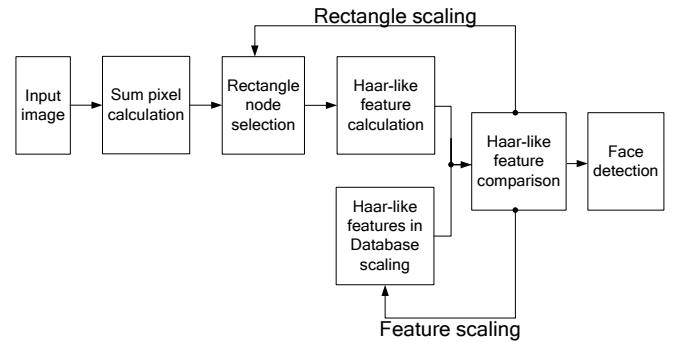


Figure 2: Flow diagram of the face detection.

We trained an upright detector using 2000 manually cropped 20x20 pixel faces and 2000 background (non-face)

patches. All profile faces were derotated so that the faces were looking approximately straight right. The resulting cascade has 11 layers of classifiers with the first six classifiers having 9, 9, 3, 7, 10 and 9 features, respectively.

We trained only one detector for frontal faces. Therefore we rotate the picture to be detected. The rotation angle is 30 degrees and we make 12 in-plane rotations so that together, the 12 pictures cover the full 360 degrees of possible rotations. We made translations of pixel coordinates for image rotation. Though there are 12 translations, in fact we only need two pair of coordinates, which are $(0.866x-0.5y, 0.866y+0.5x)$ and $(0.5x-0.866y, 0.5y+0.866x)$ ((x,y) is the pixel coordinate before rotation), other translated coordinates are simply the reverse or mirror of the above 3 pair coordinates.

The input images are preprocessed using histogram equalization to alleviate luminance variance. The achieved face detection rate is 95% with 0.1% false positive rate. Figure 3 gives some examples of face detection results. Rotated face can be detected correctly (Fig. 3(b)) for both color and gray-scale images. It takes less than 0.3 seconds in a Pentium IV 2.8GHz machine to execute the software implementation of our face detection algorithm for a 320x240 image.

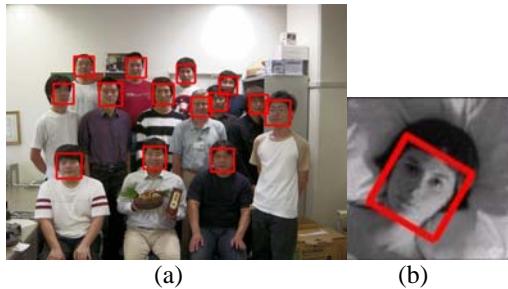


Figure 3: Results of Human face detection.

3. Face Recognition System

We also implemented haar-like feature based algorithm for the face recognition purpose. Different with face detection which needs only one training procedure for detection of all faces, each person's face should be trained in the face recognition step. The face size for training is chosen as 30x30 pixels. We use one person's faces under different conditions as positive samples and use other persons' faces as negative samples. In the face recognition step, we only process the detected face region (Fig. 4) of the complete picture.

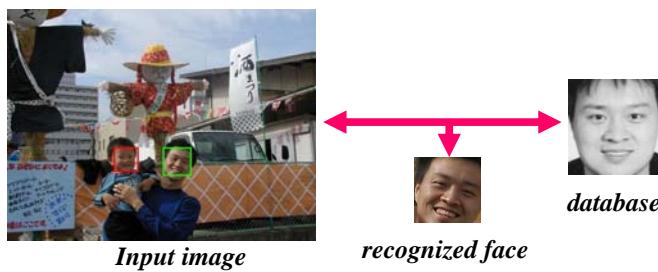


Figure 4: Face recognition example.

To decrease the false positive rate, the threshold of the final classifier is increased. This unfortunately also reduces the recognition rate. To increase the recognition rate again (now accompanies by a higher false positive rate), classifier layers

are removed from the end of the cascade. This is done simultaneously for all of the classification stages of the recognition system. Finally we achieved 75% correct face recognition rate with 15% false positive rate in less than 0.1 seconds recognition time, with a Pentium IV 2.8GHz machine.

4. Hardware Realization

Figure 5 shows the hardware structure of face detection as well as recognition system. It consists of memories, counters, adders, multipliers, comparators and peripheral circuits. Because the Haar-feature based algorithm doesn't use any nonlinear equations such as integral or differential, it's very easy to be implemented into an FPGA chip. Meanwhile because we use the same type algorithm for face detection and recognition; it may be possible to construct a unified face detection and recognition hardware. The complexity of the hardware structure is related to the input image size.

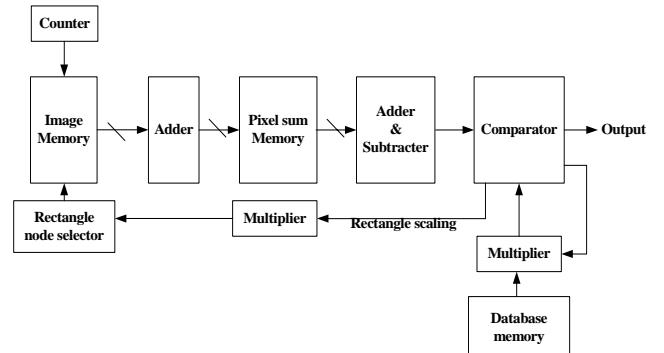


Figure 5: Proposed hardware structure of face detection and recognition system.

5. Conclusions

We have demonstrated the possibility of a unified face detection and recognition system for in-plane rotated faces based on haar-like features. The face detection rate is 95% with 0.1% false positive rate and the face recognition rate achieves 75% with 15% false positive rate at the present development stage. The execution time of the whole system takes is shorter than 0.7 seconds for a QVGA size image on a 2.8GHz Pentium 4 PC. The proposed method works well and has the speed advantage compared with other methods. We also described a possible hardware structure for the proposed system.

References

- [1] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *International Conference on Computer Vision*, 2000.
- [2] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, HI, December 2001.
- [3] H. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 38–44, 1998.
- [4] R. Schapire and Y. Singer. Improving boosting algorithms using confidence-rated predictions, 1999.
- [5] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *International Conference on Computer Vision*, 1998.

Introduction and Background

Definition of Face Detection:

Given an arbitrary image, the goal of face detection is to determine whether or not there are any faces in the image and, if present, return the image location and extent of each face.

Challenges associated with face detection

1. Pose

Frontal, 45 degree, profile, upside down

2. Presence or absence of structural components

Beards, mustaches, glasses, scarf

3. Facial expression

4. Occlusion

5. Image orientation

6. Imaging conditions

Lighting, camera characteristics (sensor, response, lenses)

Haar-like features for face region detection

The Haar-like feature is specified by its shape, position and the scale.



If $i(R_1) - i(R_2) > C$

$$i(R_1) = \sum_{(x,y) \in R_1} i(x, y)$$

$$i(R_2) = \sum_{(x,y) \in R_2} i(x, y)$$

C is a constant threshold.

2-rectangle filters



3-rectangle filters

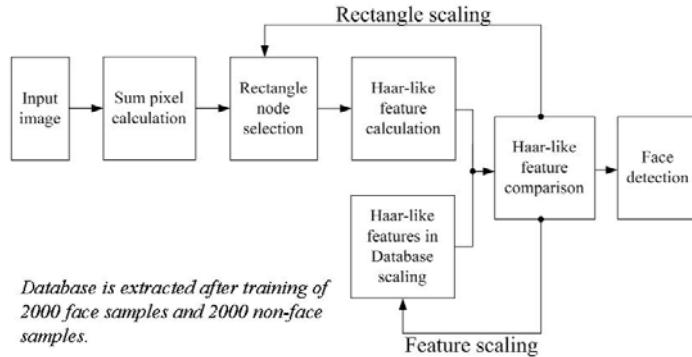


4-rectangle filter



$i(x, y)$ is pixel luminance value.

Haar-like Face detection Algorithm



Rotated face detection issue

Rotate the input image by $\alpha=0, 30, 60\dots$ and 330 degrees

$$(x, y)=(r\cos \theta, r\sin \theta)$$

$$(x', y')=(r\cos(\theta+\alpha), r\sin(\theta+\alpha))$$

Based on the correlation of the coordinates, we need only to calculate 4 values:

$$0.866x-0.5y, 0.866y+0.5x, 0.5x-0.866y, 0.5y+0.866x$$

Because input image shape is symmetric, we only calculate 1/4 of all pixels for each rotation.



Haar-like face detection examples



Haar-like face recognition example

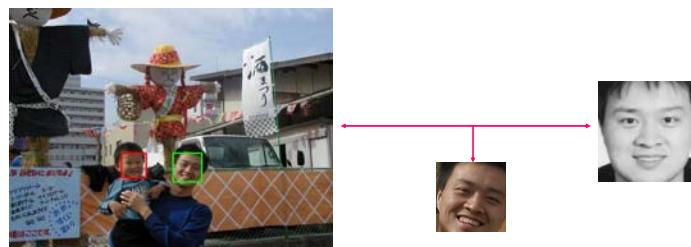
Definition of Face Recognition:

- biometric identification by scanning a person's face and matching it against a library of known faces.

Training data:

Positive samples: one person's faces under different conditions

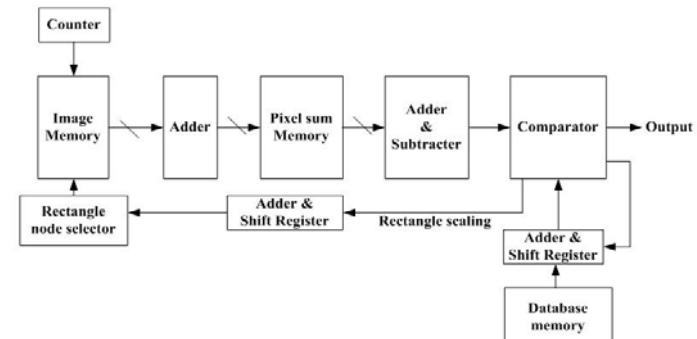
Negative samples: other persons' faces



Conclusions

- A Unified face detection and recognition system for in-plane rotated faces based on Haar-like features is proposed.
- Illumination improvement for face detection by use of histogram normalization method.
- A training detection rate of 95% with false positive rate of 0.1% is achieved. Recognition rate of 75% is achieved.
- The execution time of the whole system is shorter than 0.7 seconds for a QVGA size image on a 2.8GHz Pentium-4 PC
- A hardware structure of this system is described.

Hardware Architecture of unified face detection and recognition system



Scaling factor=1.125,

Scaling operation is realized with an Adder and a Shift Register.

The face detection and recognition system based on Haar-like features can be implemented into hardware with simple arithmetic units, even without multipliers!

Future work

Solving Convergence problem for face recognition with Haar-like method.

Adding self-learning function to face detector and recognizer.

Hardware Realization of motion face recognition system.