이학석사 학위논문

Face detection using AdaBoost and template matching

(AdaBoost와 템플릿 매칭을 이용한 얼굴 검출)

2014년 2월

서울대학교 대학원

수리과학부

방 미 림

Face detection using AdaBoost and template matching

(AdaBoost와 템플릿 매칭을 이용한 얼굴 검출)

지도교수 강 명 주

이 논문을 이학석사 학위논문으로 제출함

2013년 12월

서울대학교 대학원

수리과학부

방 미 림

방 미 림의 이학석사 학위논문을 인준함

2013년 12월

위 원 장 ______________________ (인)

부 위 원 장 ______________________ (인)

위 원 ______________________ (인)

Face detection using AdaBoost and template matching

A dissertation
submitted in partial fulfillment
of the requirements for the degree of
Master of Science
to the faculty of the Graduate School of
Seoul National University

by

Mirim Bang

Dissertation Director : Professor Myungjoo Kang

Department of Mathematical Sciences
Seoul National University

February 2014

Abstract

This work contains the existing face detection method on gray-scale static image. Discrete-AdaBoost algorithm which is proposed by Viola-Jones is one of the typical real-time face detection methods with superior speed and performance. Then Maio and Maltoni's paper is reviewed, and it suggested the template matching algorithm which enhances the face detection rate.

The Viola-Jones face detector contains three main ideas that make it possible to build a successful face detector that can run in real time: the integral image, classifier learning with AdaBoost, and the attentional cascade structure. Maio and Maltoni's method approximately detects the image positions where the probability of finding a face is high; next, the location accuracy of the candidate faces is improved and their existence is verified.

Key words: Face detection, Viola-Jones, Template Matching, AdaBoost, Directional image, Generalized Hough transform
Student Number: 2011-23205

Contents

List of Figures

Chapter 1

Introduction

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 [1]. Face detection has received substantial attention in pattern recognition, video surveillance, human computer interface, biometric systems. There are several factors which contribute to making this work complicated, especially in the case of applications requiring to operate on gray-scale static image. The difficulty associated with face detection can be attributed to many variations in scale (different distances between the human and the camera), size, location, pose and head rotation, facial expression, lighting conditions, complex backgrounds, etc.

1.1 Previous work

There have been hundreds of reported approaches to face detection. Yang [1] grouped the various methods into four categories:

• Knowledge-based methods : These rule-based methods encode human knowledge of what constitutes a typical face. Usually, the rules capture the relationships between facial features. These methods are designed mainly for face localization.

• Feature invariant approaches : These algorithms aim to find structural features that exist even when the pose, viewpoint, or lighting conditions vary, and then use the these to locate faces. These methods are designed mainly for face localization.

• Template matching methods : Several standard patterns of a face are stored to describe the face as a whole or the facial features separately. The correlations between an input image and the stored patterns are computed for detection. These methods have been used for both face localization and detection.

• Appearance-based methods : In contrast to template matching, the models (or templates) are learned from a set of training images which should capture the representative variability of facial appearance. These learned models are then used for detection. These methods are designed mainly for face detection.

1.2 Overview

The Viola-Jones algorithm

The Viola-Jones face detector contains three main ideas that make it possible to build a successful face detector that can run in real time: the integral image, classifier learning with AdaBoost, and the attentional cascade structure.

In chapter 2.1 the integral image is introduced which could be used to compute simple Haar-like features, and chapter 2.2 explains AdaBoost learning algorithm selecting efficient classifier. Chapter 2.3 contains the attentional cascade which combines classifiers and in chapter 2.4 we report the experimental results.

The Maio and Maltoni algorithm

This technique is based on a location information which begins by approximately detecting the candidate positions where the probability to finding a face is high and then for each of them enhances the location exactness and verifies the existence of a real face.

In chapter 3.1 the directional image is defined and some comments about its computation are reported. Chapter 3.2 describes the approximate location which is based on the search of elliptical blobs in the directional image by means of the generalized Hough transform(module AL). In chapter 3.3 explains the fine location and face verification using dynamic-mask(module FLFV) and in chapter 3.4 we discuss how to practically combine AL and FLVF. Chapter 3.5 reports the results of our experimentation over a 40 image database.

Finally, in chapter 4, we present our conclusions.

Chapter 2

The Viola-Jones algorithm

2.1 The integral image

Integral image is an algorithm for fast evaluating the sum of values in a rectangle subset of a grid. It is very similar to the summend area table which was first introduced to the computer graphics field by Crow [5] for use in mipmaps. Viola and Jones applied the integral image for rapid computation of Haar-like features.

The integral image is constructed as follows:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

where $ii(x, y)$ is the integral image at pixel location (x, y) and $i(x', y')$ is the original image. Using the integral image to compute the sum of any rectangular area is extremely efficient, as shown in figure 2.1. The sum of pixels in rectangle region ABCD can be calculated as:

$$\sum_{(x,y) \in ABCD} i(x, y) = ii(D) + ii(A) - ii(B) - ii(C)$$

which only requires four array references.

The integral image can be used to compute simple Haar-like rectangu-

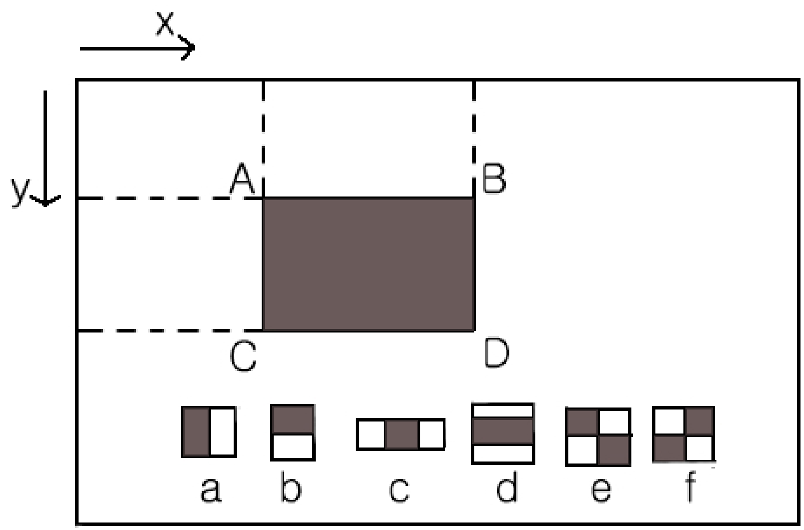

Figure 2.1: Illustration of the integral image and Haar-like rectangle features (a-f).

lar features, as shown in figure 2.1 (a-f). The features are defined as the (weighted) intensity difference between two to four rectangles.

2.2 AdaBoost learning

A simple and efficient classifier which is built using the AdaBoost learning algorithm (Freund and Schapire, 1995) to select a small number of important features from a very large set of potential features. In any image sub-window, there are so many Haar-like features, far larger than the number of pixels. In order to ensure fast classification, the learning process must leave out a most of the available features, and concentrate on a small set of important features. In Viola-Jones paper, feature selection is performed using the AdaBoost learning algorithm by restricting on each weak classifier to depend on only a single feature. So each step of the boosting process selects a new weak classifier and can be viewed as a feature selection process. AdaBoost provides an effective learning algorithm and strong bounds on generalization performance (Schapire et al., 1998).

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:

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

The boosting algorithm for learning. 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.

- Given example images $(x_1, y_1), ..., (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, ...,$ T:

1. Normalize the weights, $w_{t,i} \leftarrow \dfrac{w_{t,i}}{\sum n_{j=1} w_{t,j}}$
2. Select the best weak classifier with respect to the weighted error

$$\epsilon_t = min_{f,p,\theta} \sum_i w_i |h(x_i, f, p, \theta) - y_i|$$

3. Define $h_t(x) = h(x, f_t, p_t, \theta_t)$ where $f_t, p_t,$, and θ_t are the minimizers of ϵ_t .
4. Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \dfrac{\epsilon_t}{1 - \epsilon_t}$.

- The final strong classifier is:

$$C(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_t = log\frac{1}{\beta_t}$.

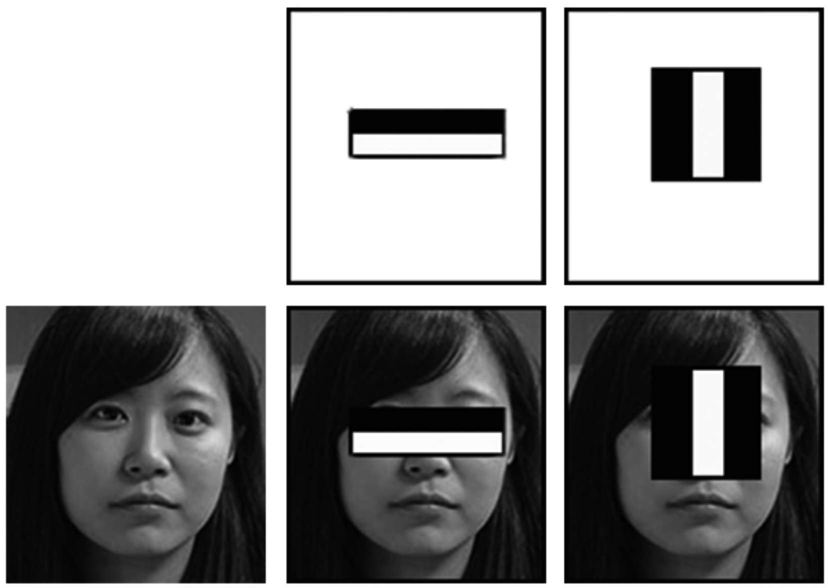

Figure 2.2: The two features selected by AdaBoost. They are overlayed on a training face in the bottom row. The first feature measures the difference in intensity between the region of the eyes and a region across the upper cheeks. The second feature compares the intensities in the eye regions to the intensity across the bridge of the nose.

The AdaBoost learning algorithm:

For each feature, the feature values of all the images are sorted in a list. Four sums are maintained and evaluated: the total sum of positive image weights T^+ , the total sum of negative image weights T^-, the sum of positive weights below the current image S^+ and the sum of negative weights below the current image S^-. The error for a threshold which splits the range between the current and previous image in the sorted list is: $e = min(S^+ + (T^- - S^-), S^- + (T^+ - S^+))$. These sums are easily updated as the search proceeds.

2.3 The attentional cascade structure

The attentional cascade is an important component in the Viola-Jones algorithm. The key insight is that smaller, and thus more efficient, boosted classifiers can be built which reject a lot of the negative sub-windows while keeping almost all the positive samples. Consequently, most of the sub-windows will be rejected in forward stages of the detector, making the detection process extremely efficient.

The overall process of classifying a sub-window thus forms a degenerate decision tree, which was called a cascade [6]. As shown in figure 2.3, a series of classifiers are employed to every sub-window. The initial classifier removes a lot of negative examples with very little processing. Following layers remove additional negatives but need extra computation. After several stages of processing the number of sub-windows have been eliminated radically. Further processing can take any form such as additional stages of the cascade (as in our detection system) or an alternative detection system.

The cascade structure also has an influence on the training process. Face detection is a rare event detection task. Therefore, there are usually most of

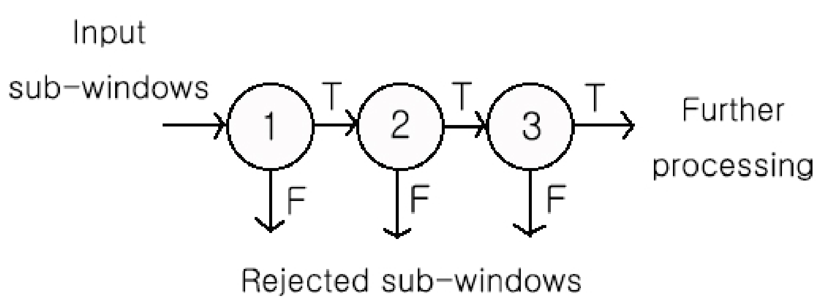

Figure 2.3: The attentional cascade.

negative examples needed in order to train a high performance face detector. Viola and Jones [6] used a bootstrap process to handle the huge amount of negative training examples. In other words, at each node, a threshold was manually chosen, and the partial classifier was used to scan the negative example set to find more accepted negative examples for the training of the next node. In addition, each node is trained independently, as if the previous nodes does not exist. One argument behind such a process is to force the addition of some nonlinearity in the training process, which could improve the overall performance.

Training algorithm for building a cascaded detector:

- User selects values for f, the maximum acceptable false positive rate per layer and d, the minimum acceptable detection rate per layer.
- User selects target overall false positive rate, F_{target}.
- P = set of positive examples
- N = set of negative examples
- $F_0 = 1.0; D_0 = 1.0$
- $i = 0$
- while $F_i > F_{target}$

– $i \leftarrow i + 1$

– $n_i = 0; F_i = F_{i-1}$

– while $F_i > f \times F_{i-1}$

  ∗ $n_i \leftarrow n_i + 1$

  ∗ Use P and N to train a classifier with ni features using AdaBoost

  ∗ Evaluate current cascaded classifier on validation set to determine F_i and D_i

  ∗ Decrease threshold for the ith classifier until the current cascaded classifier has a detection rate of at least $d \times D_{i-1}$(this also affects F_i)

– $N \leftarrow \emptyset$

– If $F_i > F_{target}$ then evaluate the current cascaded detector on the set of non-face images and put any false detections into the set N

2.4 Experimental results

We quote the result in Viola and Jones [2].

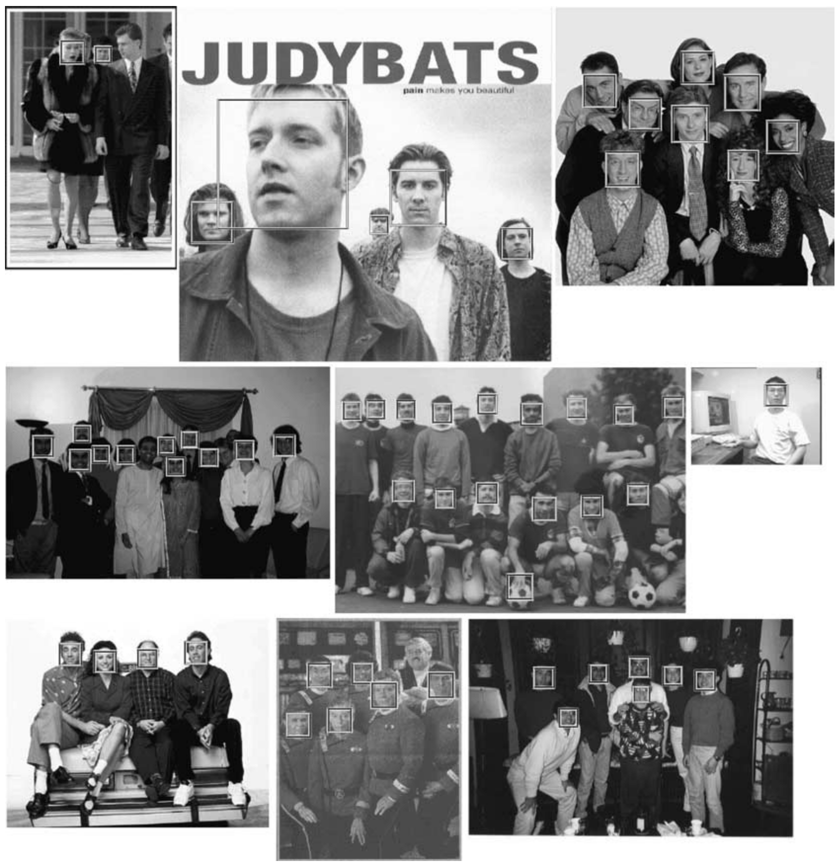

Figure 2.4: Output of Viola-Jones face detector on test images in the MIT+CMU test set (Viola, Jones, 2004).

Chapter 3

The Maio and Maltoni algorithm

3.1 Directional image

A directional image is a matrix whose elements represent the local direction of the image edges with the grid nodes. Each element is a vector lying on xy plane. The vector direction denotes the tangent to the image edges in a neighborhood of the point, and its modulus is decided as a weighted sum of the contrast and the consistency. We compute the directional image through the method proposed by Donahue and Roklin [11]. Each directional image element is calculated over a local window where a gradient-type operator is employed to extract several directional estimates (2d sub-vectors), which are averaged by least-squares minimization to control noise.

3.2 Approximate location

This section explains the search of candidate face positions in the image. When a face exists in an image the corresponding directional image region is marked by vectors showing an elliptical blob. From this idea, to

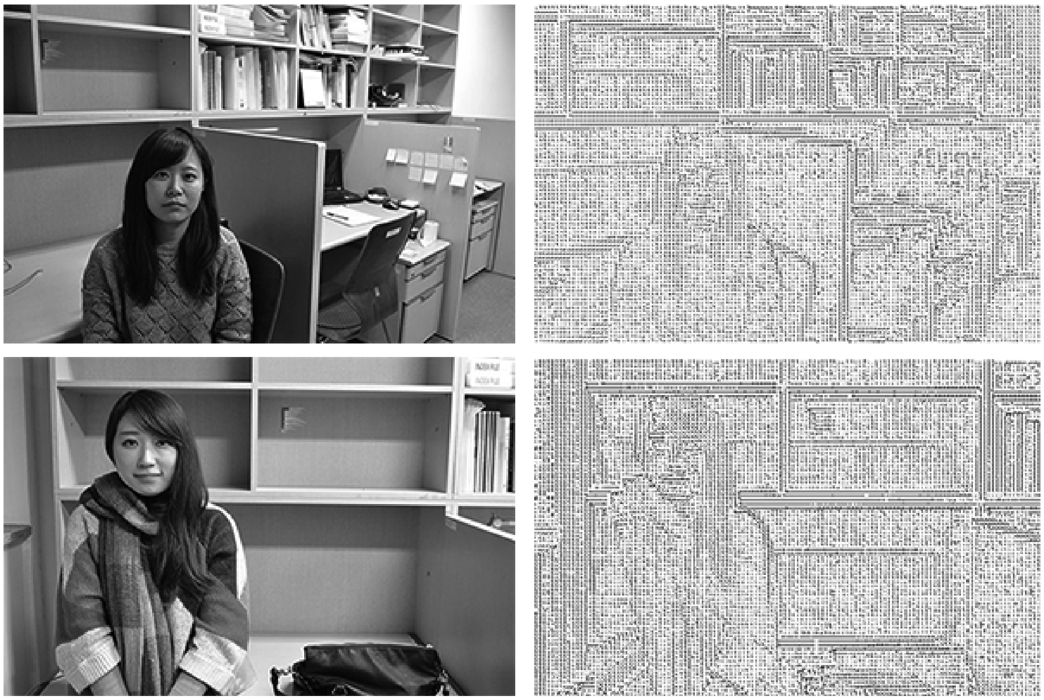

Figure 3.1: Two images and the corresponding directional images. The vector lengths are proportional to their moduli.

detect approximately is based on the search for ellipses on the directional image. We take the generalized Hough transform [13] which could detect the approximate position of all the ellipses in a definite range of variation allowed rotation changes. The main concept is we use an elliptical annulus C as template with a generalized Hough transform.

Let a and b be the lengths of the semi-axes of an ellipse used as reference, and let ρ_r and ρ_e be respectively the reduction and expansion coefficients defining the scale range (and hence the elliptical annulus C): $a_{min} = \rho_r \cdot a$, $b_{min} = \rho_r \cdot b$, $a_{max} = \rho_e \cdot a$, $b_{max} = \rho_e \cdot b$ (figure 3.1). Let D be the directional image and let A be the accumulator array, then the algorithm can be sketched as:

Reset A
$\forall$ vector d $\in$ D

$\{[x_0, y_0] = origin(\text{d});$
$\quad \varphi = direction(\text{d});$
$\quad \sigma = modulus(\text{d});$
$\quad \text{T} = current\_template([x_0, y_0], \varphi);$
$\quad \forall \text{ pixel } [x, y] \in \text{T}$
$\quad \{ \text{ A}[x, y] = \text{A}[x, y] + \sigma \cdot weight_{\text{T}}([x, y]); \}$
$\}$

The high-score A cells are good candidates for ellipse centers.

the *direction*(d) and the *modulus*(d) of the directional elements d are calculated off-line as described in Donahue and Roklin [11].

current_template($[x_0, y_0], \varphi$) determines the current template T as a function of the direction φ of the vector centered in $[x_0, y_0]$. The points $[x_1, y_1]$ and $[x_2, y_2]$ in figure 3.2 are the only two points where an ellipse tangent to d in $[x_0, y_0]$ with semi-axes a, b could be centered. Since we are interested in all the ellipses whose semi-axes are in the range $[a_{min} \dots a_{max}, b_{min} \dots b_{max}]$, we must take into account all the points lying on the two segments determined by the intersection between the straight line defined by $[x_1, y_1]$, $[x_2, y_2]$ and the elliptical annulus C. Finally, if we assume a maximum angular variation θ on the directional information, the geometric locus T of the possible centers becomes:

$$\text{T}=\left\{ [x, y] \middle| \rho_r^2 \leq \left(\frac{x - x_0}{a}\right)^2 + \left(\frac{y - y_0}{b}\right)^2 \leq \rho_e^2, \Delta\left(arctg\left(\frac{y - y_0}{x - x_0}\right), \phi\right) \leq \frac{\theta}{2} \right\}$$

where Δ angle $(\alpha, \beta) : [-90°, +90°] \times [-90°, +90°] \to [0°, 90°]$ returns the smaller angle determined by the directions α, β the angle ϕ can be computed as a function of φ by deriving the tangent vector expression by the parametric equation of the ellipse:

$$\phi = arctg\left(-\frac{b}{a \cdot tg(\varphi)}\right)$$

The function $weight_T$: T $\to [0, 1]$ associates, to each point $[x, y]$ of T, a weight which decreases with the angular distance between the straight line defined by $[x, y]$, $[x_0, y_0]$ and the direction φ:

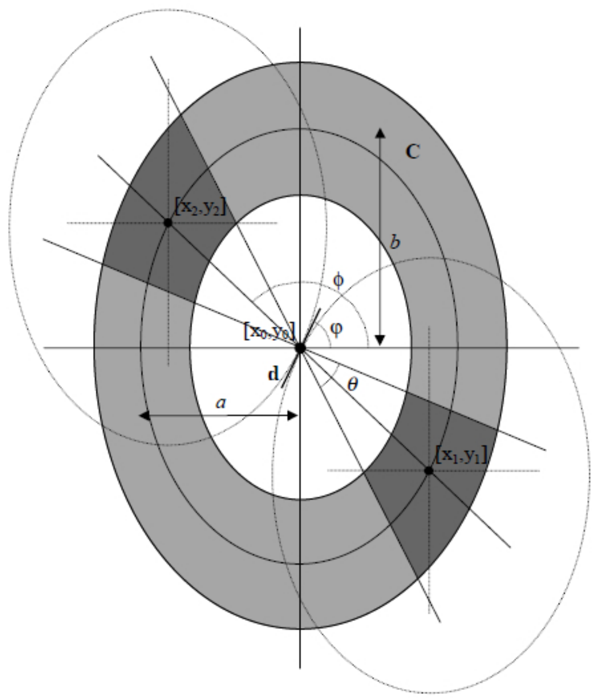

Figure 3.2: The template T (in dark gray) is constituted by those points which are possible centers of ellipses capable of originating in $[x_0, y_0]$ a vector d with direction φ (Maio, Maltoni, 2000).

$$weight_{\mathrm{T}}(x,y) = 1 - \frac{2 \cdot \Delta angle\big(arctg\left(\frac{y-y_0}{x-x_0}\right), \phi\big)}{\theta}$$

An efficient implementation of the approximate location has been obtained by adopting the following tricks:

- The grid which defines the accumulator array A has been set same to that defining D. Especially the granularity used is 7×7 pixels, that is, the directional image elements are computed every 7×7 pixels. Thus, the size of both the directional image and the accumulator array associated to an X×Y pixel image is $[X/7] \times [Y/7]$.
- The directions of the elements in D have been discretized (256 values).
- The templates T have been pre-computed (in our experiments we chose

$a = 34, b = 45, \rho_r = 0.6, \rho_e = 1.2, \theta = 30°$); by using relative coordinates according to the ellipse center, the number of different templates corresponds to the number of different directions.

- The algorithm has been performed in integer arithmetic.

3.3 Fine location and face verification

Actually, since particular physiognomies, hairstyles or illuminations sometimes make the face only coarsely elliptical, an accurate face location cannot be acquired by using an elliptical template all the time. Otherwise, a reliable face verification method strongly depends on the face location exactness. Consequently, in order to discuss a greater robustness to this chapter, we developed a global approach which attempts to meet the two different aims at the same time.

By starting from a candidate position resulting from AL, the face is locally searched in a small portion of the directional image; to this purpose, we apply a mask $\mathcal{F}$ describing the global aspect of a human face. The local search is achieved by way of an orientation-based correlation between $\mathcal{F}$ and the D portion of interest. In fact, after the AL step, D is locally refined in a neighborhood of the candidate position with respect to a multiresolution approach (in the following, D' denotes the new directional image portion).

The mask $\mathcal{F}$ is defined as a set of directional elements n_i $i = 1 \ldots n_{max}$ each of which is characterized by an origin, an unoriented direction (in the range $[-90°, 90°]$) and a modulus. A different number n_{max} of elements is created varying a and b, since the template is parametrically defined according to the sizes a and b, whereas the grid granularity is fixed.

All the elements within the mouth, eyes and eyebrows regions have horizontal direction, the nose elements have vertical direction and each element belonging to the border region has the direction of the tangent (in that point) to the external ellipse. By discretizing a and b in the range $[a_{min} \ldots a_{max}, b_{min} \ldots b_{max}]$ it is possible to pre-compute a set $\mathcal{F} = \{F_1, F_2, \ldots F_m\}$ of static masks (figure 3.4), where the element positions are given in relative coordinates according to the mask center.

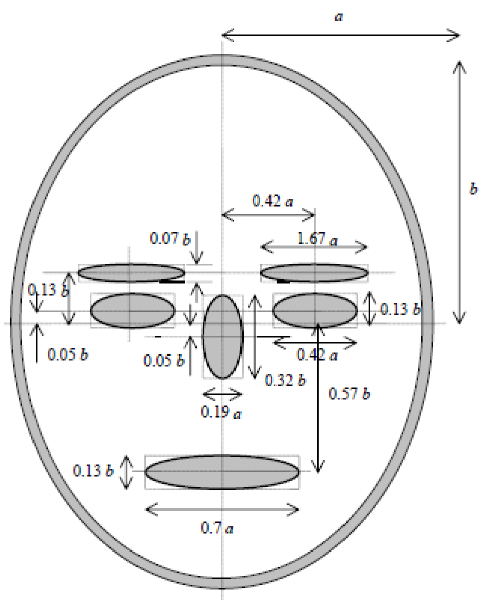

Figure 3.3: The template used for the construction of the mask $\mathcal{F}$ used by the FLVF. (Maio, Maltoni, 2000).

The masks $\mathcal{F}$ in allow a correlation degree at each position in D' to be efficiently computed. Matching directional elements requires an ad-hoc correlation (or distance) operator capable of dealing with the discontinuity ($-90° \leftrightarrow 90°$) in the definition of directions (e.g. Crouzil [15]). Good results have been obtained in this work with a distance operator defined as an average sum of direction-difference absolute values. The distance between the mask $F_i \in \mathcal{F}$ and the portion of D' centered in $[x, y]$ is computed as:

Distance(F_i, D',$[x, y]$)
{ Distance = 0 ; ModuliSum = 0 ;
  $\forall$ element n $\in F_i$
  {$[x_n, y_n] = origin$(n) ;
    Let d $\in$ D' be the element with origin $[x, y] + [x_n, y_n]$;
    $\varphi_n = direction$(n); $\sigma_n = modulus$(n);
    $\varphi_d = direction$(d); $\sigma_d = modulus$(d);
    Distance = Distance + $\sigma_n \cdot \sigma_d \cdot \Delta angle(\varphi_n, \varphi_d)$;
    ModuliSum = ModuliSum + $\sigma_n \cdot \sigma_d$;

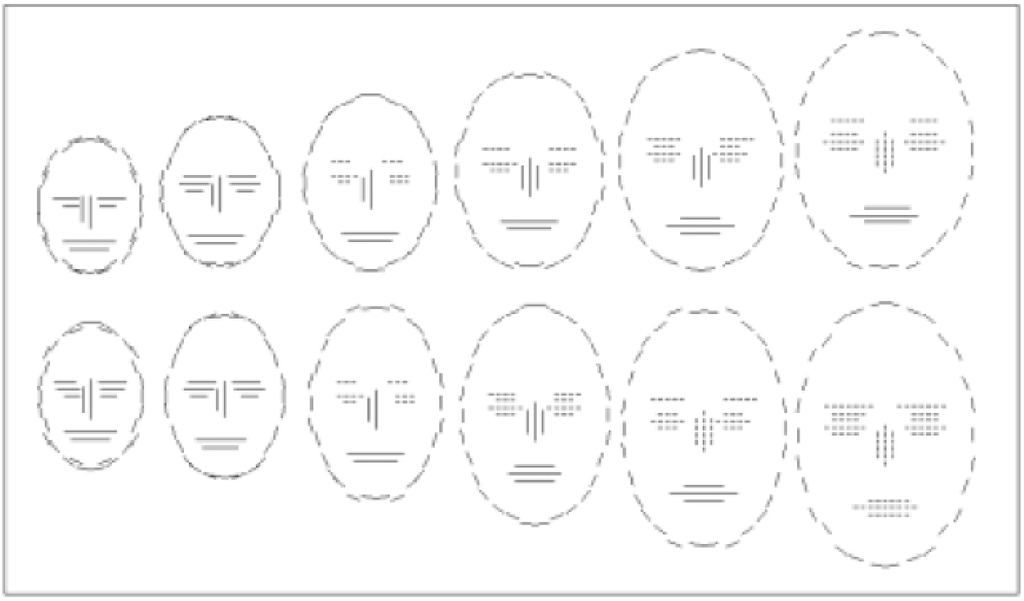

Figure 3.4: A graphical representation of the 12 masks constituting the set $\mathcal{F}$ (m=12) used in our experimentation (the length of the mask elements is proportional to their modulus) (Maio, Maltoni, 2000).

```
        }
    Distance = Distance / ModuliSum ;
}
```

Let $[x_0, y_0]$ be a candidate position as resulting from the module AL, then FLFV determines the face position $[x^*, y^*]$ and the semi-axes a^* and b^* by minimizing the Distance function over a discrete state space:

$$\mathrm{d}_{min} = \min_{\substack{F_i \in \mathcal{F} \\ x \in [x_0 - \Delta x, x_0 + \Delta x] \\ y \in [y_0 - \Delta y, y_0 + \Delta y]}} \{Distance(F_i, \mathrm{D'}, [x, y])\}$$

where Δx and Δy define a neighborhood of $[x_0, y_0]$, and a^*, b^* coincide with the semi-axes of the best fitting mask. In our simulations we set $\Delta x = \Delta y = 4$ and therefore the total number of states is $12 \cdot 9 \cdot 9 = 972$. Once the best fitting position has been determined, by comparing the distance

d_{min} with a pre-fixed threshold, the face verification sub-task can be simply achieved.

Figure 3.5 shows the results of the intermediate steps of a face location example.

Figure 3.5: The results of the intermediate steps in a face location example: directional image computation, generalized Hough transform and selection of the best candidate position, directional image local refining, determination of the best fitting mask and position.

3.4 Combining AL and FLFV

There are various ways of adapting and combining AL and FLFV modules. In particular, the algorithm finds for just one face in the image; it returns the face position $[x_f, y_f]$ and sizes a_f, b_f, in case of detection, and null otherwise. A pseudo-code version of the whole face detection method is reported:

Compute directional image D ;
Perform generalized Hough transform ;
$[x_0, y_0]$ = get first candidate position on A ;
$\mathrm{d}_f = \infty$;
While ($[x_0, y_0]$ is not NULL) and $\mathrm{d}_f > \mathrm{T}_1$)
{ D' = Refines the directional image in a neighborhood of $[x_0, y_0]$;
$\mathrm{d}_{min} = \min\limits_{\substack{F_i \in \mathcal{F} \\ x \in [x_0 - \Delta x, x_0 + \Delta x] \\ y \in [y_0 - \Delta y, y_0 + \Delta y]}} \{Distance(F_i, \mathrm{D'}, [x, y])\}$
if ($\mathrm{d}_{min} < \mathrm{d}_f$) { $\mathrm{d}_f = \mathrm{d}_{min}; [x_f, y_f] = [x^*, y^*]; a_f = a^*; b_f = b^*;$ }
$[x_0, y_0]$ = get next candidate position on A ;
}
if ($\mathrm{d}_f < \mathrm{T}_2$) { return $[x_f, y_f], a_f, b_f;$ }
else return null ;

The algorithm uses two different thresholds T_1 and T_2 ($\mathrm{T}_1 < \mathrm{T}_2$) : when a good match has been found, T_1 stops the iterative search process ($\mathrm{d}_f \leq \mathrm{T}_1$). After all the candidate positions have been examined, the iterative process is disturbed, and if the smallest distance computed is less than T_2, a valid face is returned. Using a pair of thresholds allows many candidate positions to be examined in case the current one is not sufficiently dependable, and at the same time if no more-likely candidates have been found, the impermanent rejected faces can be reconsidered.

3.5 Experimental results

In the paper, experimental results have been produced on a database of 70 images, and we use 40 images. All the images contains one human face and they are in offices and laboratories. The faces were required to gaze the camera.

The approximate location

The approximate location module performed the correct face position as global maximum of A (first candidate position) in 36 cases . In the remaining 4 images the face position was detected as the second or third candidate position. Figure 3.6 shows some images and the corresponding transforms, where the global maximum determined by the elliptical face shape is well visible. Figure 3.7 reports an example where the transform global maximum does not coincide with the face position. This false alarm, as can be seen in figure 3.8 (first row, first column), is removed by the module FLFV.

The whole approach : AL+FVFL

The total face detection algorithm has been applied to the 40 images: in 40 cases was correctly detected. 12 false alarms (in 4 images) generated by the AL were correctly discarded by FLFV.

Figure 3.8 shows some examples; the thin-border ellipses denote the AL outputs; the false alarms produced by AL can be easily noted since no masks are associated to the corresponding ellipses.

Figure 3.6: The approximate location module(AL).

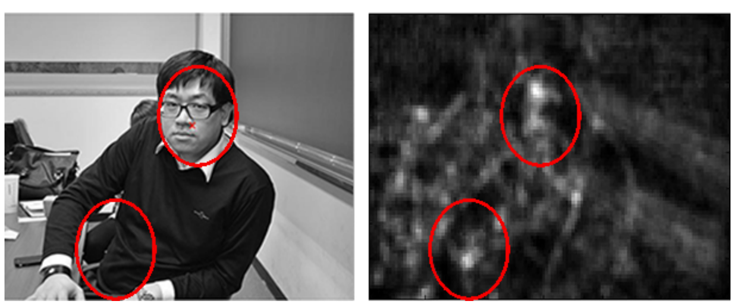

Figure 3.7: False alarms.

Figure 3.8: Some examples of final result.

Chapter 4

Conclusion

Viola and Jones have presented a method for face detection which reduces computation time while performing high detection accuracy. This detector can use for other objects, such as pedestrians or automobiles.

A new algorithm using the integral image computes efficiently a large set of image features. In order to perform true scale invariance, almost all face detection systems must compute on multiple image scales. However, using the integral image, face detection is finished in almost the same time as it takes for an image pyramid to be computed.

Using AdaBoost for feature selection is a simple and efficient classifier built from computationally efficient features. Given an effective tool for feature selection, the system designer is free to define a very huge and very complex set of features as input for the learning process. The resulting classifier is nevertheless computationally efficient, since only a small number of features need to be evaluated during run time.

A technique for making a cascade of classifiers extremely reduces computation time while enhancing detection accuracy. Forward stages of the cascade are devised to reject a majority of the image in order to focus subsequent processing.

Maio and Maltoni propose a two-stage modules to face location on gray-scale static images with complex backgrounds. Both the modules perform on the elements constituting the directional image, which has been proved to be very effective in providing reliable information even in the presence of critical illumination and semidarkness.

The approximate location module searches for the most likely positions in the image by means of a certain implementation of the generalized Hough transform. Great computational saving is obtained with respect to a correlation-based technique. In fact, in the first case just one directional image scan allows the "hot" positions to be extracted, whereas in the second some elliptic templates (resembling the different elliptical shapes and sizes) should be moved everywhere on the directional image to detect the high correlation points.

The fine location and face verification module examines small refined portions of directional image trying to detect the precise position and size of a face together with a reliance value about its presence. Since the masks are defined in terms of directions, their matching is robust and very little biased by light conditioning, this is performed by means of a set of masks resembling the human face stereotype.

Bibliography

[1] M.-H. Yang, D. J. Kriegman, and N. Ahuja, *Detecting faces in images: A survey*, IEEE Trans. on PAMI, 24(1):34–58 (2002)

[2] P. Viola and M. Jones, *Robust Real-Time Face Detection* International Conference on Computer Vision, **57**, no. 2, 137-154 (2004).

[3] D. Maio and D. Maltoni, *Real-Time Face Location on Gray-Scale Static Images*, Pattern Recognition, **33**, no. 9, 1525-1539 (2000).

[4] C. Zhang and Z. Zhang, *A survey of recent advances in face detection*, Technical report, Microsoft Research (2010).

[5] F. Crow, *Summed-area tables for texture mapping* In Proc. of SIGGRAPH, **18**, 207-212 (1984).

[6] P. Viola and M. Jones, *Rapid object detection using a boosted cascade of simple features* In Proc. of CVPR (2001).

[7] P. Viola and M. Jones, *Robust real-time object detection*, International Conference on Computer Vision, 12-14 (2001).

[8] R. Chellappa, S. Sirohey, C.L. Wilson and C.S. Barnes, *Human and Machine Recognition of Faces: A Survey*, Technical report CS-TR-3339, Computer Vision Laboratory, University of Maryland (1994).

[9] G. Yang and T.S. Huang, *Human face detection in a complex background*, Pattern Recognition, **27**, 53-63 (1994).

[10] S. A. Sirohey, *Human face segmentation and identification*, Technical report CAR-TR-695, Center for Automation Research, University of Mariland (1993).

[11] M. J. Donahue and S. I. Rokhlin, *On the use of Level Curves in Image Analysis*, Image Understanding, **57**, 185-203 (1993).

[12] P. Seitz and M. Bichsel, *The Digital Doorkeeper - Automatic Face Recognition with the Computer*, proc. of 25th IEEE Carnahan Conference on Security Technology (1991).

[13] D. H. Ballard *Generalizing the Hough transform to detect arbitrary shapes*, Pattern Recognition, **3**, 110-122 (1981).

[14] R. Brunelli and T. Poggio, *Face Recognition: Features versus Templates*, IEEE Tr. on PAMI, **15**, 1042-1052 (1993).

[15] A. Crouzil, L. Massip-Pailhes and S. Castan, *A New Correlation Criterion Based on Gradient Fields Similarity*, proc. 13th ICPR, v. A, 632-636 (1996).

[16] C. Zhang and Z. Zhang, *A Survey of Recent Advances in Face Detection*, Technical report MSR-TR-2010-66, 185-203 (2000).

국문초록

본 논문은 기존의 그레이 스케일의 정지된 영상에서 얼굴을 검출하는 방법으로 구성되어 있다. Viola-Jones가 제안한 AdaBoost를 이용한 얼굴 검출 방법은 뛰어난 속도와 성능으로 실시간 얼굴 검출에 사용되는 대표적인 알고리즘 중의 하나이다. 그리고 얼굴 검출 방법의 또 다른 방법으로 Maio와 Maltoni의 논문에서 제공하는 템플릿 매칭 방법에 대해 알아본다.

Viola-Jones의 얼굴 검출기는 세 가지 아이디어를 포함하는데, 누적 영상, AdaBoost를 이용한 학습 분류기, 그리고 cascade 구조이다. Maio 와 Maltoni의 방법은 근사적으로 얼굴을 발견할 확률이 높은 이미지의 위치를 감지한다. 그 다음 후보 얼굴의 위치의 정확성이 향상되고 존재가 확인된다.

주요어휘: 얼굴 검출, Viola-Jones, 템플릿 매칭, AdaBoost, 방향 이미지, 일반화된 허프 변환
학번: 2011-23205