A Real-time Face Detector*

SI-CHENG ZHANG
Center for Media Technology
City University of Hong Kong
Hong Kong, China
rockie.cheung@student.cityu.edu.hk

ZHI-QIANG LIU
Center for Media Technology
City University of Hong Kong
Hong Kong, China
smzliu@cityu.edu.hk

Abstract – *In this paper, we present a real-time face detector using a hybrid template that characterizes face-like shapes and face texture patterns. The face detector consists of two main stages: (1) face shape detection based on a modi ed fast ellipse detection algorithm and (2) face candidate veri cation in which we de ne a set of face texture patterns and extend a texture description method to verifying the presence of a face in the face shape candidate region. We present our experimental results which shows that the proposed approach is able to achieve high accuracy with a real-time performance.*

Keywords: Face Detection, Real-time Performance, Ellipse-like, Face Texture Pattern

1 Introduction

For the last few decades, numerous face detection approaches have been proposed. Recently, Yang et al [5] grouped the existing face detection algorithms into four catalogues: Knowledge-based, Feature invariant, Template matching and Appearance-based approaches. Feature invariant approaches are the most popular ones, including skin color, face size, face shape, face texture and etc.

Among the facial features, the shape of face/head is the most distinct biologic characteristic. Human faces have ellipse-like shapes, which provide a basic and stable evidence for face location. Many researchers have applied ellipse detection techniques to finding face candidates, e.g., polygonal approximation [4]. A common issue is that they may suffer from computational inefficiency. Face texture is of interest for rule-based face detection, typically mosaic images [2]. However, it is computationally costly.

In this paper, we present a real-time algorithm for face detection based on robust ellipse-like shape detection and rapid texture pattern matching. This method is able to handle semi-frontal faces under various image conditions (noise and lighting). We have carried out extensive experiments using standard face databases to show its efficiency and accuracy.

*0-7803-8566-7/04/$20.00 © 2004 IEEE.

2 Face Template

2.1 Review

The face template plays an important role in face detection, especially knowledge-based and template-matching approaches. Different sub-models such as edge, color, mesh and texture are also developed for specific purposes [2, 4]. A common issue of a single template is the balance of detection accuracy and efficiency. In our face detector, we uses a hybrid face template consisting of an edge sub-model and a texture sub-model. This is motivated by the fact that edges and textures are the most salient features to the human visual system [5].

2.2 Face Model

The face model outlines the facial components of a generic face. There are two kinds of facial components, namely internal components, including eyes/eyebrows, nose/nostrils, mouth/lips, cheeks/forehead, and external ones, including face/head outline, hairs, ears etc.

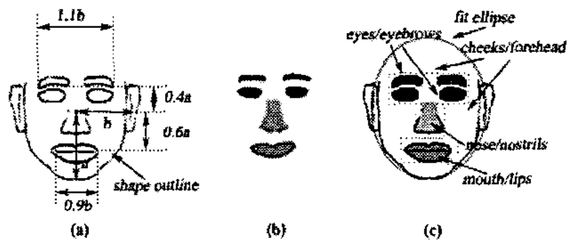


Figure 1: Face template consisting of two sub-models and facial components: (a) face edge sub-model with geometric relationships; (b) face texture sub-model with relative brightness difference; and (c) facial components and the associated rectangular regions.

As shown in Fig. 1(a), the edge sub-model primarily consists of two parts, the outline of the face/head shape and shapes of the internal facial components. Since the boundary of objects may be blurred, a texture sub-model is necessary for supporting purpose.

Fig. 1(b) shows the texture sub-model for gathering the evidence of facial features. The texture here is denoted as pixel density in gray scale. Generally, human face has an apparent natural contrast in face texture, which enables the detection of face texture pattern.

The two sub-models are merged into a combined face template. Fig. 1(c) shows a 2-D template with descriptive information of edge and texture in frontal upright view. We consider two aspects: constrained face poses and various face shapes. Here we restrict the detectable face poses within a reasonable range, usually $\pm 45°$ rolls (rotation in x-y plane), $\pm 30°$ tilts (rotation in y-z plane) and $\pm 30°$ in slants (rotation in x-z plane).

2.3 Properties for Face Shape (PFS)

In face detection, a face-like shape may indicate possible existence of a face in the scene. In order to handle various deformations of ellipse-like shapes, we adopt a 2-D displacement filed [1] to deform the edge sub-model. The operations result in a large number of deformed face instances, from which we summarize the general properties of face shapes (PFS):

PFS(1) The face shape is in principle an upright ellipse-like shape, with reasonable rotations;
PFS(2) The face shape has a limited variation range of semi-axis ratio;
PFS(3) The edge convexity changes more sharply at the bottom than on the left or right sides;
PFS(4) The edge at the top usually contains fragmented curves while that at the bottom is usually continuous, and the elliptic feature may be severely damaged when the face has a large slant (e.g., profile views).

2.4 Properties for Face Texture (PFT)

In practice, we use intensity contrast to describe the texture feature. For compactness, we use rectangle regions when comparing the intensity difference between facial textures. Some knowledge-based rules are derived from the texture sub-model and the edge sub-models:

PFT(1) The eye region is darker than its upper and lower regions;
PFT(2) The left and right parts of the eye region are darker than its middle;
PFT(3) The mouth region is darker than its upper and lower regions;
PFT(4) The nose region is darker than its left and right regions;
PFT(5) The eye region is approximately parallel to the mouth region, and the nose region is approximately orthogonal to them;
PFT(6) The regional intensities of eye, mouth, nose and cheeks regions are ascending one by one.

3 Face Detection

3.1 Face Shape Detector

3.1.1 Detection of Ellipse

Ellipse detection is the basis of detecting the ellipse-like shapes. It has been a challenging task in shape recognition and rarely is it efficient. Based on the fast ellipse Hough transform (HT), we have developed an algorithm, real-time ellipse Hough transform [7], which is able to detect ellipses in real time while maintaining an excellent accuracy. Broadly, the algorithm has six steps:

(1) Extracting object boundaries using a refined robust edge detector; (2) Estimating edge convexity at individual points; (3) Selecting point pairs by a Stream's Scan process; (4) Locating all possible ellipse centers by the decomposition of the HT parameter space; (5) Voting for the semi-axis ratio and orientation; and (6) Estimating the long axe a.

With each center (a_0, b_0), a closely related point set Ra_0b_0 is generated by a '4-2-1' rule from the selected point pairs in step (4) and consistently used in steps (5) and (6) (see [7] for details).

3.2 Detection of Ellipse

Ellipse-like shapes have three characteristics: elliptic, deformable and fragmental; hence we make two improvements to previous ellipse detection algorithm:

(1) Local Convexity Adjustment. To accommodate the contour irregularities presenting in face shapes, we adjust the estimated convexity to better represent local details. After the estimation step (2), we adjust the convexity at every edge point within a local window (7×7 or 9×9) using Equation (1). Then, the convexity matrix is updated by new values.

$$\begin{aligned} C_i' &= \max_{k=1}^{8}\{\frac{\sum_{j=1}^{N_i} L_j^k}{N_i}\}, \\ L_j^k &= \begin{cases} 1 \text{ if } C_j = k, \\ 0 \text{ if } C_j \neq k, \end{cases} \end{aligned} \tag{1}$$

where C_j and C_i' are the initially estimated convexity and the adjusted convexity respectively at point i, and N_i is the number of edge points within the window centered on point i.

(2) Verification of Ellipse-like Candidate. We use an energy function Equation (2) as a verification measure to count for distortion and noise that may lead to fragmented ellipses. By minimizing it over the Ra0b0, we can find the new position of the ellipse candidate. Alternatively, if the minimized value is below a threshold T_e, it is treated as a false alarm.

$$\begin{aligned} \min E_{a_0b_0} = | \sum_{(x,y)\in Ra_0b_0} & \big((x - a_0 + \delta)^2 b^2 \cos(\rho) + \\ & +(y - b_0 + \varepsilon)^2 a^2 \sin(\rho) - a^2b^2\big) | \end{aligned} \tag{2}$$

where $-r_a < \delta, \varepsilon < r_b$, and $(r_a \times r_b)$ is the region for center varying.

3.2.1 Detection of Face Shape

With the rules for detecting face shapes, after some modifications the ellipse-like shape detector can be used as a flexible face shape detector:

1. According to PFS (1), the algorithm handles a specific case where long semi-axis a is approximately vertically orientated. This contributes to refining the algorithm in several places.
2. According to PFS (1) and PFS (2), the algorithm constrains the variation of the semi-axis ratio and the orientation within a range, i.e., $b/a \in [1.0, 1.5]$ and $\rho \in [-45°, +45°]$.
3. According to PFS (3), the search region in the convexity matching is reduced in width and expanded in height respectively.
4. With PFS (4) we may improve the fitting process. The parameters in Equation (2) are adjusted accordingly in different parts of the ellipse-like shape.

3.3 Face Texture Veri cation

3.3.1 Texture Description

To apply the PFT to the verification process, we adopt an efficient approach to calculating the average intensity of the rectangle region. The concept of Integral Image (II) proposed in [6] enables an intermediate representation for images using Sum of Pixel Intensity (SPI) instead of direct access. In order to generalize the idea, we devised an efficient approach to handling arbitrarily rotated rectangles. First we calculate a Rotated Integral Image (RII) with rotation (ρ) from X axis. As shown in Fig. 2(a), RII at location (x,y) represents the SPI contained by the rotated region. It can be accumulatively computed by Equation (3), where Pu denotes the pixels of the upper partition and Pl denotes the lower (see Fig. 2(b)).

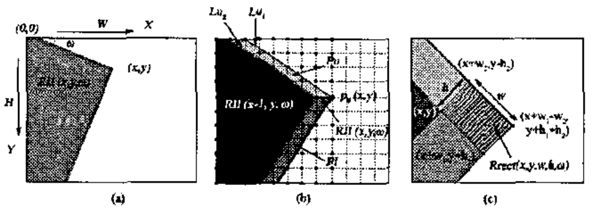


Figure 2: Rotated integral image and the SPI of rotated rectangular.

$$RII(x,y,\omega) = RII(x-1,y,\omega) + I(x,y) +$$
$$+Pu\left[(x-1,y),(x,y),\omega\right] + Pl\left[(x-1,y),(x,y),\omega\right],$$

$$Pu\left[(x_1,y_1),(x_2,y_2),\omega\right] = \sum_{\frac{y'-y_2}{x'-x_2} \le \tan(\omega) < \frac{y'-y_2}{x'-x_2},\, x'<x_2,\, y'<y_2} I(x',y'),$$
$$Pl\left[(x_1,y_1),(x_2,y_2),\omega\right] = \sum_{\frac{y'-y_2}{x_2-x'} \le \cot(\omega) < \frac{y'-y_2}{x_2-x'},\, x'<x_2,\, y_2<y'} I(x',y'), \quad (3)$$

where Pu denotes the SPI belonging to the upper partition and Pl those to the lower, and $(x,y) \in [0,W] \times [0,H]$.

$$Rrect(x,y,w,h,\omega) =$$
$$RII(x,y,\omega) + RII(x+w_1-w_2, y+h_1+h_2,\omega) \quad (4)$$
$$-RII(x+w_1,y+h_1,\omega) - RII(x+w_2,y-h_2,\omega),$$

where $h_1 = w \cdot \sin(\omega)$, $w_1 = w \cdot \cos(\omega)$, $h_2 = h \cdot \cos(\omega)$, $w_2 = h \cdot \sin(\omega)$, and $0 < \omega \le 90°$.

Given the rotation (ω) from the X axis of the rotated rectangular region, the Pu partition contains a sequence of pixels $Seq_p : (p_1, p_2, \ldots, p_m)$ between the two parallels Lu_1 and Lu_2, whose appearance order Do are fixed. If we calculate the sequence of pixels Seq_p with rotation (ω) on off-line and store the appearance order Do in a lookup table, we can readily compute any Pu according to Equation (3). In this way, we may compute a rotated rectangle $Rrect(x,y,w,h,\omega)$ using Equation (5) with only several indexing operations and additions (see Fig. 2(c)). With RII, we can calculate the regional intensity of any rotated rectangular region efficiently by dividing SPI using the region's area.

3.3.2 Feature Generation and Selection

According to PFT, we can generate and select useful regional texture patterns for the matching process. The concept of integral image enables us to quickly generate the Harr-like feature set introduced in [3]. A Haar-like pattern is composed of several basis rectangles that are upright with fixed dimensions. A basis rectangle is marked black or white to represent the regional intensity. Various combinations of the basis rectangles will generate a large set of patterns.

However, the Haar-like pattern set is inadequate for the detection of complex texture, e.g., face texture, since it is too simplistic for representing general textural features in the scene. In this paper, we propose a new pattern generation scheme for the task at hand. As Fig. 3 shows, the proposed pattern is similar to the Haar-like pattern, which consists of basic rectangles marked with black or white. We call it *scalable prototype* in the sense that: (1) the basis rectangles can be arbitrarily rotated; (2) the dimension of each basis rectangle is variable, which helps represent various geometric structural contents; and (3) the contrast is variable; that is, the basis rectangles don't have to be black or white. In this way, we can define the content of each scalable prototype pattern flexibly. The proposed patterns are sufficient to handle most situations in face detection.

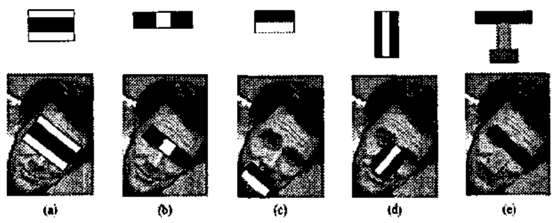


Figure 3: Five face texture patterns used to compose a FTP set.

According to PFT, we select five scalable prototype patterns as face texture patterns (FTP) that represent the characteristics of the face texture. Fig. 3 shows these patterns (a-e) on a sample image: (a) is the most apparent pattern that fits the feature of eye regions according to PFT(1); (b) is the supplement pattern for (a) according to PFT(2); (c) and (d) describe the features of mouth and nose regions according to PFT(3) and (4) respectively; and (e) is responsible for the global feature of the face texture according to PFT(5) and (6). Patterns (a) to (e) with unique rotation (ϕ) are grouped sequentially into a set (FTP set) used in the face texture verification process.

3.3.3 Matching Face Texture Patterns

The face verification is achieved by pattern matching that uses RII to evaluate the similarity is relatively simple and effective. In our algorithm, an FTP is controlled by a set of scalable prototype parameters: $\{N^i, A^i_k, \phi, (t^i_1, \ldots, t^i_j), (\alpha^i_{min}, \alpha^i_{max})\}$, where N^i is the number of basis rectangles A^i_k, ϕ is the rotation angle of the FTP, $\{t^i_j\}$ is a set of variables defining the scales of rectangles, and $(\alpha^i_{min}, \alpha^i_{max})$ is the coefficient range of contrast. Fig. 4 illustrates the geometric structure of the first two FTPs, which are represented by the SPP. For the ith FTP in the set, the matching process superposes it on the test image and compute the similarity by the measurement function Fm^i according to Equation (5), where $Rrect(\cdot)$ and $S(\cdot)$ are the SPI and the area of basis rectangles A^i_k respectively, and (x^i, y^i) is corresponding to the top-left corner of the pattern. Equation (6) constrains the brightness contrast within a range $[\theta^i_{min}, \theta^i_{max}]$, where V is the intensity variance given by Equation (7).

To search for a match, we need to maximize the measurement function over the image space by varying the top-left point of the pattern (x^i, y^i) and the scaling parameters $\{t^i_j\}$, i.e., maximizing Equation (5) while satisfying Equation (6). With the rotated FTP sets, we can also handle the texture of the potential rotated face. In general to increase the accuracy of verification, we may employ more rotated FTP sets. However, they may also increase false positive detections and computational cost. As a practical trade-off, we use five rotated FTP sets with rotation angles (ϕ) $-30°, -15°, 0°, +15°, +30°$ in turn. After applying all the rotated FTP sets respectively, each matching process (using one FTP set) yields its face candidate with an associated measurement value $\sum_i \frac{Fm^i}{(\theta^i_{max} - \theta^i_{min})N^i}$. The face candidate with the maximum value is considered as the final verified face. Non-face is declared if its value is below a given preset threshold.

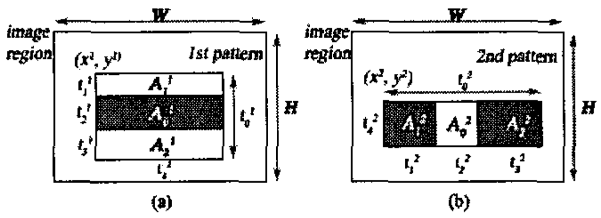


Figure 4: The scalable prototype parameters of the first two FTP in Fig. 3.

$$Fm^i\left(\{A^i_k\}; \{t^i_j\}, (x^i, y^i)\right) = \sum_{k=1}^{N^i} \left| \frac{Rrect(A^i_0)}{S(A^i_0)} - \frac{Rrect(A^i_k)}{S(A^i_k)} \right| \tag{5}$$

$$\alpha^i_{max} \cdot V \equiv \theta^i_{max} \geq \left| \frac{Rrect(A^i_0)}{S(A^i_0)} - \frac{Rrect(A^i_k)}{S(A^i_k)} \right| \geq \theta^i_{min} \equiv \alpha^i_{min} \cdot V \tag{6}$$

$$V = \sigma^2 \equiv \sum_{(x,y)} E\left(\left(I(x,y) - \frac{Urect(0,0,W,H)}{W \times H} \right)^2 \right) \tag{7}$$

4 Experiment Results

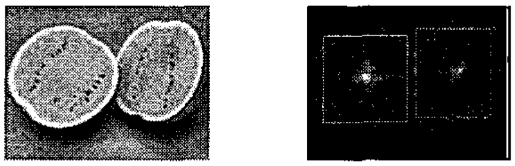
To evaluate the performance of the algorithm, we have conducted extensive experiments. The algorithm is implemented using C++ on a desktop PC with Intel P4 1GHz processor. For real-time face detection, we are especially concerned with its computational efficiency and detection accuracy.

Fig. 5 shows the results of the ellipse-like shape detection, from which we can see that the system is able to detect ellipse-like shapes and locate them accurately.

Fig. 6 shows some detection results of face shapes. 1395 test images from the FGnet face set were grouped into 93 cases (15 persons, 93 images per person). Fig. 6 (c-d) shows that our face shape detector has enough tolerance to hair styles.

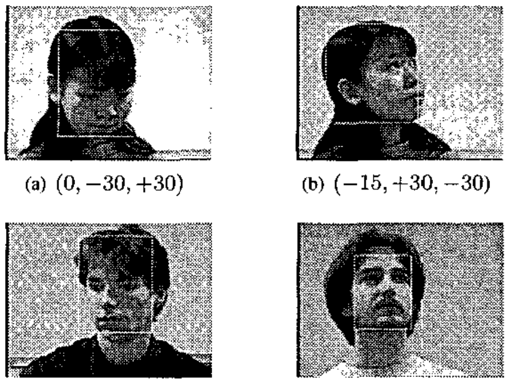
Table 1: Detection results of faces in six groups.

Face type	Background	Faces/Images	Hit Rate	False Rate	Computational Time	
					Stage(1)	Stage(2)
(a) Single, frontal	simple	50/50	100.0 %	2.0 %	42 ±5ms	46 ±5ms
(b) Single, arbitrary	simple	50/50	96.0 %	6.0 %	44 ±5ms	62 ±5ms
(c) Single, frontal	complex	50/50	98.0 %	4.0 %	53 ±5ms	54 ±5ms
(d) Single, arbitrary	complex	50/50	92.0 %	6.0 %	56 ±5ms	66 ±5ms
(e) Multiple, frontal	simple	36/30	91.7 %	13.9 %	53 ±5ms	59 ±5ms
(f) Multiple, arbitrary	arbitrary	64/40	90.1 %	14.10 %	59 ±5ms	64 ±5ms



(a) (b)

Figure 5: Detection of ellipse-like shapes.



(a) $(0, -30, +30)$ (b) $(-15, +30, -30)$

(c) $(0, -15, -15)$ (d) $(0, +30, +15)$

Figure 6: Detection of face shapes with angles of (roll, tilt and slant).

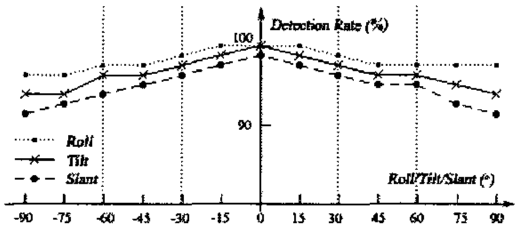


Figure 7: Detection rate of face shapes that even in the worst cases is above 92%.

Fig. 7 shows the face shape detector performs stably and reliably when dealing with face poses within the ranges of roll $[-45°, +45°]$, tilt $[-30°, +30°]$ and slant $[-30°, +30°]$, which we encounter in most real-world applications.

Fig. 8 shows the detection results of six groups of face images with respect to single/multiple faces, frontal/arbitrary views and simple/complex background. Fig. 8 (f) contains a false negative detection of the face detection algorithm which is located by the face shape detector however excluded by the face texture verifier, as the internal facial features have been completely occluded by the binocular, which is reasonable in practical applications). The test images were taken from the CMU set, the MIT set and classified into six groups manually with a normalization size of 320x240 pixels (cropped if needed). The corresponding results are reported in Table 1, which shows that our face detector performs well both in the detection rate and computational cost.

References

[1] A.K. Jain, Y. Zhong, and S. Lakshmanan, Object matching using deformable templates, IEEE Trans. on Pattern Analysis and Machine Intelligence, 18(3), pp. 267-278, 1996.

[2] C. Kotropoulos and I. Pitas, Rule-based face detection in frontal views, Proc. IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing, pp. 2537-2540, 1997.

[3] C. Papageorgiou, M. Oren, and T. Poggio, A general framework for object detection, Proc. Int'l Conf. on Computer Vision, pp. 555-562, 1998.

[4] I. Craw, D. Tock, and A. Bennett. Finding face features, Proc. the First European Conf. on Computer Vision, pp. 92-96, 1992.

[5] M.H. Yang, D.J. Kriegman, and N. Ahuja, Detecting faces in images: A survey, IEEE Trans. on Pattern Analysis and Machine Intelligence, 24(1), pp. 34-58, 2002.

[6] P. Viola and M. Jones, Robust real-time object detection, Proc. IEEE Int'l Conf. Computer Vision Workshop, 2001.

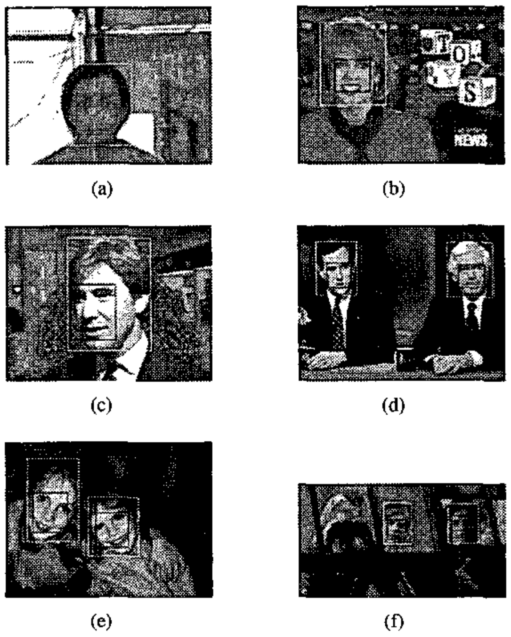


Figure 8: Face detection results with six groups of face images, where the white and black boxes indicate the face shape region and the verified face texture region respectively.

[7] Si-Cheng Zhang and Zhi-Qiang Liu, A robust, real-time ellipse detector, Pattern Recognition, to appear, 2004.