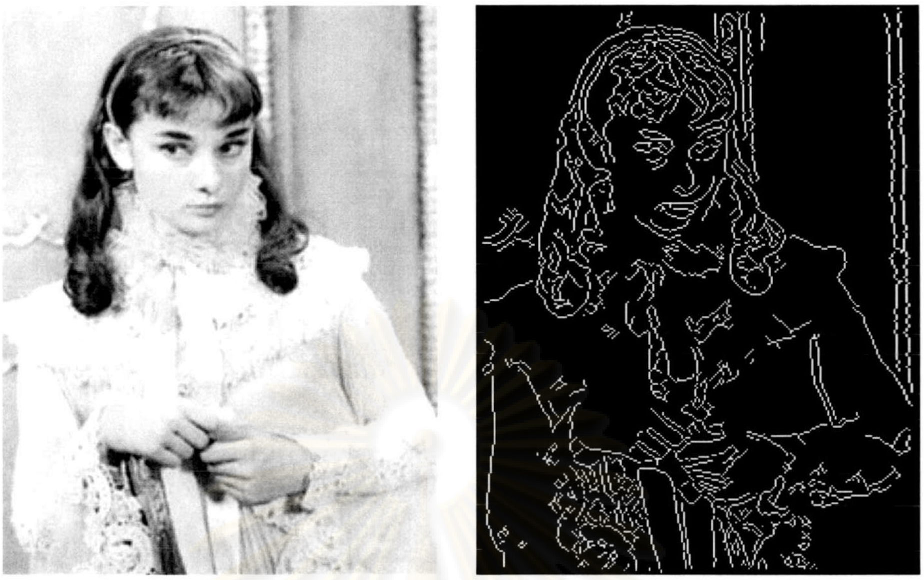
CHAPTER III

Face Detection

Face detection is the first essential stage of an automated face recognition system, since a face must be located before recognized. The existing face detection methods require intensity information. However, the results so obtained may contain errors in face positions when input images are distorted or occluded by some objects. To overcome such shortcomings, intensity information is thus avoided. This chapter itemizes the major components of face detection method, without taking intensity information into consideration. Two preprocessing steps, i.e., edge detection and finding mean face template, are performed to extract the faces from any given image.

3.1 Edge Detection

Since this method aims to extract face from arbitrary images without image intensity usage, the inputs are obtained from pixel values of edge image that is independent of the intensity information. Canny edge detection is selected in this process to find the edge image from color, gray, or black-and-white input images, except sketchy images that consist of only edge lines. Canny method [57] finds the edges by looking for local maxima of the gradient of pixel values. The gradient is calculated using the derivative of a Gaussian filter. The method differs from other edge detection methods in that it uses two different thresholds to detect strong and weak edges. Weak edges may present in the output only if they are connected to the strong edges. This method is therefore



(a) (b)

Figure 3.1: An edge detection example. (a) Original image. (b) Edge image by using Canny edge detection.

less likely than the others to be "fooled" by noise, and more likely to detect true weak edges. Figure 3.1 shows the result of edge detection.

3.2 Mean Face Template

Mean face template can be generated from average intensity of faces at same size in the databases [5–7] by the following equations:

$$\widetilde{I^r}(i,j) = \frac{1}{Q} \times \sum_q I_q^r(i,j) \; 1 \leq i \leq m, 1 \leq j \leq n \tag{3.1}$$

$$\widetilde{I^g}(i,j) = \frac{1}{Q} \times \sum_q I_q^g(i,j) \; 1 \leq i \leq m, 1 \leq j \leq n \tag{3.2}$$

$$\widetilde{I^b}(i,j) = \frac{1}{Q} \times \sum_q I_q^b(i,j) \; 1 \leq i \leq m, 1 \leq j \leq n \tag{3.3}$$

where

$I_q^r(i,j)$: red(r) component of image I_q at position (i,j)

$I_q^g(i,j)$: green(g) component of image I_q at position (i,j)

$I_q^b(i,j)$: blue(b) component of image I_q at position (i,j)

$\widetilde{I}^r(i,j)$: red(r) component of average image $\widetilde{I}$ at position (i,j)

$\widetilde{I}^g(i,j)$: green(g) component of average image $\widetilde{I}$ at position (i,j)

$\widetilde{I}^b(i,j)$: blue(b) component of average image $\widetilde{I}$ at position (i,j)

m : image vertical length

n : image horizontal length

q : image index

Q : the number of images in the databases

To solve multi-view task, three types of prototype views, consisting of frontal view, left-near-profile view, and right-near-profile view, are defined. For each view, a mean face template must be converted to black-and-white version and edge version using Canny edge detection, yet retaining similarity with input image. Figure 3.2 depicts three prototype views of mean face templates. A black-and-white version and an edge version of each prototype image are illustrated in Figures 3.2(b), 3.2(e), 3.2(h), and Figures 3.2(c), 3.2(f), 3.2(i), respectively. From multi-view face image set example [39] in Figure 3.3, the views vary horizontally from −90° to 90° in yaw and vertically from −30° to 30° in tilt. Figure 3.4 shows view segmentation mapped from Figure 3.3. The view space is divided into three major segments: left-profile, frontal, and right-profile in the horizontal direction (yaw). Each segment can be divided to three subsegments in the vertical direction (tilt) and applied by appropriate-view face template. For the upper and lower frontal views (subsegments (2) and (8) in Figure 3.4) in the vertical direction (tilt), the faces can be directly detected by frontal-view face template. Unlike frontal view, the faces in upper and lower of left or right-profile view (subsegments (1), (3), (7), and (9)) are detected by an appropriate-view face template with rotation. For example, a face with -90° in yaw and 30° in tilt can be detected by left-near-profile-view face template with 30 degree in clockwise direction. Moreover, the mean face template is resized to be compatible with all sizes of face detection. In this study, the first mean face template comes from the average normalized faces at 64×64 pixels and is resized to thirteen different sizes. The smallest size is 24×24. The other following sizes are derived by multiplying a constant in power of 1.25 to the length and width of the smallest size. Hence, the set of all considered sizes is $\{24 \times 24, 30 \times 30, 38 \times 38, 47 \times 47, 59 \times 59, 73 \times 73, 92 \times 92, 114 \times 114, 143 \times 143, 179 \times 179, 223 \times 223, 279 \times 279, 349 \times 349\}$. A template of size $s \times s$ will be used to cover an image of sizes $[s/1.125] \times [s/1.125]$ to $[1.125s] \times [1.125s]$, where $[]$ is rounding operation.

3.3 Proposed Algorithms

This section presents the detection algorithms. The input image can be color, gray, black-and-white, or edge image. Each black region of a black-and-white version of the face template is called a *black hole*. An example of a black hole is shown in Figures 3.6(a) and 3.6(c). This black hole will be used to filter the essential features of a face. The concept of face detection is based on two following significant observations.

1. Suppose a face is correctly located. The number of white pixels of the edge version of the considered face must be close to the number of white pixels of the edge version of the face template. Figure 3.5 shows an example of the edge version of the face template and an example of the edge version of a considered face.

2. When performing an *OR* operation between the black-and-white version of the face template and the edge version of the considered face, there will be some white pixels appearing in the black hole regions. These white pixels are possibly the

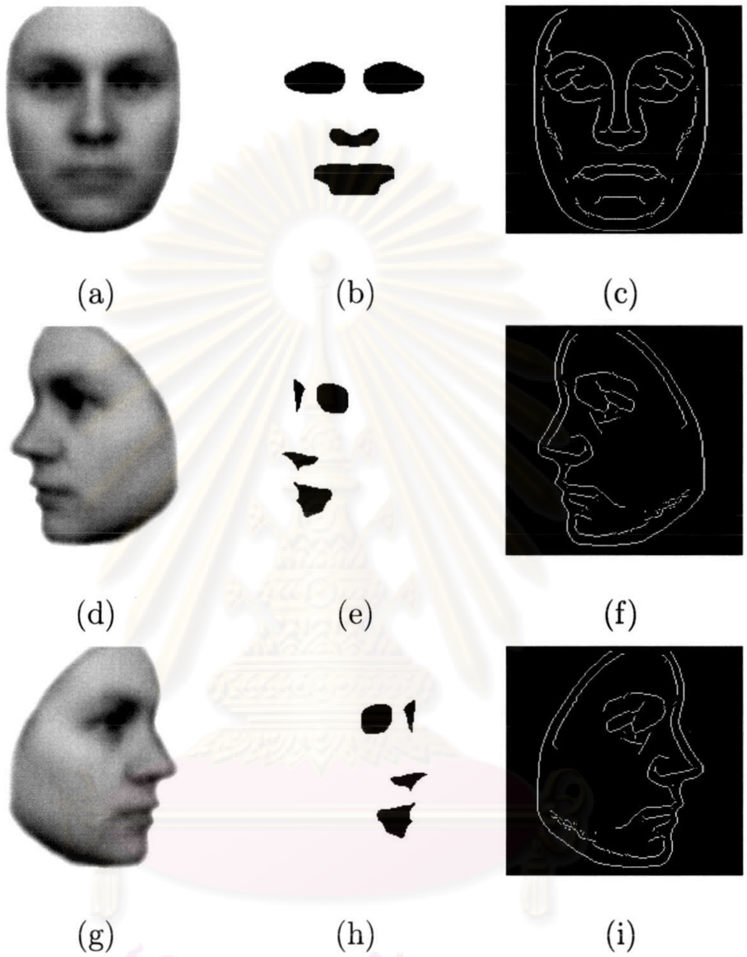


Figure 3.2: Mean face template. (a) Mean frontal-view face template (0° in yaw) at 96 × 96 pixels. (b) Black-and-white version of (a). (c) Edge version of (a). (d) Mean left-near-profile-view face template ($-60°$ in yaw) at 96×96 pixels. (e) Black-and-white version of (d). (f) Edge version of (d). (g) Mean right-near-profile-view face template (60° in yaw) at 96 × 96 pixels. (h) Black-and-white version of (g). (i) Edge version of (g).

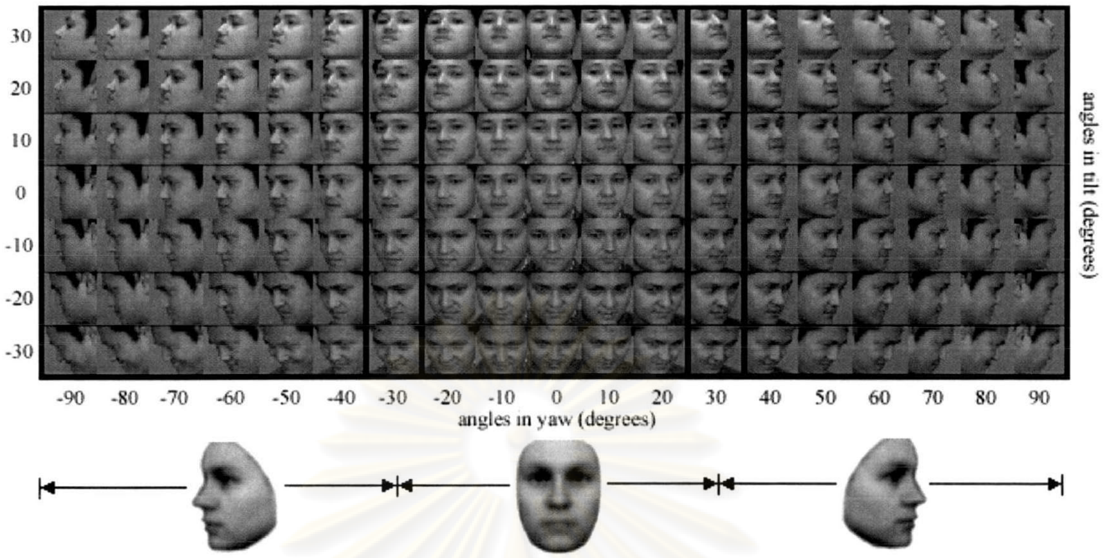


Figure 3.3: An example of multi-view face image set. The images vary from −90° to 90° in yaw and from −30° to 30° in tilt at 10° intervals. There are three templates: left-near-profile-view face template (−60° in yaw) for detecting face images of views from −90° to −30° in yaw, frontal-view face template (0° in yaw) for detecting face images of views from −30° to 30° in yaw, and right-near-profile-view face template (60° in yaw) for detecting face images of views from 30° to 90° in yaw.

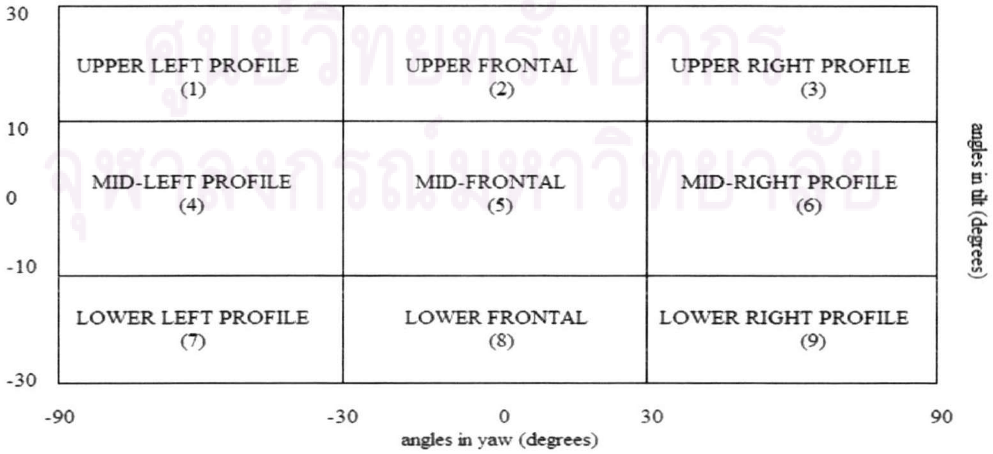


Figure 3.4: Multi-view face segmentation.

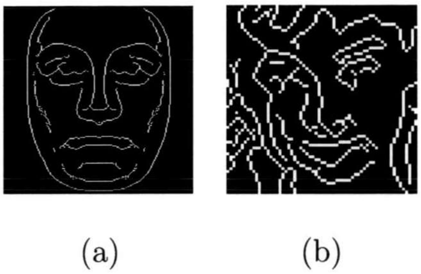


Figure 3.5: A comparison between the edge version of the face template and an actual face. (a) Edge version of the face template. (b) Edge version of an actual face.

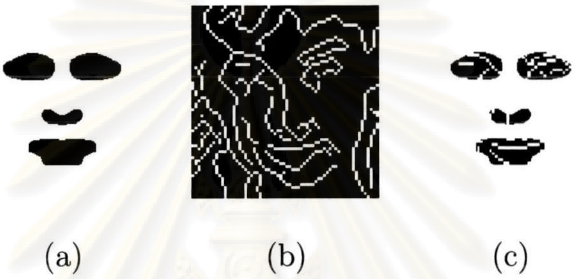


Figure 3.6: A comparison between the black-and-white version of the face template and the edge version of an actual face. (a) Black-and-white version of the face template. (b) Edge version of an actual face. (c) Features are apparent in the black holes.

essential features of the face. The density of these white pixels are high within the black hole region. Figure 3.6 shows an example of the edge version of a considered face whose features appear in the black hole regions.

Since a given face can be rotated by any angle, a set of face templates rotated at different angles must be specified to cope with these possibilities. Here, eight clockwise rotational angles {0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°} are defined. These different angles are enough to roughly estimate any rotational angle. A face template with rotational angle close to the actual rotational angle of the face will be selected. The actual rotational angle of the face will be computed afterwards by using Radon transform. Figures 3.7 and 3.8 show the black-and-white version and the edge version of the face templates with all defined rotational angles. Details of face detection algorithm are as follows:

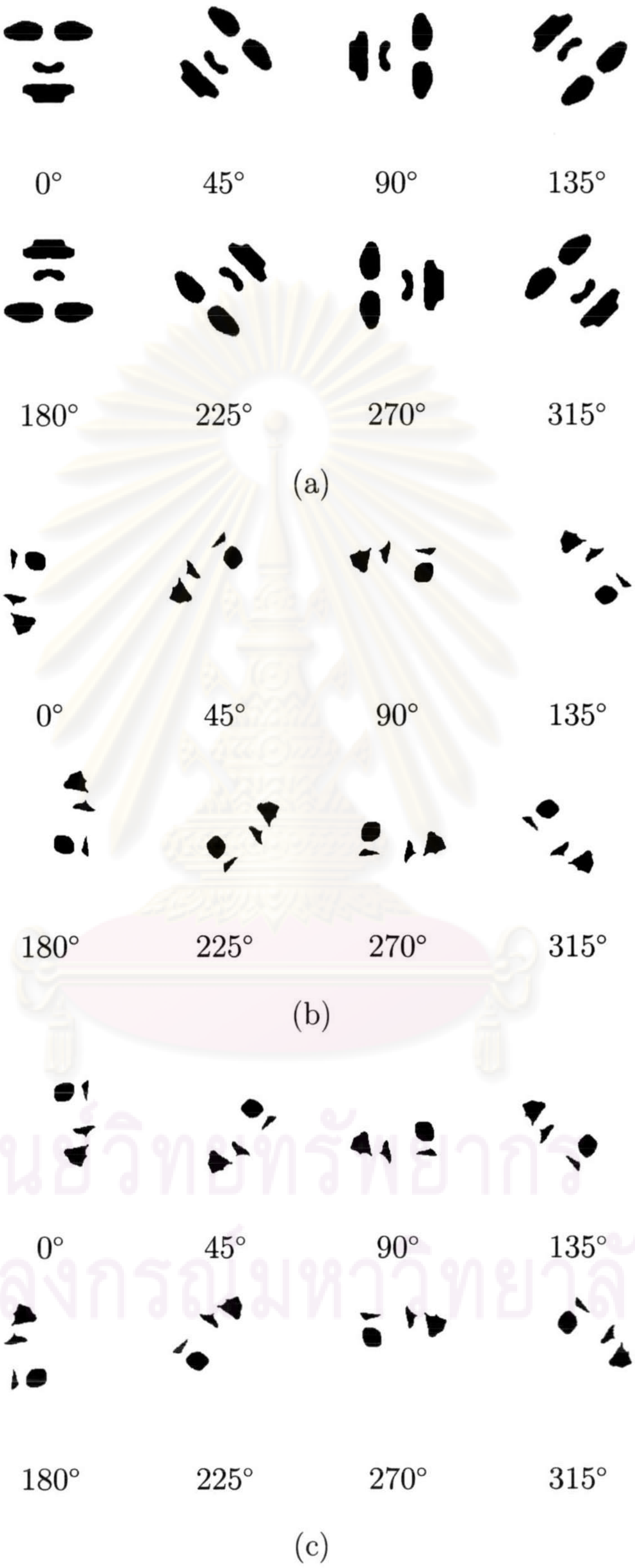


Figure 3.7: Black-and-white version of the face template rotated in eight different angles at 64 × 64 pixels (a) Frontal view. (b) Left-near-profile view. (c) Right-near-profile view.

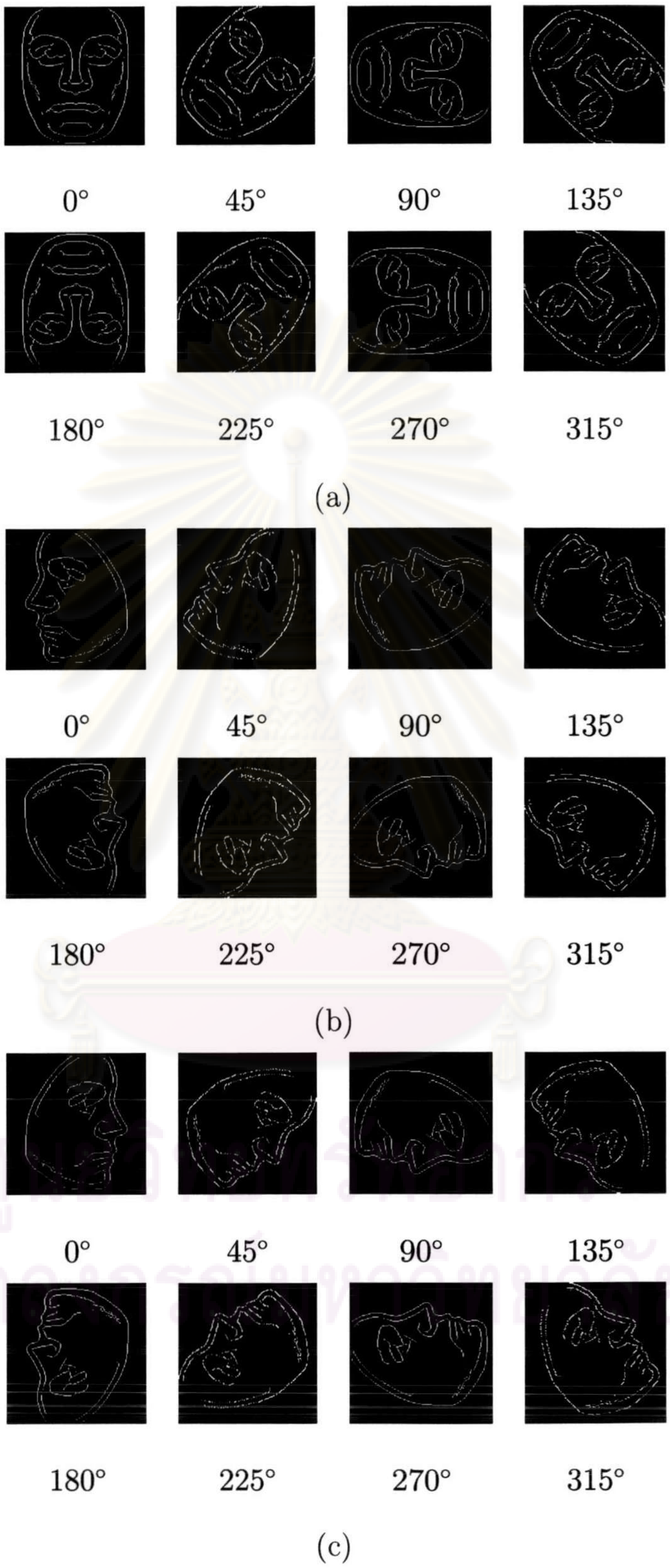


Figure 3.8: Edge version of the face template rotated in eight different angles at 64 × 64 pixels (a) Frontal view. (b) Left-near-profile view. (c) Right-near-profile view.

Algorithm 1: Face Detection

1. Convert a given color (24 bits/pixel with 8 bits/each color component), gray (8 bits/pixel) or binary (2 bits/pixel) input image into an edge (2 bit/pixel) image by using Canny edge detection as described in Section 2.1. In case of binary image, there are two types of image. The first type is a sketchy image or a drawing while the second type is a black-and-white image. The ratio of black and white pixels of the first type is rather low but the ratio of the second type is rather high. Only the second type requires edge detection.

2. Choose the smallest size of mean face, black-and-white face, and edge face templates (24 × 24 pixels) at 0°.

3. Let γ be the number of white pixels in the edge face template.

4. Slide the edge face template throughout the image from left to right and from top to bottom. For each region **a** of the image covered by the edge template, let $\phi_{\mathbf{a}}$ be the number of white pixels of region **a**. If $\phi_{\mathbf{a}} < 2\gamma$, then consider this region as the first candidate face region.

5. From the first candidate region **a** in step 4, let $\epsilon_{\mathbf{a}}$ be the number of feature pixels appearing in the black holes of the black-and-white face template.

6. Mark every region **b** for which

$$\epsilon_{\mathbf{b}} \quad > \quad 0.8\max_{\mathbf{a}}(\epsilon_{\mathbf{a}}) \;\; ;\forall \mathbf{a}, \mathbf{b} \tag{3.4}$$

   These regions are the second candidate face regions.

7. Suppose the mean face template is of size $l \times l$. Let $\phi_{i,j}$ and $\epsilon_{i,j}$ be the number of white pixels and the number of feature pixels in the region bounded by rows

$i-l+1$ to i and columns $j-l+1$ to j, respectively. Normalize $\phi_{i,j}$ and $\epsilon_{i,j}$ using the following formulae

$$\begin{aligned} \hat{\phi}_{i,j} &= \frac{\phi_{i,j} - \min_{i,j}\phi_{i,j}}{\max_{i,j}\phi_{i,j} - \min_{i,j}\phi_{i,j}} && (3.5) \\ \hat{\epsilon}_{i,j} &= \frac{\epsilon_{i,j} - \min_{i,j}\epsilon_{i,j}}{\max_{i,j}\epsilon_{i,j} - \min_{i,j}\epsilon_{i,j}} && (3.6) \end{aligned}$$

Compute the matching value, $\lambda_{i,j}$, of this region.

$$\lambda_{i,j} = (1-\hat{\phi}_{i,j}) + \hat{\epsilon}_{i,j} \qquad (3.7)$$

8. Mark every region whose $\lambda_{i,j}$ is at least $0.95\max_{i,j}\lambda_{i,j}$ as the face region.

9. Repeat steps 3 to 8 for other rotational angles.

10. Repeat steps 3 to 9 for other template sizes.

11. Repeat steps 3 to 10 for other views.

There are four thresholds used in this algorithm. The first two thresholds are in Canny edge detection method. The first candidate face region is the edge region having the number of white pixels less than twice of the number of white pixels in the edge face template (step 4). Such a wide range coverage can handle the effect of the thresholds. The remaining thresholds, namely, 0.8 and 0.95 from Eq.(3.4) and step 8, respectively, are tuned for good implementation. Higher thresholds yield detector with fewer false positives and lower detection rate. Lower thresholds yield classifiers with more false positives and higher detection rate. Unfortunately, finding the optimal thresholds is a difficult problem.

Based on the first observation, step 4 in the algorithm rejects about 70% of non-face regions. The remaining regions become initial candidate regions. Subsequently, the second candidate regions are retained by the second observation in step 6 and more

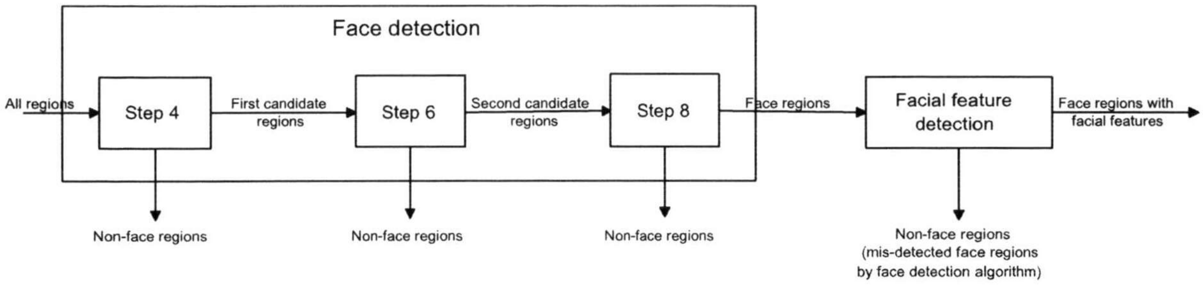


Figure 3.9: Face detection process. A large number of non-face regions (negative examples) are discarded by steps 4, 6, and 8 in the algorithm. Subsequently, some mis-detected face regions are eliminated by facial feature detection process.

than 99% of non-face regions are rejected. The face regions are extracted in step 8, which combine two observations together. However, all face regions will be analyzed again in the facial feature detection process. Some face regions will be discarded if there is no feature detected from facial feature detection algorithm. This will be useful to filter all the regions being mis-detected as face regions by face detection algorithm. The overall face detection process is shown in Figure 3.9. Figure 3.10 and Figure 3.11 depict some face detection examples.

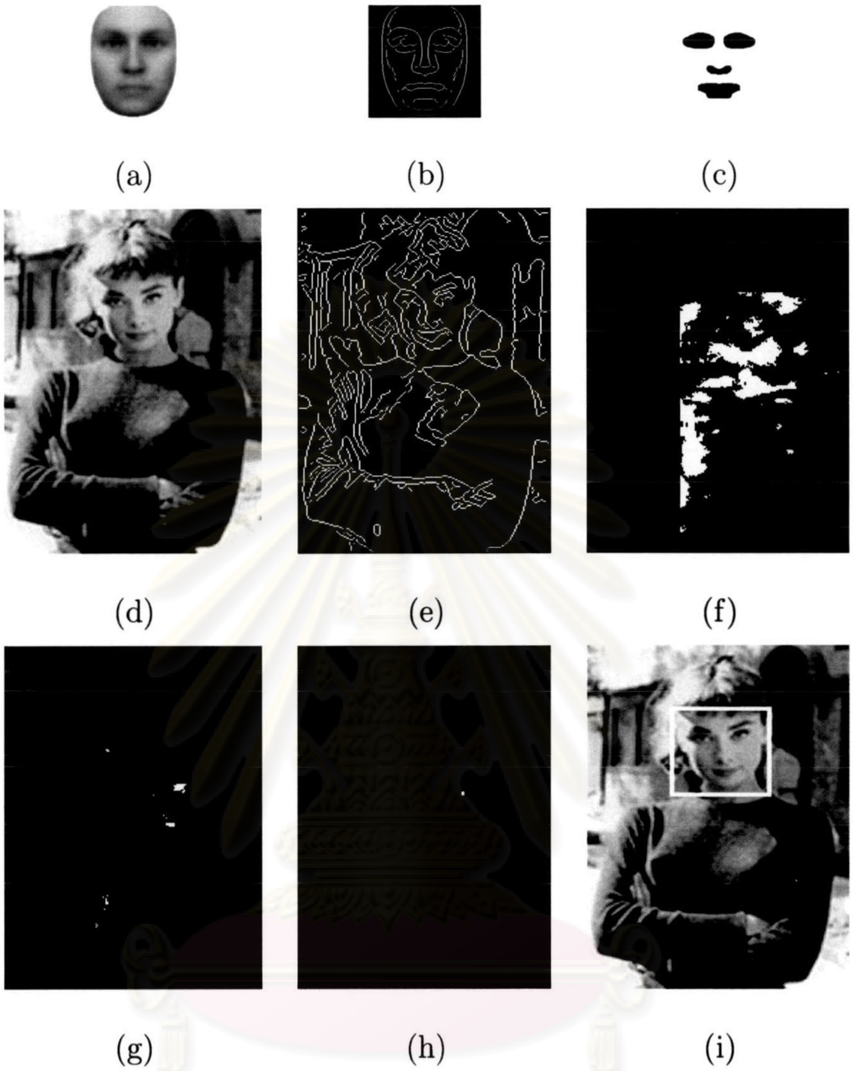


Figure 3.10: Face detection steps. (a) Mean face template. (b) Edge version of template applied to steps 3 and 4. (c) Black-and-white version of template applied to steps 5 and 6. (d) Original image. (e) Edge image. (f) The first candidate face regions by step 4. (white pixel at position (i, j) corresponds to the square candidate region between vertical pixels $i - l + 1$ and i, and horizontal pixels $j - l + 1$ and j, where the template size is $l \times l$). (g) The second candidate face regions by step 6. (h) From step 8, a white pixel at position (i, j) corresponding to a detected region between vertical pixels $i - l + 1$ and i, and horizontal pixels $j - l + 1$ and j, where the template size is $l \times l$. (i) Face detection result.

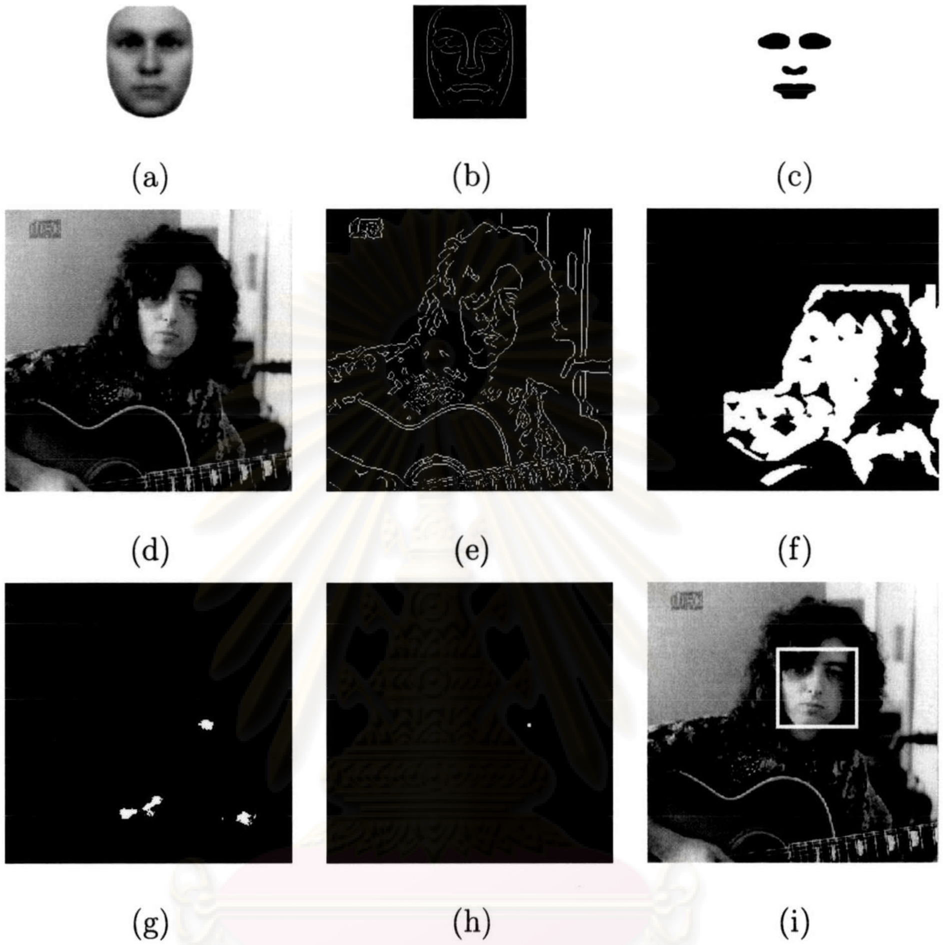


Figure 3.11: Face detection steps. (a) Mean face template. (b) Edge version of template applied to steps 3 and 4. (c) Black-and-white version of template applied to steps 5 and 6. (d) Original image. (e) Edge image. (f) The first candidate face regions by step 4. (white pixel at position (i, j) corresponds to square candidate region between vertical pixels $i - l + 1$ and i, and horizontal pixels $j - l + 1$ and j, where the template size is $l \times l$). (g) The second candidate face regions by step 6. (h) From step 8, a white pixel at position (i, j) corresponding to a detected region between vertical pixels $i - l + 1$ and i, and horizontal pixels $j - l + 1$ and j, where the template size is $l \times l$. (i) Face detection result.