Available online at www.sciencedirect.com
ScienceDirect
Image and Vision Computing 25 (2007) 741–753
www.elsevier.com/locate/imavis

Face detection and facial feature localization without considering the appearance of image context

S. Phimoltares [a,*], C. Lursinsap [a], K. Chamnongthai [b]

[a] Advanced Virtual and Intelligent Computing (AVIC) Center, Department of Mathematics, Chulalongkorn University, 254 Phyathai Road Patumwan Bangkok 10330, Thailand

[b] Department of Electronics and Telecommunication Engineering, King Mongkut's University of Technology Thonburi, 91 Prachauthit Road Thung Kharu Bangkok 10140, Thailand

Received 10 June 2004; received in revised form 7 April 2006; accepted 31 May 2006

Abstract

Face and facial feature detection plays an important role in various applications such as human computer interaction, video surveillance, face tracking, and face recognition. Efficient face and facial feature detection algorithms are required for applying to those tasks. This paper presents the algorithms for all types of face images in the presence of several image conditions. There are two main stages. In the first stage, the faces are detected from an original image by using Canny edge detection and our proposed average face templates. Second, a proposed neural visual model (NVM) is used to recognize all possibilities of facial feature positions. Input parameters are obtained from the positions of facial features and the face characteristics that are low sensitive to intensity change. Finally, to improve the results, image dilation is applied for removing some irrelevant regions. Additionally, the algorithms can be extended to rotational invariance problem by using Radon transformation to extract the main angle of the face. With more than 1000 images, the algorithms are successfully tested with various types of faces affected by intensity, occlusion, structural components, facial expression, illumination, noise, and orientation.

Keywords: Face detection; Face location; Facial feature detection; Neural visual model; Pattern recognition

1. Introduction

In recent years, detecting human faces and facial features has become an important task in computer vision with numerous potential applications including human computer interaction, video surveillance, face tracking, and face recognition. For this reason, the detection method should be able to handle various practical and artificial problems. The objective of face detection is to determine whether or not there are any faces in the image and, if any, return the face location. In the case of facial feature localization, the goal is to detect the presence and location of features after the locations of faces are extracted by using any face detection methods. The challenges associated with face and facial feature detection can be attributed to the following factors [1].

- *Intensity.* There are three types of intensity: color, gray, and binary.
- *Pose.* Face images vary due to the relative camera-face pose (frontal, 45°, profile), and some facial features such as an eye may become partially or wholly occluded.
- *Structural components.* Facial features such as beards, mustaches, and glasses may or may not be presented.
- *Image rotation.* Face images directly vary for different rotations.
- *Poor quality.* Image intensity in poor-quality images, for instance, blurry images, distorted images, and images with noise, becomes unusual.

* Corresponding author. Tel.: +66 24152486; fax: +66 28982766.
E-mail addresses: Suphakant.P@Student.chula.ac.th (S. Phimoltares), lchidcha@chula.ac.th (C. Lursinsap), kosin.cha@kmutt.ac.th (K. Chamnongthai).

0262-8856/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2006.05.017

- *Facial expression.* The appearance of faces depends on a personal facial expression.
- *Unnatural intensity.* Cartoon faces and rendered faces from 3D model have unnatural intensity.
- *Occlusion.* Faces may be partially occluded by other objects such as hand, scarf, etc.
- *Illumination.* Face images vary due to the position of light source.

The existing techniques are not applicable for all images with any of the above factors or their combinations. In addition, the accuracy of those techniques are based on the usage of intensity information as input. To overcome this limitation for extending to any face images, the input should be obtained from the position and face shape information that is low sensitive to intensity change. In this paper, face and facial feature detection algorithms are developed that are able to handle a wide range of variations in face images with above attributes, based on neural network and image processing techniques with the position and face shape information.

In recent surveys, there are five categories of face and facial feature detection.

(1) *Geometry-based methods.* These methods utilize geometrical information [2–4]. Each feature is demonstrated as a geometrical shape. They can accurately detect face and facial features, but cannot handle large variations of the face images such as images with some occluded facial features and images with noise.
(2) *Color-based approaches.* These approaches face difficulties in robustly detecting skin colors in the presence of complex background and different illuminations [5–8]. These algorithms are applicable only for color images.
(3) *Appearance-based methods.* These methods use the models learned from a set of training images [9–13]. Gray value (intensity) is the most important parameter for the detection. They are not able to perfectly detect face images with poor quality in intensity, some occlusions, and unnatural intensity.
(4) *Motion-based methods.* Face and facial features are detected from the image sequence [14–16]. Using such methods, facial features cannot be detected using only a single still image from one view.
(5) *Edge-based methods.* In the last class of methods [17–20], faces are detected without information due to intensity and motion. The edge information is used as input. These methods can handle large variations of the face images. However, these methods detect only face [20] or facial features [17–19].

It is obvious that no method can be applied to all intensity types, poses, illuminations, invariant rotation, and face appearances.

In this paper, to avoid a limitation of intensity usage, position and face shape information is used as input. There are two main stages: face detection and facial feature detection as shown in Fig. 1. For the face detection stage, the face is detected by the proposed algorithm based on two observations. First, the number of white pixels of the edge version of the considered face must be close to those of the edge version of the face template. The first candidate face regions are obtained by this first filtering. Next, the second observation is that the density of the white pixels is high within the facial feature-like areas. The regions obtained from the second filtering are the second candidate face regions. Then, the matching value, which is the combination of both observations, is calculated and applied to find the actual face as the third filtering. For the facial feature detection stage, the detection is applied to the face obtained in the previous step. A proposed neural visual model (NVM) is used to recognize all possibilities of facial feature positions. With the extracting facial parameters algorithm, input parameters are obtained from the positions of facial features and the face characteristics that are low sensitive to intensity change. Finally, to improve the results, image dilation is applied for eliminating some irrelevant regions. Additionally, the algorithms can be extended to rotational invariance problem by using Radon transformation to extract the main angle of the face. The remainder of the paper is organized as follows: Section 2 describes the proposed face detection. Section 3 presents the proposed facial feature detection. The experimental results are shown in Section 4. Section 5 concludes the paper.

2. Proposed face detection

Since the existing face detection methods require intensity information, there are some errors of face positions when input images are distorted or occluded by some objects. To break this limitation, the intensity information should be avoided. This section itemizes the major components of face detection method with preprocessing that is performed to extract the faces rather independent from the intensity information.

2.1. Preprocessing

To avoid a limitation of intensity usage, the edge image that can be obtained from arbitrary images is considered to be used as input. Canny edge detection [21] is selected in this process to find the edge image from color, gray, or black-and-white image. The method differs from the other edge detection methods in that it includes the weak edges in the output only if they are connected to the strong edges.

The edge image is further compared with our introduced template, namely, mean face template. The mean face template can be generated from average intensity of faces at same size in database [22]. Then, the mean face template is converted to black-and-white version and edge version by using Canny edge detection, yet retaining similarity with

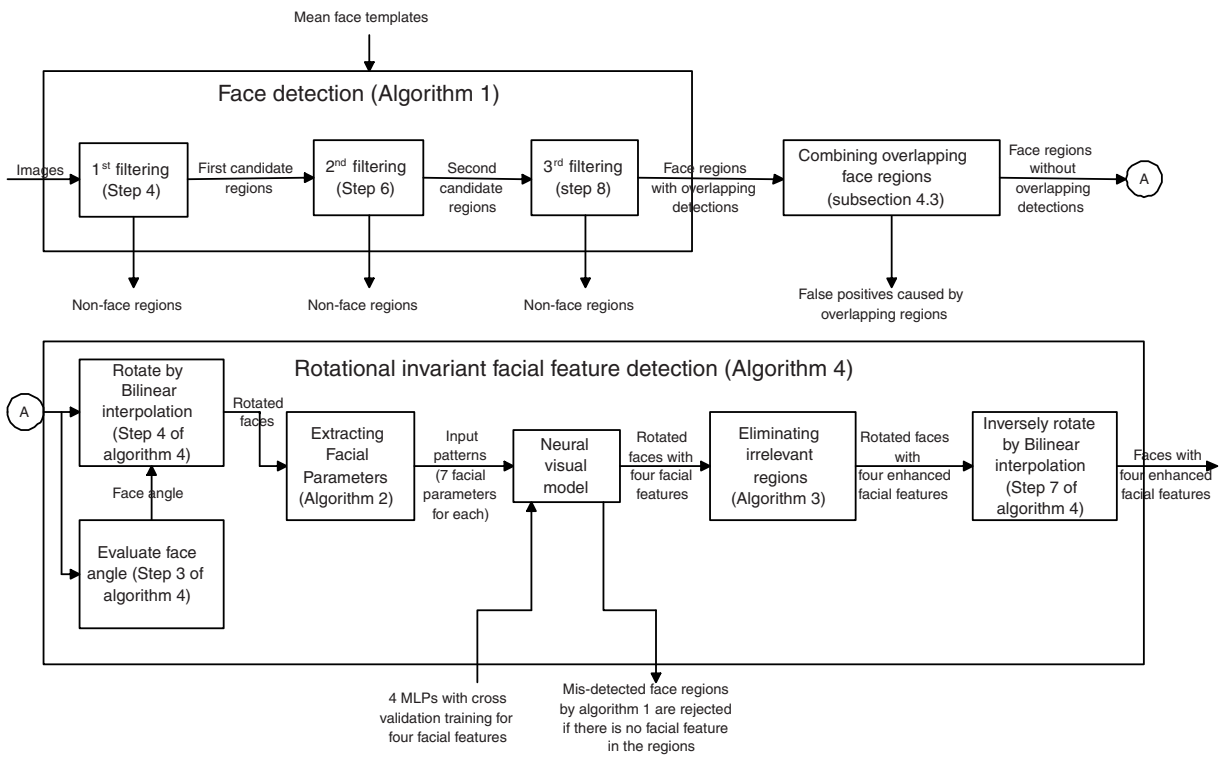

Fig. 1. The overall process of our proposed face and facial feature detection.

input image. Fig. 2 depicts the mean face template. From multi-view face image set example [23] in Fig. 3, the views vary horizontally from −90° to 90° in yaw and vertically from −30° to 30° in tilt. The frontal-view mean face template can support the images varying horizontally from −30° to 30° in yaw.

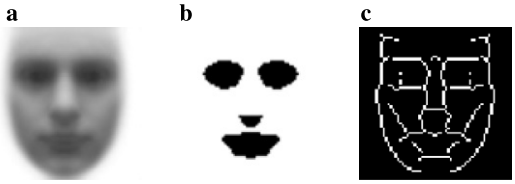

Fig. 2. Mean face template. (a) Mean face template at 92×92 pixels. (b) Black-and-white version of (a). (c) Edge version of (a).

To apply the proposed method to faces with size variation, the mean face template is resized to various sizes in order to be compatible with all sizes of face detection. In this study, the first mean face template comes from the average normalized faces at 256×256 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 $\{24 \times 24, 30 \times 30, 38 \times 38, 47 \times 47, 59 \times 59, 73 \times 73, 92 \times 92, 114 \times 114, 143 \times 143, 179 \times 179, 224 \times 224, 279 \times 279, 349 \times 349\}$. A template of size $s \times s$ will be used to cover

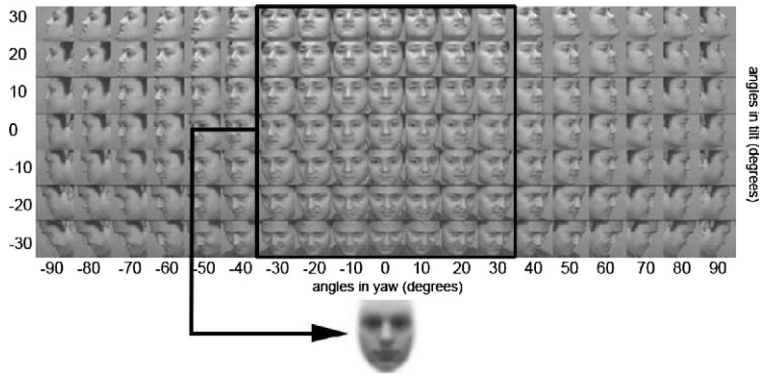

Fig. 3. An example of multi-view face image set. The images are varied from −90° to 90° in yaw and from −30° to 30° in tilt at 10° intervals. Frontal-view mean face template can be applied to images of views from −30° to 30° in yaw.

an image of sizes $[s/1.125]\times[s/1.125]$ to $[1.125s]\times[1.125s]$, where the square brackets denote the rounding operation.

Moreover, a set of face templates rotated with different angles must be specified to cope with the possibilities of rotated faces. 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 transformation.

2.2. Proposed algorithm

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* as shown in Figs. 4(a) and (c). This black hole will be used to filter the essential features of a face. The concept of our proposed 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.
(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 holes. These white pixels are possibly the essential features of the face. The density of these white pixels is high within the black holes. Fig. 4 shows an example of the edge version of a considered face whose features appear in the black holes.

Details of the face detection algorithm are as follows:

Algorithm 1: Face detection

(1) Convert an input image into an edge image by using Canny edge detection.
(2) Generate the first mean face template by averaging intensity of faces at same size in database [22]. From the first template, generate other templates in thirteen different sizes and eight rotational angles. Then, convert all mean face templates to black-and-white and edge versions. Next, 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
$$\epsilon_{\mathbf{b}} > 0.8\max_{\mathbf{a}}(\epsilon_{\mathbf{a}}); \forall \mathbf{a}, \mathbf{b} \tag{1}$$
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 second candidate 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:
$$\hat{\phi}_{i,j} = \frac{\phi_{i,j} - \min_{i,j}\phi_{i,j}}{\max_{i,j}\phi_{i,j} - \min_{i,j}\phi_{i,j}} \tag{2}$$
$$\hat{\epsilon}_{i,j} = \frac{\epsilon_{i,j} - \min_{i,j}\epsilon_{i,j}}{\max_{i,j}\epsilon_{i,j} - \min_{i,j}\epsilon_{i,j}} \tag{3}$$
Compute the matching value, $\delta_{i,j}$, of this region.
$$\delta_{i,j} = (1-\hat{\phi}_{i,j}) + \hat{\epsilon}_{i,j} \tag{4}$$
(8) Mark every region whose $\delta_{i,j}$ is at least $0.95\max_{i,j}\delta_{i,j}$ as the face region.
(9) Repeat steps 3–8 for the other rotational angles.
(10) Repeat steps 3–9 for the other template sizes.

Fig. 4. 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 appearing in the black holes.

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 as described in 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. (1) and step 8, respectively, are tuned for good implementation. Higher thresholds yield detector with fewer false positives and lower detection rate. Lower thresholds yield detector with more false positives and higher detection rate. Unfortunately, finding the optimal thresholds is a difficult problem.

Based on the first observation, step 4 of the algorithm rejects about 70% of non-face regions. The remaining regions are the first candidate regions. Based on the second observation, the second candidate regions are retained by

step 6 and more than 99% of non-face regions are rejected. The face regions are extracted by 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 by facial feature detection algorithm. This will be useful to filter all the regions being mis-detected as face regions by the face detection algorithm. The overall form of face detection process is shown in Fig. 5. Fig. 6 represents the face detection example.

3. Proposed facial feature detection

The goal of facial feature detection is to detect the presence and location of features (left eye, right eye, nose, and mouth). Before applying facial feature detection algorithm, the size of the face from the face detection process is normalized to 128×128 pixels. The proposed approach consists of three main parts. First, facial features are coarsely detected by a neural visual model (NVM). To be free from the usage of intensity information, the input of the NVM is defined from the position and face shape information. The results from the NVM are further enhanced by applying image dilation algorithm in the second part. In the last part, Radon transformation is used to evaluate the face angle of a rotated image.

3.1. Neural visual model (NVM)

Neural visual model (NVM) is used for detecting facial features. There are two parts in the NVM. The first part is input preparation. The network input contains facial parameters obtained from the position and face shape information. The other part is model construction.

3.1.1. Facial parameters

Seven facial parameters obtained from face databases are extracted from the position and face shape information. The first two parameters denote the position of any given image pixel with respect to the center of face image in polar coordinates. The remaining five parameters represent to the width of the face and the Euclidean distances measured from four corners, upper left, upper right, lower left, and lower right, to the center of face image, respectively. The parameters are illustrated in Fig. 7. The normalized size of the training image is 128×128 pixels.

The following geometric values and notations are referred in the NVM.

(1) (x_o, y_o): Position of the face center O in rectangular coordinates. The values of x_o and y_o are dependent on the length and width of the face, respectively. Since the face length L_l has no effect on the analysis of facial features by the NVM, the value of x_o is set to $L_l/2$. However, the value of y_o depends upon the actual width of a given face and must be derived from the image.
(2) (r, θ): Position of any point (x, y) in polar coordinates with respect to the face center.
(3) L_w: Face width defined as the distance of a line connecting two boundary points (E and F) of the face.
(4) L_A, L_B, L_C, L_D: Distance between the center (O) and the far upper left (A), upper right (B), lower left (C), and lower right corner (D), respectively.

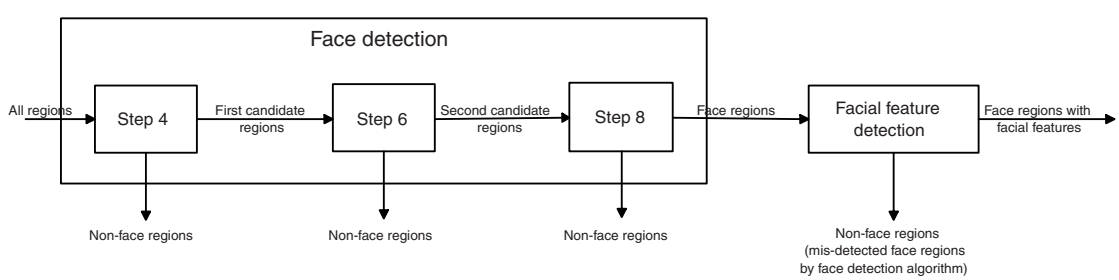

Fig. 5. Face detection process. A large number of non-face regions (negative examples) are discarded by steps 4, 6, and 8 in the algorithm. Subsequently, some mis-detected face regions are eliminated by facial feature detection process.

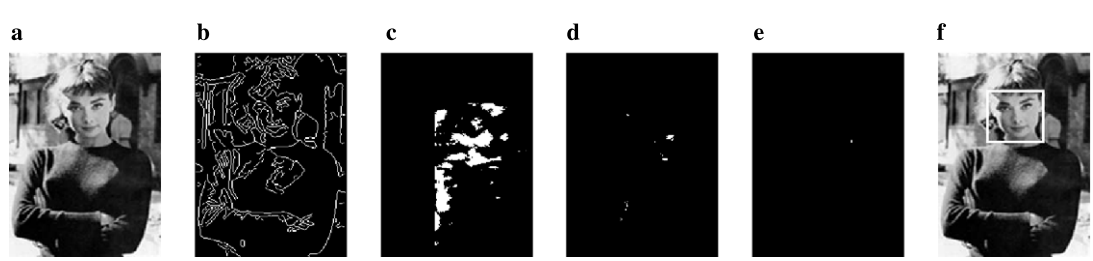

Fig. 6. Face detection steps. (a) Original image. (b) Edge image. (c) The first candidate face regions by step 4. (white pixel at position (i,j) corresponds to the square candidate region between vertical pixels $i - l + 1$ and i, and horizontal pixels $j - l + 1$ and j, where the template size is $l \times l$). (d) The second candidate face regions by step 6. (e) 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 $l \times l$. (f) Face detection result.

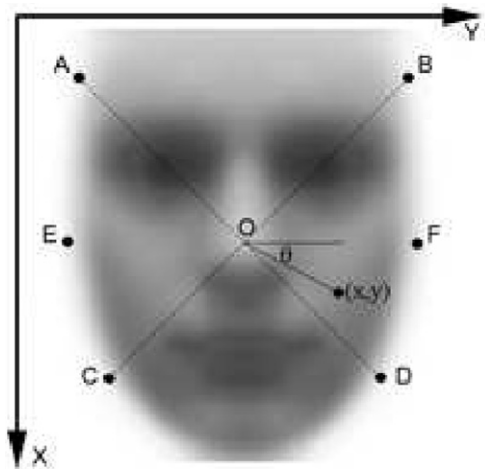

Fig. 7. Notation used for the facial parameters. O is a face center. Line EF is parallel to the Y-axis. The length of line EF is called face width. Lines AD and BC are perpendicular to each other. The angles $B\widehat{O}D$ and $A\widehat{O}C$ are divided into two halves by line EF. A, B, C, and D are at four corners of the face.

All of the above parameters can be derived from a given image by the following algorithm.

Algorithm 2: Extracting facial parameters

(1) Extract face by the face detection algorithm and normalize it to 128×128 pixels.
(2) Convert the face image from step 1 into edge face image.
(3) Generate face template of size 128×128 by resizing face template of size 24×24 and dilate the face boundary of the face template.
(4) Apply *AND* operation to the dilated template and the edge face image.
(5) Dilate the face boundary from step 4 in order to connect all pixels.
(6) Apply thinning method to dilated face boundary.
(7) Consider two pixels at the 64th row. L_w is measured between both pixels and face center is the center of the line between both pixels.
(8) Extract L_A, L_B, L_C, L_D by measuring between the face center and a white pixel on thinned image of step 6 in four directions.
(9) Compute (r, θ) at any point by converting rectangular to polar coordinates with respect to the face center from step 7.

An example of extracting the face center and the face width is shown in Fig. 8.

These facial parameters are very low sensitive to the facial color, illuminations and any face appearances such as wearing the spectacles or sunglasses and covering the mouth by a scarf. Only parameters r, θ, L_w, L_A, L_B, L_C, L_D are used in our NVM for locating the essential facial features.

3.1.2. Implementation of the neural visual model

Facial feature detection model is constructed using multilayer perceptron (MLP) networks with back-propagation learning algorithm because they are simple and often quick. Four essential facial features, i.e. left eye, right eye, nose, and mouth, are considered. Each feature will be located using one MLP network. Hence, the neural visual model (NVM) consists of four MLP networks introduced for all facial features. Each input pattern consists of seven facial parameters as described in the previous section. The problem of locating facial feature is considered as a problem of classifying a given set of facial parameters into one of two classes, locating in an interested facial feature or not locating in an interested facial feature. In addition, the cross validation technique [24] is applied to obtain an optimal generalization performance.

3.2. Eliminating irrelevant regions

In this paper, image dilation [25] is an operator chosen to eliminate some irrelevant regions remaining after using the NVM. The basic effect of the operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels (i.e. white pixels, typically). The algorithm for eliminating the irrelevant regions is depicted as follows:

Algorithm 3: Eliminating irrelevant regions

(1) Filter the original image by a threshold, and invert the threshold filtered image. The black-and-white face image is obtained from this step.
(2) Apply dilation to the image from step 1 by

$$I_D(x,y) = I(x,y) \oplus d(x,y) \tag{5}$$

where $\oplus$ is dilation sign, (x,y) is position of pixel in rectangular coordinates, $I(x,y)$ is the black-and-white face image from step 1, $d(x,y)$ is a structuring element

Fig. 8. Steps for finding face width and face center. (a) Normalized face from face extracted by face detection. (b) Edge version of (a). (c) Dilated boundary from face template. (d) Result of applying *AND* operation to (b) and (c). (e) Thinned version of dilated face boundary. (f) Two pixels at the 64th row. The face width is the distance between both pixels and the face center is the center of the line between both pixels.

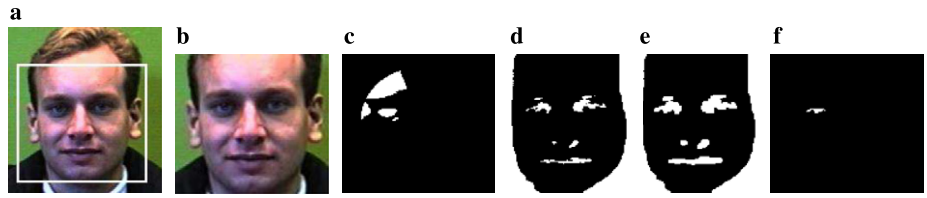

Fig. 9. Left-eye region detection steps using image dilation. (a) Face detection. (b) Normalized face from (a). (c) Left-eye region detection by the NVM. (d) Black-and-white image obtained from original image with thresholding filter (step 1). (e) Dilated image. (f) Left-eye region after eliminating the irrelevant regions using image dilation.

defined by the 3×3 square image with white intensity, and $I_D(x,y)$ is the dilated image.

(3) Combine the dilated image and the detected image from the NVM with an *AND* operation by

$$I_R(x,y) = AND(I_D(x,y), I_{\mathrm{NVM}}(x,y)) \tag{6}$$

where $I_{\mathrm{NVM}}(x,y)$ is the detected image from the NVM, and $I_R(x,y)$ is the detected image after eliminating the irrelevant regions.

An example of how the algorithm works is shown in Fig. 9.

3.3. Rotational invariance

Radon transformation [26] transforms a two-dimensional image with lines into a domain of possible line parameters. Each line in the image gives a peak position at the corresponding line parameters such as the distance to the center of image (ρ), and the angle of a line (θ). For our algorithm, some lines of rotated edge image provide an important line parameter called *face angle*.

In the face detection stage, a face is detected by face templates with eight angles. However, the real face angle is re-calculated from the maximum intensity of the transformed image. The algorithm for facial feature detection with rotational invariance is as follows:

Algorithm 4: Rotational invariant facial feature detection

(1) Construct the edge image using Canny edge detector.
(2) Eliminate the face boundary by removing outside white pixels from the possible face region. The face region is in the circle region with radius α and center at face center (x_o, y_o). The radius is calculated by

$$\alpha = 0.3 \times \max(m,n) \tag{7}$$

where $m \times n$ is image size. The coefficient 0.3 in the above equation is a yield value which retains the eyes, nose, and mouth regions but face boundary. This coefficient is statistically tuned for good implementation.
(3) Apply Radon transformation to the edge image from the previous step. The face angle ($\beta°$) is evaluated from the position having the highest intensity on the transformed image.
(4) Rotate the original image by $-\beta°$ in clockwise direction using a bilinear interpolation.
(5) Detect face and facial features from the re-rotated image using face and facial feature detection algorithms in Sections 2, 3.
(6) Bound the regions of facial features by rectangular blocks.
(7) Inversely rotate the rectangular blocks by $\beta°$ in clockwise direction and superimpose the blocks on the original image.

An example of rotated face and facial feature detection is shown in Fig. 10.

4. Experimental results

This section illustrates the face database, the speed of the detector, and the experiments on real-world face images.

4.1. Face database

The face and facial feature detection algorithms are applied to detect generic faces from several face image databases as follows:

(1) ESSEX-face database [22] which contains head or head and shoulder pictures.
(2) Purdue AR database [27] which contains face images with different facial expressions and occlusions under different illuminations.
(3) MIT-CMU database [28] which contains gray mugshot-style images.
(4) Max Planck face database [29] which contains seven views of face images without hair.
(5) Private gallery which contains images from the World Wide Web, magazines, newspapers,and photographs.

A set of mean face templates for face detection algorithm is generated from the gray images in the ESSEX-face database, which contains face with single color background. Since these face images have single color background, it is easy to extract facial parameters from them. Hence, these images are also used for training the NVM. The test set for this study is obtained from the combination among those five databases. For

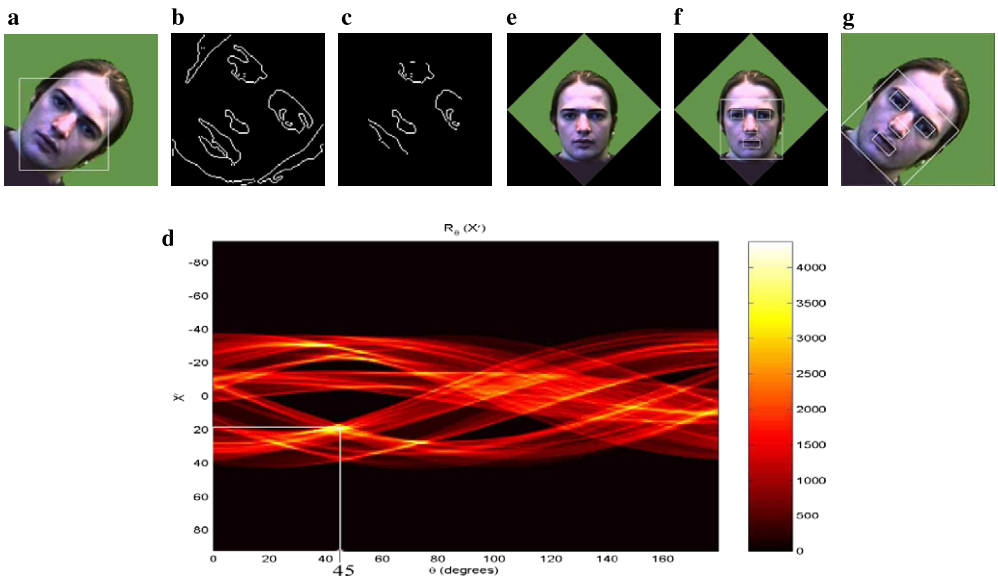

Fig. 10. Detection steps of rotated image. (a) Face detection. (b) Edge image of normalized face. (c) Edge image without face boundary. (d) Radon transformation of (c) which the highest intensity is at 45°. (e) Re-rotated image. (f) Face and facial feature detection of re-rotated image. (g) Face and facial feature detection of rotated image.

the ESSEX face database, there are 20 faces varied in facial expression, scaling, and translation for each person, therefore only one image is selected at random for each person. Like the ESSEX database, 300 test images are randomly selected from the Purdue AR database under different illuminations and occlusions. For the MIT-CMU database, the test set is originally separated and all images in the test set are used. The Max Planck face database contains seven views of color face images. Only face images with 0°, −30°, and 30° in yaw are chosen in various appearances.

4.2. Speed of the algorithms

Our method was carried out on Matlab V 6.0 with 1.4 GHz Pentium 4 PC. Training time for each MLP network of the NVM was about 20 h. The face detection can process an image of size 1280×1024 within 20 s. The facial feature detection takes about 15 s to locate all facial features in a face. This is 2 times slower than Rowley–Baluja–Kanade method [28] and 24 times faster than Schneiderman–Kanade method [30]. There are three main circumstances that affect the processing time as follows:

(1) It should be faster if our proposed method is applied to an integral image [12], a new image representation for very fast calculation, instead.
(2) The parallel processing can be applied to improve the speed of face detection algorithm because each image is separated into subimages and they can be processed independently.
(3) Matlab is somewhat slower than any other programming languages such as C++.

The proposed method will be improved to reduce processing time in the future work.

4.3. Combining overlapping face regions

Overlapping detections usually occur around each face, but the result should be one detection per face. To find the final result of face detection, postprocessing is applied to combine overlapping detected faces into a single detected face. In this study, the conventional technique [12] is used as the postprocessing. The set of overlapping detected faces can be partitioned into disjoint subsets. Multiple detected faces are in the same subset if their boundary regions overlap. Finally, since there should be only one detected face for each partition, the four corners of the final boundary region are the average of the corners of all detected faces in the subset. In addition, reducing multiple detected faces into a single detected face decreases the number of false positives.

4.4. Experiments on real-world face images

All color and gray images are converted to edge version in the first step of face detection. The normalized face size after face detection is 128×128 pixels. The NVM consists of four MLP networks for detecting four essential features. There are seven nodes with respect to facial parameters in input layer and eighty nodes in hidden layer for each facial feature. Since the algorithms are based on the position and face shape information, faces and facial features can be detected without regarding image appearance. The results can be categorized by the factors that affect the appearance of image as shown in Table 1. Fig. 11 shows the examples of the face and facial feature detection results corresponding the factors in Table 1.

Table 1
The scope of the factors affecting the image appearance

Attributes	Scope
Intensity	Color image (Fig. 11(a)), gray image (Fig. 11(b)), binary image (Figs. 11(c) and (d))
Structural components	Face with glasses (Figs. 11(e) and (g)), face with beards and mustaches (Figs. 11(f) and (g))
Rotation	Rotated face with 0°-360° in clockwise direction (Figs. 11(h)–(k))
Poor quality	Blurry image (Fig. 11(l)), distorted image (Fig. 11(m)), noise-added image (Fig. 11(n))
Facial expression	Angry face (Fig. 11(o)), excited face (Fig. 11(p)), surprised face (Fig. 11(q)), happy face (Fig. 11(r))
Unnatural intensity	Cartoon image (Fig. 11(s)), rendered image (Fig. 11(t)), mosaic image (Fig. 11(u))
Occlusion	Occluded mouth (Fig. 11(v)), occluded eyes (Fig. 11(w))
Illumination	Image with any direction of light source (Figs. 11(x)–(z))

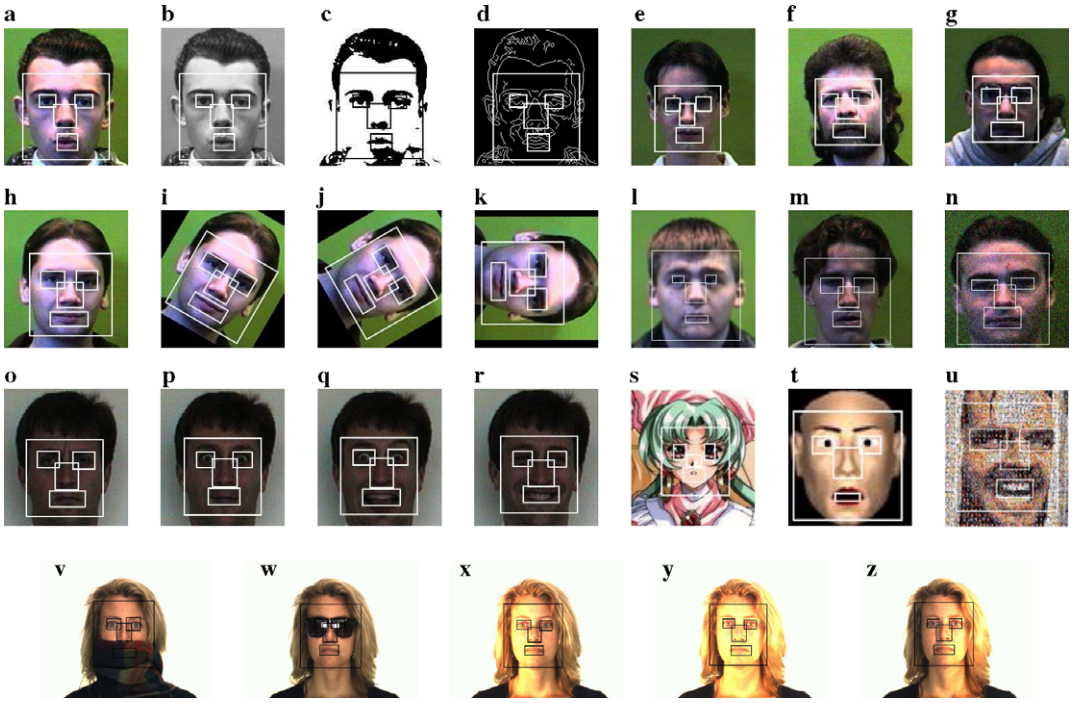

Fig. 11. Detection results of images. (a) Color image. (b) Gray image. (c) Black-and-white image. (d) Edge image. (e) Face with glasses. (f) Face with beards and mustaches. (g) Face with glasses, beards, and mustaches. (h) 0° rotated face. (i) 30° rotated face. (j) 60° rotated face. (k) 90° rotated face. (l) Blurry image. (m) Distorted image. (n) Image with 10% Gaussian noise. (o) Angry face. (p) Excited face. (q) Surprised face. (r) Happy face. (s) Colored cartoon image. (t) Rendered image. (u) Mosaic image. (v) Face with scarf. (w) Face with sunglasses. (x) Image with left light source. (y) Image with front light source. (z) Image with right light source.

Generally, there are more than one factors simultaneously affecting the appearance of face image. Fig. 12 shows the detection results of such images. Some images contain many faces as in Figs. 12(e), (j), (m), and (n). Some include rotated faces as in Figs. 12(b)(−3°), (e)(−10°), (i)(−4°), (j)(7°), (k)(6°), (m)(5°), and (n)(8°, 6°, 2°, −4°). Some contain slightly posed faces as in Figs. 12(h) and (i). Some contain unnatural human faces as in Figs. 12(a), (e), (h), (k), and (m). Some contain some occlusions as in Figs. 12(d), (j), (l), and (o), or structural components as in Fig. 12(f) (glasses). Some contain illumination effect as in Fig. 12(b). Some contain facial expressions as in Figs. 12(f), (k), (m), and (n). Some contain dark skins as in Figs. 12(g) and (l). Some are like a negative film as in Fig. 12(c).

Typically, edge appearance, which is used as input of face detection, is sensitive to illumination, complex background and skin wrinkles. On the other hand, the face detection algorithm is robust to those factors as follows:

- *High illumination.* The illumination can be categorized into high and low illuminations. In case of high illumination, higher illumination generates image with higher brightness (intensity) and lower contrast. Details on face, e.g. facial features and skin wrinkles, are faded. For an ordinary edge detection, a small number of edges are detected from image with high brightness. However, the Canny method is more prominent edge detection than others because it can detect weak edges, which are obtained from the regions with high brightness. This roughly reduces the effect of high illumination. The comparison of edge detection results is shown in Fig. 13. In addition,

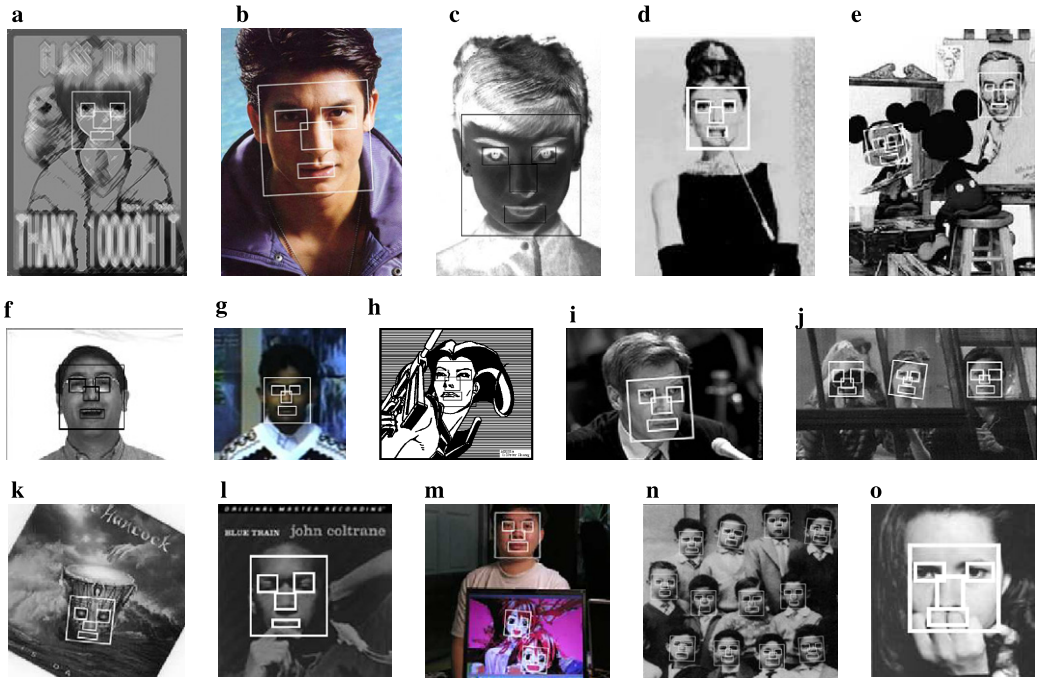

Fig. 12. More results. (a) Gray cartoon by colored pencil. (b) Rotated face with illumination effect. (c) Inverted black-and-white face like film negative. (d) Gray image. (e) Cartoon and painted face with rotation and facial expression. (f) Gray face with glasses and facial expression. (g) Color dark-skin face. (h) Sketchy image. (i) Slightly posed face with rotation. (j) Multifaces with occlusion and rotation. (k) Rotated unnatural face. (l) Gray dark-skin face with dark background. (m) Multifaces including cartoon faces. (n) Multifaces with rotation. (o) Occluded face.

Fig. 13. The comparison of edge detection results. (a) Original image with high illumination. (b) Edge image by using Canny method. (c) Edge image by using Sobel method.

the effect of high illumination is more discarded in the face detection process. The face is detected by the algorithm based on two following observations: (1) the number of white pixels of the edge version of the considered face must be close to those of the edge version of the face template and (2) the density of the white pixels is high within the facial feature-like areas. Step 4 of the face detection algorithm refers to the first observation. The first candidate face region is the region that has the number of white pixels less than twice of number of white pixels in the edge face template. In this step, the face with faded edges is still be considered as the first candidate face region. The number of feature pixels appearing in the facial feature-like areas for each first candidate face region is counted in step 5. In step 6, which is based on the second observation, the number of feature pixels of each first candidate face region is proportionally compared to that of the region that has the maximum number of feature pixels. This can tolerate the effect of illumination that causes the number of edge pixels smaller or larger. Like step 6, the matching value in step 8 is calculated and proportionally compared to that of the region that has the maximum matching value. Fig. 14 shows face detection results of images with high illumination and normal illumination.

- *Low illumination and skin wrinkles.* With low illumination, the details on face, especially skin wrinkles, evidently appear. Moreover, some faces contain skin wrinkles in normal illumination. The number of white pixels of edge face with low illumination or skin wrinkles is naturally large, but it is still less than twice of number of white pixels in the edge face template (step 4). From this reason, the edge face is considered as the first candidate region. However, the white pixels appearing outside the facial feature-like

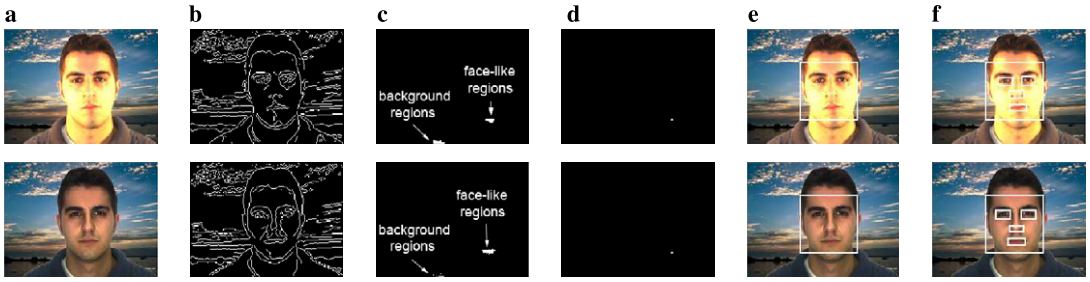

Fig. 14. The comparison of face detection results of images with high illumination (upper row) and normal illumination (lower row). (a) Original images with high and normal illuminations. (b) Edge images by using Canny method. The prominent difference is cheek skin wrinkles disappeared in image with high illumination. (c) The second candidate face regions by step 6 of the face detection algorithm (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 $l \times l$). Some second candidate regions are background regions. (d) Each white pixel corresponding to each detected region. (e) Face detection results corresponding to white pixels in (d). (f) Face and facial feature detection results.

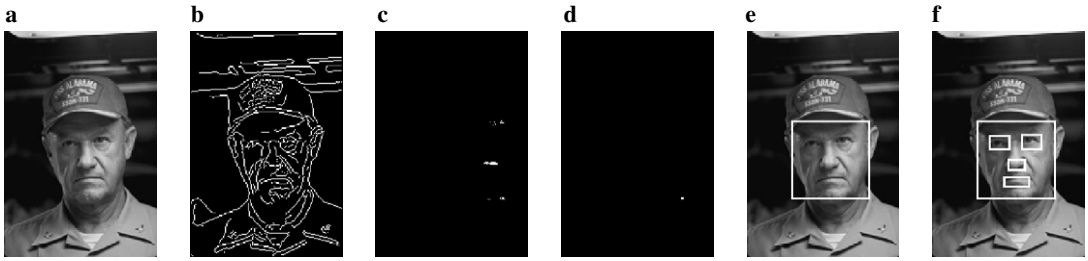

Fig. 15. The result of image with skin wrinkles. (a) Original image. (b) Edge image by using Canny method. The wrinkles undoubtedly appear. (c) The second candidate face regions by step 6 of the face detection algorithm (white pixel at position (i,j) corresponds to the square candidate region between vertical pixels $i-l+1$ and i, and horizontal pixels $j-l+1$ and j, where the template size is $l \times l$). (d) A white pixel corresponding to a detected region. (e) Face detection result corresponding to a white pixel in (d). (f) Face and facial feature detection result.

areas are not considered as the essential features, so the effect of skin wrinkles is neglected in steps 5 and 6. Fig. 15 shows the results of image with skin wrinkles.

- *Complex background.* Usually, the region containing complex background is the first candidate face region if this region contains the number of white pixels less than twice of number of white pixels in the edge face template. Moreover, the number of white pixels in the facial feature-like areas is large, so this region can also be the second candidate face region as appeared in Fig. 14(c). In step 7, Eq. (4) is used to evaluate the matching value, δ. This value of any considered region will be high when the number of white pixels, ϕ, is small and the number of feature pixels, ϵ, is large. This means that the face-like region is the region that most of white pixels appear in the facial feature-like areas while a small number of white pixels, which may be some skin wrinkles, are outside those areas. However, both of the number of white pixels and the number of feature pixels of the region containing complex background are large, so the matching value is not high enough to mark the region as the face-like region. Fig. 14(d) shows the images that the background regions are eliminated.

A conclusion of the face and facial feature detection results (including detection rates and number of false positives) is presented in Fig. 16 and Table 2. To create the

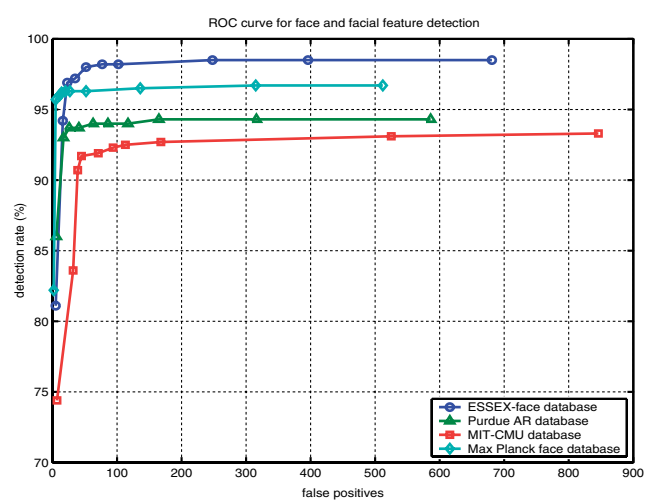

Fig. 16. ROC curves for the proposed face and facial feature detection (FFD) on each database.

receiver operating characteristic (ROC) curves of the proposed face and facial feature detection (FFD), two thresholds in steps 6 and 8 of the face detection algorithm (algorithm 1) and four thresholds of four MLP classifiers used for detecting the facial features are adjusted in the range between 0 and 1. Higher thresholds yield detector with fewer false positives and lower detection rate. Lower thresholds yield detector with more false positives and higher detection rate. For each database, the best results

Table 2
The results of both face detection (FD) and face and facial feature detection (FFD) of each database

Test set	ESSEX	Purdue-AR	MIT-CMU	Max-Planck
No. of images	397	300	130	600
No. of faces	397	300	507	600
No. of FPs (FD)	38	25	52	8
No. of FPs (FFD)	23	17	39	5
FD rate (%)	97.7	95.7	97.5	98.3
FFD rate (%)	96.9	93.0	90.7	95.7

FP is shorted from false positive.

Table 3
Comparative face detection performance of the existing methods and the proposed method on MIT-CMU testset containing 130 images and 507 faces

Method	Detection rate (%)	False detections
Proposed method	97.5	52
Schneiderman–Kanade	94.4	65
Rowley–Baluja–Kanade	90.1	167
Viola–Jones	93.1	65
Xiao–Li–Zhang	92.0	135
Huang–Shimizu–Hagihara–Kobatake	86.0	53

Facial feature detection is not considered here.

of both face detection (FD) and face and facial feature detection (FFD), which is a single point on each ROC curve, are shown in Table 2. A detected face is a correct detection if the detected locations of features and square block bounding a face are extracted with a small amount of tolerance, otherwise the false positive is accounted. The face and facial feature detection rate is calculated from the ratio of the number of correct detections in a database to those of all faces in the database. Unlike face and facial feature detection, the face detection rate is calculated from the ratio of the number of correct face detections without considering any facial features to those of all faces in a database. Usually, the face and facial feature detection rate is lower than the face detection rate. In the case of false positive, the error is caused by the edge regions that is similar to edge face in face detection stage, but it can be reduced in facial feature detection stage. For the AR [27] and the Max Planck [29] databases, the number of false positives is very small because each image has one face and single color background.

Failure cases can be categorized into five major classes. The first failure class comes from the unusual positions of facial features or unnatural faces as depicted in Fig. 17(a). The second class consists of the faces without face boundaries. Some facial parameters such as face width are not able to be obtained by the extracting facial parameters algorithm. An example of this class is shown in Fig. 17(b). The third class occurs when the boundary regions of multiple faces are overlapped. A single detected face is computed by averaging the corners of all detected faces. This causes failure detection as shown in Fig. 17(c). The fourth class comes from partial face, where face information is not enough for detection as the rightmost face in Fig. 17(c). Finally, the last failure class is a set of very small images with low contrast such as some small faces in Fig. 17(d).

Unlike Table 2 where both of face detection and face and facial feature detection are considered, Table 3 shows only the face detection rate and the number of false positives for our method as well as other existing methods. The Schneiderman–Kanade method [30] achieves 94.4% with 65 false positives. For the Rowley–Baluja–Kanade results [28], there are a number of different versions of their detector tested yielding a number of different results, but their best detection rate is 90.1% with 167 false positives. For the Viola–Jones method [12], detection results are between 81.1% and 93.7% with a number of different values of false positives varying from 10 to 422. The detection rate with 65 false positives is selected. For the Xiao–Li–Zhang [9], the best detection rate is 92% with 135 false positives. For the Huang–Shimizu–Hagihara–Kobatake [13], the best classifier yields 86.0% detection rate and 53 false positives. Our detection rate is 97.5% with 52 false positives.

5. Conclusions

This paper presents a novel face and facial feature detection algorithms that are applicable to any image types under various conditions and rotations with any appearances. The algorithms can be applied to any face images with high detection rate and low false positive rate. The algorithms separate into two main stages. The first stage is face detection. The existing face detection methods require intensity information causing some errors of face positions when input images are distorted or occluded by some objects. To overcome such shortfalls, the intensity information should be avoided. Mean face templates obtained from a large collections of face in various rotated

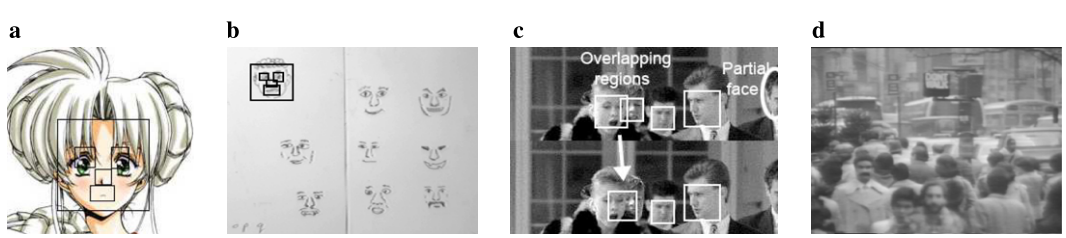

Fig. 17. Failure cases. (a) Unusual eye positions. (b) Faces without face boundaries. (c) Overlapping boundary regions (upper image) causing a failure detected face (lower image) and partial face(rightmost face). (d) Very small faces with extremely low contrast.

angles and sizes (from 21×21 to 393×393 pixels), are used to scan all around edge image to find face-like region. The second stage is facial feature detection and is applied to detected faces. Facial features are coarsely extracted from neural visual model. Some irrelevant regions from the NVM are eliminated by applying mathematical morphology called dilation. Furthermore, the proposed algorithms are extended to cover rotational invariance problem by using Radon transformation to extract the main face angle. Detection results on several databases have been performed in all classes of face images with detection rate more than 94% while the detection rate of other existing techniques is about 90%.

Acknowledgements

This research is financially supported by Thailand Research Fund through the Royal Golden Jubilee Ph.D. Program. We thank Dr. Petr Musilek for providing some computing facilities and the reviewers for their valuable comments.

References

[1] M. Yang, D.J. Kriegman, N. Ahuja, Detecting faces in images: a survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (1) (2002) 34–58.
[2] S. Zhang, Z. Liu, A robust, real-time ellipse detector, Pattern Recognition 38 (2) (2005) 273–287.
[3] K. Anderson, P.W. McOwan, Robust real-time face tracker for cluttered environments, Computer Vision and Image Understanding 95 (2) (2004) 184–200.
[4] F.Y. Shih, C. Chuang, Automatic extraction of head and face boundaries and facial features, Information Sciences 158 (1) (2004) 117–130.
[5] Z. Liu, J. Yang, N.S. Peng, An efficient face segmentation algorithm based on binary partition tree, Signal Processing: Image Communication 20 (4) (2005) 295–314.
[6] F. Tsalakanidou, S. Malassiotis, M.G. Strintzis, Face localization and authentication using color and depth images, IEEE Transactions on Image Processing 14 (2) (2005) 152–168.
[7] M. Soriano, B. Martinkauppi, S. Huovinen, M. Laaksonen, Adaptive skin color modeling using the skin locus for selecting training pixels, Pattern Recognition 36 (3) (2003) 681–690.
[8] R.L. Hsu, M. Abdel-Mottaleb, A.K. Jain, Face detection in color images, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 696–706.
[9] R. Xiao, M. Li, H. Zhang, Robust multipose face detection in images, IEEE Transactions on Circuits and Systems for Video Technology 14 (1) (2004) 31–41.
[10] C. Liu, A Bayesian discriminating features method for face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (6) (2003) 725–740.
[11] Y. Li, S. Gong, J. Sherrah, H. Liddell, Support vector machine based multi-view face detection and recognition, Image and Vision Computing 22 (5) (2004) 413–427.
[12] P. Viola, M.J. Jones, Robust real-time face detection, International Journal of Computer Vision 57 (2) (2004) 137–154.
[13] L. Huang, A. Shimizu, Y. Hagihara, H. Kobatake, Gradient feature extraction for classification-based face detection, Pattern Recognition 36 (11) (2003) 2501–2511.
[14] M. Liévin, F. Luthon, Nonlinear color space and spatiotemporal MRF for hierarchical segmentation of face features in video, IEEE Transactions on Image Processing 13 (1) (2004) 63–71.
[15] E. Loutas, I. Pitas, C. Nikou, Probabilistic multiple face detection and tracking using entropy measures, IEEE Transactions on Circuits and Systems for Video Technology 14 (1) (2004) 128–135.
[16] S. Spors, R. Rabenstein, A real-time face tracker for color video, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001, pp. 1493–1496.
[17] S. Phimoltares, C. Lursinsap, K. Chamnongthai, Locating essential facial features using neural visual model, in: Proceedings of the First IEEE International Conference on Machine Learning and Cybernetics, 2002, pp. 1914–1919.
[18] S. Phimoltares, C. Lursinsap, K. Chamnongthai, Facial feature extraction with rotational invariance using neural visual model, in: Proceedings of the Third International Conference on Intelligent Technologies, 2002, pp. 226–234.
[19] S. Phimoltares, C. Lursinsap, K. Chamnongthai, Tight bounded localization of facial features with color and rotational independence, in: Proceedings of the IEEE International Symposium on Circuits and Systems, 2003, pp. V-809–V-812.
[20] F. Fleuret, D. Geman, Coarse-to-fine face detection, International Journal of Computer Vision 41 (1–2) (2001) 85–107.
[21] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (6) (1986) 679–698.
[22] L. Spacek, Essex face database, 2000. Available from: <http://cswww.essex.ac.uk/allfaces/index.html/>.
[23] J. Ng, S. Gong, Composite support vector machines for detection of faces across views and pose estimation, Image and Vision Computing 20 (5–6) (2002) 359–368.
[24] L. Prechelt, Automatic early stopping using cross validation: quantifying the criteria, Neural Networks 11 (4) (1998) 761–767.
[25] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley, Reading, MA, 1993.
[26] P. Toft, The Radon transform, 1997. Available from: <http://eivind.imm.dtu.dk/staff/ptoft/Radon/Radon.html/>.
[27] A.M. Martinez, R. Benavente, The AR face database, CVC Technical Report#24, 1998. Available from: <http://rvl1.ecn.purdue.edu/~aleix/aleix_face_DB.html/>.
[28] H.A. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1) (1998) 23–38.
[29] V. Blanz, T. Vetter, A morphable model for the synthesis of 3D faces, in: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, 1999, pp. 187–194.
[30] H. Schneiderman, T. Kanade, A statistical method for 3D object detection applied to faces and cars, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2000, pp. I:746–I:751.