Automatic Facial Feature Extraction by Genetic Algorithms

Chun-Hung Lin and Ja-Ling Wu, *Senior Member, IEEE*

***Abstract*— An automatic facial feature extraction algorithm is presented in this paper. The algorithm is composed of two main stages: the *face region estimation* stage and the *feature extraction* stage. In the face region estimation stage, a second-chance region growing method is adopted to estimate the face region of a target image. In the feature extraction stage, genetic search algorithms are applied to extract the facial feature points within the face region. It is shown by simulation results that the proposed algorithm can automatically and exactly extract facial features with limited computational complexity.**

***Index Terms*— Facial feature extraction, genetic search algorithm, head-and-shoulder images, low bit rate video coding.**

I. Introduction

FACIAL feature extraction plays an important role in many applications, such as very low bit-rate video coding [1], [2] and human face recognition [3]–[5]. In very low bit-rate video coding, the variation of facial features (the so-called local motion) carries much more information than the other parts of the target image do. In order to obtain higher coding gain, the facial feature areas are usually separated from the target image, and then special coding schemes are performed on these areas. As for human face recognition, facial features are extracted as the major analytical targets because they compose the characteristics of a human face.

It is still a complex issue to extract facial features automatically nowadays, although it might not be difficult for people to perceive human faces and facial features from a digital image. In order to reduce the complexity of facial feature extraction systems, the input images are usually restricted to typical head-and-shoulder images with plain background. In general, the tasks of facial feature extraction can be divided into two stages: the *face region estimation* stage and the *feature extraction* stage.

In the face region estimation stage, the face region is extracted from a target image. For a head-and-shoulder image, the face region is located roughly in the center of the image. Moreover, the face region usually exhibits higher contrast than its surrounding neighborhoods do. It can therefore be extracted by region segmentation or edge detection techniques.

In the feature point extraction stage, the desired facial features, such as eyes, nose, and mouth, will be extracted from the face region (found in the above stage) by the methods of template matching. One or several feature templates can be previously defined based on the characteristics of human facial features. By using the feature templates, all the candidate points within the face region are evaluated based on the predefined cost functions. Those points that have the best matching values are extracted as the feature points.

Although many facial feature extraction algorithms have been proposed so far [6]–[9], facial feature extraction is still a handicap to the prescribed applications due to its high complexity. The computational cost of facial feature extraction is dominated by the following three sources : 1) the estimation of the face region, 2) the complexity of the evaluation of cost functions, and 3) the searching for the feature points.

It is common to apply region segmentation methods or edge detection methods on the whole image to estimate the face region, such as chrominance thresholding and combining in [7], and edge pixel counting in [9]. Although they were shown to be able to extract the face region in a facial image, the computational cost is high. On the other hand, many of the computations performed on the background could be avoided because it has been known that the face region is located roughly in the center of a facial image. In this work, a second-chance region growing method is used, which will be detailed in Section II-A. That method can extract the face region with relatively low computational complexity.

Another critical issue is the design of feature templates. Because feature templates are employed to calculate the matching value for each possible feature point within the face region, the complexity of the feature templates dominates the computational cost of the overall feature extraction algorithm. An obvious drawback of complex feature templates is that they would not be robust when some subregions of the face exhibit low contrast, for example, the contrast between irises and lenses might be low due to the shadow of hair. In this paper, a generic feature template is proposed to extract all the critical feature points. The structure of the adopted generic feature template is simple, hence low computational complexity can be expected.

In the existing algorithms, all the pixels within the face region were evaluated to find the best ones, that is, they did not apply any fast searching method. Genetic algorithms (GA's) [10]–[12] are adopted in our work to solve the most troublesome task of facial feature extraction, the template matching. Due to its good reputation in the applications of searching and

Manuscript received March 25, 1996; revised July 9, 1998. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Eric Dubois.

The authors are with the Communication and Multimedia Labortory, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 106, Taiwan, R.O.C. (e-mail: wjl@cmlab.csie.ntu.edu.tw).

Publisher Item Identifier S 1057-7149(99)04063-4.

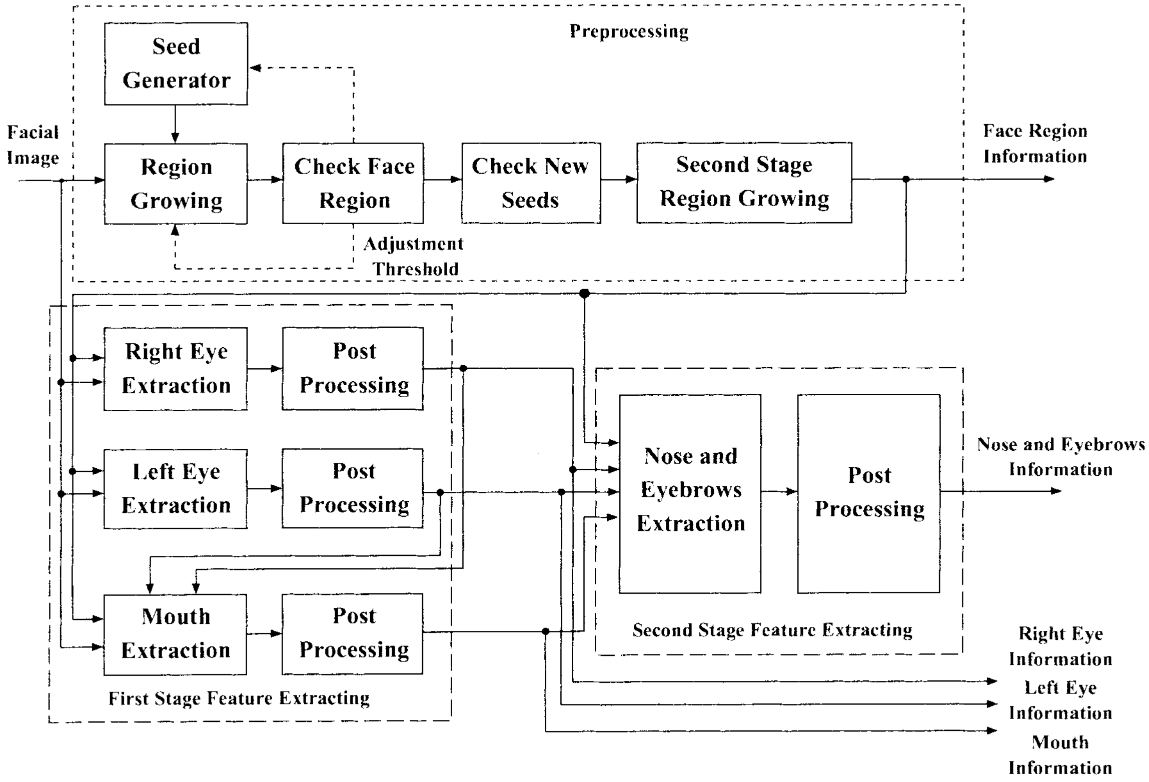


Fig. 1. Block diagram of the proposed automatic feature extraction algorithm.

optimization, good template matching performance (in terms of speed and matching error) can be expected. Although the results are not guaranteed to be global optima by applying GA's in searching, the expected feature points can always be extracted, in our simulations, with the aid of some intrinsic geometric relations among facial feature points.

This paper is organized as follows. In Section II, the system architecture of the proposed algorithm is introduced. The generic feature template and its associated cost functions are also elaborated. In Section III, we depict the method of applying GA's in feature point searching. The experimental results are shown in Section IV, and conclusions are given in Section V.

II. An Automatic Facial Feature Extraction Algorithm

Fig. 1 shows the block diagram of the proposed facial feature extraction algorithm. The system is composed of three main components: 1) *preprocessing,* 2) *first stage feature extracting,* and 3) *second stage feature extracting*. In the preprocessing stage, the face region is estimated by the second-chance region growing method. Then, facial features are searched within the estimated face region according to their matching values. In the first stage feature extracting, the main features (eyes and mouth) are extracted from the face region. When the main features have been found, in the second stage feature extracting, the other features (eyebrows and nose) are also extracted based on the intrinsic information of the main features. The details are described in the following subsections.

A. Preprocessing

The purpose of the preprocessing stage is to identify the face region. The feature extraction algorithm can then be applied only on this region. It therefore improves the efficiency of the system. The second-chance region growing method is used here. In our simulations, we found that the face region can be estimated well by the second-chance region growing method, which will be described as follows. Notice that, the computational complexity of this method is very low.

Because the face region is located roughly in the center of a head-and-shoulder image, the region growing algorithm [14] can be performed to estimate that region by selecting the central point of the image as an initial seed. Assume the initial seed is represented as s_0, and the region grown from s_0 is R_0. The size of R_0 must be checked to make sure that it will be a reasonable face region. The size of R_0, denoted by $|R_0|$, should be restricted to a bounded range, that is $T_l \leq |R_0| \leq T_h$, where T_l and T_h are predefined constants.

Sometimes it is possible that the central point of the facial image belongs to a feature region (e.g., nostril). In this case, s_0 will be the feature region, instead of the face region, and $|R_0|$ is small. On the other hand, the size of the grown region depends on the predetermined growing threshold. When the contrast between the face region and the background region is low and the threshold is set larger than a suitable value, the associated grown region will cover some background regions. In this case, $|R_0|$ will be relatively large.

If $|R_0|$ is beyond the range $[T_l, T_h]$, either a new seed should be selected (a new seed s_1 can be selected from the neighboring pixels of s_0), or the thresholds should be adjusted, to

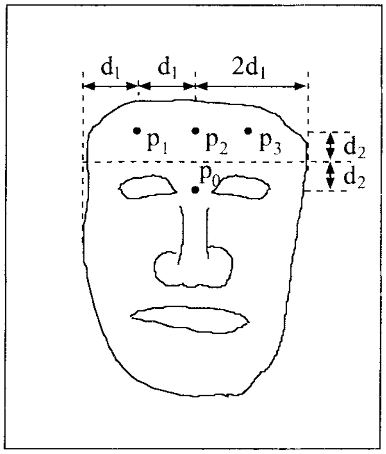


Fig. 2. Pivot point p_0 and the three possible seed points p_1, p_2, and p_3.

generate a new region R_1. The size of R_1 is checked again, and another new region should be generated if $|R_1| < T_l$ or $|R_1| > T_h$. This process will be iterated until a region R_i is found, where $T_l \leq |R_i| \leq T_h$. If the desired face region is represented as R, R_i will be equal to R or be a subset of R, i.e., $R_i \subseteq R$.

It might be difficult to detect the forehead region by the region growing method because that region is usually separated from the face region by hair and eyebrows (or eye glasses). In order to make sure that the forehead region is included in R, the intensities of the three extended points (as shown in Fig. 2) are compared with that of the pivot point p_0. If their intensity difference is less than a threshold, i.e., $|I(p_j) - I(p_0)| \leq T_r$, $j = \{1, 2, 3\}$, p_j will be selected as a new seed. The new region R_{p_j} that is grown from p_j, is checked and combined with R_i if it satisfies the following conditions: 1) its width is less than that of R_i, i.e., $W(R_{p_j}) < W(R_i)$; and 2) its height is less than half of the height of R_i, i.e. $H(R_{p_j}) < \frac{1}{2}H(R_i)$. In other words, $R = R_i \cup R_p$, where R_p could be one of the eight subsets of the set $\{R_{p_1}, R_{p_2}, R_{p_3}\}$. The checking process prevents from including the hair region or the background region into the face region. Because parts of the forehead region might be covered by hair, it is comprehensible to consider the three possible seeds at the same time. This second-chance region growing process avoids the possibility of losing the forehead region from the face region.

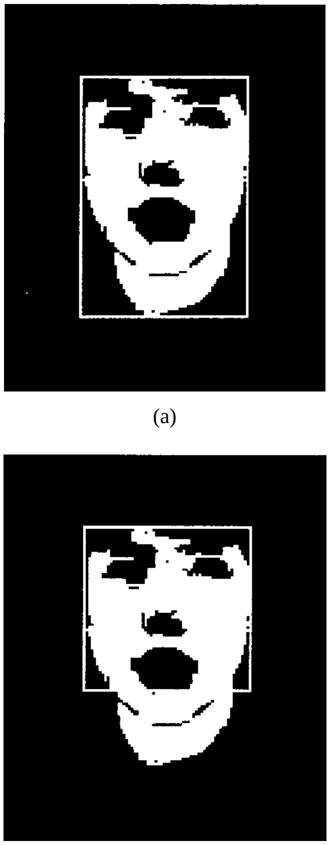
When the combined region R is extracted, a minimum rectangle can be found to surround R as shown in Fig. 3(a). It can be found from Fig. 3(a) that R may contain the neck region. To exclude the neck region from R, the lower boundary of the rectangle is adjusted, so that the height of the rectangle is equal to the its width. The result of the rectangle adjustment is shown in Fig. 3(b). The following feature extraction algorithm will be performed within the so-obtained square area to find the facial features.

Assume $(C_x(A), C_y(A))$ represents the central point of the target image A; the second-chance region growing algorithm is summarized at the bottom of the page.

B. Generic Feature Template and Its Associated Cost Functions

As shown in Fig. 4, a generic feature template is defined to extract all the critical facial features, including eyes, mouth,

```
$i := 0$
$s_i := (C_x(A), C_y(A))$
$R_i :=$ Region_Growing$(s_i)$
While $|R_i| < T_l$ or $|R_i| > T_h$ Loop
    $i := i + 1$
    $m := \lfloor \sqrt{i} \rfloor$
    $x := (-1)^m([i - m(m+1)] \cdot [1 - (\lfloor 2\sqrt{i} \rfloor \text{ mod } 2)] + \lceil \frac{1}{2}m \rceil)$
    $y := (-1)^m([i - m(m+1)] \cdot [\lfloor 2\sqrt{i} \rfloor \text{ mod } 2] - \lceil \frac{1}{2}m \rceil)$
    $s_i := (C_x(A) + x, C_y(A) + y)$
    $R_i :=$ Region_Growing$(s_i)$
End Loop
$p_0 := (C_x(R_i), C_y(R_i) - H(R_i)/2 + d_2)$
$p_1 := (C_x(R_i) - d_1, C_y(R_i) - H(R_i)/2 - d_2)$
$p_2 := (C_x(R_i), C_y(R_i) - H(R_i)/2 - d_2)$
$p_3 := (C_x(R_i) + d_1, C_y(R_i) - H(R_i)/2 - d_2)$
for j:=1 to 3 Loop
    If $|I(p_j) - I(p_0)| \leq T_r$ Then
        $R_{p_j} :=$ Region_Growing$(p_j)$
        If $W(R_{p_j}) \geq W(R_i)$ Or $H(R_{p_j}) \geq \frac{1}{2}H(R_i)$ Then
            $R_{p_j} := \phi$
        End If
    Else
        $R_{p_j} := \phi$
    End If
End Loop
$R := R_i \cup R_{p_1} \cup R_{p_2} \cup R_{p_3}$.
```



(a)

(b)

Fig. 3. Extraction of the face region. (a) The face region is bounded by a rectangle. (b) The rectangle area is adjusted to exclude the neck region.

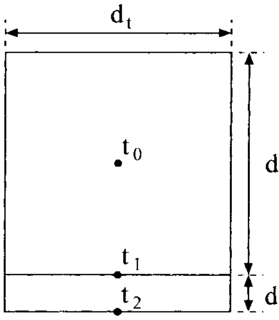


Fig. 4. Generic feature template.

eyebrows, and nose. Because the template is adapted to different facial features, it is difficult to determine the exact sizes for the width and the height of the template. Moreover, the shape of the template is not important in our approach. Therefore, a square template is used for matching. When the target feature is found, the shape can be adjusted in the postprocessing stage later.

The size of the template depends on the width of the face region. In our implementation, d_t equals one quarter of the width of the face region, and the length of the extended area, d_e, is set to be $\frac{1}{8}d_t$. Assume the central point of the square area is t_0, whose coordinate is (x, y), two pivot points t_1 and t_2 below t_0 are selected, where $t_1 = (x, y + d_t/2)$ and $t_2 = (x, y + d_t/2 + d_e)$. Every possible feature point within the feature region is evaluated based on the cost functions given in (1) and (11), and its matching value depends on the intensity values of the neighboring points masked by the feature template. The candidate point with the best matching value in each feature subregion is selected as the associated feature point.

The cost function that will be used in the first stage feature extracting is defined as

$$F(t_0) = C_m(t_0) + U_{I(t_1)-I(t_0)-K} \cdot U_{I(t_2)-I(t_0)-K} \cdot C_c(t_0) \tag{1}$$

where t_0 is the evaluated candidate feature point, C_m is the mean crossing function (which is used to accumulate the intensity variation within the masked area), C_c is the central weighting function (which is used to compute the intensity difference between the central part and the boundary parts of the masked area), K is a constant, $I(p)$ means the intensity of the point p, and U is a unit step function. Only the points covered by the square part of the template will affect the function values of C_m and C_c.

The mean crossing function is applied to detect the intensity variation within the square area. We can take $(d_t + 1)$ one-dimensional (1-D) signals from the square area row-by-row, and another $(d_t + 1)$ 1-D signals column-by-column [as shown in Fig. 5(a)]. The corresponding mean crossing values for these 1-D signals are then calculated. The total mean crossing value C_m is defined to be the sum of these mean crossing values, that is

$$C_m(x, y) = \sum_{i=-d_t/2}^{d_t/2} [V_r(x, y + i) + V_c(x + i, y)] \tag{2}$$

where (x, y) is the coordinate of the evaluated point, and V_r and V_c are the mean crossing values of the row signal and the column signal which are defined, respectively, as

$$V_r(x, y) = \sum_{i=-d_t/2+1}^{d_t/2} U_{-S_r(x+i,y)\cdot S_r(x+i-1,y)-1}, \tag{3}$$

$$V_c(x, y) = \sum_{i=-d_t/2+1}^{d_t/2} U_{-S_c(x,y+i)\cdot S_c(x,y+i-1)-1} \tag{4}$$

and (5) and (6), shown at the bottom of the next page.

As shown in Fig. 5(b), the mean value μ_m is computed first for each of the evaluated 1-D signals, and then a bar area, $(\mu_m - K_m, \mu_m + K_m)$, is formed, where K_m is a constant. The number of μ_m-crossings of the 1-D signal within the bar area is calculated. If the intensity variation of the signal is large, the mean crossing value will also be large. Because the crossing value is computed on the bar area, the slight intensity variation caused by noise will be neglected. While comparing with the other nonfeature areas, the intensity variations in the feature areas are relatively large and so are the corresponding mean crossing values. For the sake of saving computations, we only extract ten lines (five for each direction) near the center of the square area to calculate the crossing values, as shown in Fig. 5(c).

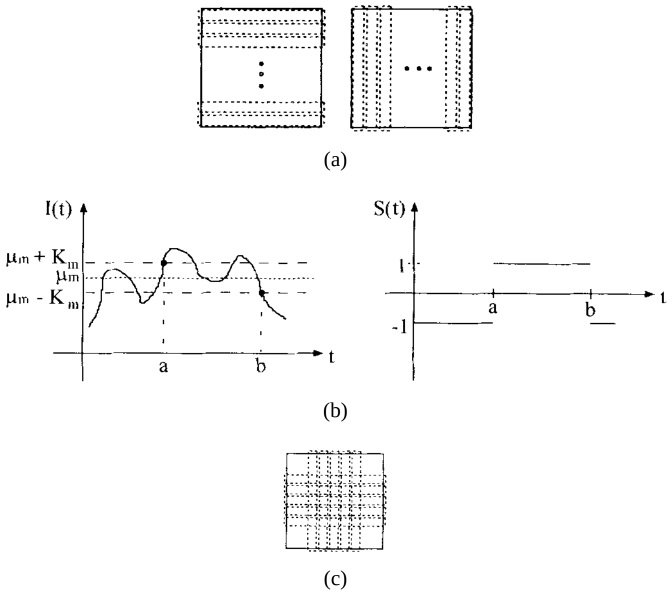


Fig. 5. Mean crossing evaluation. (a) One-dimensional signals can be obtained from the square area, row-by-row and column-by-column. (b) The mean crossing numbers are calculated for each of the 1-D signal. (c) The simplified 1-D signal extraction pattern is adopted in the implementation.

The central weighting function is included in the cost function because of the fact that the intensity difference between the central part and the boundary parts of the feature area is large. As shown in Fig. 6, three mean values μ_c, μ_u, and μ_l are calculated for the three different subregions (B_0, B_1, and B_2) within the square area. The central weighting function is defined as

$$C_c(t_0) = \left[K_c \cdot \left(\frac{\mu_u + \mu_l}{2}\right) + \mu_l - \mu_u\right] \Big/ \mu_c \tag{7}$$

where

$$\mu_c = \frac{1}{|B_0|} \sum_{p \in B_0} I(p), \tag{8}$$

$$\mu_u = \frac{1}{|B_1|} \sum_{p \in B_1} I(p), \tag{9}$$

$$\mu_l = \frac{1}{|B_2|} \sum_{p \in B_2} I(p) \tag{10}$$

and K_c is a constant.

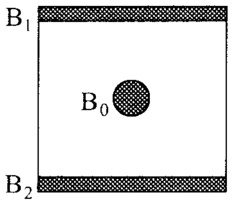


Fig. 6. Evaluation of the central weighting.

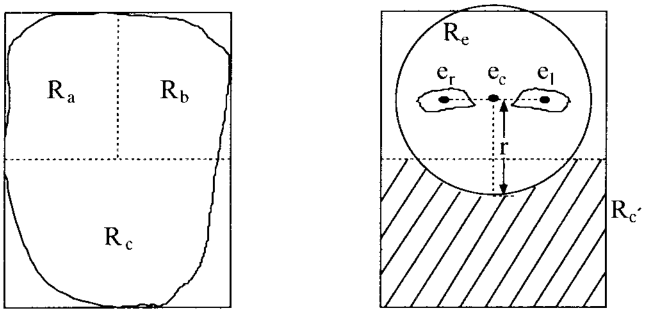


Fig. 7. (a) Three subregions are selected from the face region to extract the main features in first stage feature extracting. (b) The subregion R_c is shrunk to exclude the nose feature point.

In the first stage feature extracting, main features are extracted by applying the cost function defined in (1). It is obvious that the intensity difference between main feature points and those points below and near the main feature areas are large. Therefore, unit step functions are involved in (1) to prevent from selecting the other features (such as eyebrows and nose) in the first stage process.

The cost function defined in (1) is slightly modified to extract the other feature points in the second stage feature extracting. The cost function for the second stage feature extracting is defined as

$$F'(t_0) = C_m(t_0) + C_c(t_0) \tag{11}$$

where the unit step function terms have been taken off.

C. First Stage Feature Extracting

Basically, the face region can be divided into three subregions, as shown in Fig. 7(a). The subregion R_a is expected to contain the right eye feature point, the left eye feature point is expected to appear in R_b, and the mouth feature point is in R_c.

$$S_r(x+i,y) = \begin{cases} 0, & i = -\frac{d_t}{2}, \\ 1, & I(x+i,y) - \mu_m > K_m, \quad \text{and} \quad i > -\frac{d_t}{2}, \\ -1, & I(x+i,y) - \mu_m < -K_m, \quad \text{and} \quad i > -\frac{d_t}{2}, \\ S_r(x+i-1,y), & |I(x+i,y) - \mu_m| \le K_m, \quad \text{and} \quad i > -\frac{d_t}{2}, \end{cases} \tag{5}$$

$$S_c(x,y+i) = \begin{cases} 0, & i = -\frac{d_t}{2}, \\ 1, & I(x,y+i) - \mu_m > K_m, \quad \text{and} \quad i > -\frac{d_t}{2}, \\ -1, & I(x,y+i) - \mu_m < -K_m, \quad \text{and} \quad i > -\frac{d_t}{2}, \\ S_c(x,y+i-1), & |I(x,y+i) - \mu_m| \le K_m, \quad \text{and} \quad i > -\frac{d_t}{2} \end{cases} \tag{6}$$

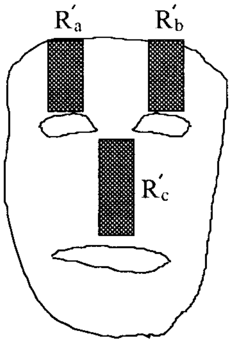


Fig. 8. Specified subregions R'_a, R'_b, and R'_c.

In the subregions R_a and R_b, the matching values of the eyebrow feature points, calculated by (1), are much lower than that of the eye feature points. The possibility of mistaking eyebrows for eyes is therefore very low in the first stage of feature extracting.

However, in the subregion R_c, the matching values of the nose feature point and the mouth feature point are very similar sometimes. Hence, it is possible that the nose feature point instead of the mouth feature is extracted from R_c. This situation can be avoided by separating the two features into two different subregions.

To exclude the nose feature point from R_c, the subregion should be modified after both eye feature points have been found. As shown in Fig. 7(b), the point e_c is the central point of the two eye feature points (e_r and e_l), and R_e is a round region with the central point e_c and a radius of r, where r is equal to the distance between e_r and e_l. The modified region [the shadow part in Fig. 7(b)] is

$$R_{c'} = R_c - R_e. \tag{12}$$

The mouth feature point is then searched in the subregion $R_{c'}$.

Because the face rotation angle is normally not large, the main feature points will not cross the boundaries of their associated subregions. By applying GA's on searching the possible feature points and finding the one with the best matching value in each subregion, the three main feature points of a human face can always be extracted.

D. Second Stage Feature Extracting

After the main features have been extracted, the other feature points are searched in the three specified subregions R'_a, R'_b, and R'_c, as shown in Fig. 8. The points with the best matching values, which are evaluated based on (11), are extracted as the second feature points.

E. Postprocessing

When the feature points have been extracted, we want to determine rectangle windows with different sizes to specify the feature areas. The adjustment of the windows is done in this postprocessing stage.

Fig. 9(a) shows a region grown image generated from the preprocessing stage. It can be used to estimate the initial window size for each feature area. As shown in Fig. 9(a), we can find all the black regions that contain the feature points, and specify rectangle windows to surround these black regions. The imprecise feature areas are then specified, as shown in Fig. 9(b).

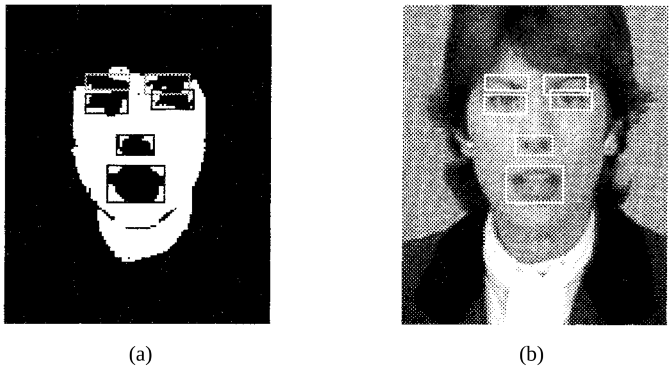


Fig. 9. The region grown image is used to estimate the feature areas, (a) the binary version of the target image (where the feature areas are bounded by rectangle windows), and (b) the target image and the corresponding feature areas.

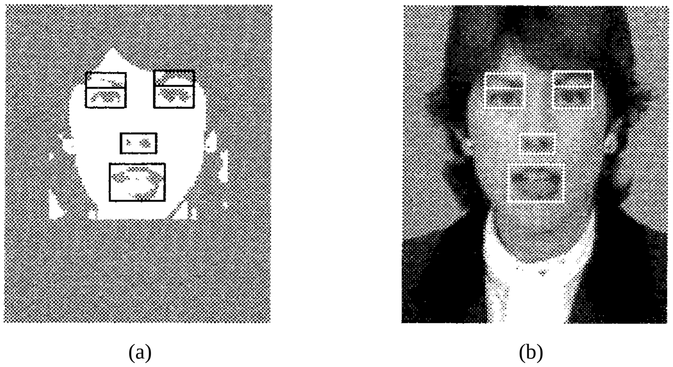


Fig. 10. The thresholded image is used to refine the feature areas, (a) the thresholded image (where the feature areas are bounded by refined windows), and (b) the target image and the corresponding feature areas.

In addition to the region grown image, a thresholded image is generated by converting the intensity values of the points within the face region to binary values. The conversion rule is

$$B(i,j) = \begin{cases} 255, & I(i,j) < T, \\ 0, & I(i,j) \geq T \end{cases} \tag{13}$$

where T is a constant. The thresholded image can be used to refine the feature areas. Fig. 10(a) shows the thresholded image that is generated from the same source of Fig. 9(a).

As an example, assume one of the central points of the initial feature windows is at (x, y), then the intensity accumulation of the four boundaries of the window that contains (x, y) can be calculated as

$$G_0 = \sum_{i=x-\frac{W}{2}}^{x+\frac{W}{2}} B\left(i, y - \frac{H}{2}\right), \tag{14}$$

$$G_1 = \sum_{i=x-\frac{W}{2}}^{x+\frac{W}{2}} B\left(i, y + \frac{H}{2}\right), \tag{15}$$

$$G_2 = \sum_{j=y-\frac{H}{2}}^{y+\frac{H}{2}} B\left(x - \frac{W}{2}, j\right), \tag{16}$$

$$G_3 = \sum_{j=y-\frac{H}{2}}^{y+\frac{H}{2}} B\left(x + \frac{W}{2}, j\right) \tag{17}$$

where W and H are the width and the height of the rectangle window, respectively. The rectangle window is shrinked until the condition (18) is satisfied:

$$G_0 \cdot G_1 \cdot G_2 \cdot G_3 \neq 0. \tag{18}$$

The other feature windows are also shrunk by following the same rule. By proper adjusting of the feature windows, feature areas with clear boundaries can be obtained as shown in Fig. 10(b).

III. Genetic Algorithms

Let S be a solution space where all the elements in S have their associated fitness values. The straightforward way to find the element with the maximum fitness value is to search among all the elements and to compare their fitness values. However, the computational complexity will be very high if the space size is large. In order to reduce the computational complexity, an efficient search algorithm should be applied.

If GA's are applied to search for the global maximum in S, a population P is maintained which consists of N elements, where N is the population size. Each element in P is called a chromosome which is composed of a list of genes. The population P will evolve into another population P' by performing some genetic operations. The chromosomes with higher fitness values will have more probability to be kept in the population of the next generation, and to propagate their offspring. On the other hand, the weak chromosomes whose fitness values are small will be replaced by other stronger chromosomes. Therefore, the quality of the chromosomes in the population will be better and better. After a suitable number of generations, the mature population will be expected to contain the element with the global maximum value.

In this application, the solution space S is a two-dimensional (2-D) image subregion, and each image pixel in the subregion is an element in S. Assume the central point of S is located at $(\hat{x}, \hat{y})$. The ith chromosome C_i in the population is defined as

$$C_i = \begin{bmatrix} m_i \\ n_i \end{bmatrix} = \begin{bmatrix} a_{i,k-1} & a_{i,k-2} & \cdots & a_{i,1} & a_{i,0} \\ b_{i,k-1} & b_{i,k-2} & \cdots & b_{i,1} & b_{i,0} \end{bmatrix}, \quad i = 0, 1, \cdots, N-1 \tag{19}$$

and the relative location is

$$(m_i, n_i) = (x_i - \hat{x}, y_i - \hat{y}) \tag{20}$$

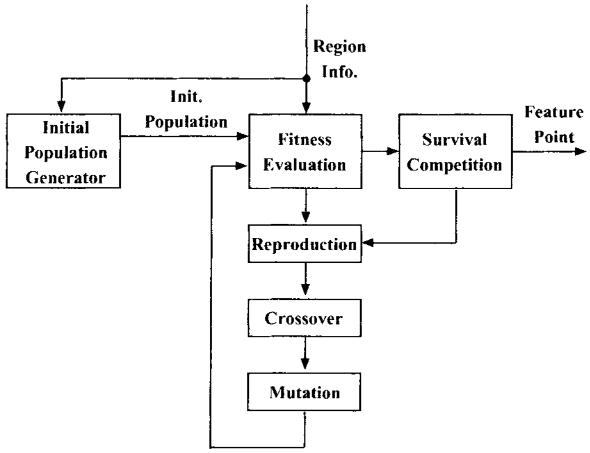


Fig. 11. Block diagram of the adopted genetic evolution.

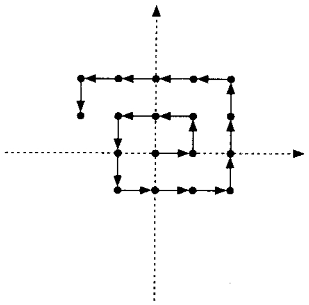


Fig. 12. Selection of initial chromosomes (which is based on the above spiral function).

where (x_i, y_i) represents the location of the central point of the template, and the codeword size k depends on the region size. If the maximum value of the width and the height of the region is w, the value k will be $\lceil \log_2 w \rceil$. The values of the genes are derived from the coordinate of the chromosome (x_i, y_i), that is

$$a_{i,j} = \left\lfloor \frac{m_i + 2w}{2^j} \right\rfloor \bmod 2, \tag{21}$$

$$b_{i,j} = \left\lfloor \frac{n_i + 2w}{2^j} \right\rfloor \bmod 2 \tag{22}$$

where $\bmod$ denotes the modulo operation and $\lfloor \cdot \rfloor$ is a floor function. The relative location (m_i, n_i) can be calculated from the values of the genes by

$$m_i = \sum_{j=0}^{k-1} a_{i,j} \cdot 2^j - 2a_{i,k-1}w, \tag{23}$$

$$n_i = \sum_{j=0}^{k-1} b_{i,j} \cdot 2^j - 2b_{i,k-1}w. \tag{24}$$

Fig. 11 depicts the block diagram of the genetic evolution implemented in our simulation. An initial population is formed before the evolution. In most GA's based applications, the initial population is created by randomly selecting elements from the solution space. To reduce the iteration number, in our approach, the initial chromosomes are selected from the solution space as shown in Fig. 12. The coordinate of the ith initial chromosome (x_i, y_i) is

$$x_i = \hat{x} + \Delta \cdot \tilde{x}_i, \tag{25}$$

$$y_i = \hat{y} + \Delta \cdot \tilde{y}_i, \tag{26}$$

$$\Delta = \left\lceil \frac{d}{4\max(\tilde{x}_{N-1}, \tilde{y}_{N-1})} \right\rceil, \tag{27}$$

$$\tilde{x}_i = (-1)^l \left([i - l(l+1)] \cdot [1 - (\lfloor 2\sqrt{i} \rfloor \bmod 2)] + \left\lceil \frac{1}{2} l \right\rceil \right), \tag{28}$$

$$\tilde{y}_i = (-1)^l \left([i - l(l+1)] \cdot [\lfloor 2\sqrt{i} \rfloor \bmod 2] - \left\lceil \frac{1}{2} l \right\rceil \right), \tag{29}$$

$$l = \lfloor \sqrt{i} \rfloor \tag{30}$$

where d is the length of the region, and $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ are the ceiling function and the floor function, respectively.

Each chromosome has an associated fitness value that is defined as

$$f_i = U_{F(x_i, y_i) - \hat{F}_\tau} \cdot (F(x_i, y_i) - \hat{F}_\tau) + \delta_{F(x_i, y_i) - \hat{F}_\tau} \tag{31}$$

where $F(x_i, y_i)$ is the matching value of the ith chromosome, and $\hat{F}_\tau$ is the τth maximum matching value among all the N matching values, and δ is a delta function. The constant τ determines how many chromosomes could be selected at most as the seeds in the reproduction stage for producing a rival population. The chromosomes with larger fitness values in the current population have higher probability to be selected as seeds of the next generation. This probabilistic scheme for selecting the seeds of new generations is known as *probabilistic reproduction.*

The reproduction method used in this work is similar to the weighted roulette wheel method [10]. For each chromosome, an interval r_i is calculated as

$$r_i = \left[\frac{\sum_{j=0}^{i-1} f_j}{\sum_{j=0}^{N-1} f_j}, \frac{\sum_{j=0}^{i} f_j}{\sum_{j=0}^{N-1} f_j} \right) \tag{32}$$

where f_j is the fitness value of the jth chromosome in the population, and '[' and ')' denote closing and opening boundaries, respectively. When the interval of each chromosome has been determined, N real numbers, α_ν, for $\nu = 0, 1, \cdots, N-1$, are randomly generated, where $0 \leq \alpha_\nu < 1$. The value of α_ν will be bounded by some r_ω, that is, $\alpha_\nu \in r_\omega$. The chromosome C_ω is then selected as a seed to generate the rival population. It is possible that one chromosome can be selected twice or more. Because N real random numbers are generated, N seeds could be selected and placed in the mating pool. The seeds will be processed by the genetic operations of crossover and mutation.

The crossover method used here is the so-called uniform crossover [13]. For every two seeds, C'_l and C'_{N-1-l}, $0 \leq l \leq \frac{N-1}{2}$ selected from the mating pool, two new chromosomes are produced by performing uniform crossover operations as

$$m'_i = (m_i \cap M_x) \cup (m_j \cap \bar{M}_x), \tag{33}$$

$$m'_j = (m_i \cap \bar{M}_x) \cup (m_j \cap M_x), \tag{34}$$

$$n'_i = (n_i \cap M_y) \cup (n_j \cap \bar{M}_y), \tag{35}$$

$$n'_j = (n_i \cap \bar{M}_y) \cup (n_j \cap M_y) \tag{36}$$

where $[m'_i \; n'_i]^t$ and $[m'_j \; n'_j]^t$ represent the new chromosomes, $[m_i \; n_i]^t$ and $[m_j \; n_j]^t$ are the original chromosomes selected from the mating pool, M_x and M_y are two randomly generated bit masks, and $\bar{M}_x$ and $\bar{M}_y$ are the complements of M_x and M_y, respectively.

The crossover operations are applied to explore new elements from the search space. In this work, it is applied only when the mutation operation is performed on the most significant genes (MSG's) of the chromosomes (e.g., a_{k-1}, b_{k-1}), because the violent evolution by applying crossover is not proper when the chromosomes are expected to converge to the maximum.

After the crossover stage, each chromosome in the mating pool will be processed and transferred into a candidate chromosome of the new generation. Assume the current seed to be processed is $[m_i \;\; n_i]^t$, where $m_i = [a_{i,k-1} a_{i,k-2} \cdots a_{i,0}]$ and $n_i = [b_{i,k-1} b_{i,k-2} \cdots b_{i,0}]$. In the jth generation, the two genes $a_{i,z}$ and $b_{i,z}$ are varied, where $z = k - 2 - (j \bmod k)$. There are eight mutation operators, $\{(\zeta_p, \eta_p) \mid p = 0, 1, \cdots, 7\}$, which can be applied in our implementation, that is

$$a'_{i,z} = a_{i,z} + \zeta_p, \tag{37}$$

$$b'_{i,z} = b_{i,z} + \eta_p \tag{38}$$

where p is a random integer number whose value is between zero and seven. Because the chromosomes are randomly selected and put on the mating pool, it is not necessary to generate a random number for determining the p's value. We simply set p to be $(i \bmod 8)$. The mutation operators are therefore defined as

$$\zeta_p = (-1)^l \left([p + 1 - l(l+1)] \cdot [1 - (\lfloor 2\sqrt{p+1} \rfloor \bmod 2)] + \left\lceil \frac{1}{2} l \right\rceil \right), \tag{39}$$

$$\eta_p = (-1)^l \left([p + 1 - l(l+1)] \cdot [\lfloor 2\sqrt{p+1} \rfloor \bmod 2] - \left\lceil \frac{1}{2} l \right\rceil \right), \tag{40}$$

$$l = \lfloor \sqrt{p+1} \rfloor. \tag{41}$$

By applying these mutation operators, the neighboring points of the seeds are included in the rival population to be evaluated for their fitness values. The chromosomes with larger fitness values will have more copies in the mating pool, so more neighboring points of them will be included. On the contrary, less neighboring points will be included for those chromosomes with smaller fitness values. In other words, the number of included neighboring points are determined by the fitness values.

When the mutation operations are performed on the MSG's, the chromosomes which are far from the original ones in the search space are generated. Whereas, the nearby chromosomes are generated when the mutation operations are performed on the least significant genes (LSG's).

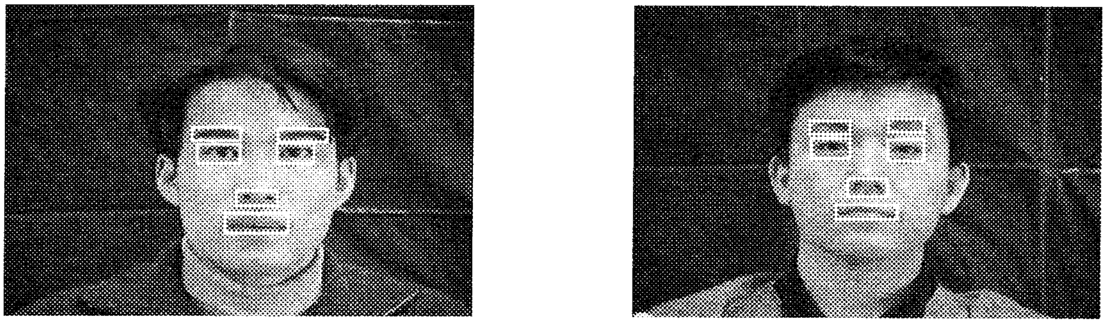


Fig. 13. Simulation results of two captured images.

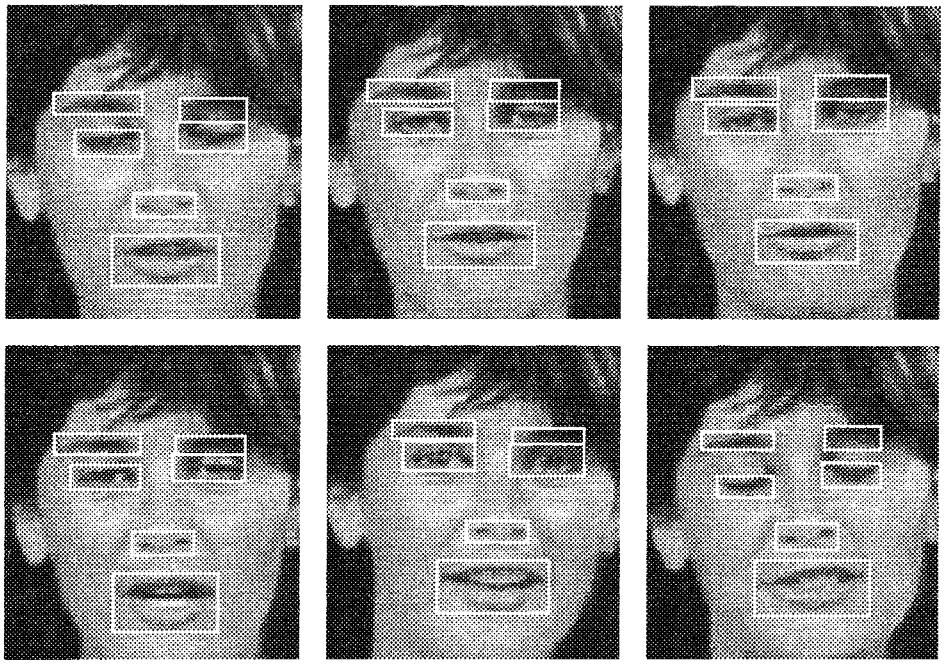


Fig. 14. Simulation results of the eighth, tenth, forty-fifth, fiftieth, eighty-fifth, and ninety-eighth frames from the Miss America sequence. Notice that the eyes in the ninety-eighth frame are closed.

There are N chromosomes in the mating pool after performing the genetic operations. Along with the original chromosomes in the current generation, N chromosomes are selected from these $2N$ chromosomes according to their fitness values. Each chromosome can only be selected once, at most. The chromosomes with larger fitness values will be picked up as the members of the population in the next generation, and go through the next iterations of the genetic evolution. Although the sorting operation is needed in the survival competition stage, the overhead is not high because the population size is usually not large in this case. This stage is added in the proposed algorithm to prevent the chromosomes from being destroyed by the new ones with poorer fitness values, because the new chromosomes generated from the original ones are not guaranteed to have larger fitness values in GA's.

The chromosome with the maximum fitness value is selected from the current population as the possible solution. The possible solution might be replaced by the other ones from a generation to the other generations. The iteration will be terminated if the solution is not updated for a predetermined period of iterations.

IV. Experimental Results

The proposed algorithm has been implemented, on a SUN SPARC-10 workstation under Unix environment, at the Communication and Multimedia Laboratory, National Taiwan University. Figs. 13–15 show the experimental results selected from our simulations. Although the materials are originally color images, only the luminance components are shown because the chrominance components are not utilized in the proposed algorithm.

The thresholds used in our implementation are static for each input image. Their values are listed in Table I.

The two images (352 × 240 pixels) in Fig. 13 were captured by a normal V8 camera and transferred into digital form by an A/D converter. There was no special lighting when the images were captured. That is, the images were captured in a normal environment without any special equipment. With the proposed algorithm the facial features of the two images can be extracted successfully, as shown in Fig. 13.

We also applied the proposed algorithm to analyze typical head-and-shoulder images, two CIF image sequences, Miss

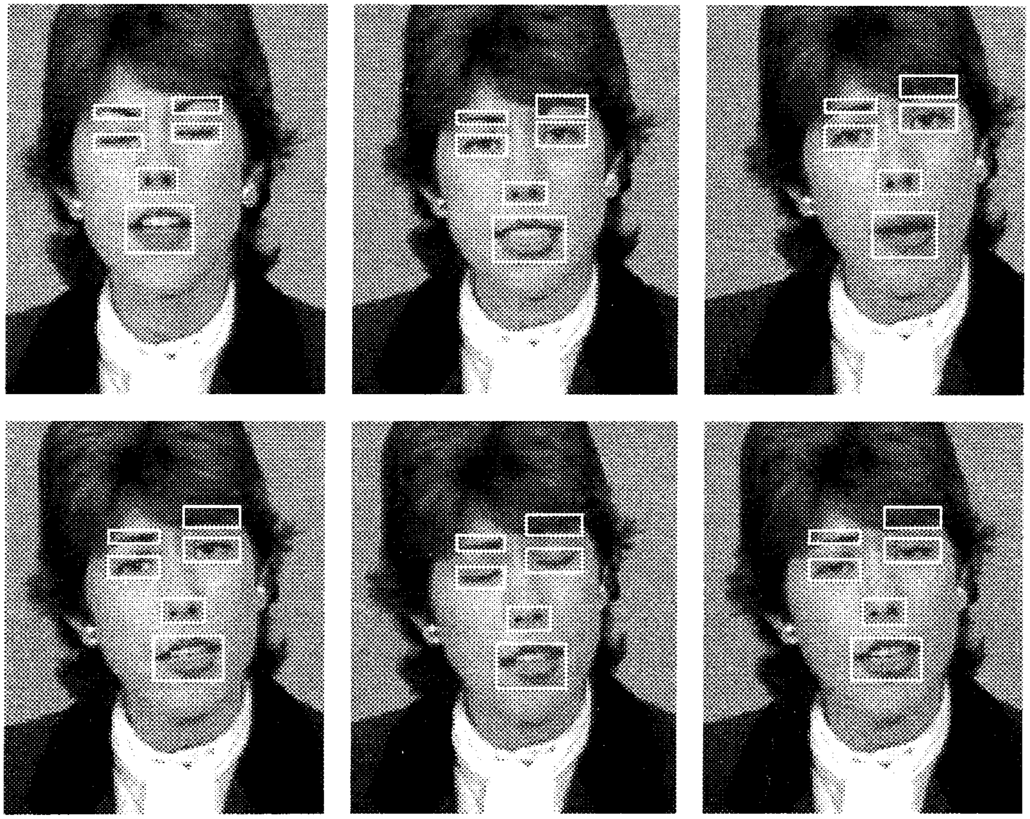


Fig. 15. Simulation results of the eighth, thirty-first, seventy-first, seventy-sixth, eighty-third, and eighty-sixth frame from the Claire sequence. Notice that the eyes in the 8th and the 83th frames are closed. Moreover, the eyebrows of the second layer images are covered by hair.

TABLE I
SUMMARY OF CHOSEN THRESHOLDS (R REPRESENTS THE FACE REGION)

Threshold	Value
T_l	$0.024 \times \|R\|$
T_h	$0.39 \times \|R\|$
T_r	35
K	50
K_m	10
K_e	10
T	110

TABLE II
AVERAGED AREAS OF THE FACE REGIONS IN DIFFERENT IMAGES

Image	Area (pixels)
Claire	3600
Miss America	8836
Chou	17689
Tseng	15376

America and Claire. Their resolution is 360 × 288 pixels. Each image in these two sequences was processed by applying the proposed algorithm and as expected the facial features had been extracted exactly. Figs. 14 and 15 show some of the simulation results. Notice that the eye features can still be extracted even though the eyes on the target images are closed.

Among the image sequences, the patterns of eyebrows and nostrils are nearly the same, but that of eyes and mouth varied violently in different image frames. Because eyes and mouth can be opened and closed in much freedom, the patterns might be totally different. In addition to the distortion of these features, the face of a head-and-shoulder image is nearly rigid. Since the features are extracted by template matching, the nonrigid nature of faces does not raise problems while comparing with the other methods. From the simulation results, it follows that the generic feature template defined in this paper works on this problem well.

The average areas of the face regions in different images are listed in Table II. In general, the necessary processing time for extracting facial features is proportional to the area of the face region.

Tables III and IV show the average ratios of execution time in different processing stages. The percentage of the execution time in the post-processing stage is small, so it is

TABLE III
AVERAGED RATIOS (%) OF EXECUTION TIME IN DIFFERENT PROCESSING STAGES, WHERE GA'S ARE NOT APPLIED TO SECOND STAGE FEATURE EXTRACTING

Stage	Claire	Miss America	Chou	Tseng
Pre-Processing	15.2	16.8	14.4	17.4
1st stage extracting	65.2	50.5	39.6	36.1
2nd stage extracting	19.6	32.7	46.0	46.5

TABLE IV
AVERAGED RATIOS (%) OF EXECUTION TIME IN DIFFERENT PROCESSING STAGES, WHERE GA'S ARE APPLIED TO SECOND STAGE FEATURE EXTRACTING

Stage	Claire	Miss America	Chou	Tseng
Pre-Processing	14.9	18.9	18.8	23.1
1st stage extracting	63.8	56.7	52.0	47.7
2nd stage extracting	21.3	24.4	29.2	29.2

TABLE V
RATIOS OF THE AVERAGED NUMBER OF SEARCHED POINTS TO THE TOTAL NUMBER OF CANDIDATE POINTS

Stage	Claire	Miss America	Chou	Tseng
Right Eye	0.36	0.17	0.08	0.10
Left Eye	0.37	0.17	0.10	0.09
Mouth	0.57	0.33	0.17	0.14

not included in the table. In Table IV, GA's are applied to the second stage feature extracting, whereas, GA's are not applied to that stage in Table III. It follows from the two tables that the computational cost in the second stage feature extracting is largely reduced by applying GA's in feature searching when the search range is large. Nevertheless, it will not benefit from GA's if the search range is small, for example, the computational complexity is not improved in the Claire sequence.

Table V shows the ratios of the average number of searched points to the total number of candidate points. According to the table, it is known that most of the unnecessary point evaluations are avoided. Therefore, the computational complexity is largely reduced, although overheads are needed for applying GA's in the searching process.

V. CONCLUSION

In this paper, an automatic facial feature extraction algorithm is proposed. In order to extract the facial features from a head-and-shoulder image, the face region is first estimated by the second-chance region growing method. The advantage of this method is that its computational cost is very low. The facial feature points are then searched based on genetic algorithms. We also defined a generic feature template and the associated cost functions. Because the complexity of calculating the cost functions is not high, and genetic algorithms are adopted to reduce the necessary search numbers, the whole system is very efficient. The proposed algorithm has been applied to many head-and-shoulder images. It is shown by computer simulations that the facial features can always be extracted exactly by applying the proposed algorithm.

This work focuses on extracting facial features from a single image. The temporal information between neighboring images has not been taken into consideration. When the feature extraction algorithm is applied to very low bit-rate video coding applications, the temporal relation of neighboring images in a video sequence can be employed. Because the movement of a human face in a scene is normally not violent, the positions of facial features in neighboring images are close to each other. By means of the temporal information, new techniques might be exploited to make the feature extraction algorithm even more efficient. This is one of our future work directions.

REFERENCES

[1] K. Aizawa and T. S. Huang, "Model-based image coding: Advanced video coding techniques for very low bit-rate applications," *Proc. IEEE*, vol. 83, pp. 259–271, Feb. 1995.

[2] M. Wollborn, "Prototype prediction for color update in object-based analysis-synthesis coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 236–245, June 1994.

[3] A. Samal and P. A. Iyengar, "Automatic recognition and analysis of human faces and facial expressions: A survey," *Pattern Recognit.*, vol. 25, pp. 65–77, Jan. 1992.

[4] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1042–1052, Oct. 1993.

[5] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proc. IEEE*, vol. 83, pp. 705–740, May 1995.

[6] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *Int. J. Comput. Vis.*, vol. 8, pp. 99–111, Feb. 1992.

[7] H. C. Huang, M. Ouhyoung, and J. L. Wu, "Automatic feature point extraction on a human face in model-based image coding," *Opt. Eng.*, vol. 32, pp. 1571–1580, July 1993.

[8] F. Lavagetto and S. Curinga, "Object-oriented scene modeling for interpersonal video communication at very low bit-rate," *Signal Process.: Image Commun.*, vol. 6, pp. 379–395, June 1994.

[9] L. C. Desilva, K. Aizawa, and M. Hatori, "Detection and tracking of facial features," in *SPIE Symp. Visual Communications and Image Processing*, Taipei, Taiwan, R.O.C., May 1995, pp. 1161–1172.

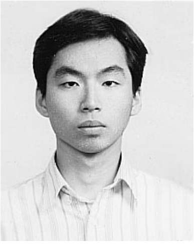
[10] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[11] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *IEEE Comput. Mag.*, vol. 27, pp. 17–26, June 1994.

[12] K. S. Tang, K. F. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications," *IEEE Signal Processing Mag.*, vol. 13, pp. 22–37, Nov. 1996.

[13] G. Syswerda, "Uniform crossover in genetic algorithms," in *Proc. 3rd Int. Conf. Genetic Algorithms and Applications*, 1989, pp. 2–9.

[14] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision, Vol. 1*. Reading, MA: Addison-Wesley, 1992.



Chun-Hung Lin was born in Kaohsiung, Taiwan, R.O.C., in 1968. He received the B.S. degree in information engineering from Tatung Institute of Technology, Taipei, Taiwan, in 1991, and the M.S. and Ph.D. degrees in computer science and information engineering from National Taiwan University, Taipei, in 1993 and 1997, respectively.

His research interests include image and video coding, multimedia systems, and genetic algorithms.

Ja-Ling Wu (A'86–SM'98) received the B.S. degree in electronic engineering from Tamkang University, Tamshoei, Taiwan, R.O.C., in 1979, and the M.S. and Ph.D. degrees in electrical engineering from Tatung Institute of Technology, Taipei, Taiwan, in 1981 and 1986, respectively.

From 1986 to 1987, he was an Associate Professor of the Electrical Engineering Department at Tatung Institute of Technology. Since 1987, he has been with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, where he is currently a Professor. From August 1996 to July 1998, he transferred to the National Chi Nan International University, Puli, Taiwan, as the Head of the Department of Information Engineering. He is the co-founder of the Communications and Multimedia Laboratory, where he has been Director since January 1991. His research interests include digital signal processing, multimedia data compression, and digital watermarking techniques. He has published more than 200 technical and conference papers.

Dr. Wu was the recipient of the 1989 Outstanding Youth Medal of the Republic of China and the Outstanding Research Award sponsored by the National Science Council, from 1987 to 1994, the Long-Term Medal for Ten Distinguished Researchers (1996), the Best Long-Term Paper Award (1995), the Special Long-Term Award for Collaboratory Research (1994), sponsored by the Acer Corporation, and the Award for Distinguished Information People of the Year (1993).