

Chin Contour Extraction Based on an Auto-Initialized Shape-Enhanced Snake

Kuo-Yu Chiu[†], Shih-Che Chien[†], and Sheng-Fuu Lin[†]

[†]Department of Electrical Engineering, National Chiao Tung University,
1001 Ta Hsueh Road, Hsinchu, Taiwan 300, ROC

Summary

In this paper, an improved method to extract the chin contour under different face poses is proposed. Several previous approaches have used active contour models(ACM) or snakes in front-view facial images. However, successful chin contour extraction for a wide range of face poses has seldom been mentioned. An algorithm based on an improved snake for chin contour extraction is presented here and it is able to be operated successfully with wide range of different face poses. Since lots of algorithms have been proposed for extracting face region and important facial features like eyes and lips, the algorithm in this paper assumes that the face region and locations of these two features are known as prior knowledge. At the first stage, the face is normalized and face pose is roughly estimated according to the geometric relationship between eyes and lips. Then, the snake for chin contour is initialized with two parabolas according to different face poses. The shape-enhanced snake model is employed to fit the chin contour at last. Experimental results show that the proposed algorithm can extract the chin contour under different face poses with good accuracy.

Key words:

Active contour model, chin contour extraction, facial feature, face pose

1. Introduction

Chin contour extraction, or fitting the chin, is a major requirement for face modeling, face pose estimation, face replacement, and face recognition. In fact, chin contour is an important feature to distinguish the face region from the neck region. Besides, the shape of a chin also offers some information for face recognition and face pose estimation. While the shape of a chin is an important feature for a human, the chin contour is rarely used as a feature. One reason is that an estimated chin contour may deviate from the true chin contour due to the weak boundary between face region and neck region. With noise or complicated background, the estimated chin contour would easily be led into false result. Another reason is that the chin contour varies greatly while the face pose is different. When the face is profile, turn 90 degrees left or right, the chin contour is totally different from the frontal pose. Hence,

chin contour extraction is one of the most difficult tasks of facial feature extraction. The shadows, variations of the skin color, clothing, hair, beards, and noisy background may easily reduce the accuracy of chin contour extraction. In order to overcome these problems, the prior information such like skin color region and facial features, such as eyes and lips, are important and necessary. Using these information, the typical shape of chin contour can be modeled and the possible locations of chin contour can be estimated.

Since facial features extraction is an important and primary step for systems such as interaction systems and identification systems, lots of algorithms and methods [1]-[5] have been proposed for extracting the facial features, which include skin color region, eyes, and lips region. Hence, we do not focus on how to extract and localize these facial features in this paper. There are also some chin contour estimation or fitting the chin algorithms. For example, the authors in [6] use color-based segmentation along with morphological operation to approximate the chin contour. The resulting set of discontinued segments and fragmented curves are processed, thinned, classified, and finally the chin contour is reconstructed. In [7], the chin contour is estimated by a deformable template consisting of four parabolas. Costfunctions are established and minimized to find the best fit of the template to the chin and cheek. A more accurate algorithm based on three models is proposed in [8]. Three perfect models, circular, triangular, and trapezoidal models, are used here along with piecewise linear model for curve fitting. The experiments show high accuracy for extraction and classification on a large data set. Another algorithm based on two curves with polynomials of five degrees are employed in [9][10]. By using modified Canny edge detector, some false edges with unfit gradient directions are excluded and the results of fitting are improved. There are other approaches based on active contour model [11]-[16]. As the snake model is an energy-minimizing spline influenced by image features and external forces which pull the spline toward real edges, the results using a snake model for fitting would be much

closer to the real chin contour. However, when the chin contour appears loosely marked due to complicated background or profile face pose, the reliability of snake model is greatly reduced. Furthermore, the performance is affected seriously by the initialization and external force of snake model. The algorithm proposed in [17] shows good accuracy when dealing with different face poses. By training 10 flexible face models based on different face poses, the initialization problem is hence solved.

In this paper, an algorithm based on snake for chin contour extraction which separates the chin region from the neck region is presented. This algorithm is applicable to both frontal and profile face images based on the information that the skin color region and the locations of eyes and lips are known. There are three main stages to accomplish the chin contour extraction. In the first stage, the face is first normalized so that the roll angle is rotated to be zero. The line across two tips of the lips with profile face pose or the line across two eyes with frontal face pose is rotated to be parallel to the x -axis. The purpose of normalizing face image is to facilitate locating the possible chin contour region. Then, the normalized face image is roughly classified into 27 classes, according to the face turning angle and face looking up, down, or straight as the second stage. The geometric relationship between eyes and lips is used as the input features for classification. The snake model is differently initialized according to these two classes in this stage. Finally, the shape-enhanced snake model is employed in the last stage to fit the chin curve.

The remainder of this paper is organized as follows. In Section 2, the steps of face image pre-processing including face image normalization and face pose classification are given. Then, the proposed auto-initialized shape-enhanced snake algorithm for fitting the chin contour is introduced in Section 3. Some experimental results and discussions are shown in Section 4. Finally, the conclusions are given in Section 5.

2. Face Image Pre-processing

In this section, the face image pre-processing steps including face image normalization and face pose classification will be introduced. Active contour model or snake is an energy minimizing curve that deforms to fit image features. However, there are some limitations, such like the model usually incorporate edge information, ignoring other image characteristics such as texture and color. Besides, the snake must be initialized close enough to the feature of interest to avoid being distracted by noise and clutter. By using the geometrical and statistical information between facial features, the chin curve can be roughly initialized with two parabolas. In this paper, the

skin color region and locations of eyes and lips, represented by their mass centers, are supposed to be known as prior information to utilize the estimation.

The chin curve, as it can be seen, transforms greatly with different face poses. When turning left or right, or looking up or down, these changes will lead to different chin curves as shown in Fig.1. A fixed model used as initialization of snake is not appropriate to describe the variety of chin contours. Since the snake is so sensitive to the initialized curve, different initial snakes should be built by considering different face poses, instead of using one chin contour model. In this paper, face poses are roughly divided into two classes, namely frontal pose and profile pose. These two classes are distinguished by the number of eyes. When two eyes are found in a face image, the face pose is defined as frontal, while one eye represents profile face pose. Since there may be two eyes or one eye in a face image and the face pose may be frontal or profile, these conditions are discussed and processed under different criteria. In this section, the face image is first normalized by setting the roll angle to be zero. Later, the face pose can be roughly estimated and classified according to the geometrical and statistical information between prior facial features. Then the estimated chin curve can be initialized close enough to the real one.

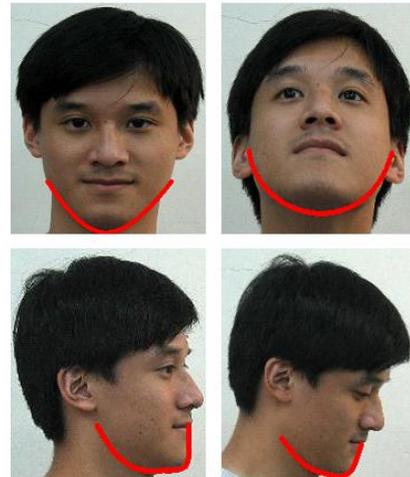


Fig.1 Chin contour varies greatly with the change of face poses. When turning left or right, or looking up or down, these changes will lead to different chin contours.

2.1 Face image normalization

The purpose of face image normalization is to rotate the face image so that the face is centrally located with zero roll angle. Most of the lines crossing important face features, such like eyes, lips, or nose, are parallel to the x -axis or y -axis after normalization, as shown in Fig.2. The roll angle in this paper is determined in two ways

according to the number of eyes. With the prior information that the locations of eyes and lips are known, the roll angle θ is defined as the included angle between x -axis and the extended line across two eyes when two eyes are found,

$$\theta = \tan^{-1}\left(\frac{y_{r-eye} - y_{l-eye}}{x_{r-eye} - y_{l-eye}}\right), \quad (1)$$

where y_{r-eye} and x_{r-eye} are the y -coordinate and x -coordinate of the right eye respectively, while y_{l-eye} and x_{l-eye} are the y -coordinate and x -coordinate of the left eye respectively. When there is only one eye in the face image, which means the profile face pose, the roll angle is defined as the included angle between x -axis and the extended line across the right and left tips of the lips,

$$\theta = \tan^{-1}\left(\frac{y_{r-lip} - y_{l-lip}}{x_{r-lip} - y_{l-lip}}\right), \quad (2)$$

where y_{r-lip} and x_{r-lip} are the y -coordinate and x -coordinate of the right tip of lips, while y_{l-lip} and x_{l-lip} are the y -coordinate and x -coordinate of the left tip of lips.

After the roll angle is retrieved, the face can be normalized to be horizontal by executing rigid transformation. The rigid transformation matrix M_{rig} is defined by

$$M_{rig} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

Bilinear interpolation using the four nearest neighbors of a point is adopted here to modify the rotated image. Let the point (x', y') denote the coordinate of interpolated pixel and $v(x', y')$ denote the RGB value assigned to it. For bilinear interpolation, the assigned RGB value is given by

$$v(x', y') = ax' + by' + cx'y' + d, \quad (4)$$

where four coefficients are determined from the four equations in four unknowns that can be written using the four nearest neighbors of point (x', y') [19].

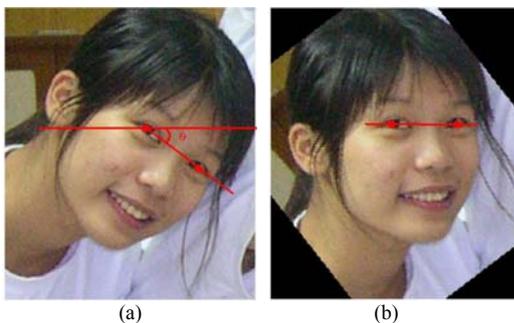


Fig.2 (a) Original image. (b) Rotated image with zero roll angle after normalization.

2.2 Face pose estimation

After the face image is normalized, the face pose owing to turning left or right and looking up or down is estimated. The face pose can be divided into two major classes, namely frontal pose or profile pose, according to the number of eyes found in a face image. In an input face image, face pose is defined to be frontal if two eyes are found, while profile face pose means that only one eye is found, as shown in Fig.3. Since the chin contour varies greatly when the face pose is different, face poses are further discussed and classified in this section. With the prior information of skin color region and locations of eyes and lips, 15 sub-classes in frontal pose and 12 sub-classes in profile pose are classified by using a 3-layer neural network. Three parameters which derive from the geometrical relationship of facial features are used as the input of the neural network. These parameters will be introduced as follows.

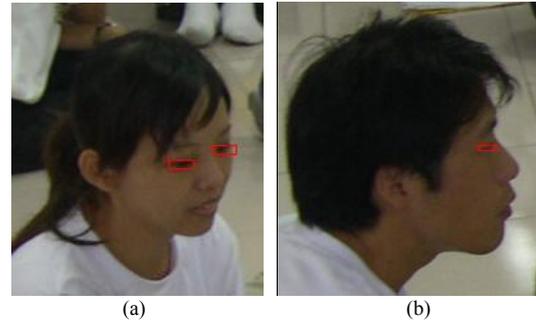


Fig.3 (a) Face pose is defined to be frontal if two eyes are found. (b) Face pose is defined to be profile if only one eye is found.

Since the resolution and size may be different from each other, by comparing with the face image in data base, a *scale factor*, S_f is defined here to represent the relative size of face image or face region. When the face pose is frontal, or two eyes are found, the scale factor S_f is defined by

$$S_f = \sqrt{(D_{e-e}^2 + D_{l-ee}^2)} = \sqrt{\square \mathbf{E}_L - \mathbf{E}_R \square^2 + \square \mathbf{T}_p - \mathbf{L}_c \square^2}, \quad (5)$$

where D_{e-e} is the distance between two eyes, \mathbf{E}_L and \mathbf{E}_R , and D_{l-ee} is the vertical distance from the centroid of the lips, \mathbf{L}_c , to the point, \mathbf{T}_p , which is the intersection of the line segment of two eyes and the vertical line crossing the centroid of the lips. These feature points are illustrated in Fig.4(a). From the statistical experiment results, the value of these two parameters, D_{e-e} and D_{l-ee} , are approximately proportional to the face size and nearly equal to each other when the face is frontal without turning left or right and looking up or down, while D_{e-e} varies with the turning angle and D_{l-ee} varies when looking up or down. Hence, the scale factor S_f is defined by using these two parameters. Though the skin color region is known as prior information, the size of skin color region is not used here

to represent the size of face image since skin color region extracted by skin color may contain neck region. Hence, the size of skin color region is not a valid parameter for face size. When the face pose is profile, or one eye is found, S_f is defined with lesser facial information as

$$S_f = D_{l-e} = \|\mathbf{E}_o - \mathbf{L}_c\| \quad (6)$$

where D_{l-e} is the distance between the only eye, \mathbf{E}_o , and the centroid of lips, \mathbf{L}_c .

After S_f is determined, we use some parameters as input features for face pose classification. From our experiments, three more robust and simpler parameters are used to describe the face pose. With frontal face pose, these features are defined as

$$\begin{cases} p_1 = \left(\frac{D_{e-e}}{D_{l-ee}}\right), \\ p_2 = \left(\frac{D_{TE-r} - D_{TE-l}}{S_f}\right) = \frac{\|\mathbf{T}_p - \mathbf{E}_R\| - \|\mathbf{T}_p - \mathbf{E}_L\|}{S_f}, \\ p_3 = \left(\frac{F_c - FF_c}{S_f}\right), \end{cases} \quad (7)$$

where D_{TE-r} is the distance between \mathbf{T}_p to right eye \mathbf{E}_R and D_{TE-l} is the distance between \mathbf{T}_p to left eye \mathbf{E}_L . When the face is turning left (from the viewpoint of an observer), D_{TE-r} would be larger than D_{TE-l} and the result of p_2 will be positive. The difference value between D_{TE-r} and D_{TE-l} is divided by the factor S_f to represent a relative turning degree. In last equation, the mass centers of face region and facial features are utilized. We use F_c to represent the x -coordinate of skin color region centroid. The x -coordinate facial feature centroid determined by the mass center of \mathbf{E}_L , \mathbf{E}_R , and \mathbf{L}_c is defined as facial feature centroid FF_c . These two features are illustrated in Fig.4(a) and Fig.4(b) respectively. Since only x -coordinate is concerned in this equation, the skin color region containing neck region or not does not matter. The skin color region with neck region changes the centroid of skin color region center in y -axis much more than that in x -axis. With the face turning left, the facial feature centroid moves left than the skin color region centroid does. A positive p_3 represents that the face is turning left and the magnitude show the turning degree.

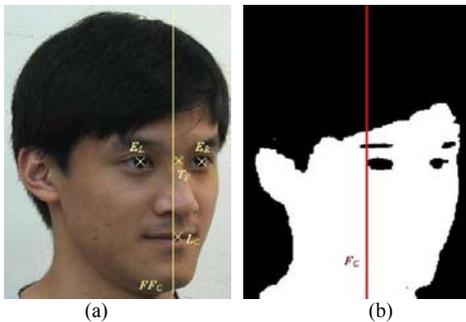


Fig.4 (a) Facial features and the centroid of facial features of frontal face pose. (b) Face centroid is defined as the centroid of skin color region.

When dealing with profile face poses, we first determine whether the head is turning left or right by comparing the x -coordinates of skin color region centroid and facial feature centroid, which is derived from the centroid of lips only. Then, the chin tip, C_T , is extracted by

$$C_T = \max_{(x,y) \in F} |xy|, \quad (8)$$

where F is the set of points of skin color region. By setting the y -coordinate of the highest skin-color point and the x -coordinate of the most right skin-color point when turning left or the most left skin-color point when turning right as origin, the variable x and y are the relative x -coordinate and y -coordinate of points of skin color region. Connecting chin tip and lips tip as L_1 and connecting chin tip and the eye as L_2 , we define another three features for classification. These features are defined by

$$\begin{cases} p_1 = \left(\frac{D_{Eo-L_1}}{S_f}\right), \\ p_2 = \theta_{L_1}, \\ p_3 = \theta_{L_2}, \end{cases} \quad (9)$$

where D_{Eo-L_1} is the distance from the eye to L_1 . When the face pose is more profile, the distance from the eye to L_1 will be shorter. The parameter θ_{L_1} is the included angle between L_1 and x -axis, while θ_{L_2} is the included angle between L_2 and x -axis. These two features vary with the face looking up or down and turning right or left. All the features are shown in Fig. 5.

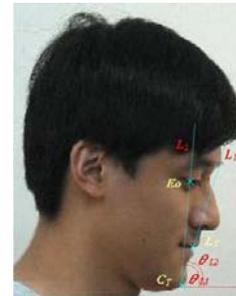


Fig.5 Facial features and some feature points or lines of profile face pose.

Now, for each face image, we have three features derived from the geometrical relationship and treat them as the input of neural network. The outputs of neural network are the types of face pose. When the face pose is frontal, there are 15 classes composed of 5 types of face turning with looking up, down, and straight. The 5 types are $-50^\circ \sim -30^\circ$, $-30^\circ \sim -10^\circ$, $-10^\circ \sim 10^\circ$, $10^\circ \sim 30^\circ$, and $30^\circ \sim 50^\circ$ respectively. The positive degree represents that the face is turning right from the view of an observer, while negative degree means turning left. The face pose is defined as looking straight if the tilt angle owing to looking up or down is between -20° and 20° . Hence, the tilt angle

greater than 20° means looking up and less than -20° represents looking down. When the face pose is profile, there are 4 types of face turning with looking up, down, and straight, totally 12 classes. The 4 types of face turning are $-90^\circ \sim -70^\circ$, $-70^\circ \sim -50^\circ$, $50^\circ \sim 70^\circ$, and $70^\circ \sim 90^\circ$. The definition of looking up, down and straight is the same as before. Figure Fig.6 shows an example of all the 27 different face poses. Each face in Fig.6 belongs to one of these classes. After training the neural network with test data from both real face images and virtual face images by using the features defined before, the face pose of an input face image can be roughly classified.

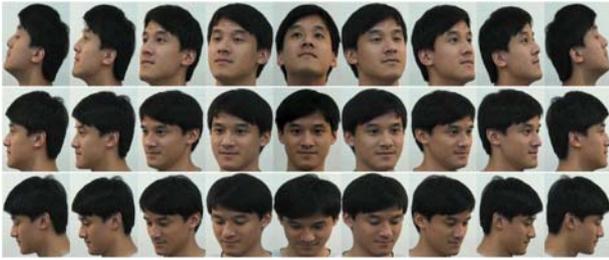


Fig.6 The representative face images of 27 different face pose classes.

3. Chin Contour Extraction

In this section, chin contour extraction based on auto-initialized shape-enhanced snake is introduced. The snake model is an energy-minimizing active contour composed of image features, internal constraints and external forces. Snakes lock on to local minima in the potential energy generated by processing an image. However, there are limitations such like that the models must be initialized close to the feature of interest to avoid being distracted by noise and clutter. In chin contour extraction problem, chin contour is weak and indistinct, and it varies greatly when the face pose is different. Besides, there is some noise nearby such like beards or collars. With the information of face pose obtained before, the snake can be initialized appropriately to the real chin contour to solve these problems. A chin contour database of various face pose is first built. We use two parabolas to describe the chin contour based on the chin tip. Left chin and right chin are estimated by a parabola respectively. These parabolas are estimated by using least square error (LSE) method with the chin contour points extracted by man. Then, the initial control points can be set with this information. At last, an iterative shape-enhanced snake is adopted here to fit the chin contour, which focuses on the initial shape and the distance between initial position and final position. These will be introduced in the following sub-sections.

3.1 Snake auto-initialization

Generally, the initial snake can be set arbitrarily in any shape, such as circle or square. However, if the shape of snake is similar to the target and the position is near to the target, a better and faster result will be obtained. The purpose of snake auto-initialization is to fit the chin contour as similar and near as possible with the face pose information.

Before initializing the snake, a chin contour database based on face pose and statistical results is built. The initial snake control points are the linear transformation of the chin contour stored in database. We determine one typical chin contour as the average face pose for each class of face pose, and there are therefore 27 typical chin contours in the database. For example, the chin contour with 60° turning right looking straight face pose will represent all the class of $50^\circ \sim 70^\circ$ turning right looking straight face pose, while 20° looking up and -20° looking down will represent the class of looking up and looking down. Each typical chin contour is extracted by man. We define the *matching center*, M_c , as the center of lips when the face pose is frontal or as the interior tip of lips when the face pose is profile. Two parabolas, f_L and f_R , separated by the x -coordinate of M_c with frontal face pose and C_T with profile face pose are constructed to fit the typical chin contour, as shown in Fig. 7. Using least square error (LSE) method, the parabola on the left, f_L , can be derived from the following equation

$$y = f_L(x) = \left[\left[X_L^T X_L \right]^{-1} X_L^T Y_L \right]^T \begin{bmatrix} x^2 \\ x \\ 1 \end{bmatrix}, \quad (10)$$

where

$$X_L = \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_c^2 & x_c & 1 \end{bmatrix}, \quad Y_L = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_c \end{bmatrix}, \quad (11)$$

and (x_k, y_k) , $k = 1, \dots, c$, is the coordinate of typical chin contour on the left extracted by man. The right parabola can also be derived in a similar way. These two parabolas of each typical chin contour are built in the database for snake initialization.

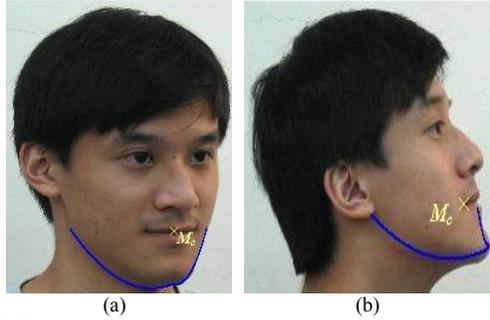


Fig.7 The matching center and the reconstructed chin contour with two parabolas in (a) frontal face pose, and (b) profile face pose.

However, since the scale and the position of input face image may not be the same as those in database, scaling and shifting is needed. When the input image face is in high resolution, more control points are needed as shown in Fig.8(a), while few control points are enough to represent the chin contour in face images with poor resolution. To maintain accuracy of chin estimation and simplicity of calculation, the number of control points N_{cp} is adaptive to the scale of input face image and is determined by

$$N_{cp} = 7 + 2 \left\lfloor \frac{S'_f}{40} \right\rfloor, \quad (12)$$

with frontal face pose, or is determined by

$$N_{cp} = 9 + 2 \left\lfloor \frac{S'_f}{30} \right\rfloor, \quad (13)$$

with profile face pose where S'_f is the scale factor of input face image. These equations and parameters are derived from statistical experiment results. Because the profile face pose is more complicated with more curves and details, more control points are also needed to express the chin contour, as shown in Fig.8(b).

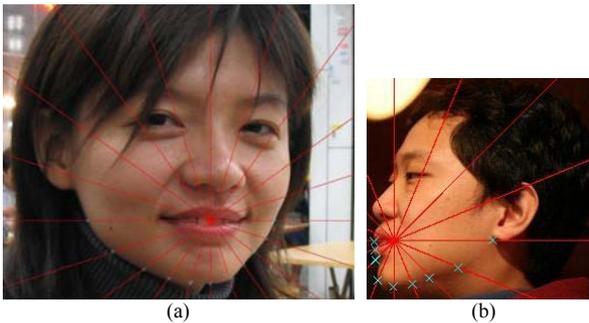


Fig.8 (a) The face image with higher resolution need more control points to represent the chin contour. (b) There are more details on the chin contour of profile face pose, so more control points are also needed.

The initial control points are linear transformation of some points of parabolas stored in database. Suppose $\mathbb{S}' = \{s'_1, s'_2, \dots, s'_{N_{cp}}\}$ is the set of the initialized snake

control points containing N_{cp} elements as shown in Fig. 9(b), each of the control point is auto-initialized by

$$s'_k = \frac{S'_f}{S_f} [s_k - \mathbf{M}_c] + \mathbf{M}'_c, \quad k = 1, 2, \dots, N_{cp}, \quad (14)$$

where s'_k , \mathbf{M}'_c , \mathbf{M}'_c , and S'_f are the coordinates of control points, matching center, and scale factor of input face image, while s_k , \mathbf{M} , \mathbf{M}_c , and S_f are those of typical chin contour with the same class of face pose. The control point s_k of typical chin contour is defined by

$$s_k = \left\{ (x, y) \left| \tan^{-1} \left(\frac{y_{M_c} - y}{x_{M_c} - x} \right) = (k-1) \frac{\pi}{N_{cp} - 1}, (x, y) \in [f_x, f_y] \right. \right\}, \quad (15)$$

$$k = 1, 2, \dots, N_{cp},$$

where (y_{M_c}, x_{M_c}) is the coordinate of \mathbf{M}_c , as shown in Fig.9(a).

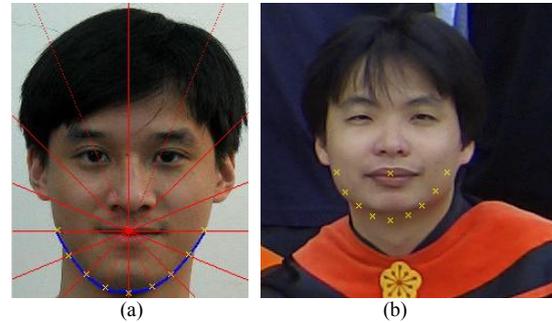


Fig.9 (a) The control points of typical chin contour. (b) The auto-initialized control points derived from the typical chin contour.

3.2 Shape-enhanced snake

The basic snake model is an energy-minimizing spline. A snake is represented as a parametric curve $\mathbf{v}(k) = (x(k), y(k))$, where the arc length, $k = 1, 2, \dots, N_{cp}$, is an index. The energy function of snake is defined as [18]

$$E_{snake} = \int (\alpha E_{internal}(\mathbf{v}(k)) + \beta E_{image}(\mathbf{v}(k)) + \gamma E_{external}(\mathbf{v}(k))) ds, \quad (16)$$

where $E_{internal}$ is the internal energy of the contour composed of tension and stiffness, E_{image} represents the energy of the image, $E_{external}$ is the energy of the external constraint, α , β , and γ are weighting parameters. The internal energy incorporates regularizing constraints that give the model tension and stiffness, and is defined as

$$E_{internal} = \alpha_1 E_{tension}(\mathbf{v}(k)) + \alpha_2 E_{stiffness}(\mathbf{v}(k)), \quad (17)$$

$$= \alpha_1 \mathbf{v}'(k) + \alpha_2 \mathbf{v}''(k)$$

where $\mathbf{v}'(k)$ and $\mathbf{v}''(k)$ controlled by weighting parameters α_1 and α_2 are first- and second-order terms that produce tension and stiffness respectively. The image energy drives the model towards salient features, which is usually generated by processing the image to enhance edges, terminations, or features. The image energy is defined as

$$E_{image} = -|\nabla I(x, y)|, \quad (18)$$

where $\nabla I(x,y)$ is the gradient image. The external energy represents external constraints imposed by high-level sources such as human operators. Here, we consider two constraints as snake external energy. First, since the snake is appropriately initialized, the real chin contour would be close to the initialized snake. When the chin contour is not salient with profile face pose or when there is lots of noise nearby, the auto-initialized snake should be used as important information. The distance of each control point between the final snake and the initialized snake should be minimized. With this constraint, the chance that the snake falls into another contour such as the neck of clothes will be reduced. Second, when there are errors in feature extraction, transformations of control points including shift or size change are predictable. Under this condition, all the control points will trend to the same direction. With above two considerations, the modified external energy of shape-enhanced snake in this paper is defined as

$$E_{external} = \gamma_1 |i(k)| + \gamma_2 \sigma_k, \quad (19)$$

where γ_1 and γ_2 are the weighting parameters, $i(s)$ is the searching index whose sign is the searching direction and magnitude is the distance between snake control point of initialization and current iteration, and σ_k is the standard deviation of $i(k)$. Since the chin contour is properly initialized, the result of estimation should be close to the initialized chin contour. The distance between control points of the initial snake and the final result is minimized. Hence, the first term strengthens the shape of initialized snake and a smaller $|i(k)|$ indicates the snake after iterations is close to the initial one. The second term focuses on the consistency. A smaller σ_i would be derived if most of the searching indexes including direction and magnitude are similar. When there are errors between facial extraction or feature matching, the initialized chin contour is a transformation of the final results. A minimized σ_k implies that the degree of shift and translation is similar.

Since the real chin contour is near the initialized snake, the possible searching range can be confined with the information of errors and scale. The errors include the estimated error between the typical chin contour model and all the chin contours in the same class, and the matching error brought from the unprecise location of matching centers. These errors will be larger with high face image resolution. Hence, after the snake is initialized, the possible searching range of one control point can be determined by

$$\mathbf{v}(k) = \bar{\mathbf{v}}(k) + i(k)\bar{\mathbf{n}}(k), \quad -I_{max}(k) \leq i(k) \leq I_{max}(k), \quad (20)$$

where $\bar{\mathbf{v}}(k)$ and $\bar{\mathbf{n}}(k)$ are the corresponding initial control point and unit normal vector respectively. The unit normal vector means that the possible searching region follows the direction of normal vector. The maximum searching index I_{max} determined by the information of errors and scale is defined as

$$I_{max}(k) = \varepsilon(k) \frac{S'_f}{S_f} \Delta_m, \quad (21)$$

where $\varepsilon(k)$ is the maximum error for control point $\bar{\mathbf{v}}(k)$ between the typical chin contour model and all the chin contour in the same class, while Δ_m is a constant error which includes the matching error and the localization error of facial features.

After the searching region is determined, the auto-initialized shape-enhanced snake will lead to estimated chin contour by minimizing the energy iteratively. The result chin contour is obtained when the energy is minimized or several iterations are run, such as 100 iterations. We use cubic spline to express the final results, as shown in Fig.10. A cubic spline is a spline constructed by piecewise third-order polynomials which pass through a set of control points. By defining the cubic spline as

$$\mathbf{v}(k) = \mathbf{a}_3 \mathbf{s}_k^3 + \mathbf{a}_2 \mathbf{s}_k^2 + \mathbf{a}_1 \mathbf{s}_k + \mathbf{a}_0, \quad (22)$$

the parameters, \mathbf{a}_0 , \mathbf{a}_1 , \mathbf{a}_2 , and \mathbf{a}_3 can be derived by differentiating the cubic spline equation. These parameters are expressed as

$$\begin{cases} \mathbf{a}_3 = 2\mathbf{v}(k) - 2\mathbf{v}(k+1) + \mathbf{v}'(k) + \mathbf{v}'(k+1), \\ \mathbf{a}_2 = -3\mathbf{v}(k) + 3\mathbf{v}(k+1) - 2\mathbf{v}'(k) - \mathbf{v}'(k+1), \\ \mathbf{a}_1 = \mathbf{v}'(k), \\ \mathbf{a}_0 = \mathbf{v}(k), \end{cases} \quad (23)$$

$$k = 1, 2, \dots, N_{cp} - 1,$$

and the spline function is obtained by

$$\begin{aligned} \mathbf{v}(t) &= \begin{bmatrix} \mathbf{v}(k)^3 & \mathbf{v}(k)^2 & \mathbf{v}(k) & 1 \end{bmatrix} \begin{bmatrix} \mathbf{a}_3 \\ \mathbf{a}_2 \\ \mathbf{a}_1 \\ \mathbf{a}_0 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{s}_k^3 & \mathbf{s}_k^2 & \mathbf{s}_k & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}(k) \\ \mathbf{v}(k+1) \\ \mathbf{v}'(k) \\ \mathbf{v}'(k+1) \end{bmatrix}, \quad (24) \\ t &\in [k \quad k+1], \end{aligned}$$

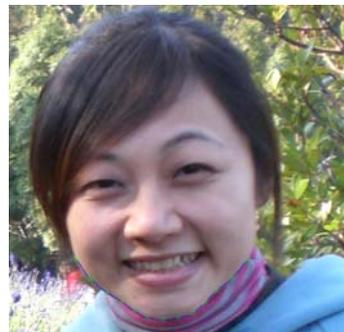


Fig.10 The cubic spline is a continuous function constructed by piecewise third-order polynomials which pass through all the control points.

4. Experiment Results and Discussions

Our experiment results are divided into two parts. One is the face pose estimation, and the other one is chin contour estimation. For the results of face pose estimation, we use a database containing 270 real face images from 10 people and 810 virtual face images. Since there are total 27 classes of face pose in this paper, 10 real face images and 30 virtual face images are used as training data for each class of face pose. The real images offer the variety of face shape, such as the fatter face, thinner face, or face with double chin. The virtual face images, on the other hand, give various face poses since the degree of turning and looking up or down can be easily set by adjusting the parameters. All the 1080 face images are used as input training data for a 3-layer neural network, and the output are the 27 classes of face pose. Each class of face pose is described by a typical chin contour model which consists of two parabolas. Since the face pose of real face image is unable to be correctly classified, we use another 540 virtual face images of random face poses for testing, which means 20 test images are used for each class.

The results of face pose estimation are shown in Fig.11. From the experimental results, we can see that the face pose estimation algorithm proposed in this paper gives an acceptable average accuracy as about 91%. Besides, analyzing the wrong results, all the misclassified face poses are classified into the adjacent class of the correct class. Since the estimated face poses are used as snake initialization, the estimation results are acceptable and useful for the further chin contour estimation.

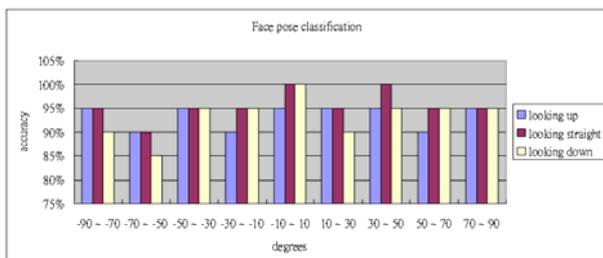


Fig.11 The accuracy of face pose classification. The average accuracy of face pose classification is 94.07%.

After the face pose is classified, the snake model can be initialized. The maximum errors between the initialized chin contour and the real chin contour are derived from the testing 540 virtual face images. We do not use the real face image because the chin contour of real face image can be extracted by man only. The task is time consuming and the chin contour extracted by man is a little different time by time. Contrarily, the chin contour of virtual image could be extracted easily by program and the points of chin contour can be determined previously. Hence, the maximum errors are derived from the virtual face image. With this

information, the searching range for chin contour could be set reasonably.

Some cases are concerned in our experiments, which include faces with different resolution, sizes, and face poses. In this paper, we defined the scaling factor S_f which offers the information of relative face size and image resolution. As shown in Fig.12, the resolution of input face image is much higher than the face image in database. By using the information of relative scaling factor, the initialized snake can be resized to fit the real chin contour. When the face is wider or narrower and longer in shape, the chin contour can be extracted more precisely under the limited searching range as shown in Fig.13. While dealing with different face poses, they are roughly classified into 27 classes and each of them has a typical initialized chin contour composed of two parabolas. In this way, this method proposed in this paper greatly improves the snake initialization drawback when coping with chin contour extraction. Besides, chin contour varies severely under different face pose and extraction for chin contour of profile face poses is seldom mentioned in previous papers because of the weak boundary. The face pose based auto-initialization shape-enhanced snake provides a way to approach the chin contour with profile face poses as illustrated in Fig.14.

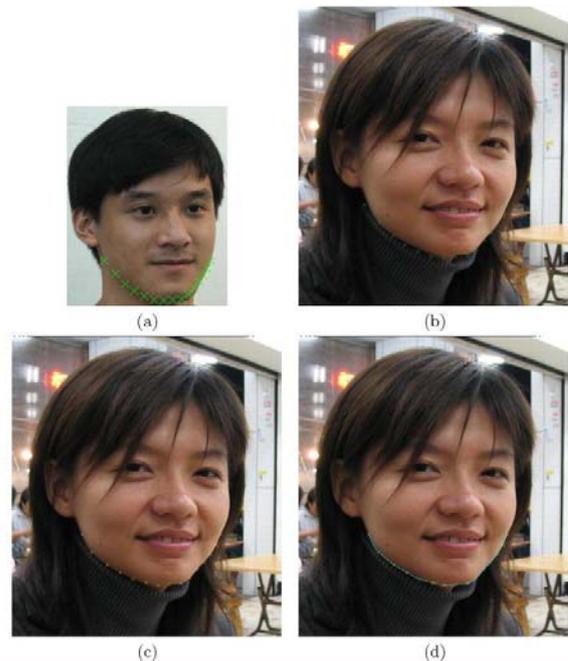


Fig.12 (a) The typical chin contour model stored in database. (b) The initialized control points. (c) The control points after several iterations. (d) The reconstructed chin contour.

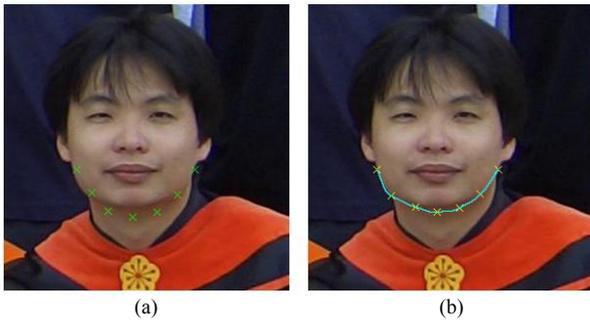


Fig.13 (a) The typical chin contour model stored in database is thinner than the input face. (b) The chin contour can be properly estimated under limited searching range.

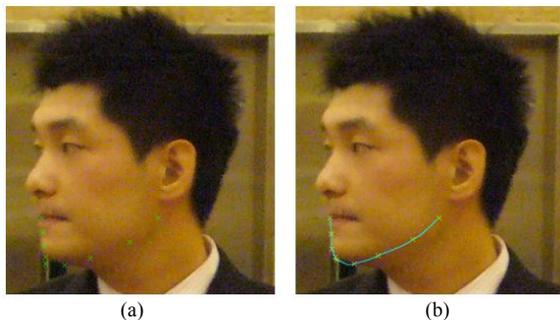


Fig.14 (a) The initialized control points of chin contour. (b) The auto-initialized shape-enhanced snake offers an approach to the profile face pose.

Also, we consider some special cases. In Fig.15, a fatter face with double chin, profile face pose, looking down, and complex background is tested. The wrinkle formed because of face pose or fatty face shape leads into a local minimum. The face image is more complicated and with more details on it. From the experimental results, the proposed method in this paper offers an acceptable approach to the test input image. Another contrary example is given in Fig.16. The input image is a face with long thin chin and the initial chin contour is hence far away from the real chin contour. After several iterations, the proposed snake successfully converges to the real chin contour. When there are some objects or noise near the chin contour, the proposed method also has good performance. For example, consider the face image with a cell phone near the face as shown in Fig.17(a), the integrity and the smoothness of the estimated chin contour is maintained. Except the internal constraint of stiffness and tension, this is not only because that the auto-initialized shape-enhanced snake offers a prototype of chin contour but because that the normal vector is used to narrow the searching region to a line segment instead of an area, which reduces the opportunity of falling into false minimum. Hence, a proper chin contour is derived by the proposed method, as shown in Fig.17(b).

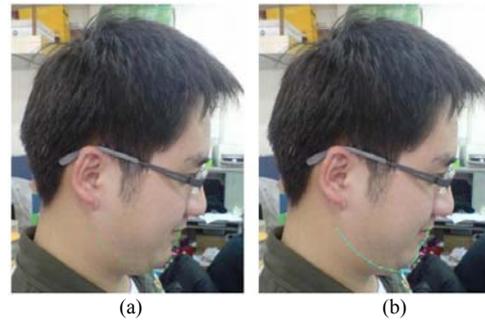


Fig.15 (a) The input face image is a fatter face with double chin, profile face pose, looking down, and complex background. (b) The reconstructed chin contour.

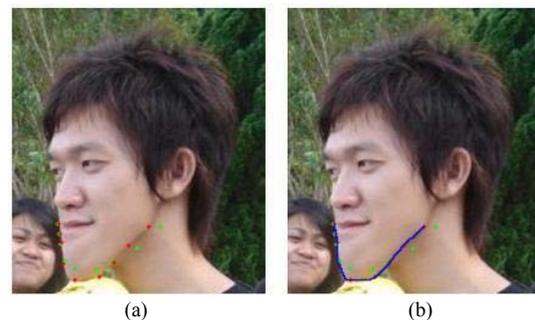


Fig.16 (a) Face image with a long and thin chin. Dots in green are the initial control points, while dots in red are the estimated chin contour. (b) An estimated chin contour is established when the chin is thinner and longer than the one in data base.

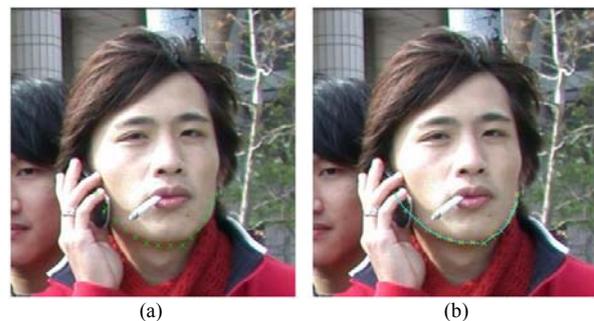


Fig.17 (a) Face image with a cell phone near the face (b) The chin contour is estimated with the information of initialized snake.

5. Conclusions

In this paper, a chin contour estimation algorithm is proposed, which is based on an auto-initialized shape-enhanced snake. By roughly estimated the face poses, the control points of snake can be appropriately initialized according to the 27 different face poses. The drawback of snake initialization is therefore improved. When dealing with the profile face poses, the auto-initialized snake along with the enhanced snake shape offers an approach to the profile face poses, even the boundary of profile face poses are very poor. The

experimental results show that the proposed method can estimate the chin contour with reliability and robustness under various face poses and several situations.

References

- [1] U. Qayyum and M. Y. Javed, "Real time notch based face detection, tracking and facial feature localization," in Int. Conf. on Emerging Technologies, 2006. ICET'06, Nov. 2006, pp. 70-75.
- [2] R. Belaroussi and M. Milgram, "Face tracking and facial feature detection with a webcam," in 3rd European Conf. on Visual Media Production, 2006. CVMP 2006, Nov. 2006, pp. 122-126.
- [3] R. Hefers, G. Verghese, K. Derpanis, R. McCready, J. Maclean, A. Levin, D. Topalovic, L. Wood, A. Jepson, and J. Tsotsos, "Detection and tracking of faces in real environments," in Proc. Int. Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, Sep. 1999, pp. 96-104.
- [4] M. H. Yang and N. Ahuja, "Detecting human faces in color images," in Proc. Int. Conf. on Image Processing, vol. 1, Oct. 1998, pp. 127-130.
- [5] L. V. Praseeda, S. Kumar, and D. S. Vidyadharan, "Face detection and localization of facial features in still and video images," in First Int. Conf. on Emerging Trends in Engineering and Technology, ICETET '08, Jul. 2008, pp. 95-99.
- [6] S. Boursas and H. Yan, "Chin extraction in colour frontal-view face images," in Digital Image Computing Techniques and Applications, Melbourne, Australia, Jan. 2002, pp. 1-6.
- [7] M. Kampmann, "Estimation of the chin and cheek contours for precise face model adaptation," in Proc. ICIP'97, Oct. 1997, pp. 300-303.
- [8] J. Y. Wang, G. D. Su, and X. G. Lin, "The research of chin contour in fronto-parallel images," in Proc. Second Int. Conf. on Machine Learning and Cybernetics, vol. 5, Mar. 2003, pp. 2814-2819.
- [9] Q. R. Chen, W. K. Cham, and K. K. Lee, "Chin contour estimation for face recognition," in IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP03), vol. 6, Apr. 2003, pp. 761-764.
- [10] Q. R. Chen, W. K. Cham, and K. K. Lee, "Extracting eyebrow contour and chin contour for face recognition," Pattern Recognition, vol. 40, pp. 2292-2300, Aug. 2007.
- [11] K. M. Lam and H. Yan, "An analytic-to-holistic approach for face recognition based on a single frontal view," in Proc. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 20, Jul. 1998, pp. 673-686.
- [12] R. L. Rudianto and K. N. Ngan, "Automatic 3d wireframe model fitting to frontal facial image in model-based video coding," Picture Coding Symposium (PCS '96), pp. 585-588, Mar. 1996.
- [13] C. L. Huang and C. W. Chen, "Human facial feature extraction for face interpretation and recognition," Pattern Recognition, vol. 25, pp. 1435-1444, Dec. 1992.
- [14] M. Hu, S. Worrall, A. Sadka, and A. Kondo, "A fast and efficient chin detection method for 2-d scalable face model design," in IEE Int. Conf. on Visual Information Engineering, UK, Jul. 2003, pp. 121-124.
- [15] P. Kuo and J. M. Hannah, "Improved chin fitting algorithm based on an adaptive snake," in IEEE Int. Conf. on Image Processing, Atlanta, US, Oct. 2006, pp. 205-208.
- [16] J. Y. Wang, G. D. Su, and X. G. Lin, "Improved chin fitting algorithm based on an adaptive snake," in Proc. Int. Conf. on Intelligent Computing, ICIC 2005, Hefei, China, Aug. 2005, pp. 203-212.
- [17] X. L. Ge, J. Yang, Z. L. Zheng, and F. Li, "Multi-view based face chin contour xtraction," Engineering Applications of Artificial Intelligence, vol. 19, 2006, pp. 545-555.
- [18] M. Kass, A. Witkin, and D. Terzopoulos, "Snake: active contour models", Int. J. Comouter Vision, vol. 1(4), 1987, pp. 321-331.
- [19] R. C. Gonzalez and R. E. Wood, Digital Image Processing, 2nd ed. Prentice Hall, 2001.



Kuo-Yu Chiu was born in Hsinchu, R.O.C., in 1981. He received the B.E. degree in electrical and control engineering from National Chiao Tung University, Hsinchu, Taiwan, R.O.C, in 2003. He is currently pursuing the Ph. D. degree in the Department of Electrical and Control Engineering, the National Chiao Tung University, Hsinchu, Taiwan. His current research interests include image processing, face recognition, face pose estimation, face replacement, intelligent transportation system, and machine learning.



Shih-Che Chien was born in Chiayi, R.O.C., in 1978. He received the B.E. degree in electronic engineering from the Nation Chung Cheng University, in 2002. He is currently pursuing the M.E. and Ph.D. degree in the Department of Electrical and Control Engineering, the National Chiao Tung University, Hsinchu, Taiwan. His current research interests include image processing, image recognition, fuzzy theory, 3D image processing, intelligent transportation system, and animation.



Sheng-Fuu Lin (S'84-M'88) was born in Tainan, R.O.C., in 1954. He received the B.S. and M.S. degrees in mathematics from National Taiwan Normal University in 1976 and 1979, respectively, the M.S. degree in computer science from the University of Maryland, College Park, in 1985, and the Ph.D. degree in electrical engineering from the University of Illinois, Champaign, in 1988. Since 1988, he has been on the faculty of the Department of Electrical and Control Engineering at National Chiao Tung University, Hsinchu, Taiwan, where he is currently a Professor. His research interests include image processing, image recognition, fuzzy theory, automatic target recognition, and scheduling.