Chapter 2
Cartoon Face Modeling from Image Feature Point Transformation

2.1 Introduction

To generate a talking cartoon face, the proposed system includes two major parts: a facial feature tracker and a face model transformer. The facial feature tracker tracks facial feature points from sequential facial images. Before doing this, a face model must be constructed first. Generally speaking, the feature points in the image and the control points in the face model are independent. Therefore, we may track a man's facial features and use a woman's face model to generate a talking cartoon face of the woman. This in turn means that a transformation between image feature points and face model control points is needed by the face model transformer. An overall configuration of the proposed system is shown in Figure 2.1

In this chapter, several techniques are proposed to achieve the purpose mentioned above. First, a review of Chen and Tsai [1] about constructing a face model is given in Section 2.2. Second, a technique is proposed to transform image feature points into face model control points in Section 2.3. In Section 2.4, a technique is proposed to create the cartoon face in two ways.

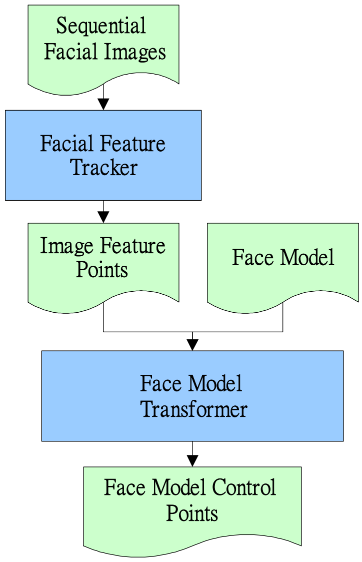


Figure 2.1 An overall configuration of proposed talking cartoon face generation system

2.2 Review of Adopted Cartoon Face Model

Chen and Tsai [1] proposed an automatic method for generation of cartoon faces from a neutral facial image. In their method, there are three major steps: extraction of facial feature regions, extraction of facial feature points, and face model creation. In the first step, background, hair, and face regions in images are extracted by hierarchical bi-level thresholding. A flowchart of the hierarchical bi-level thresholding method is shown in Figure 2.2. Then, the eyes are detected by finding all region pairs and matching the symmetry of the two regions in each pair. With the positions of the detected eye-pair as a reference, the facial feature regions are then extracted by a knowledge-based edge detection process.

Before extracting facial feature points, 72 feature points in a face were defined by simplifying the 84 feature points defined in the MPEG-4 standard. The 72 feature points and some *facial animation parameter units* (FAPUs), also defined in the MPEG 4 format, were set up as a new face model as shown in Figure 2.3. In the face model, some control points used to control cartoon face expressions are listed as follows:

1. *Eyebrow Control Points*: there are 8 control points both in the left and in the right eyebrow, namely, 4.2, 4.4, 4.4a, 4.6, 4.1, 4.3, 4.3a, and 4.5.
2. *Eye Control Points*: there are 4 control points in eyes, namely, 3.1, 3.3, .3.2, and 3.4.
3. *Mouth Control Points*: there are 4 control points in the mouth, namely, 8.9, 8.4, 8.3, and 8.2, by which other mouth points are calculated.

These control points in this study are the so-called *face model control points* mentioned previously. According to the face model, facial feature points are extracted from the above-mentioned facial feature regions.

Finally, two curve drawing algorithms are used to create cartoon faces. One is a corner-cutting subdivision algorithm, in which the basic concept is to repeatedly chop off corners of a polygon until reaching a certain condition, as shown in Figure 2.4. The other is a cubic Bezier curve approximation algorithm used to produce curves which appear reasonably smooth with a simple polynomial descriptor, as shown in Figure 2.5.

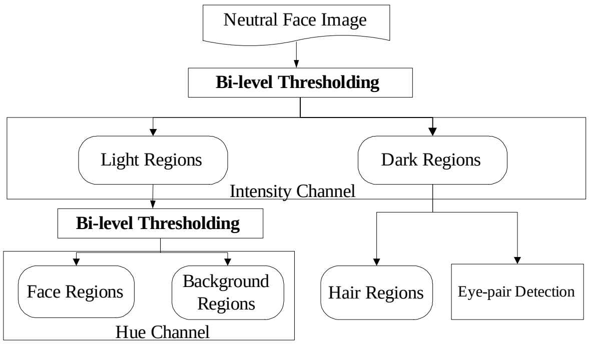


Figure 2.2 Flowchart of hierarchical bi-level thresholding method in Chen and Tsai [1].

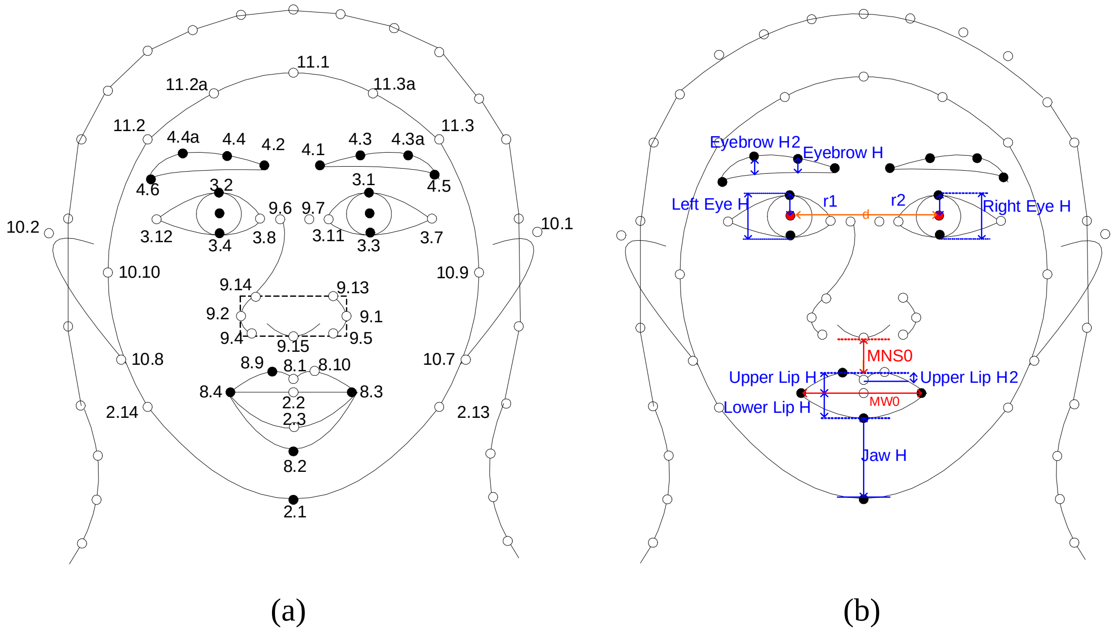


Figure 2.3 A face model. (a) Proposed 72 feature points. (b) Proposed facial animation parameter units in Chen and Tsai [1].

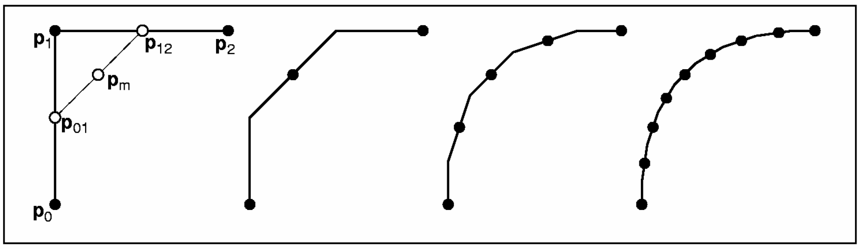


Figure 2.4 An illustration of corner-cutting algorithm in Chen and Tsai [1].

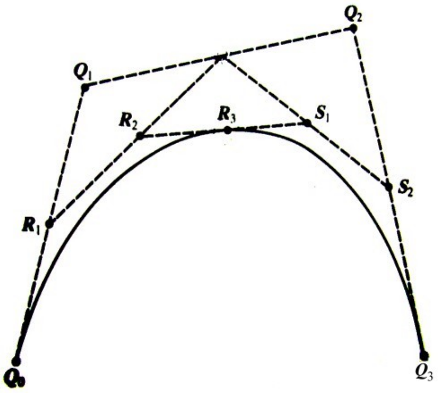


Figure 2.5 Cubic Bezier curve in Chen and Tsai [1].

2.3 Transformation of Image Feature Points into Face Model Control Points

2.3.1 Transformation Technique

Based on the face model mentioned in Section 2.2, a technique is proposed to transform image feature points into face model control points. The technique can be divided into two parts: one is for transforming eye feature points and the other for transforming mouth feature points. The basic idea is to compute the parameters for the transformation according to the ratios of face model feature values to image feature values. For example, we may require the ratio of the width of the mouth in the face model to the width of the mouth in a neutral image to be 3:2. The neutral image means the first frame of sequential facial images in this study. If now the width of the mouth in the sequential facial images is 24, it can be checked that the width of mouth should be 36 in the face model.

2.3.2 Transformation Involving Eye Feature Points

The image feature points about eye features in sequential facial images are two FAPUs, h_1 and h_2. The value of h_1 denotes the distance between eye points 3.2 and 3.4 in sequential facial images. Similarly, h_2 denotes the distance between eye points 3.1 and 3.3. Besides, some parameters about the facial features are used. Let H_{eye} denote the normal height of an open eye in the image, which is calculated using a neutral facial image. Let H'_{eye} denote the normal height of an open eye in the face model. The detailed transformation method is described in the following algorithm.

Algorithm 2.1. *Image feature point transformation for eyes.*

Input: two FAPUs h_1 and h_2.

Output: Four model eye points 3.2_{model}, 3.4_{model}, 3.1_{model}, and 3.3_{model}.

Steps:

1. Compute the height of the open left eye in the face model h_1' in the following way according to the geometric ratio principle:

$$h_1' = h_1 \times \frac{H'_{eye}}{H_{eye}}.$$

2. Set the point 3.2_{model} in the following way:

   $3.2_{\text{model}} = (x_1, y_2)$, where

   x_1 = the x-position of the left pupil;

   y_1 = the y-position of the left pupil – $h_1/2$.

3. Set the point 3.4_{model} in the following way:

   $3.4_{\text{model}} = (x_2, y_2)$, where

   x_2 = the x-position of the left pupil;

   y_2 = the y-position of the left pupil + $h_1/2$.

4. The points 3.1_{model} and 3.3_{model} are computed in a similarly way.

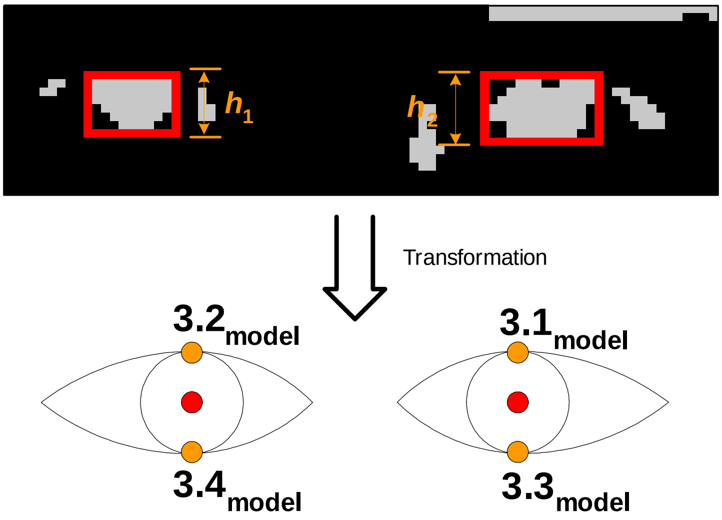


Figure 2.6 An illustration of image feature point transformation for eyes.

2.3.3 Transformation Involving Mouth Feature Points

For mouth point transformation, a new point is defined in the face model. A *basic mouth point* is defined to be the center of the closed mouth in the face model. The basic mouth point is used to correct the *y*-position of the mouth points. More details of the computation of the basic mouth point will be described in the next chapter. Besides, some parameters about the facial features are used. Let W_{mouth} and H_{mouth} denote the normal width and the normal height of the mouth in the images, respectively, which are calculated in the neutral facial image. Let W'_{mouth} and H'_{mouth} denote the normal width and the normal height of the mouth in the face model, respectively. Let H_{upperlip} denote the height of the upper lip in the neutral facial image and $H'_{upperlip}$ denote the height of the upper lip in the face model. The detail of the transformation is described in the following algorithm.

Algorithm 2.2. *Image feature point transformation for the mouth.*

Input: Four tracked mouth points 8.9_{track}, 8.4_{track}, 8.3_{track}, and 8.2_{track}, and a basic mouth point.

Output: Four model mouth points 8.9_{model}, 8.4_{model}, 8.3_{model}, and 8.2_{model}.

Steps:

1. Compute the width of the mouth in a face model *w* in the following way according to the geometric ratio principle:

$$w = (\text{the } x\text{-position of } 8.3_{\text{track}} - \text{the } x\text{-position of } 8.4_{\text{track}}) \times \frac{W'_{mouth}}{W_{mouth}}.$$

2. Based on the basic mouth point in the face model $M_{\text{model}}(x_{\text{m}}, y_{\text{m}})$ and the basic mouth point in the images $M_{\text{image}}(x_{\text{i}}, y_{\text{i}})$, set the point 8.4_{model} in the following way:

$8.4_{\text{model}} = (x_1, y_1)$, where

$x_1 = x_m - w/2$;

$y_1 = y_m +$ (the y-position of $8.4_{\text{track}} - y_i$).

3. Set the point 8.3_{model} in a similarly way.

4. Compute the open range l_1 of the upper lip in the images in the following way:

   $l_1 = y_i -$ the y-position of $8.9_{\text{track}} - H_{\text{upperlip}}$.

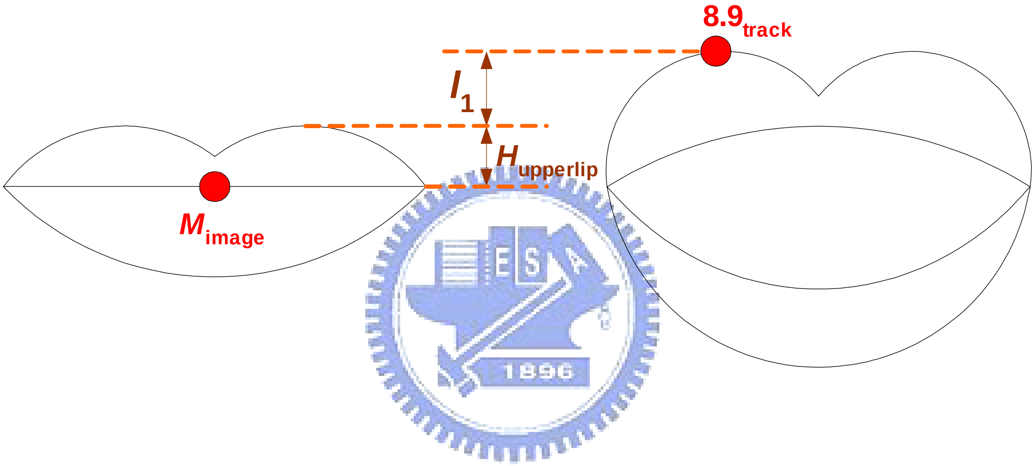


Figure 2.7 The open range of the upper lip.

5. Transform l_1 into the open range of the upper lip in the face model l_1' in the following way according to the geometric ratio principle:

$$l_1' = l_1 \times \frac{H'_{mouth}}{H_{mouth}}.$$

6. Based on the basic mouth point in the face model $M_{\text{model}}(x_m, y_m)$ and the basic mouth point in the images $M_{\text{image}}(x_i, y_i)$, set the point 8.9_{model} in the following way:

   $8.9_{\text{model}} = (x_2, y_2)$, where

   $x_2 =$ the original x-position of 8.9_{model} in the face model;;

   $y_2 = y_m - l_1' - H'_{upperlip}$.

7. Set the point 8.2_{model} in a similarly way.

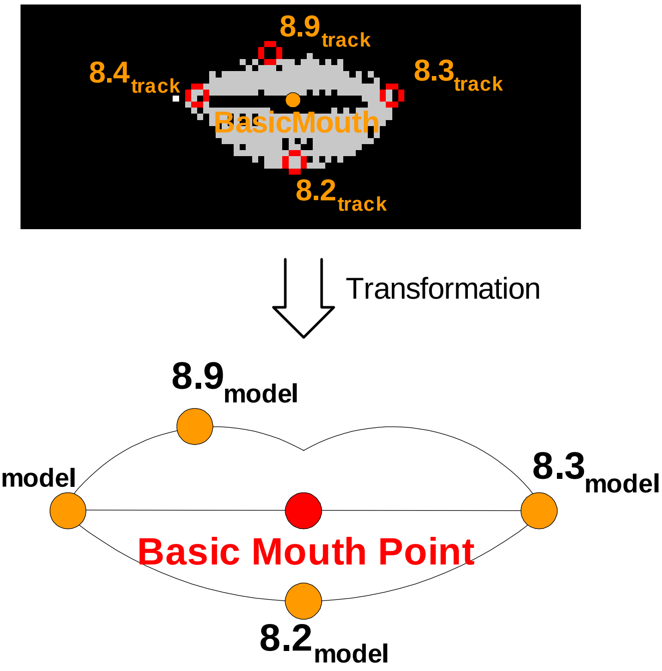


Figure 2.8 An illustration of image feature point transformation for mouth.

2.4 Creation of Cartoon Face

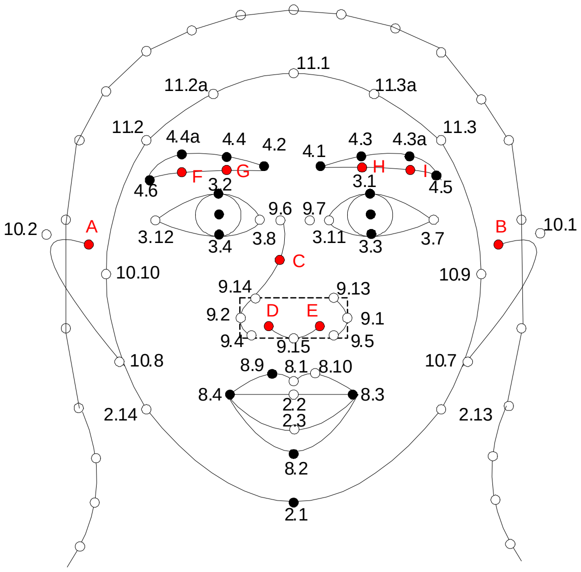


Figure 2.9 Points to help drawing.

As mentioned in Section 2.2, the cartoon face is created by the corner-cutting subdivision and cubic Bezier curve approximation algorithms. In this study, there are two types of cartoon faces: frontal cartoon face and oblique cartoon face. It is hoped to represent a head-turning talking cartoon face by the two types of cartoon faces. Some points used to help drawing are shown in Figure 2.9.

2.4.1 Creation of Frontal Cartoon Face

A frontal cartoon face is drawn by the 72 feature points of the face mentioned previously. The static parts of a cartoon face, like the hair, the contour of a face, the eyebrows, the nose, and the ears, are created first. Then the dynamic parts of the cartoon face, like the eyes and the mouth, are created. Some FAPUs in the face model are used. Let $P_{\text{mideye}}(x_{\text{m}}, y_{\text{m}})$ denote the position of the middle of the eye pairs. And let W_{ear} denote the distance between the eye and a ear in the face model. Also, let H_{eyebrow} and $H'_{eyebrow}$ denote the two heights of the eyebrows as shown in Figure 2.3. The detail of the proposed frontal face creation method is described in the following algorithm.

Algorithm 2.3. *Creation of frontal cartoon face.*

Input: 72 feature points in the face model.

Output: an image of the frontal cartoon face.

Steps:

1. Let $arc(P_1, \ldots, P_n)$ denote a curve composed by the points $P_1, \ldots, P_n$.

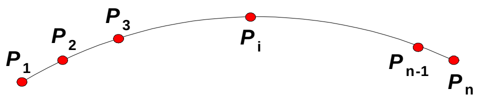


Figure 2.10 An illustration of $arc(P_1, \ldots, P_n)$.

2. Let x_{p} and y_{p} denote the *x*-position and the *y*-position of a point *P* in the face model.

3. Draw the contour of the hair by a polygon composed by 23 hair feature points.

4. Draw the contour of the face, including the forehead, cheek, and jaw, by the cubic Bezier curves *arc*(11.3, 10.9, 2.13), *arc*(2.13, 2.1, 2.14), *arc*(2.14, 10.10, 11.2), and *arc*(11.2, 11.2*a*, 11.1, 11.3*a*, 11.3).

5. Draw the contour of the left ear by the cubic Bezier curves *arc*(*A*, 10.2, 10.8), where the point $A(x_A, y_A)$ is calculated in the following way:

$$x_A = x_{\mathrm{m}} - W_{\mathrm{ear}}, \qquad y_A = (4\times y_{10.2} + y_{10.8})/5.$$

6. Draw the contour of the right ear in a similarly way.

7. Draw the contour of the nose by the cubic Bezier curves *arc*(9.6, *C*, 9.14), *arc*(9.14, 9.2, 9.4), *arc*(9.13, 9.1, 9.5), and *arc*(*D*, 9.15, *E*), where the points $C(x_{\mathrm{C}}, y_{\mathrm{C}})$, $D(x_{\mathrm{D}}, y_{\mathrm{D}})$, and $E(x_{\mathrm{E}}, y_{\mathrm{E}})$ are calculated in the following way:

$$x_{\mathrm{C}} = (x_{9.13} + x_{9.14})/2, \qquad y_{\mathrm{C}} = (y_{9.14} + y_{\mathrm{m}})/2.$$

$$x_{\mathrm{D}} = (x_{9.4} + x_{9.5})/6, \qquad y_{\mathrm{D}} = y_{9.4} - (y_{9.4} - y_{9.14})/3.$$

$$x_{\mathrm{B}} = (x_{9.4} + x_{9.5})/6, \qquad y_{\mathrm{D}} = y9.5 - (y_{9.4} - y_{9.14})/4.$$

8. Draw the contour of the left eyebrow by the corner-cutting subdivision curves arc(4.2, 4.4, 4.4*a*, 4.6) and arc(4.6, *F*, *G*, 4.2), where the points $F(x_{\mathrm{F}}, y_{\mathrm{F}})$ and $G(x_{\mathrm{G}}, y_{\mathrm{G}})$ are calculated in the following way:

$$x_{\mathrm{F}} = x_{4.4\mathrm{a}}, \qquad y_F = y_{4.4a} + H'_{eyebrow}/1.3\,.$$

$$x_{\mathrm{G}} = x_{4.4}, \qquad y_{\mathrm{F}} = y_{4.4} + H_{\mathrm{eyebrow}}/1.3.$$

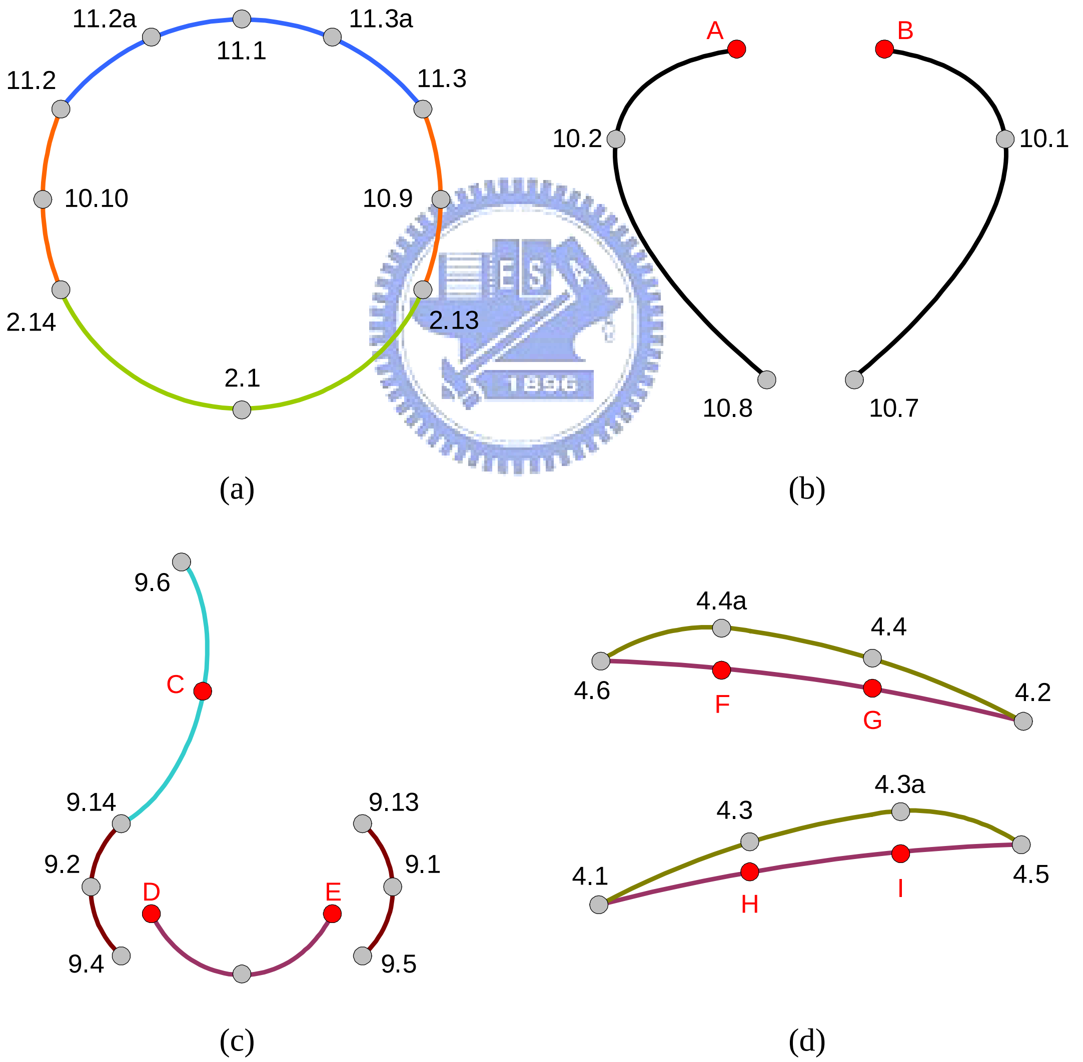
9. Draw the contour of the right eyebrow in a similarly way.

10. Draw the contour of the left eye by the cubic Bezier curves *arc*(3.8, 3.2,

3.12), *arc*(3.12, 3.4, 3.8).

11. Draw the contour of the right eye in a similarly way.
12. Draw the contour of the mouth by the cubic Bezier curves *arc*(8.4, 8.9, 8.1), *arc*(8.1, 8.10, 8.3), *arc*(8.3, 2.2, 8.4), *arc*(8.4, 2.3, 8.3), and *arc*(8.3, 8.2, 8.4).

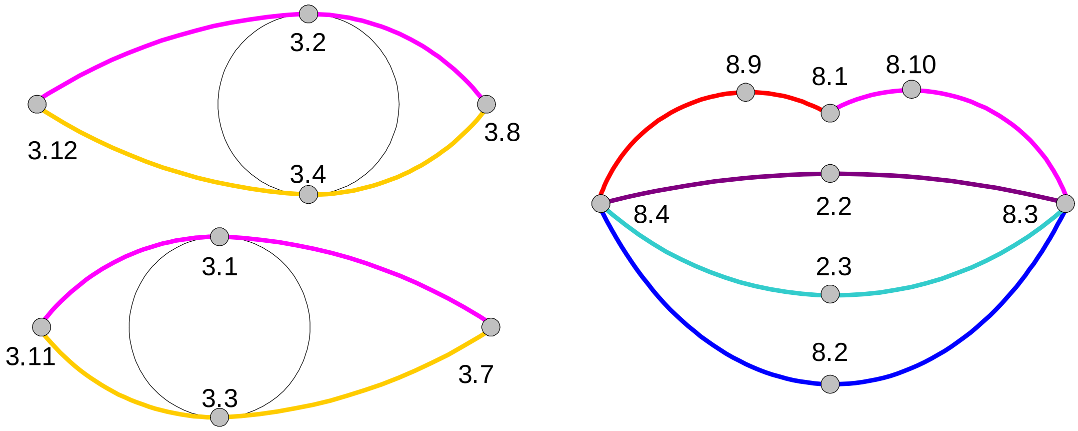
An illustration of the steps in the creation of the frontal cartoon face as described above is shown in Figure 2.11. An experimental result of the creation of a frontal cartoon face is shown in Figure 2.12.



(a) (b)

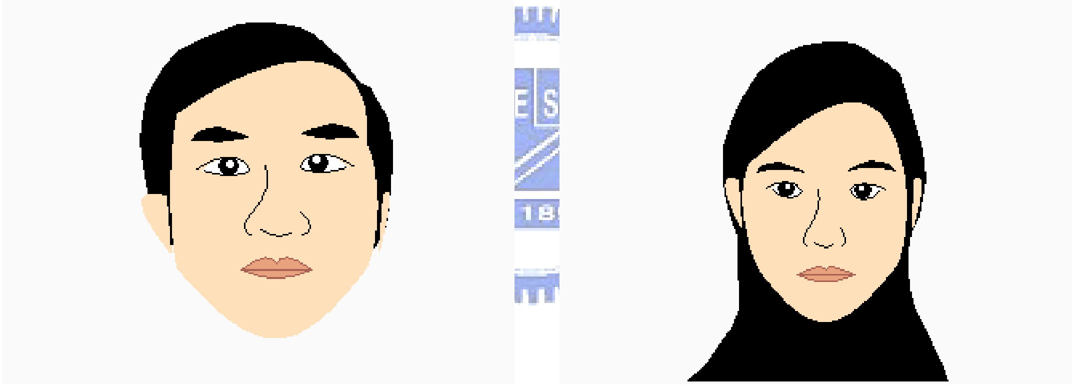
(c) (d)

Figure 2.11 An illustration of the steps in the creation of frontal cartoon face. (a) The creation of the contour of a face. (b) The creation of the ear. (c) The creation of the nose. (d) The creation of the eyebrow. (e) The creation of the eye. (f) The creation of the mouth. (continued)



(e) (f)

Figure 2.11 An illustration of the steps in the creation of frontal cartoon face. (a) The creation of the contour of a face. (b) The creation of the ear. (c) The creation of the nose. (d) The creation of the eyebrow. (e) The creation of the eye. (f) The creation of the mouth. (continued)



(a) (b)

Figure 2.12 An experimental result of the creation of frontal cartoon face. (a) A male face model. (b) A female face model.

2.4.2 Creation of Oblique Cartoon Face

The basic idea of creation of the oblique cartoon face is to shift some face model control points to change the contour of the frontal cartoon face. In this study, the shifted facial features in a face model are the eyes, mouth, eyebrows, nose, and ears. And the shifted contours of a face in a face model are the cheeks, jaw, and forehead. The face model control points are shifted according to a value *t* specifying the information about head turning. The sign of *t* denotes the direction of head turning. The absolute value of *t* denotes the range of head turning. By different ranges of the shifted value, the facial features are skewed to represent various oblique cartoon faces. The detailed method for adjusting the related points is in the following algorithm.

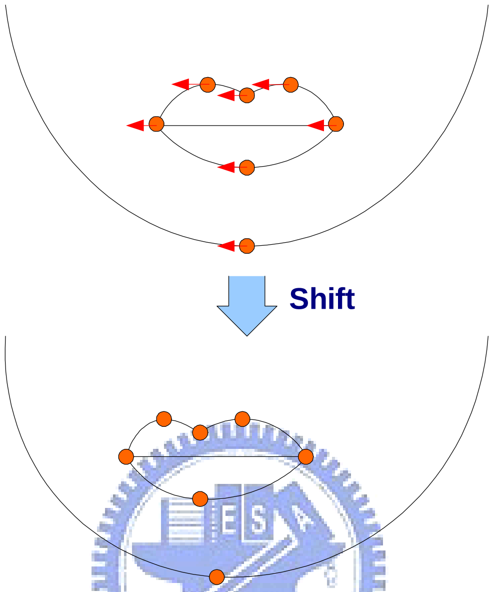


Figure 2.13 An illustration of creation of oblique cartoon face.

Algorithm 2.4. *Creation of oblique cartoon face.*

Input: 72 feature points in the face model, and a value *t*.

Output: an image of a oblique cartoon face.

Steps:

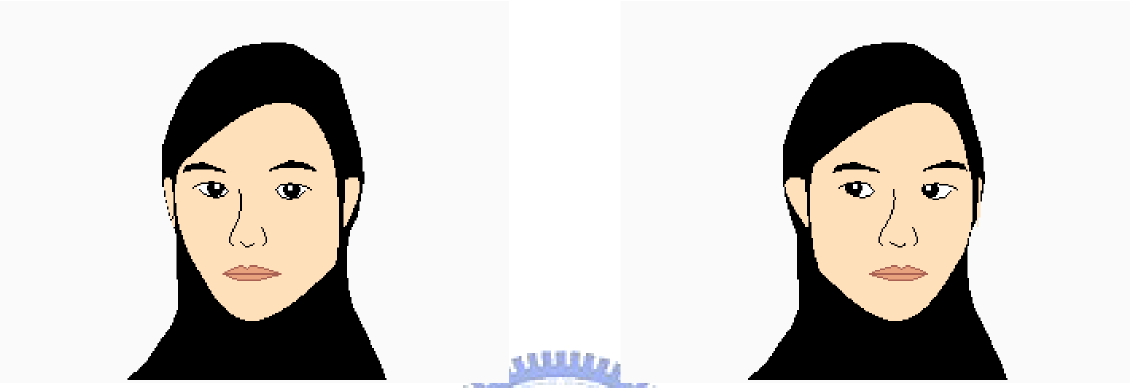
1. Adjust the cheeks by subtracting half of *t* from the *x*-position of the cheek points.
2. Adjust the forehead in the following way:

   2.1 change the middle of the forehead points by subtracting *t* from the *x*-position of the point;

   2.2 change the two extremities of the forehead points by subtracting half of *t* from the *x*-position of the points.

3. Adjust the points of the jaw, ears, nose, eyebrows, eyes, and mouth in a similarly way.
4. Apply Algorithm 2.3 to create the desired oblique cartoon face.

An illustration of the creation of oblique cartoon faces is shown in Figure 2.14

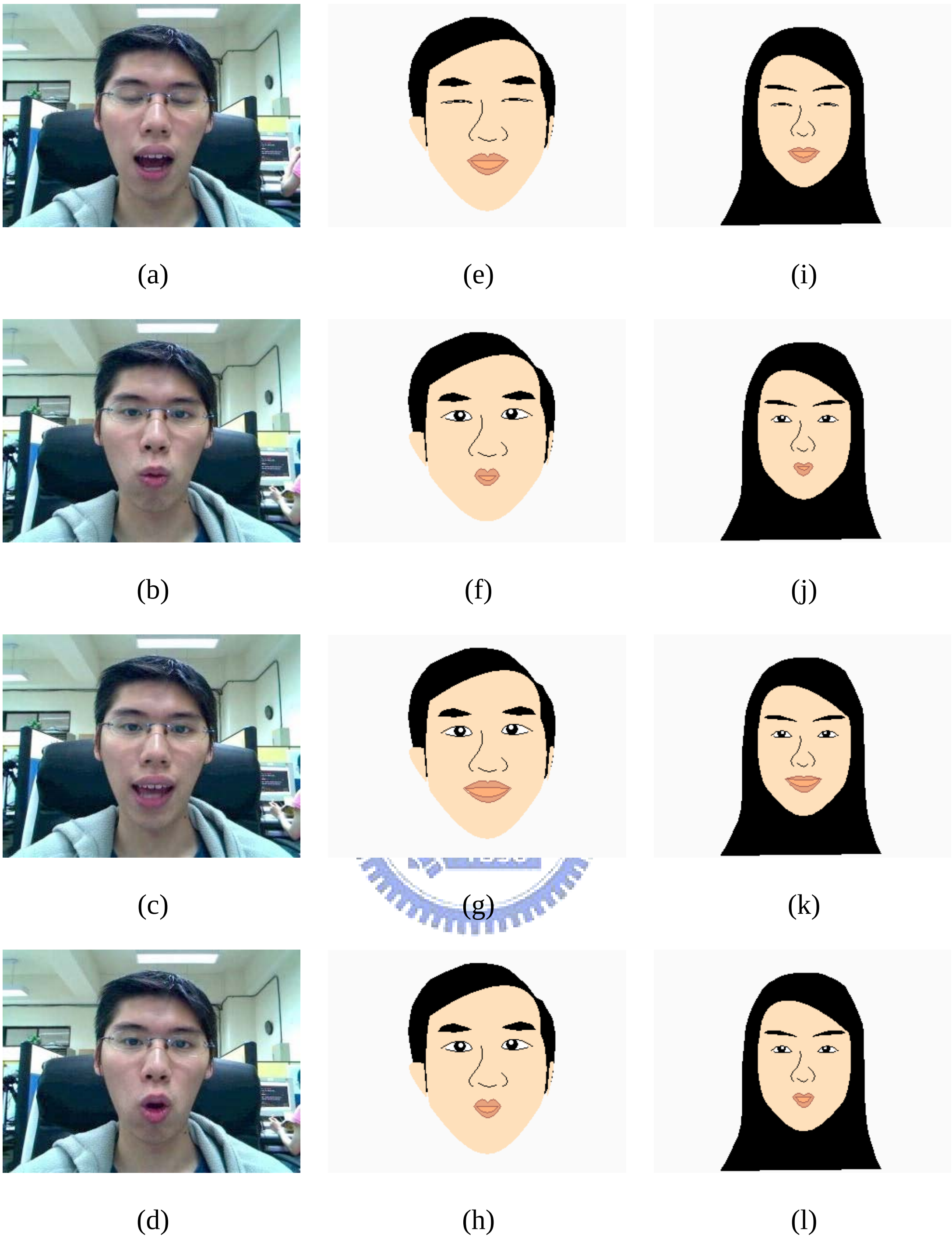


(a) (b)

Figure 2.14 An illustration of creation of oblique cartoon faces. (a) A oblique cartoon face with t = 8. (b) A oblique cartoon face with t = –8.

2.5 Experimental Results

Some experimental results of model control point transformation and creation of cartoon faces are shown here. Two different face models are controlled by the same user as shown in Figure 2.15 and Figure 2.16.



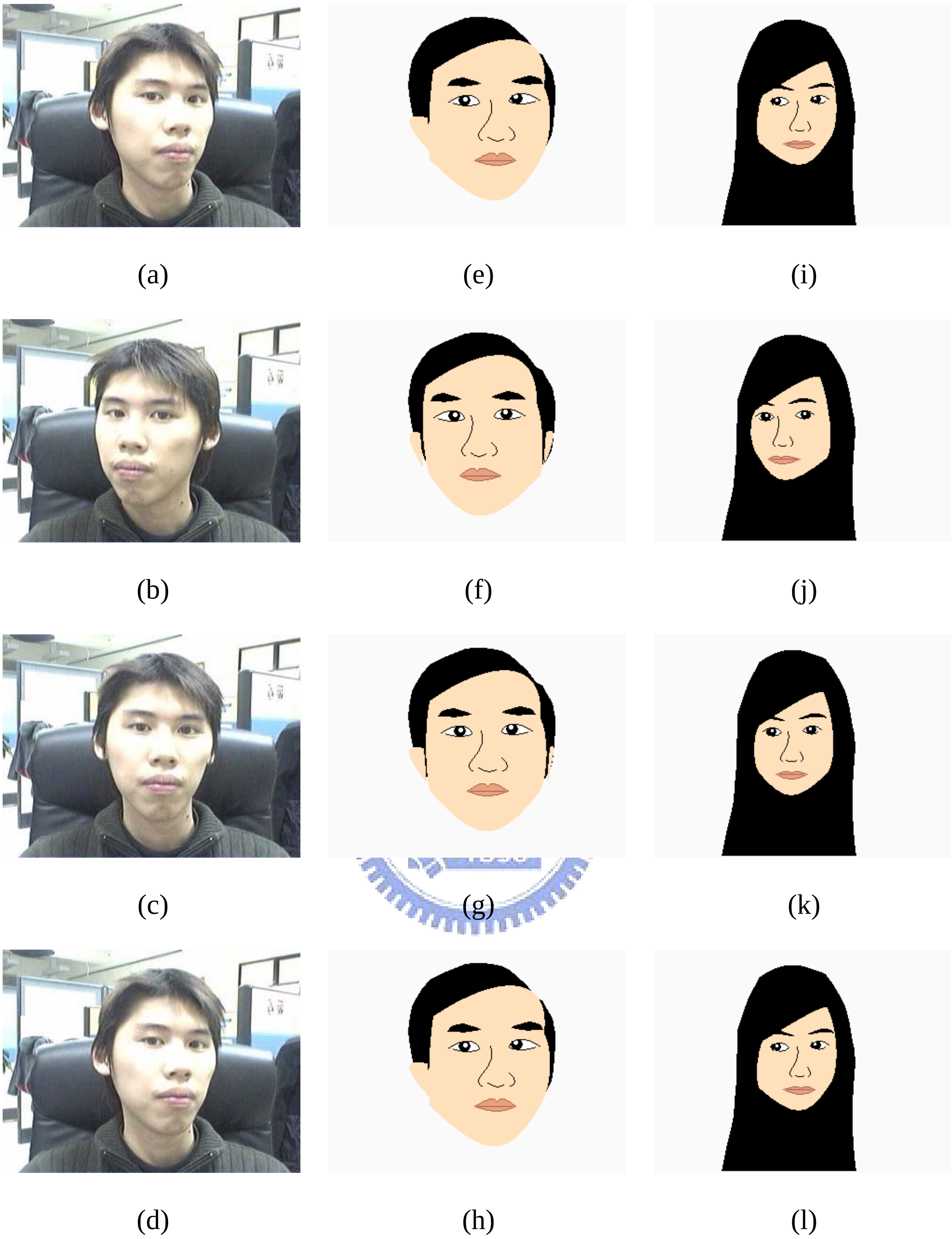
(a) (e) (i)

(b) (f) (j)

(c) (g) (k)

(d) (h) (l)

Figure 2.15 An example of experimental results for fontal cartoon faces. (a)(b)(c)(d) The input sequential facial image. (e)(f)(g)(h) The first result of cartoon face. (i)(j)(k)(l) The second result of cartoon face.



(a) (e) (i)

(b) (f) (j)

(c) (g) (k)

(d) (h) (l)

Figure 2.16 An example of experimental results for oblique cartoon faces (a)(b)(c)(d) The input sequential facial image. (e)(f)(g)(h) The first result of cartoon face. (i)(j)(k)(l) The second result of cartoon face.