Chapter 3
Facial Feature Tracking from Image Sequences in Complicated Backgrounds

3.1 Introduction

Tracking facial features from sequential facial images have been a popular research topic for many years. There are many facial features that can be tracked, such as eye, eyebrow, mouth, jaw, nose, cheek, and so on. Generally speaking, there are drawbacks in existing methods on this topic, like being expensive, inconvenient, time-consuming, and complicated. In this study, a system for tracking facial features in real time without the drawbacks mentioned above is proposed. The equipment needed in the system is only a common camera, so the cost of the system is cheap and the operation of the system is convenient. Besides, the image feature points tracked from sequential facial images to control the eyes and the mouth of the face model are only 6 points. So the system is simple and is not time-consuming. The details of the proposed method for facial feature tracking are described in this chapter. A flowchart of the method is shown in Figure 3.1.

In Section 3.2, a review of an employed tracking method is described. In Section 3.3, some proposed methods for segmentation of eye and mouth regions are described. In Section 3.4, a method proposed for eye-pair tracking in complicated backgrounds is described. In Section 3.5, a method proposed for lip tracking in complicated backgrounds is described. In Section 3.6, a method proposed for detection of head turnings is described.

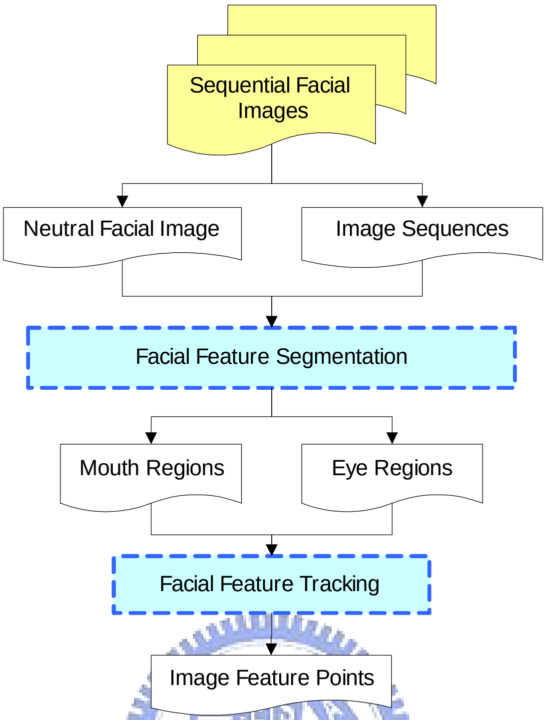


Figure 3.1 Flowchart of proposed method for facial feature tracking.

3.2 Review of Employed Tracking Method

In Chen and Tsai [1], a convenient method for facial feature tracking from sequential facial images was proposed. The first frame of the sequential facial images is chosen as a neutral image. And the eye and mouth regions are tracked in different ways. For eye regions, the neutral image in sequential facial images is analyzed to get a threshold value t_1 and an initial eye region before tracking. After tracking is started, sequential binary images are yielded by bi-level thresholding in the intensity channel using t_1. Then using the pixels of the eye regions in the previous frame as seeds and applying region growing, some possible eye regions are extracted. A new eye-pair region is obtained by an eye-pair detection process. And the positions of the pupils are calculated by an eyeball detection method. If no eye-pair is detected, the old eye-pair region is used as the new ones.

For mouth regions, three horizontal division lines, four threshold values, and an initial mouth region are obtained from analyzing the neutral image before tracking. After starting tracking, sequential binary images are obtained by thresholding the Sobel edge images using the four threshold values. Using the pixels of the previous mouth regions as seeds and applying region growing, a new mouth region is extracted. A flowchart of talking cartoon face generation from image sequences is shown in Figure 3.2.

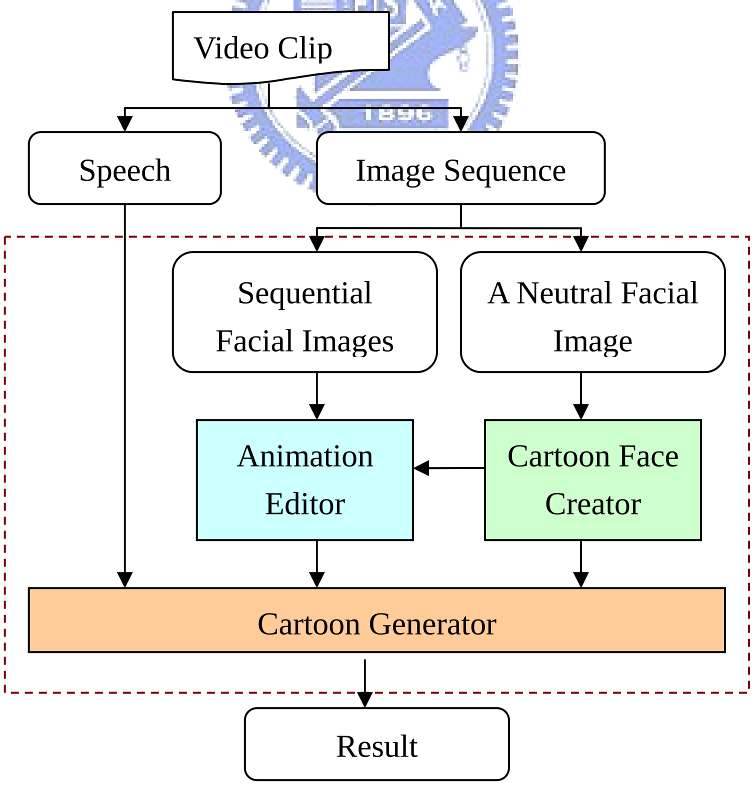


Figure 3.2 A flowchart of talking cartoon face generation from image sequences in Chen and Tsai [1]

Although the previously-mentioned method for facial feature tracking is convenient and simple, it has many drawbacks. First, it is unstable and fails easily due to existence of shadows, shaky heads, insufficient lighting, complicated grounds, and so on. Second, the resulting eye blinking and lip movement are not smooth enough. Therefore a new method for real-time facial feature tracking based on Chen and Tsai's method is proposed in this study, and described subsequently.

3.3 Segmentation of Facial Feature Regions

The hierarchical bi-level thresholding method in Chen and Tsai's method is used to separate the facial feature regions, the hairs, and the backgrounds in sequential facial images. But it is for creating a face model mainly, not for facial feature tracking. In order to let the system more stable and efficient, some methods for segmentation of facial feature regions aiming at facial feature tracking are proposed in this study. In Section 3.3.1, a method proposed for segmentation of eye regions using bi-level thresholding in the intensity channel is described. In Section 3.3.2, a method proposed for segmentation of mouth regions using chromatic color features is described.

3.3.1 Segmentation of Eye Regions

The eye-pair detection process is important in this study because the position of the eye-pair is used frequently in other facial feature detection processes. In this method, a central rectangle is used to collect the color information of a user's face. And the threshold value used to get the binary image is computed in the rectangle. The details of the method are described in the following algorithm.

Algorithm 3.1. *Segmentation of eye regions.*

Input: a neutral facial image *N*, a facial image sequence *S*, and a central rectangle *Rect* with size $m \times n$.

Output: A binary version B_{eye} of *S*.

Steps:

1. Transform *N* into a grayscale image *I*, with $I(x, y)$ denoting the intensity value at pixel of *I* at (x, y).
2. Compute the mean value m_1 and the standard deviation value s_1 of the pixels in *Rect* by the following way:

$$m_1 = \frac{1}{m \times n} \sum_{x,y \in Rect} I(x, y);$$

$$s_1 = \left(\frac{1}{m \times n - 1} \sum_{x,y \in Rect} \left(I(x, y) - m_1 \right)^2 \right)^{1/2}.$$

3. Compute a threshold value t_1 in the following way:

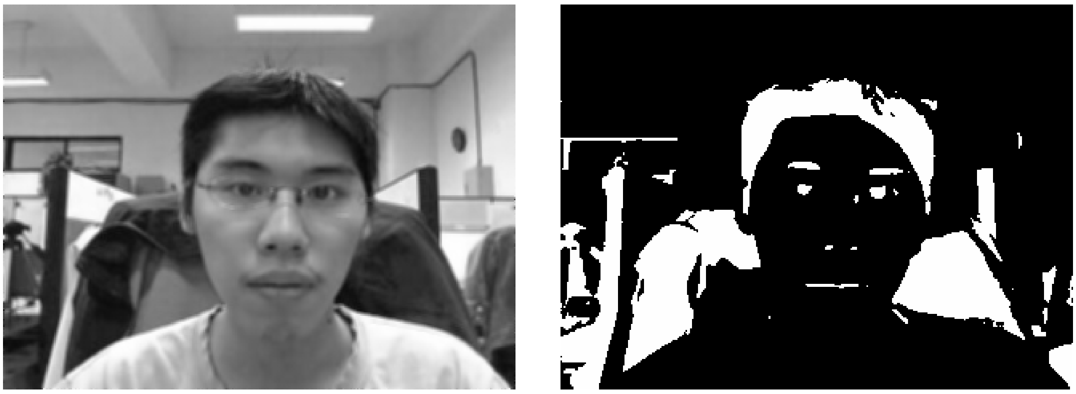
$$t_1 = m_1 - s_1.$$

4. Threshold *I* with t_1 to get a binary image $B(x, y)$ by the following way:

$$B(x, y) = \begin{cases} 0, & \text{if} \quad I(x, y) > t_1; \\ 1, & \text{if} \quad I(x, y) \le t_1. \end{cases}$$

5. Repeat Steps 6 to 7 for each image in *S*.
6. Transform S_i into a grayscale image $I_i(x, y)$, where S_i denotes the *i*-th image in *S*.
7. Threshold I_i with t_1 to get B_i, where B_i denotes the *i*-th binary image in B_{eye}.

An experimental result of the above algorithm is shown in Figure 3.3.



(a) (b)

Figure 3.3 An experimental result of segmentation of eye-pair regions in intensity channel. (a) The neutral facial image in intensity channel (b) The binary image

3.3.2 Segmentation of Mouth Regions

To reduce the errors of mouth tracking, a useful method for segmentation of mouth regions is important. Using chromatic color features, a method for this purpose is proposed in this section. According to the position of the eye-pair region, a rectangle that covers the mouth region is speculated. And an octagon included in the rectangle is computed to adapt to the contour of the mouth. Some threshold values are computed by analyzing the normalized red and green intensity values of the pixels in the octagon in a normalized chromatic space. The normalized chromatic space is defined by the following equation based on an image in the RGB space:

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}.$$

The mouth region is separated by thresholding the sequential facial images with these values. The details of the method are described in the following algorithm.

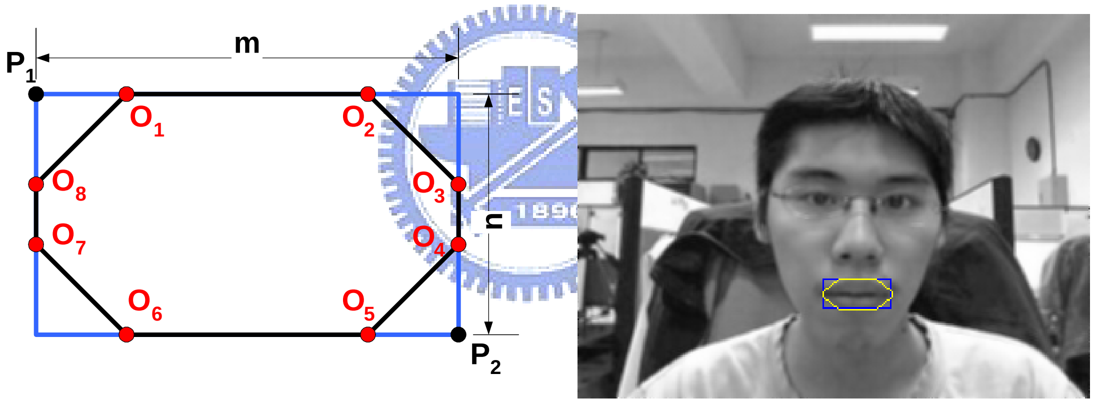
Algorithm 3.2. *Segmentation of mouth regions.*

Input: a neutral facial image *N*, and a facial image sequence *S*.

Output: A binary version B_{mouth} of *S*.

Steps:

1. Speculate one horizontal division line Div_{NL} by adding the *y*-position of the middle of the eye to the distance *d* between the eye pairs.
2. Obtain a rectangle *Rect* with size $m \times n$ according to the *x*-position of the pupils and Div_{NL}.
3. Obtain an octagon *Octa* in *Rect* by cutting the isosceles triangles at the four corners of *Rect*, as shown in Figure 3.4.



(a) (b)

Figure 3.4 An illustration of the octagon fitting in the rectangle. (a) The composing points in the octagon. (b) The octagon in a binary image.

4. Create a histogram of the normalized red intensity values *r* to analyze the pixels in *Octa*.
5. Find the distribution of the histogram and set a range of the normalized red intensity values *r*. Denote the range as $[r_{\text{lower}}, r_{\text{upper}}]$.
6. Compute a range of the normalized green intensity values *g* as $[g_{\text{lower}},$

g_{upper}] in a similarly way.

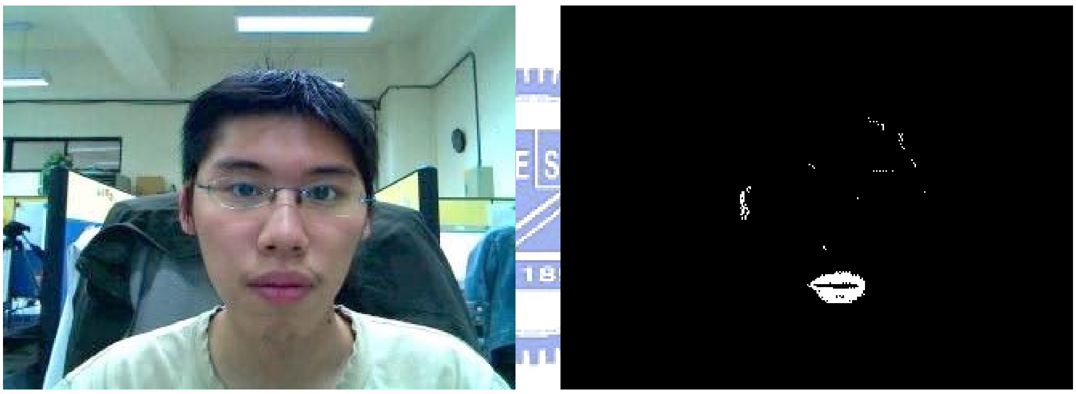
7. Threshold N to get a binary image $B(x, y)$ by the following way:

$$B(x, y) = \begin{cases} 1, & \text{if} \quad r(x, y) \in [r_{\text{lower}}, r_{\text{upper}}] \quad \text{and} \quad g(x, y) \in [g_{\text{lower}}, g_{\text{upper}}]; \\ 0, & \text{othwise.} \end{cases}$$

8. Threshold S_i with [r_{lower}, r_{upper}] and [g_{lower}, g_{upper}] to get B_i, where B_i denotes the i-th binary image in B_{mouth}.

9. Repeat Step 8 for each image in S.

An experimental result of segmentation of mouth regions is shown in Figure 3.5.



(a) (b)

Figure 3.5 An experimental result of segmentation of mouth regions. (a) The neutral facial image. (b) The resulting binary image.

3.4 Proposed Method of Eye-pair Tracking

In the process of real-time facial feature tracking, many unexpected situations are not forecasted. Therefore, some additional detection is needed to avoid these situations. In Section 3.4.1, a basic extraction process of eye-pair regions is described. In Section 3.4.2, some additional detection operations to correct the errors of the extraction are described.

3.4.1 Extraction of Eye-pair Regions

In this section, a method for extraction of the height of an open eye is proposed. Let L_{eye} and R_{eye} denote the positions of the pupils for tracking the eye regions. The initial positions of L_{eye} and R_{eye} are computed in the neutral facial image by Chen and Tsai's eye-pair detection process. The details of the method for extraction of eye-pair regions in each frame of sequential facial images are described in the following algorithm.

Algorithm 3.3. *Extraction of eye-pair regions in each frame of sequential facial images.*

Input: *a* binary image B_{eye}, a value n, and two points L_{eye} and R_{eye}.

Output: two heights h_{left} and h_{right} of the open left and right eyes, respectively.

Steps:

1. Use the pixels in the $n \times n$ square with a center at L_{eye} as seeds and apply region growing to extract the left eye region R_{lefteye} with size $m_1 \times n_1$.
2. Extract the right eye region R_{righteye} with size $m_2 \times n_2$ in a similarly way.
3. Use Chen and Tsai's eye-pair detection process to compute the positions of the pupils in the current frame.
4. If no eye-pair is detected, recover the old positions of the pupils in the

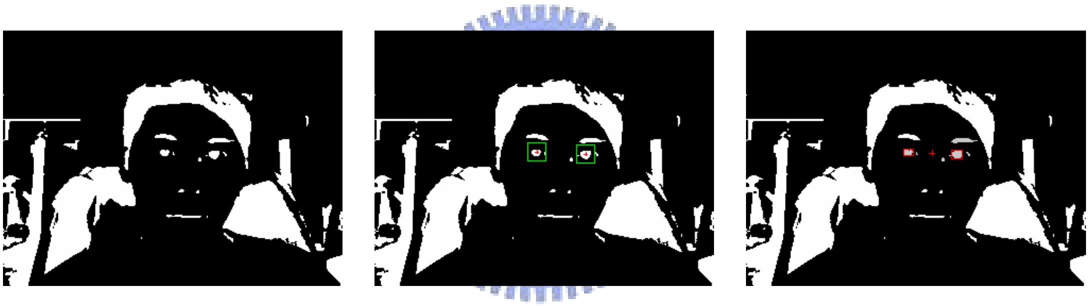
previous frame. Otherwise, set the points L_{eye} and R_{eye} to be the positions of the pupils in the current frame for the extraction of eye-pair regions in the next frame.

5. Set the values h_{left} in the following way:

$$h_{\text{left}} = \begin{cases} 0, & \text{if no eye-pair is detected;} \\ n_1 & \text{othwise.} \end{cases}$$

6. Set the values h_{right} in a similarly way.

An experimental result of the extraction of eye-pair regions is shown in Figure 3.6.



(a) (b) (c)

Figure 3.6 An experimental result of extraction of eye-pair regions. (a) The binary image B_{eye}. (b) The $n \times n$ square in B_{eye} with n = 9. (c) The final result of extraction of eye regions.

3.4.2 Correction of Region Extraction Errors

In this section, a correction process for the errors of the extraction of eye-pair regions in relative sequential facial images is described. Let P_{left} and P_{right} denote the positions of the pupils which are calculated by Algorithm 3.3. Besides, a value $y_{correct}$ is calculated to correct the errors in the extracted mouth regions in the last section. The details of the proposed method are described in the following algorithm.

Algorithm 3.4. *Correction for extraction errors of eye-pair regions in each frame of sequential facial images.*

Input: two positions of the pupils $P_{\text{left}}(x_1, y_1)$ and $P_{\text{right}}(x_2, y_2)$, and two positions of the pupils for tracking the eye regions $L_{\text{eye}}(x_3, y_3)$ and $R_{\text{eye}}(x_4, y_4)$.

Output: a value y_{correct}.

Steps:

1. Compute a value e for decision of errors in the following way according to the difference between the positions of the pupils in the current frame and in the previous frame:

$$e = \max\{|x_1 - x_3|, |x_2 - x_4|, |y_1 - y_3|, |y_2 - y_4|\}.$$

2. If $e > 5$, it is meant that a wrong eye-pair is detected. Then recover the values of P_{left}, P_{right}, L_{eye}, R_{eye}, h_{left} and h_{right} using the old values in the previous frame, where h_{left} and h_{right} denote the height of the left and right open eyes.

3. Set the point $P_{\text{mideye}}(x_{\text{mideye}}, y_{\text{mideye}})$ in the following way:

$$x_{\text{mideye}} = (x_1 + x_2)/2;$$

$$y_{\text{mideye}} = (y_1 + y_2)/2.$$

4. Compute the value y_{correct} in the following way:

   $y_{\text{correct}} = y_{\text{mideye}} - y'_{\text{mideye}}$, where y'_{mideye} denotes the original y-position of the middle of the eye.

A flowchart of the eye-pair tracking method, including the extraction of eye-pair regions and the correction of extraction, is shown in Figure 3.7.

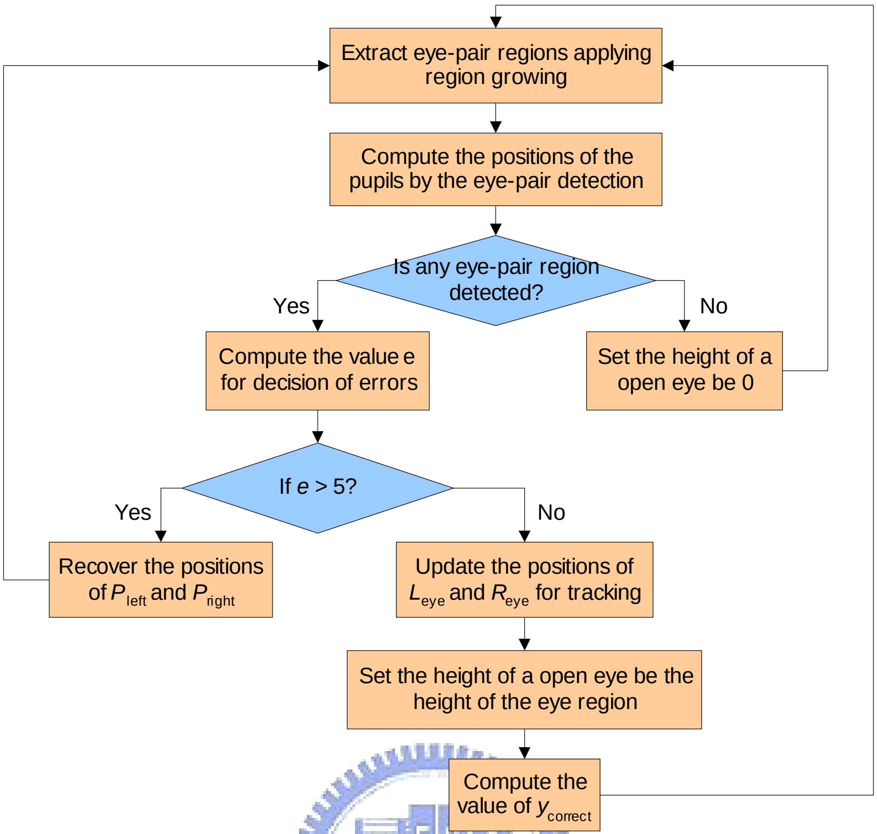


Figure 3.7 A flowchart of eye-pair tracking method.

3.5 Proposed Method of Mouth Tracking

On the whole, the method for mouth tracking is more difficult than the method for eye-pair tracking because the lip movement is very complicated. In Section 3.4.1, a basic extraction of mouth regions is described. In Section 3.4.2, some additional detection steps to correct the errors of the extraction are described.

3.5.1 Extraction of Mouth Regions

Based on Chen and Tsai's method, a mouth region and four mouth feature points are extracted from the binary facial image sequence obtained in Section 3.3.2. Some parameters are used in the method for mouth region tracking. Let H_{upperlip} and H_{lowerlip} denote the heights of the upper lip and the lower lip, respectively. Let W_{mouth} denote the normal width of the mouth. They are computed in the neutral facial image as follows in Algorithm 3.5. The details of the proposed method for extraction of mouth regions in sequential facial images are described in the following Algorithm 3.6.

Algorithm 3.5. *Preprocessing for extraction of mouth regions in the neutral facial image.*

Input: a binary neutral facial image N_{mouth}.

Output: a rectangle R_{mouth}, a basic mouth point $M_{\text{basic}}(x_{\text{basic}}, y_{\text{basic}})$, three FAPUs H_{upperlip}, H_{lowerlip}, and W_{mouth}, and four mouth feature points 8.9_{ori}, 8.2_{ori}, 8.4_{ori}, and 8.3_{ori}.

Steps:

1. Compute three FAPUs H_{upperlip}, H_{lowerlip}, and W_{mouth}, and four mouth feature points 8.9_{ori}, 8.2_{ori}, 8.4_{ori}, 8.3_{ori} by applying Chen and Tsai's mouth detection method mentioned in Section 2.2.
2. Compute a rectangle R_{mouth} in the following way:

$$R_{\text{mouth}} = [r_1, r_2, r_3, r_4]$$

where r_1 = the x-position of 8.4_{ori}, r_2 = the y-position of 8.9_{ori}, r_3 = the x-position of 8.3_{ori}, and r_4 = the y-position of 8.2_{ori}.

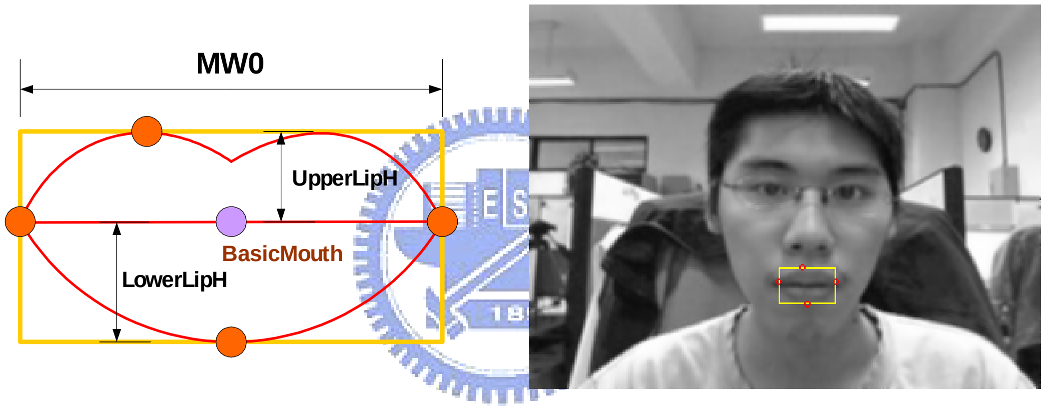
3. Set the point $M_{\text{basic}}(x_{\text{basic}}, y_{\text{basic}})$ in the following way:

x_{basic} = the x-position of 8.2_{ori};

y_{basic} = the y-position of 8.3_{ori}.

Figure 3.8 An illustration of rectangle R_{mouth}.

An experimental result of preprocessing for the extraction of mouth regions in a neutral image is shown in Figure 3.9.



(a) (b)

Figure 3.9 An experimental result of preprocessing for extraction of mouth regions. (a) An illustration of mouth region. (b) The preprocessing result of mouth region extraction.

Algorithm 3.6. *Extraction of mouth regions in each frame of sequential facial images.*

Input: a binary image B_{mouth}, a rectangle R_{mouth}, a value w, a value y_{correct}, a basic mouth point $M_{\text{basic}}(x_{\text{basic}}, y_{\text{basic}})$, and four mouth feature points 8.9_{ori}, 8.2_{ori}, 8.4_{ori}, and 8.3_{ori}.

Output: four tracked mouth feature points 8.9_{track}, 8.2_{track}, 8.4_{track}, and 8.3_{track}.

Steps:

1. Use the pixels of R_{mouth} as seeds and apply region growing on B_{mouth} to extract an $m \times n$ mouth region R'_{mouth}.
2. Set the points 8.9_{track} and 8.2_{track} in the following way:

   $8.9_{\text{track}} = (x, y)$, where

   x = the x-position of 8.9_{ori};

   y = the uppermost y-position of $R'_{\text{mouth}} - y_{\text{correct}}$;

   $8.2_{\text{track}} = (x, y)$, where

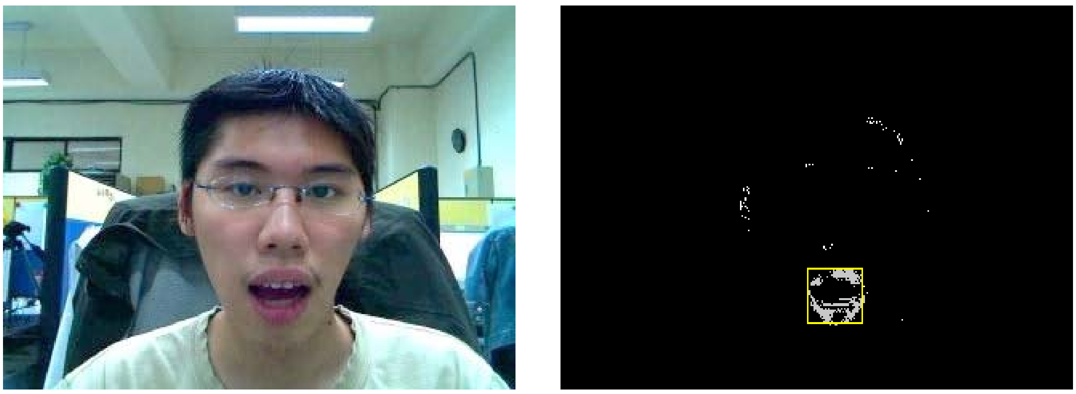
   x = the x-position of 8.2_{ori};

   y = the lowermost y-position of $R'_{\text{mouth}} - y_{\text{correct}}$.
3. Average the leftmost and rightmost w columns of the y-positions of the pixels in R_{mouth} as a value h.
4. Set the points 8.4_{track} and 8.3_{track} in the following way:

   $8.4_{\text{track}} = (x, y)$, where $x = x_{\text{basic}} - m/2$, $y = h - y_{\text{correct}}$;

   $8.3_{\text{track}} = (x, y)$, where $x = x_{\text{basic}} + m/2$, $y = h - y_{\text{correct}}$.

An experimental result of the extraction of mouth regions is shown in Figure 3.10.



(a) (b)

Figure 3.10 An experimental result of extraction of mouth regions. (a) The original image. (b) Final result of mouth region tracking

3.5.2 Correction of Region Extraction Errors

To avoid the misses of mouth tracking, some corrections are proposed in this section. Two kinds of errors are detected in this section. First, an absolute error that means the mouth region is too small to be a normal mouth region is detected. Second, a relative error that means the mouth regions have an unreasonable change in size between the two consecutive frames is detected. Let R_{mouth} denote the mouth region in the current frame. Let R''_{mouth} denote the mouth region in the previous frame. The details of the proposed method are described in the following algorithm.

Algorithm 3.7. *Correction for extraction of mouth regions in each frame of sequential facial images.*

Input: two mouth regions R_{mouth} with size $m_1 \times n_1$ and R''_{mouth} with size $m_2 \times n_2$, three FAPUs H_{upperlip}, H_{lowerlip}, and W_{mouth}.

Output: corrected mouth feature points.

Steps:

1. Detect and correct the absolute error as follows:

   if $m_1 < W_{\text{mouth}}/2$ or $n_1 < 3\times\left(H_{\text{upperlip}} + H_{\text{lowerlip}}\right)/4$, keep the values of the mouth feature points in the previous frame; otherwise, apply Algorithm 3.6 to extract new mouth feature points.

2. Detect the relative error between two consecutive mouth regions as follows:

   if $\max\{|m_1 - m_2|, |n_1 - n_2|\} > 10$, keep the values of the mouth feature points in the previous frame; otherwise, apply Algorithm 3.6 to extract new mouth feature points and set the mouth region R''_{mouth} in the following way for decision of relative errors:

$$R''_{\text{mouth}} = R_{\text{mouth}}.$$

A flowchart of the proposed mouth tracking method, including the extraction of mouth regions and the correction of extraction, is shown in Figure 3.11.

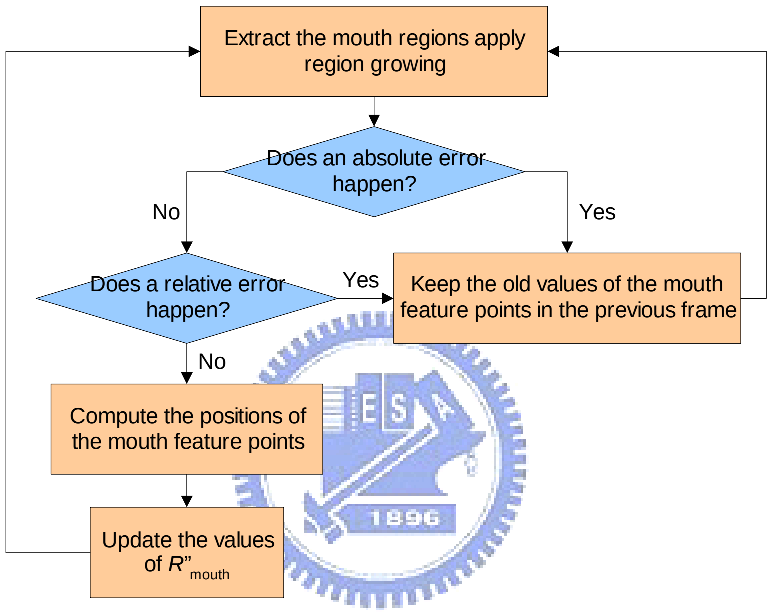


Figure 3.11 A flowchart of proposed mouth tracking method

3.6 Detection of Head Turning

In many researches for facial feature tracking, a user's head is always kept frontal without shaking. This restriction causes the generated virtual face to be stiff and toneless. Therefore, it is attempted in this study to let the talking cartoon face turn its head with the turning of a user' head. In Section 3.6.1, an analysis of head turnings and the basic idea are described. In Section 3.6.2, the process of detection of head turnings is described.

3.6.1 Basic Idea

By observing the process of a real man's head turnings, it is discovered that the distance between the cheeks and the center of the mouth changes with the range of the head turnings. In this study, some values are defined for detection of head turnings. Let D_{LfCheek} denote the distance between the center of the mouth and the right cheek, and let D_{RtCheek} denote the distance between the center of the mouth and the left cheek. Let R_{turn} denote the ratio of D_{LfCheek} to D_{RtCheek}. The details of the proposed process are described in the next Section.

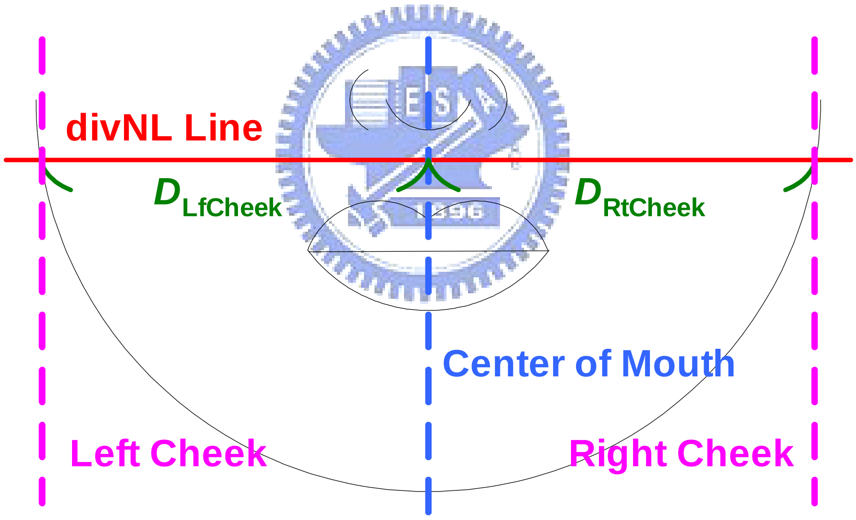


Figure 3.12 Some definitions for detection of head turning.

3.6.2 Detection Process

As mentioned above, the value of R_{turn} is computed to detect head turnings. After the detection of head turnings, an integer value t will be calculated. The value is used to create oblique cartoon faces in the way mentioned in Section 2.4.2. The details of the proposed method are described in the following algorithm.

Algorithm 3.8. *Detection of head turnings.*

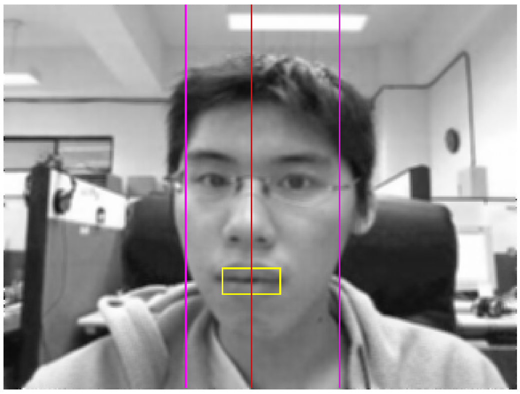
Input: a Sobel edge image E, a horizontal division line Div_{NL}, and a mouth region R_{mouth}.

Output: a value t for creation of oblique cartoon faces.

Steps:

1. Let the region R_{mouth} be denoted by $[(x_1, y_1), (x_2, y_2)]$, where (x_1, y_1) denote the leftmost and uppermost points in R_{mouth}, respectively, and (x_2, y_2) denote the rightmost and lowermost point in R_{mouth}, respectively.
2. Follow the horizontal division line Div_{NL} to find the x-position of the left and right cheeks a and b in E.
3. Compute the distance $D_{LfCheek}$ between the center of the mouth and the right cheek in the following way:
$$D_{LfCheek} = b - (x_1 + x_2)/2.$$
4. Compute the distance $D_{RtCheek}$ between the center of the mouth and the left cheek in a similarly way.
5. Compute the ratio value R_{turn} in the following way:
$$R_{turn} = D_{RtCheek}/D_{LfCheek}.$$
6. Set the value t in the following way:
$$t = \begin{cases} 8, & \text{if } R_{turn} > 2.0; \\ 2\times\left(\left(R_{turn} - 1.25\right)/0.25 + 1\right), & \text{if } 2.0 \le R_{turn} < 0.8; \\ 0, & \text{if } 0.8 \le R_{turn} \le 1.25; \\ -2\times\left(\left(0.8 - R_{turn}\right)/0.1 + 1\right), & \text{if } 0.5 \le R_{turn} < 0.8; \\ -8, & \text{if } R_{turn} < 0.5. \end{cases}$$

An experimental result of the detection of head turnings is shown in Figure 3.13.



(a) (b)

Figure 3.13 An experimental result of detection of head turnings. (a) The Sobel edge image *E* (b) The *x*-position of the left and right cheeks and the center of the mouth.

3.7 Experimental Results

Some experimental results of applying the proposed methods for extraction of facial features and detection of head turnings from sequential facial images are shown here. First, two experimental results of eye-pair region and mouth tracking in sequential facial images are shown in Figure 3.14 and Figure 3.15. Second, an experimental result for detection of head turnings in sequential facial images is shown in Figure 3.16.

Figure 3.14 An experimental result of eye-pair region tracking in sequential facial images.

Figure 3.15 An experimental result of mouth region tracking in sequential facial images.

Figure 3.16 An experimental result of detection of head turnings in sequential facial images.