AUTOMATIC COMIC FACE GENERATION FROM HUMAN FACE PICTURES

Yen-Lin Chen (陳彥霖)[1] and Wen-Hsiang Tsai (蔡文祥)[1, 2]

[1]Department of Computer and Information Science
National Chiao Tung University, Hsinchu, Taiwan, R.O.C.
[2]Department of Computer Science and Information Engineering
Taichung Healthcare & Management University, Taichung, Taiwan, R.O.C.
E-mail: {gis91529, whtsai}@cis.nctu.edu.tw

ABSTRACT

A system for automatic generation of comic faces by face image analysis is proposed. A face model of 72 facial feature points is first proposed for personal comic face drawing and modification. Accordingly, a method consisting of hierarchical bi-level thresholding and knowledge-based facial feature detection is proposed to extract facial features automatically from a given neutral face image. Finally, an editable and opened vector-based XML language of W3C (World Wide Web Consortium) standard-SVG (Scalable Vector Graphics) is used for rendering comic faces. Good experimental results show the feasibility of the proposed methods.

1. INTRODUCTION

Being able to have a comic face on behalf of somebody on computers and networks makes people feel comfortable and friendly to one another. In the past studies, comic faces were made by computers [2-5]. It is evident that there are more advantages to use comic faces than to use photorealistic ones. First, there is more freedom in designing stylish comic faces. Second, expressions can be modified and exaggerated by relocating predefined feature points. And last but not least, comic faces are more interesting but may be created with less data.

In this paper, a system for automatic generation of comic faces by analysis of face images is proposed. A 72-point face model is first proposed for personal face drawing. Hierarchical bi-level thresholding and knowledge-based facial feature detection techniques are proposed to extract facial features automatically from a given natural face image. To demonstrate the feasibility of low-cost virtual face creation, we use web-cameras to capture human faces with frontal lights and white background in this study. Then related parameters computed and comic faces are created automatically using the SVG (Scalable Vector Graphics) language for comic face rendering in the automatic creation process. Good experimental results showing the feasibility of the proposed methods are also included.

In the remainder of this paper, we first give an overview of the proposed method in Section 2. The face model used in the proposed method is described next in Section 3. The method is detailed in Sections 4 and 5, and some experimental results are given in Section 6, followed by some conclusions in Section 7.

2. OVERVIEW OF PROPOSED METHOD FOR COMIC FACE GENERATION

The proposed system for comic face generation includes two major parts: a facial feature extractor and a comic face generator, as shown in Fig. 1.

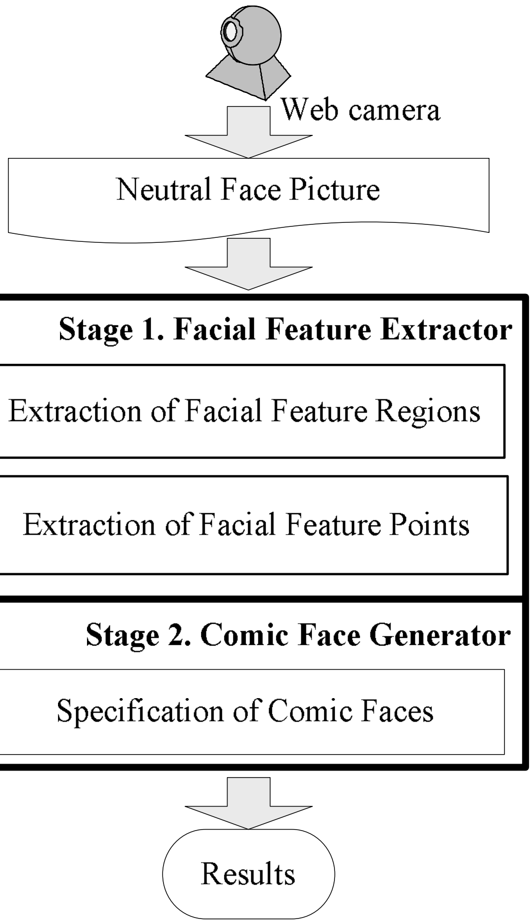


Fig. 1 Proposed system organization and two comic face generation stages.

The first part, the facial feature extractor, in which two main steps are performed, is used to analyze an input face image that includes a large frontal *neutral* face with no expression. The first step is extraction of facial feature regions and the next is extraction of facial feature points in the facial feature regions. Besides, some facial feature points are assigned to be control points for easier deformation of comic faces.

The second part, the comic face generator, is used to render the styles and expressions of a comic face by using the SVG language, which can be viewed directly by the Internet Explorer browser.

3. PROPOSED FACE MODEL

In order to set up a face model, one way is to properly fit one's face and to draw the corresponding comic face by assigning some points to the graphic functions supported by the SVG. The use of facial feature points helps us to draw and to deform a neutral comic face.

Two curve fitting algorithms (including a line drawing function and a cubic Bezier curve function) are used in this study. We apply the line drawing function to draw the hair and eyebrows, and the cubic Bezier curve function to draw the remaining facial features.

According to the MPEG-4 standard, a face model with 72 facial feature points is adopted, as illustrated in Fig. 2.

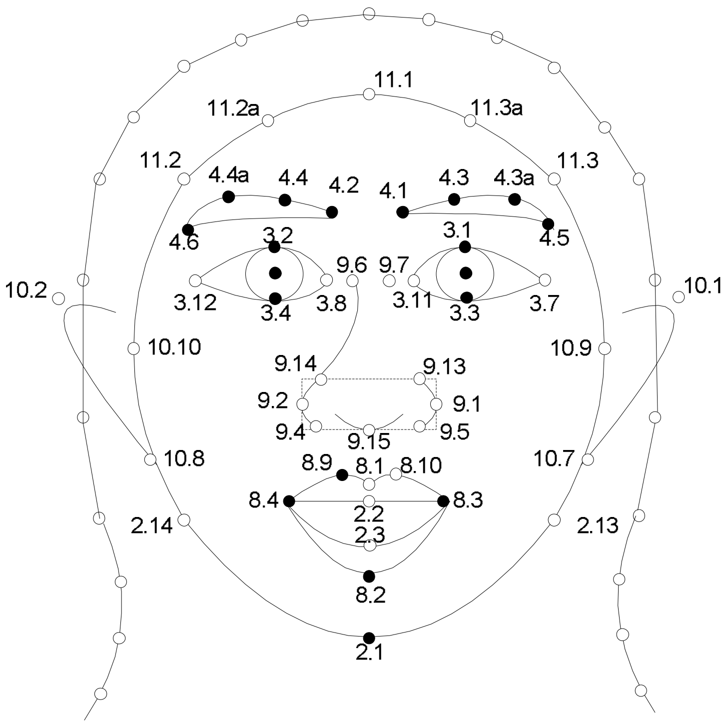


Fig. 2 Adopted face model with 72 feature points.

4. EXTRACTION OF FACIAL FEATURE REGIONS

Extracting facial feature regions help us to estimate the positions and ranges of facial features. A hierarchical bi-level thresholding method and a knowledge-based technique are proposed in this study to extract essential parts of a human face.

4.1. Hierarchical Bi-level Thresholding Method

The proposed hierarchical bi-level thresholding method is illustrated in Fig. 3.

A. First-Level Thresholding in Intensity Channel

We assume that a predetermined central rectangular range can be used to collect the grey-level information of a user's face. By computing the mean value m_1 and the standard deviation value s_1 within the central rectangle of the neutral face image, a threshold value $t_1 = m_1 - s_1$ is obtained for use to differentiate dark and light pixels. After grouping these two kinds of pixels, dark regions and light regions are extracted. An example of grey-level neutral face images and the corresponding result are shown in Figs. 4(a) and (b).

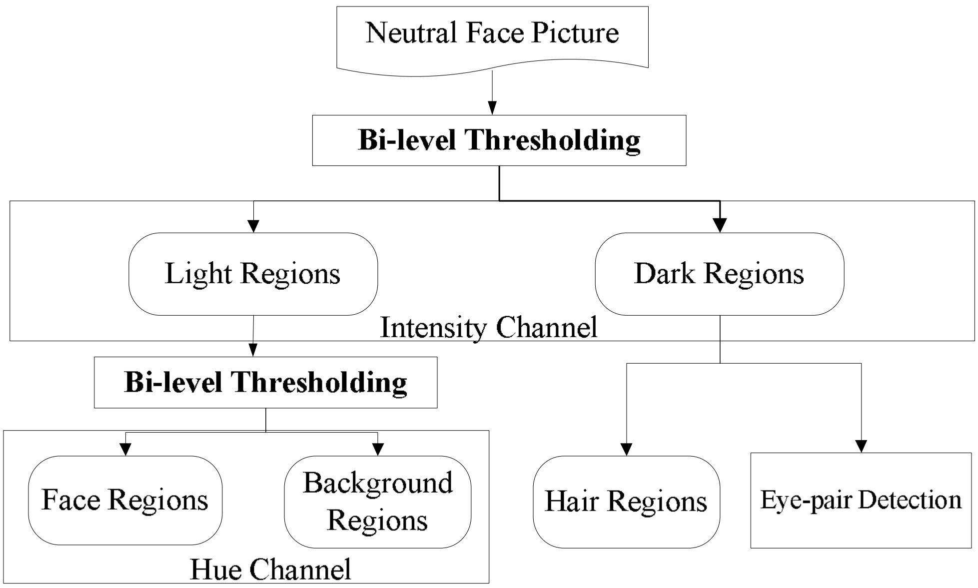
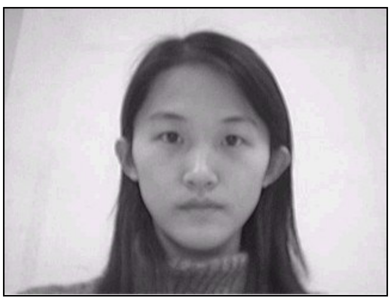
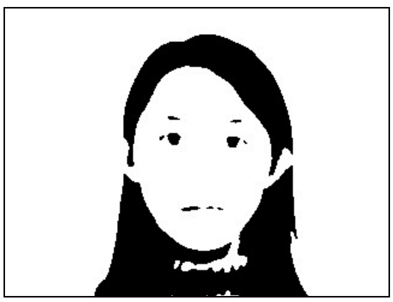


Fig. 3 A flowchart of hierarchical bi-level thresholding method.



(a)



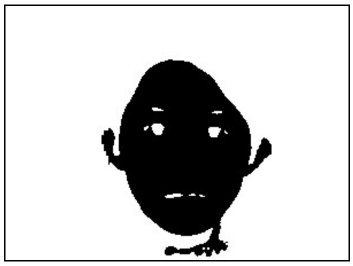
(b)

Fig. 4 Some experimental results. (a) A neutral face image in intensity channel. (b) A result of first-level thresholding.

B. Second-Level Thresholding in Hue Channel

For those light regions mentioned above, a second-level thresholding method is required to further differentiate between face regions and background regions. A mean value m_2 and the standard deviation value s_2 in the central rectangle mentioned previously in H are computed. For each pixel (x, y) belonging to the light regions, a Euclidean distance between $H(x, y)$ and m_2 is computed. Merging pixels with distances smaller than s_2, a set of face regions is obtained. A neutral face image in the hue channel and the extracted face regions are shown in Figs. 5(a) and (b).

(a)



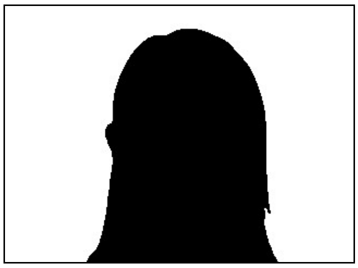
(b)

Fig. 5 Some experimental results. (a) A neutral face image in hue channel. (b) A result of second-level thresholding.

C. Region Refinement

By applying region merging techniques, a final face region and a final hair region are obtained. The background regions are then eliminated. An experimental result is shown in Fig. 6.

(a)



(b)

Fig. 6 Some experimental results. (a) A final face region in black. (b) A final hair region in black.

4.2. Knowledge-based Technique for Extraction of Facial Feature Regions

A knowledge-based technique is applied to extract detailed facial feature regions including eyebrow regions, eye regions, nose regions, and mouth regions. It is based on a concept to speculate the positions and ranges of facial feature regions by the information about the detected eye-pair.

A. Optimal Eye-pair Detection for Position Estimation

Since the dark regions consist of eye regions and hair regions, we can apply Chan's method [6] for eye-pair generation. While Chan's method yields one or more probable eye-pairs, a method is proposed in this study to select an optimal eye-pair.

Algorithm 1. *Optimal eye-pair detection.*
Input: a set of dark regions D.
Output: eye-pair regions E_j and E_k, where j and k are the region indices of an eye-pair.
Steps:

1. Apply Chan's method [6] to D to conduct eye-pair generation to get a probable set of eye-pair regions E. Let E_i denote the ith eye-pair in E.
2. Compute the areas of the two regions, denoted by $E_{i,1}$ and $E_{i,2}$ of each eye-pair E_i in the following way, where $Area_{i,1} \geq Area_{i,2}$ with $Area_{i,1}$ and $Area_{i,2}$ being normalized:

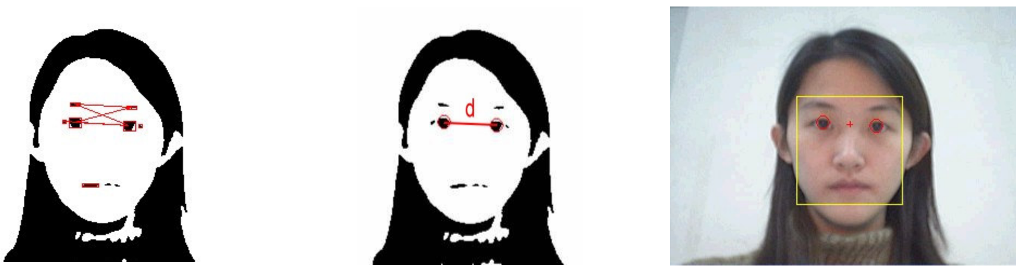
$$Area_{i,1} = E_{i,1}.width \times E_{i,1}.height;$$
$$Area_{i,2} = E_{i,2}.width \times E_{i,2}.height.$$

3. Compute a score of each eye-pair E_i by

$$Score_i = \frac{Area_{i,1} \times Area_{i,2} - S_i^{\ 2}}{2}. \quad \textbf{(1)}$$

   where $S_i = Area_{i,\,1} / Area_{i,\,2}$.
4. Find the largest $Scroe_i$, and set $E_j = E_{i,1}$ and $E_k = E_{i,2}$.

For each region of the eye-pair, a circle with the maximum number of points in it is detected to be an eyeball. After computing a Euclidean distance d between two eyeballs, a $2d \times 2d$ square region is constructed to cover the main facial features [6]. Some experimental results are shown in Figs. 7(a), (b), and (c).



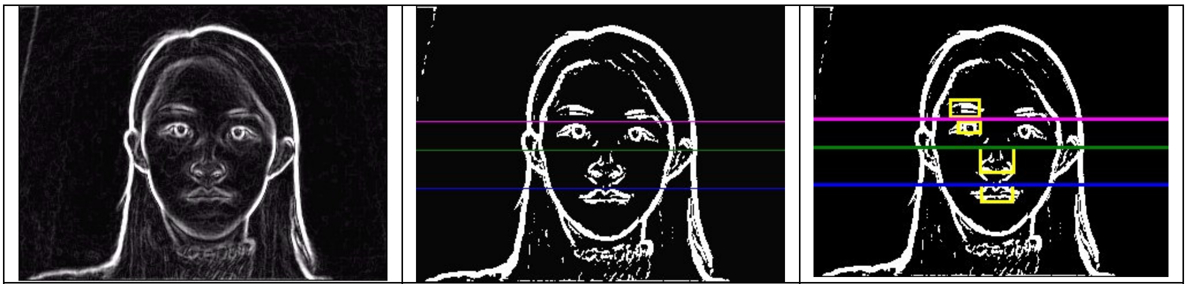
(a) (b) (c)

Fig. 7 Some results of facial feature region extraction. (a) Probable eye-pairs. (b) An optimal pair of eyeballs with a distance d between them. (c) A 2d×2d square region.

B. Knowledge-based Facial Feature Extraction

By predetermining three horizontal division lines in the $2d \times 2d$ square region mentioned above, the square regions can be separated into four parts. For each part, a local threshold value can be computed by Tsai's moment-preserving thresholding method [1] to get a binary edge image of the face image in intensity.

The main idea of the proposed knowledge-based facial feature extraction method is to speculate an initial search region of each facial feature to extract a final region in the binary edge image by a region growing method. The extracted $2d \times 2d$ square range is used here to speculate the initial position and range of each facial feature region. The details are omitted here. Some experimental results are shown in Figs. 8(a), (b), and (c).



(a)	(b)	(c)

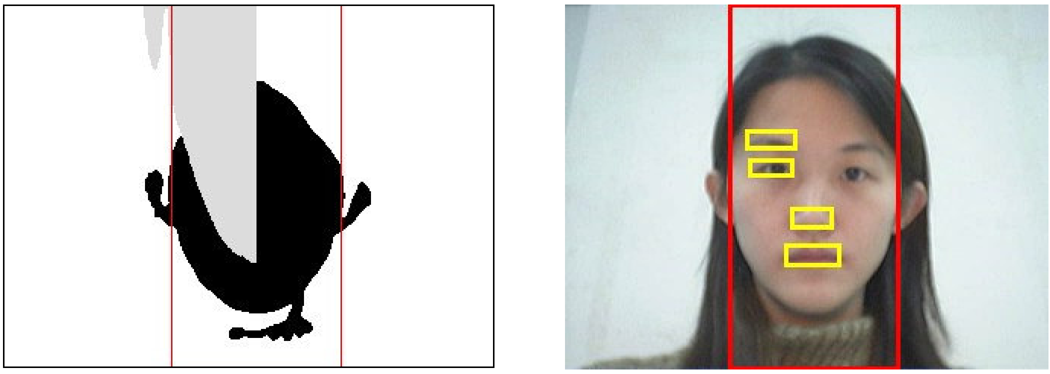
Fig. 8 Some results of facial feature region extraction. (a) Sobel edge image. (b) Binary edge image with three horizontal divisions shown on it. (c) Detected Facial Feature Regions.

4.3. Detection of Cheek Boundaries and Ear Regions

After extracting the above mentioned facial feature regions, some features are yet to be detected, like the ear regions. For this, the cheek boundaries need to be detected first. To this end, a vertical projection method is applied to the set of face regions. After mirror-mapping the right-side projection information to the left side, we find a position near the $2d \times 2d$ square range with a local minimum projection value as the desired cheek boundaries. After detecting the position of the cheek boundaries, we then divide the left and right parts of the set of face regions into ear regions and cheek regions. Some experimental results are shown in Figs. 9(e) and (f).

5. EXTRACTION OF FACIAL FEATURE POINTS

After setting up the face model with facial feature points, a method based on the above-mentioned edge image for extraction of facial feature points in a given facial image is needed for the purpose of creating a personal comic face. Since some facial features have symmetry properties, we will only detect the left-side feature points and compute the remaining right-side feature points.

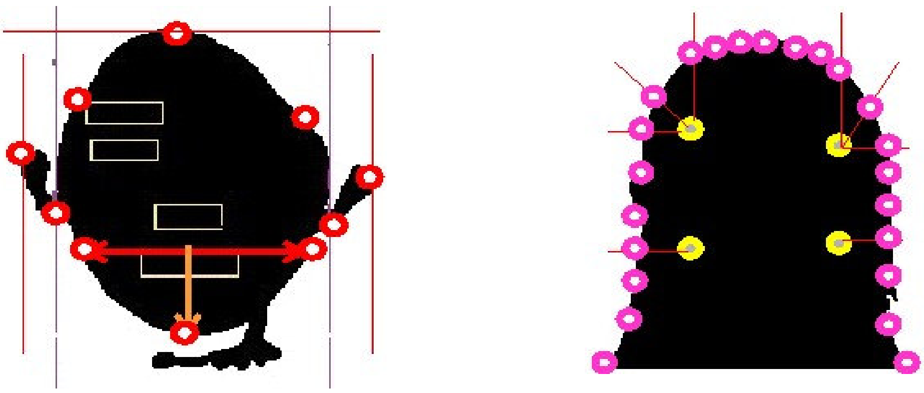


(a) (b)

Fig. 9 Some results of facial feature region extraction. (a) Projection information of face regions. (b) Final extraction results of facial feature regions and cheek boundaries.

The main idea to extract the feature points of the face, ear, and hair is to choose a start point in the feature regions and conduct search. When the start point moves to a position that satisfies a predefined rule, the search is stopped and the position is recorded as a feature point. Some illustrations are shown in Figs. 10(a) and (b).

In order to extract the feature points of the remaining facial features (including eyebrows, eyes, nose, and mouth), a local averaging method is applied in the facial feature regions to obtain the demanded feature points. Since the edge information of the nose region is not always adoptable for pixel-by-pixel detection here, the given size information of the nose region is used to obtain the desired feature points. Some illustrations are shown in Figs11. (a) through (f).



(a) (b)

Fig. 10 Extraction of facial feature points. (a) Detection of face points in face region. (b) Detection of hair points in hair region.

6. COMIC FACE GENERATION AND EXPERIMENTAL RESULTS

The automatically generated comic faces are rendered by the SVG language that is useful for describing two-dimensional mixed vector or raster graphics in XML. Most importantly, it supports the two previously-mentioned curve drawing methods used in the proposed system. The comic face generator will write the desired drawing script into an SVG file.

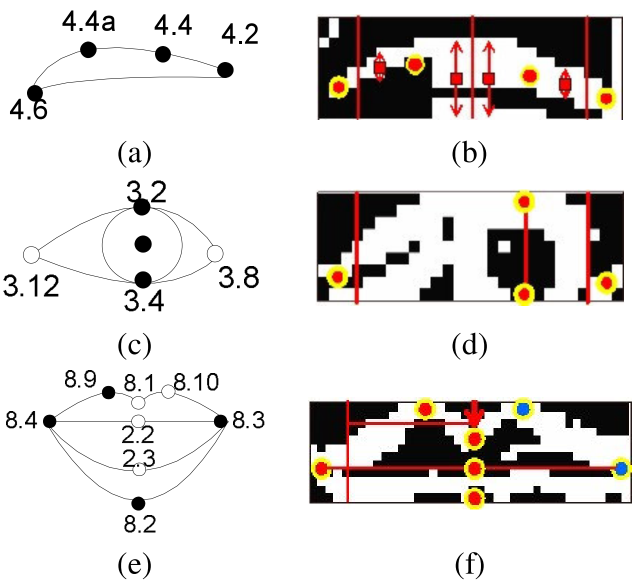


Fig. 11 Extraction of facial feature points. (a)(c)(e) Demanded feature points of the left eyebrow, left eye and mouth. (b)(e)(f) Detection of the corresponding feature points in corresponding feature regions.

In our system, twenty neutral face images of five people were used for learning and twenty for testing. The accuracy of automatic eye-pair detection is 90%. Errors occurred because of the variations of lighting and/or the shapes between eyes and eyebrows. In such cases, acceptable comic faces were created from the originally generated ones by relocating ten facial feature points in average. Some experimental results are shown in Fig. 12.

We can also modify the comic faces by specifying the shapes and styles of the SVG line drawing and color filling. It can be used for simulating kinds of emotions and different comic types. Some experimental results are shown in Figs. 13 and 14.

7. CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORKS

A system for automatic generation of comic faces has been implemented. Based on a face model of 72 facial feature points, the proposed facial feature extractor is used for automatically extracting facial feature points. Dragging theses feature points can directly deform the comic face. In the component of the comic face generator, an XML language SVG is used for rendering the personal comic face.

Finally, we mention some interesting topics for future research. In order to demonstrate low-cost processing, the facial feature detection method needs to be improved to fit more application environments. Besides, detections of glasses, wrinkles, and hair styles of given facial images should be considered. Moreover, a way to learn a caricaturist's comic style would make the comic face more attractive for more people.

REFERENCES

[1] W. H. Tsai, "Moment-preserving thresholding: a new approach," *Computer Vision, Graphics, and Image Processing*, 1985, vol. 29, pp. 377-393.

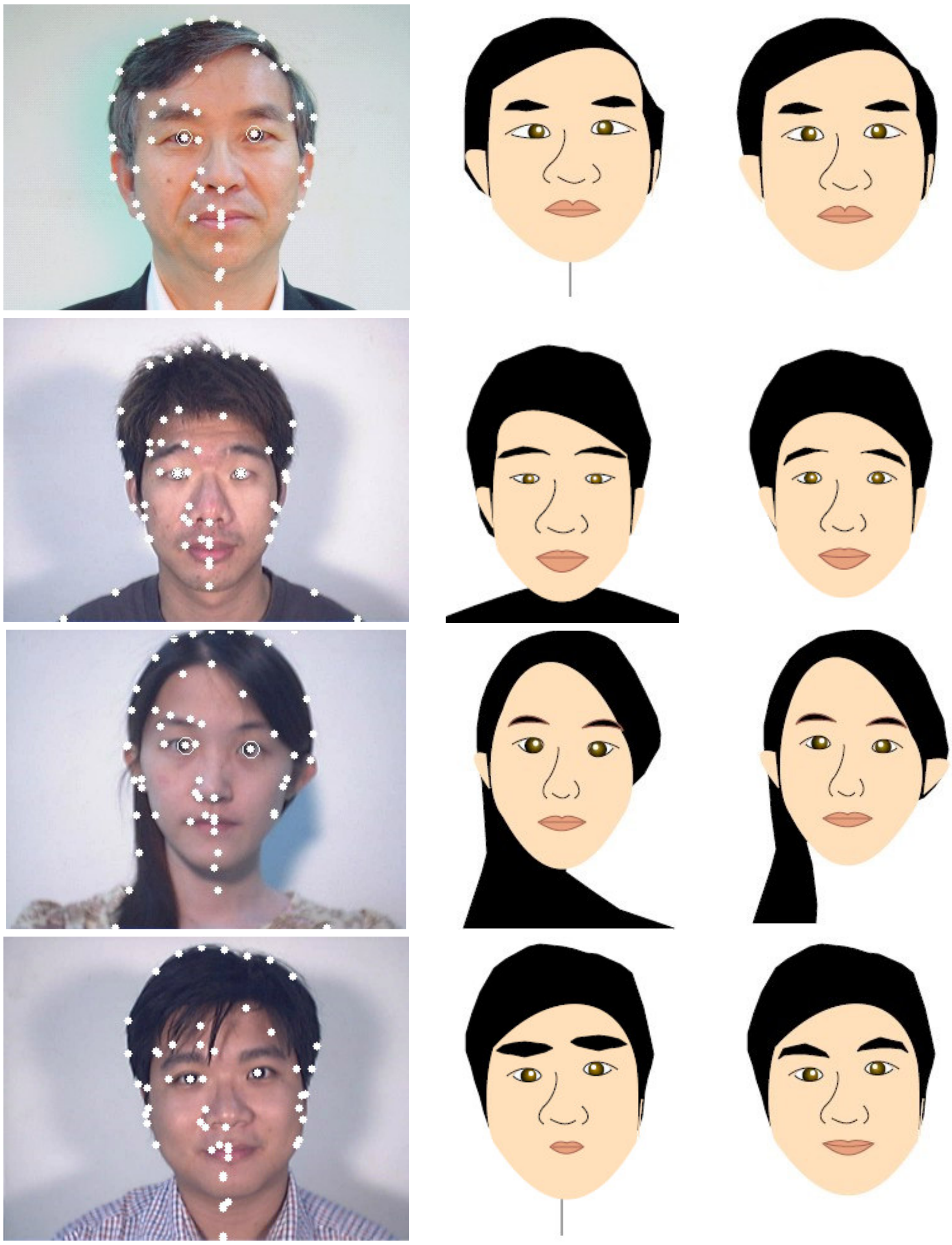
[2] Z. Ruttkay , H. Noot, "Animated CharToon faces," *Proceedings of 1st international symposium on Non-photorealistic animation and rendering*, Annecy, France, June 05-07, 2000, pp.91-100.

[3] G. J. Edwards, A. Lanitis, C. J. Taylor, T. F. Cootes, "Face recognition using statistical models," *Image Processing for Security Applications, IEE Colloquium on Image Processing for Security Applications*, UK, March 10, 1997, pp. 2/1-2/6.

[4] Rein-Lien Hsu, A. K. Jain, "Generating discriminating cartoon faces using interacting snakes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, vol. 25, pp. 1388 -1398.

[5] H. Chen, N. N. Zheng, L. Liang, Y. Li, Y. Q. Xu, H. Y. Shum, "PicToon: a personalized image-based cartoon system," *Proceedings of Tenth ACM international conference on Multimedia*, France, 2002, pp.171-178.

[6] S. C. Y. Chan , P. H. Lewis, "A Pre-filter Enabling Fast Frontal Face Detection," *Proceedings of Third International Conference on Visual Information and Information Systems*, June 02-04, 1999, pp.777-784.



(a) (b) (c)

Fig. 12 Some experimental results. (a) Results of detection of facial feature points. (b) Results of automatic generation of neutral comic faces rendered by SVG. (c) Results of a comic face with relocating some facial points.

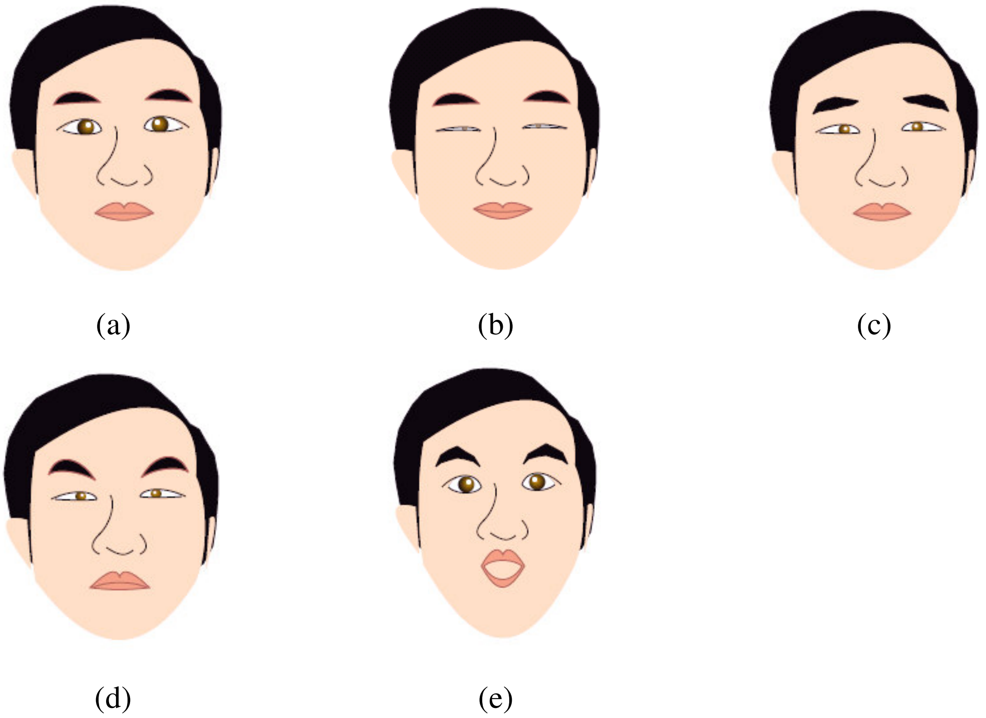


Fig. 13 Some experimental results of emotions rendered by SVG. (a) Neutral comic face. (b) Smile. (c) Sad. (d) Angry. (e) Surprise.

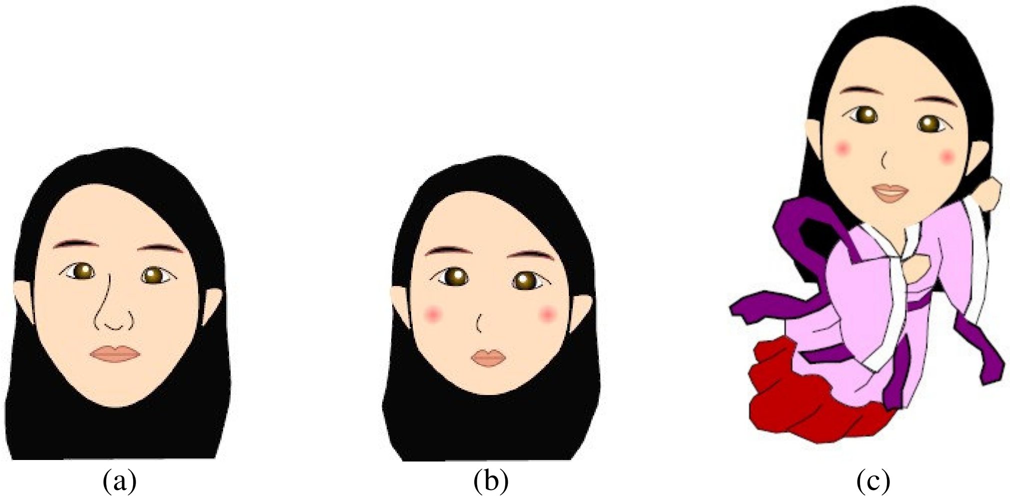


Fig. 14 Some experimental results of different types of comic faces rendered by SVG.. (a) Normal type. (b) Cute type. (c) With clothes.