AUTOMATIC 2D VIRTUAL FACE GENERATION BY 3D MODEL TRANSFORMATION TECHNIQUES

[1]*Yi-Fan Chang* (張依帆) *and* [1,2]*Wen-Hsiang Tsai* (蔡文祥)

[1]Institute of Multimedia Engineering,
National Chiao Tung University, Hsinchu, Taiwan, R. O. C.
[2]Dept. of Computer Science and Information Engineering,
Asia University, Wufeng, Taiwan, R. O. C.
E-mail: yfchang.cs94g@nctu.edu.tw, whtsai@cis.nctu.edu.tw

ABSTRACT

A system for automatic generation of talking cartoon faces is proposed, which includes four processes: cartoon face creation, speech analysis, facial expression and lip movement synthesis, and animation generation. A face model of 72 facial feature points is adopted. A method for construction of a 3D local coordinate system for the cartoon face is proposed, and a transformation between the global and the local coordinate systems is conducted. A new 3D rotation technique is applied to the cartoon face model with some additional points to draw the face in different poses. A concept of assigning control points is applied to animate the cartoon face with different facial expressions. A statistical method is proposed to simulate the timing information in simulating various facial expressions. For lip synchronization, a sentence utterance segmentation algorithm is proposed and a syllable alignment technique is applied. Twelve basic mouth shapes for Mandarin speaking are defined to synthesize lip movements. Finally, an editable and opened vector-based XML language - Scalable Vector Graphics (SVG) is used for rendering and synchronizing the cartoon face with speech. Experimental results showing the feasibility of the proposed methods are included.

Keywords: Virtual face animation, sentence utterance segmentation, facial expression simulation, lip-sync.

1. INTRODUCTION

The topic of virtual talking faces has been studied for many years. Roughly speaking, there are two main phases to generate virtual talking faces. One is the creation of head models, including 2D and 3D models. The other is the generation of virtual face animations with speech synchronization. For the first phase, the main issue is how to create a face model. Virtual face models can be created from single front-view facial images by extraction of facial features [1, 2]. Some methods were proposed to morph generic 3D face models into specific face structures based on multiple image views of a particular face [3-5].

In order to animate a virtual talking face, speech synchronization is an important issue to be concerned about. Virtual faces can be animated by an audio-visual mapping between input speeches and the corresponding lip configuration [1, 2]. In Li et al. [6], cartoon faces are animated not only from input speeches, but also based on emotions derived from speech signals.

For low-cost and efficient animation, we use existing 2D cartoon face models and focus on animation generation. We construct 3D face models based on the existing 2D models by applying a transformation between the two types of models. We define some new basic facial expressions and propose techniques to generate them. We synchronize lip movements with speeches after defining 12 basic mouth shapes and segmenting effectively sentence utterances. Also, we apply a statistical method to simulate the probabilistic head movements and facial expressions to animate personal cartoon faces more realistically. Finally, an editable and opened vector-based XML language - Scalable Vector Graphics (SVG) is used for rendering and synchronizing the cartoon face with speech. Experimental results showing the feasibility of the proposed methods are included.

The remainder of the paper is organized as follows. In Section 2, the proposed method for construction of 3D cartoon face models based on 2D cartoon face models and the proposed method for creation of virtual cartoon faces are described. In Section 3, the proposed method for speech segmentation for lip synchronization is presented. In Section 4, the proposed method for simulating facial expressions and head movements is described. And then, some animation issues such as lip movements and smoothing of talking cartoon facial animation are discussed and solved in Section 5. An integration of the proposed techniques using an open standard language SVG (Scalable Vector Graphics) to

generate web-based animations is described in Section 6. Finally, conclusions are included in Section 7.

2. CARTOON FACE GENERATION AND MODELING FROM SINGLE IMAGES

In the proposed system, four major parts are included: a cartoon face creator, a speech analyzer, an animation editor, and an animation and webpage generator, as shown in Fig. 1.

Since a 2D face model is not enough to synthesize proper head poses of cartoon faces, the cartoon face creator is designed to create personal cartoon faces, integrating the technique of 3D face model construction.

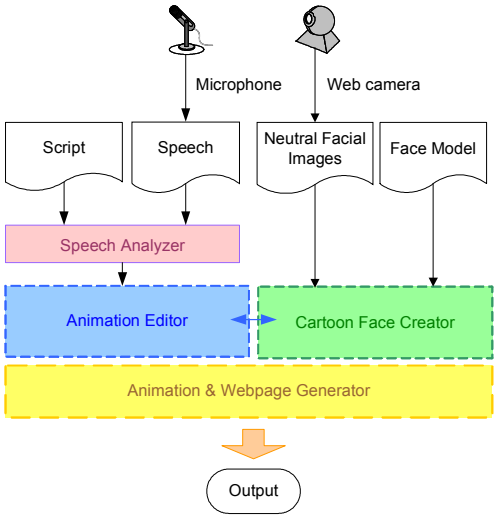


Fig. 1: Proposed system organization.

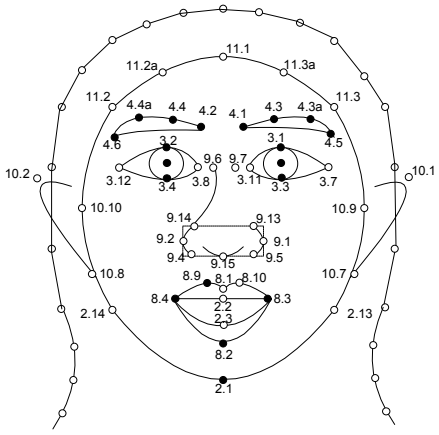
In the cartoon face creation process, three main steps are included. The first step is to assign facial feature points to a 2D face model. It can be done in two ways. One is to detect facial features of an input neutral facial image, generate the corresponding feature points, and map them to the feature points in the predefined face model [1]. The other way is to directly assign the feature points according to the input 2D face data. In this study, we adopt both ways in constructing our face model. The face model used in [1] is adopted in this study and shown in Fig. 2.

The second step is to construct a local coordinate system of the face model for applying 3D rotation techniques. By creating a transformation between the global and the local coordinate systems and assigning the positions of the feature points in the third dimension, namely, the Cartesian z-coordinate, this step can be done, and then essential head movements can be simulated. The last step is to define basic facial expression parameters for use in face animation.

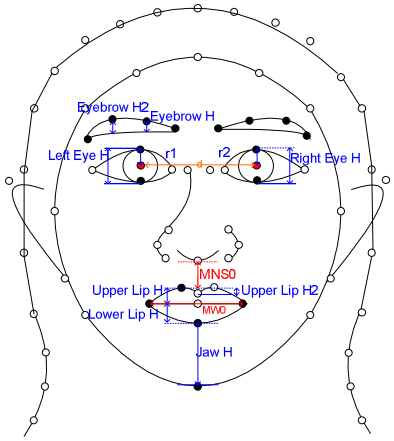
2.1. Construction of 3D Face Model Based on 2D Cartoon Face Model

Based on the face model in Fig. 2, the method proposed to construct the 3D face model can be divided into two steps: the first is to construct a local coordinate system from the global one, and the second is to assign the position of the feature points in the Cartesian z-direction.

The basic idea for constructing a local coordinate system is to define a rotation origin and transform the points of the global coordinate system into those of the local one with respect to the rotation origin. Let the position of the center between two eyeballs be denoted as *EyeMid*. And let the distance between two eyeballs be denoted as d. Then the rotation origin $O(x_o, y_o)$ can be speculated as the center of the neck by $x_o = EyeMid.x$ and $y_o = EyeMid.y + d \times 1.3$. Then each of the 72 model points $P(x_p, y_p)$ can be transformed into the local coordinate system by $x_p = x_p - x_o$ and $y_p = y_o - y_p$.



(a)

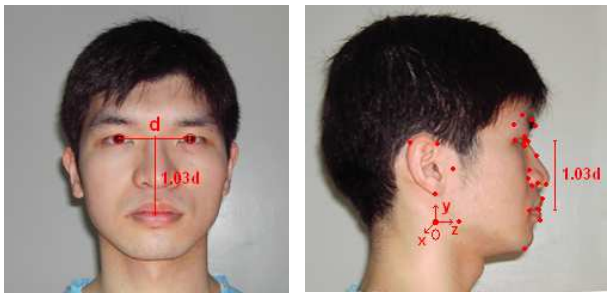


(b)

Fig. 2: Adopted 2D cartoon face model in this study. (a) 72 feature points. (b) Facial animation parameter units.

The basic idea for assigning the positions of the feature points in the Cartesian z-direction is to do the assignment based on a proposed generic model.

Although a generic model cannot represent all cases of human faces, it is practical enough in the application of generating virtual talking faces, because in real cases, one usually does not roll his/her head violently when giving a speech, and a little inaccuracy of the depth information in a face model would not affect the result much. To generate the generic model, two orthogonal photographs are used, as shown in Fig. 3. Let the distance between the y-position of *EyeMid* and the y-position of the feature point 2.2 in the front-view image be denoted as *d'*. The value *d'* can be expressed as a constant multiple of *d*. Similarly in the side-view image, the distance between *EyeMid* and the point 2.2 in the y-direction may be set to be a constant multiple of *d*. By marking all of the viewable points, including the rotation origin, and computing the distance in the z-direction between the origin and each of the points in the image, the positions of the viewable points in the z-direction can be computed as a constant multiple of *d*, too. Then we can assign the position of the feature points in the Cartesian z-direction according to their corresponding constant multiples of *d*. After the 3D face model is constructed, we can then easily generate different head poses of the face model by a 3D rotation technique.



(a) (b)

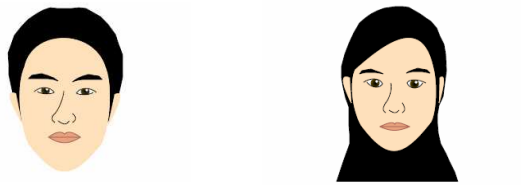
Fig. 3: Two orthogonal photographs. (a) Front view. (b) Side view.

2.2. Creation of Cartoon Faces

Two types of cartoon faces, *frontal* and *oblique* ones, are created by the corner-cutting subdivision and cubic Bezier curve approximation methods in this study. A frontal cartoon face is drawn by the 72 feature points of the face model. Two experimental results are shown in Fig. 4.

By changing some of the values of the *facial animation parameter units* (*FAPUs*) in the face model and setting up the positions of the corresponding model points, some basic facial expressions can be generated. Experimental results of generation of a smiling effect, an eyebrow raising effect, and an eye blinking effect are shown in Figs. 5, 6, and 7, respectively.

For oblique cartoon faces, we apply a 3D rotation technique to generate different head poses of the face model. We also simulate the eyeballs gazing at a fixed target while the head is turning. The basic idea for the simulation is to set up a point representative of the focus of the eyes in the local coordinate system of the face model. By speculating the radius of the eyeball, the position of the eyeball center can be computed by the position of the pupil and the focus. Then for every rotation performed in the creation process, the new position of the eyeball center can also be calculated. And the new position of the pupil can be computed by the position of the eyeball center and the focus. An illustration of the focus and eyeballs is shown in Fig. 8.



(a) (b)

Fig. 4: Experimental results of the creation of a frontal cartoon face. (a) A male face. (b) A female face.

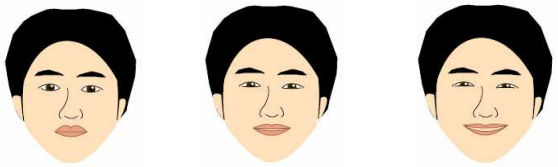


Fig. 5: An experimental result of generation of smiling effect.

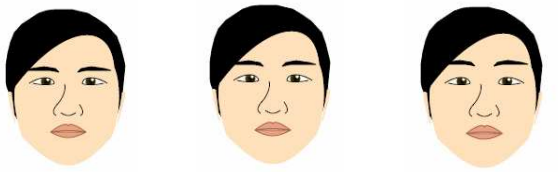


Fig. 6: An experimental result of generation of eyebrow raising effect.

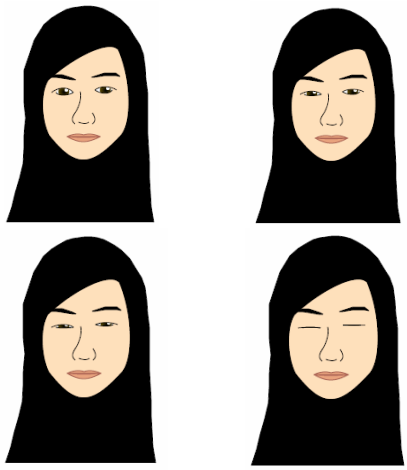


Fig. 7: An experimental result of generation of an eye blinking effect.

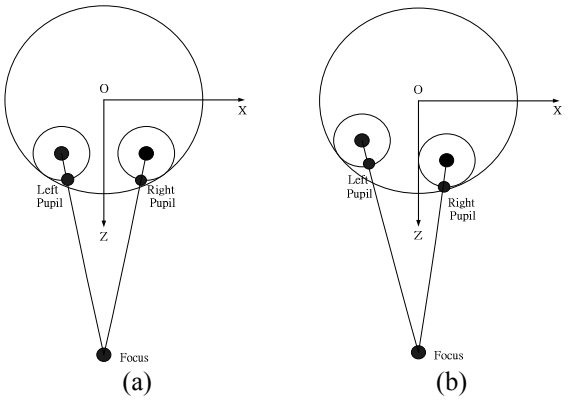


Fig. 8: An illustration of the focus and eyeballs. (a) Before rotation. (b) After rotation.

An oblique cartoon face is drawn by the 72 feature points and some additional points. The creation process is similar to that for generating the frontal cartoon face, but must be done with some additional steps, including the rotation step. Furthermore, there are some problems after applying the rotation technique. One of the problems is that the face contour will become deformed, because some of the face contour points will be hidden and not viewable after the head is turned. And the resulting points cannot represent the face contour any more. Therefore, we use some other points instead of them. Another problem is that the depth of the hair contour points is defined in a flat plane, which would look unreal after the rotation, as shown in Fig. 9. We propose a method to solve this problem, which is to change the depth of some of these points before the rotation according to the position of these points and the rotation angle. An illustration of the shift of hair contour points is shown in Fig. 10, and an illustration of oblique cartoon face creation is shown in Fig. 11.

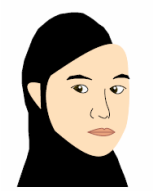


Fig. 9: An illustration of the unreality of the hair contour.

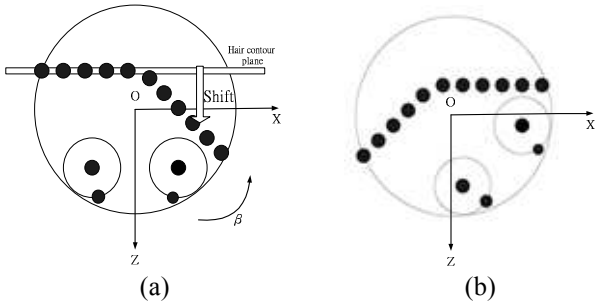


Fig. 10: An illustration of the shift of hair contour points. (a) Before rotation. (b) After rotation.

An example of experimental results of creating cartoon faces in different poses and with different facial expressions is shown in Fig. 12.

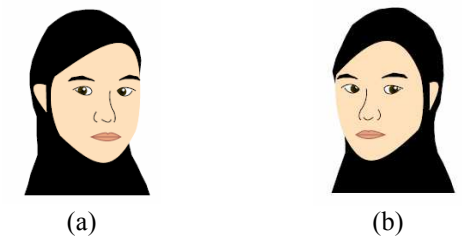


Fig. 11: An illustration of creation of oblique cartoon faces.

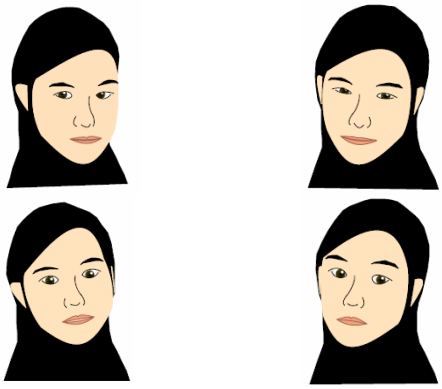


Fig. 12: Experimental results for creation of cartoon faces in different poses with different facial expressions.

3. SPEECH SEGMENTATION FOR LIP SYNCHRONIZATION

In the proposed system, the speech analyzer receives a speech file as well as a script file, called a *transcript* in the study, and applies speech recognition techniques to get the timing information of each syllable. A transcript is usually composed of many sentences. Although it is feasible to directly get the timing information of each syllable from the speech of the entire transcript without segmentation of the sentence utterances, it will take too much time to do so if the input audio is long. Therefore, by segmenting the entire audio into sentence utterances as the first step and then processing each segmented shorter sentence utterance piece sequentially to extract the duration of each syllable in the speech, the overall processing speed can be accelerated.

We propose a method to segment the speech into sentence utterances by silence features, based on Lai and Tsai's method [8]. The basic idea is to detect the silent parts between sentences and perform segmentation of sentence utterances according to these detected silent parts. To achieve this goal, the silence features, including the maximum volume of the environment noise and the duration of silent parts, must be learned

before the segmentation. In the proposed system, an interface is designed to let users select the pause between the first sentence and the second one from the input audio. Then the silence features can be learned according to this selected part of audio. The process of sentence utterance segmentation is given as an algorithm in the following.

Algorithm 1. *Segmentation of sentence utterances.*
Input: A speech file S_t of the entire transcript.
Output: Several audio parts of sentence utterances S_1, S_2, ….
Steps:

1. Select the start time t_s and the end time t_e of the first intermediate silence in S_t manually.
2. Find the maximum volume V appearing in the audio part within the selected first intermediate silence.
3. Set the minimum duration of the intermediate silence D_{pause} as $(t_e - t_s) \times c_1$, where c_1 is a pre-selected constant between 0.5 and 1.
4. Set the maximum volume of environment noise V_{noise} as $V \times c_2$, where c_2 is a pre-selected constant between 1 and 1.5.
5. Start from t_s to find all continuous audio parts $S_{silence}$ whose volume are smaller than V_{noise} and last longer than D_{pause}.
6. Find a continuous audio part $S_{sentence}$, called a *speaking part*, which is not occupied by any $S_{silence}$.
7. Repeat Step 6 until all speaking parts are extracted.
8. Break S_t into audio parts S_1, S_2, … of the speaking parts found in Step 7.

Since we assume that the speech is spoken at a steady speed, the durations of the other silent parts are considered to be close to the first one. Therefore, c_1 in Step 3 is chosen to be 0.95. Furthermore, since we assume that the speech is spoken in a loud voice and the recording environment of the input audio is noiseless, the volume of speaking parts is considered to be much larger than that of environment noise. To avoid misses of detecting silent parts, c_2 in Step 4 is chosen to be a larger value 1.45. To evaluate the algorithm, we use the *correct rate*, which is computed by the number of correctly segmented parts divided by the total number of segmented parts. After doing some experiments of segmentation for about ten audio files, the correct rate of this algorithm is computed to be 96.26%. It shows the feasibility of the proposed method. An example of selecting the first silent part in an input audio is shown in Fig. 13. The red part represents the selected silence period between the first sentence and the second one. An example of the experimental results of the proposed segmentation algorithm is shown in Fig. 14. The blue and green parts represent odd and even sentences, respectively. A process of syllable alignment is then applied to each sentence utterance to extract the timing information of the syllables in the speech. Finally, the timing information for each sentence utterance is combined into a global timeline. An example of the experimental results is shown in Fig. 15. The durations of the syllables are shown in blue and green colors.

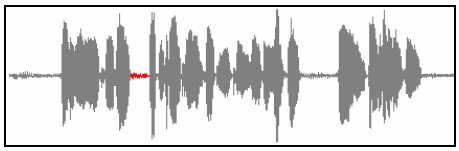


Fig. 13: An example of selecting the first silent part in an input audio.

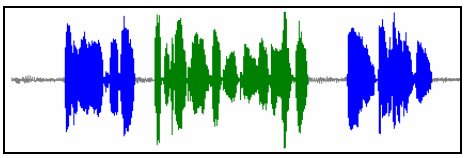


Fig. 14: An example of sentence utterances segmentation results. The blue and green parts represent odd and even speaking parts, respectively.

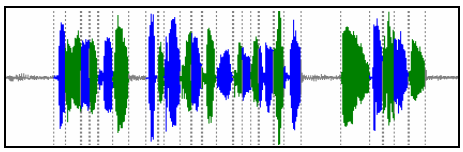


Fig. 15: Result of syllable alignment of the audio in Fig. 13.

4. ANIMATION OF FACIAL EXPRESSIONS

In Section 3, we have presented how to get the timing information of the syllables in a speech. However, the timing information of the syllables is helpful just for the control of mouth movements. In order to control movements of the other facial parts, including eye blinks, eyebrow raises, and head movements, the timing information of the facial parts must be simulated. The cartoon face can so be animated more realistically. In the paper, a statistical method is proposed to model the probabilistic functions of these facial behaviors.

For this purpose, we measured the timing information of eyebrow movements and head movements by analyzing the facial expressions of several news announcers on some TV news programs. For each behavior which we observed, we use two time stamps t_s and t_e to represent its start time and end time. Then we define two parameters to describe the timing information of the behavior. One is the *time interval* between two consecutive behaviors, which denotes the time interval between the end time of the first behavior and the start time of the second one. The other parameter is the *duration* of the behavior, which denotes the time interval between the start time and the end time of a behavior. For eyebrow movements, because an

eyebrow raising is usually followed by a corresponding eyebrow lowering, we define t_s as the start time of the eyebrow raising and t_e as the end time of the eyebrow lowering. For head movements, we define t_s as the time when the head starts moving and t_e as the time when the head stops moving.

Lin and Tsai [7] proposed a simulation method of eye blinks by the Gamma distribution, and a simulation method of eyebrow movements by the uniform distribution. We adopt their simulation method in this study. An illustration of the probability function of eye blinks in [7] is shown in Fig. 16. The one in blue color is the probability function of the Gamma distribution with parameters $\alpha = 2$ and $\theta = 1.48$. The other one in pink color is the probability function of eye blinks approximated from the analysis data of the TV News announcers.

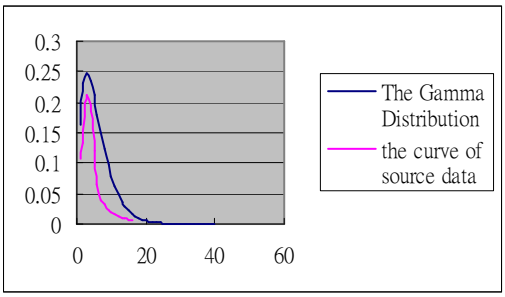


Fig. 16: The probability function of eye blinks in [7].

According to the analyzed data, we compute the mean value and the standard deviation value of the time intervals and the durations. To avoid some special cases ('outliers') which are extremely large or small and may affect the simulation result, using the mean value and the standard deviation value is better than directly choosing a number. For eyebrow movements, the mean value of the time interval is 6.72, and the standard deviation value is 5.82, so we randomly generate a variable on [6.72 − 5.82, 6.72 + 5.82] to represent the time interval between two consecutive eyebrow movements. In a similar way, the mean value of the duration is 0.75, and the standard deviation value is 0.24, so we randomly generate a variable on [0.51, 0.99] to represent the duration of eyebrow movements.

For head movements, since the time intervals and the durations of head movements are correlated to the content of speech and vary with the habit of each distinct TV news announcer, we consider that they are distributed randomly and can be simulated by the uniform distribution. Similar to the simulation of eyebrow movements, the adopted time intervals and durations of head tilting and horizontal head turning are listed in Table 1.

For vertical head turning, we have found that the time interval of vertical head turning is partially related to the times of pauses in the speech. A TV news announcer usually nods when his/her speech is coming to a pause, and breaths during the pause time with his/her head raised. Due to this finding, we collect some statistics of the time interval between the end time of the head nod (i.e. the head turning in a vertical down direction) and the start time of the pause. We also collect the duration of the nod and the duration of the head raising after the nod. The adopted simulation of intervals are listed in Table 2. By finding all silent parts $S_{silence}$ in a speech based on the method described in Algorithm 1, and using the symbol $t_{silence}$ to represent the start time of the pause for each $S_{silence}$, the occurring time of the nod can be simulated as $[t_{silence} - t_1 - d_1, t_{silence} - t_1]$, where d_1 represents the duration of the nod, and the occurring time of the head raising after the nod can be simulated as $[t_{silence}, t_{silence} + d_2]$, where d_2. represents the duration of the head raising after the nod.

Table 1: Adopted intervals for simulation of head tilting and horizontal head turning.

	Result of the interval
Time interval of head tilting (second)	[0, 6.64]
Duration of head tilting (second)	[0.4, 1.0]
Time interval of horizontal head turning (second)	[0.08, 7.92]
Duration of horizontal head turning (second)	[0.4, 0.92]

Table 2: Adopted intervals for simulation of vertical head turning.

	Result of the interval
Time interval between the nod and the pause (second)	[0.07, 0.35]
Duration of the nod (second)	[0.34, 0.94]
Duration of the head raising after the nod (second)	[0.34, 0.70]

5. TALKING CARTOON FACE GENERATION

According to the timing information of syllables mentioned in Section 3, the time about when to make the cartoon face speak is identified. However, the information about how to make a speaking motion for the cartoon face is still unknown. In this section, we propose an automatic method for talking cartoon face generation to solve this problem. In our method, twelve basic mouth shapes are defined, and syllables are translated into combinations of the basic shapes. By assigning basic mouth shapes into proper key frames in an animation and applying an interpolation technique to generate the other frames among the key frames, the animation of a talking cartoon face can be accomplished.

5.1. Definition of Basic Mouth Shapes

Chen and Tsai [1] proposed a method to reduce 21 kinds of Mandarin initials to 3 basic initial mouth shapes according to the manners of articulation. And based on the Taiwan Tongyoung Romanization, which contains a transcription of the Mandarin phonetics into a set of English alphabets, mouth shapes for Mandarin finals were also reduced to a set of combinations with 7 basic mouth shapes. According to their reduction result, any given syllable which consists of phonemes can be translated into a combination of basic mouth shape symbols. Considering that the classification of the Mandarin initials and finals in [1] was too simple to represent the differences between mouth shapes, we define instead twelve basic mouth shapes. The new classifications for the Mandarin initials and finals are listed in Tables 3 and 4, respectively.

Table 3: Six basic mouth shapes of Mandarin initials.

Mouth Shape Symbols	Members of the classes
m	ㄅ、ㄆ、ㄇ
f	ㄈ
h'	ㄉ、ㄊ、ㄋ、ㄌ、ㄐ、ㄑ、ㄒ
h	ㄍ、ㄎ、ㄏ
r	ㄓ、ㄔ、ㄕ、ㄖ
z	ㄗ、ㄘ、ㄙ

Table 4: A set of combinations with 7 basic mouth shapes of Mandarin finals.

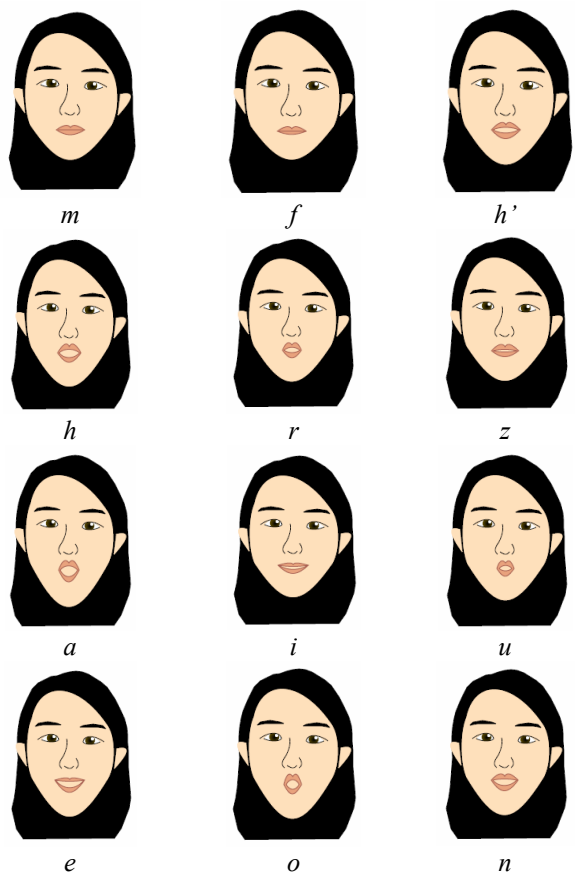
Mouth Shape Symbols	Members of classes	Combinations of Mouth Shapes	Members of classes
a	ㄚ	*au*	ㄠ
e	ㄝ	*ou*	ㄡ
i	ㄧ	*an*	ㄢ
o	ㄛ	*ah*	ㄤ
u	ㄨ、ㄩ	*ei*	ㄟ
n	ㄣ	*ai*	ㄞ
h	ㄥ、ㄦ、ㄜ		

Similarly to the method in [1], we eliminate *h'* and *h* if there is a symbol (including *i*, *u*, *e*, and *o*) behind them. For example, the syllable "ㄐㄧ" is translated into "*h'i*," where the "*i*" is right behind the "*h'*," and the final transcription of "ㄐㄧ" will be adjusted to be "*i*" automatically. However, *h'* and *h* are not eliminated if the symbol "*a*" is behind them because the pre-posture is not missed in this case. Also, due to the habit of the pronunciation for Mandarin speaking, the mouth shape of "ㄢ" will be changed from "*an*" to "*en*" if there is a Mandarin final (including "ㄧ" and "ㄩ") before it, and the mouth shape of "ㄛ" will be changed from "*o*" to "*uo*" if there is a symbol *m* before it.

5.2. Basic Mouth Shapes

An experiment was carried out to define the basic mouth shapes using some mouth points in the cartoon face model as control points, as shown in Table 5.

Table 5: Twelve basic mouth shapes for Mandarin speaking.



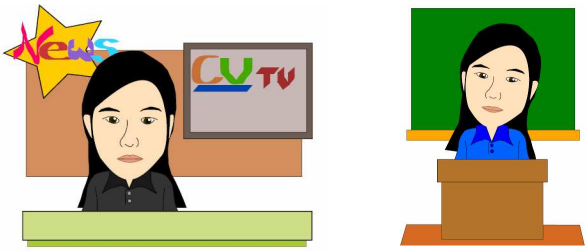
5.3. Talking Cartoon Faces Generation by Synthesizing Moving Lips

A timeline is used as a basic structure for animation. By obtaining the time information of syllables, which is done in the speech analyzer of the proposed system, arranging the timing of facial expressions, assigning basic mouth shapes for each syllable, setting up the positions of the control points in corresponding key frames in the timeline, applying an interpolation technique to generate the remaining frames among key frames, and synchronizing the frames with a speech file, the animation of the talking cartoon face can be created.

6. TALKING CARTOON FACE GENERATOR USING SCALABLE VECTOR GRAPHICS

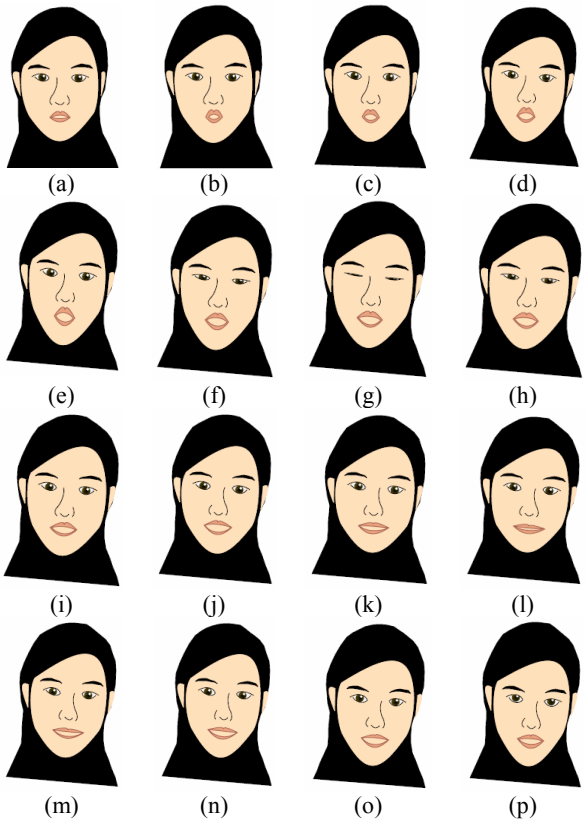
Scalable Vector Graphics (SVG) is an XML markup language which is an open standard created by the World Wide Web Consortium (W3C). It is used for rendering the talking cartoon face in the proposed system. Some experimental results are shown in Fig. 17.

Combing the techniques proposed in the previous sections, an example of experimental results of talking cartoon faces synchronized with a speech file of saying two Mandarin words "Wanyan" (Chinese) with facial expressions and head movements rendered by SVG is shown in Fig. 18. Some cartoon films of the other experimental results are offered at http://www.cc.nctu.edu.tw/~u9457518/index.html.



(a) (b)

Fig. 17: Results of cartoon faces with clothes and backgrounds rendered by SVG. (a) A virtual announcer. (b) A virtual teacher.



(a) (b) (c) (d)
(e) (f) (g) (h)
(i) (j) (k) (l)
(m) (n) (o) (p)

Fig. 18: An experimental result of the talking cartoon face speaking "Wanyan" (Chinese).

7. CONCLUSIONS

In this study, a system for automatic 2D virtual face generation by 3D model transformation techniques has been implemented. We have presented a way to automatically transform a 2D cartoon face model into a 3D one, and animate it by statistical approximation and lip movement synthesis with synchronized speech. Experimental results shown in the previous sections have proven the feasibility and applicability of the proposed methods.

Finally, we mention some interesting topics for future research. To fit more application environments for creation of face models from neutral facial images, the performance of the facial feature detection must be improved. Besides, for precise construction of 3D face models, assigning the position of the feature points in the z-direction may be combined with facial feature detection for side-view photographs. Moreover, since different TV news announcers have different habits when reporting news, more types of statistical models can be used to present different reporting styles for different TV news announcers.

REFERENCES

[1] Y. L. Chen and W. H. Tsai, "Automatic Generation of Talking Cartoon Faces from Image Sequences," *Proc. of 2004 Conf. on Computer Vision, Graphics & Image Processing*, Hualien, Taiwan, Aug. 2004.

[2] H. Chen, N. N. Zheng, L. Liang, Y. Li, Y. Q. Xu, and H. Y. Shum, "PicToon: A Personalized Image-based Cartoon System," *Proc. of 10th ACM International conf. on Multimedia*, Juan-les-Pins, France, pp. 171-178, 2002.

[3] C. Zhang and F. S. Cohan, "3-D Face Structure Extraction and Recognition From Images Using 3-D Morphing and Distance Mapping," *IEEE Trans. Image Processing*, Vol. 11, No. 11, pp. 1249-1259, Nov. 2002.

[4] T. Goto, S. Kshirsagar, and N. Magnenat-Thalmann, "Automatic Face Cloning and Animation Using Real-Time Facial Feature Tracking and Speech Acquisition," *IEEE Signal Processing Magazine*, Vol. 18, No. 3, pp. 17-25, May 2001.

[5] M. Zhang, L. Ma, X. Zeng, and Y. Wang, "Imaged-Based 3D Face Modeling," *Proc. of International Conf. on Computer Graphics, Imaging and Visualization*, pp. 165-168, July 26-29 2004.

[6] Y. Li, F. Yu, Y. Q. Xu, E. Chang, and H. Y. Shum, "Speech Driven Cartoon Animation with Emotions," *Proc. of 9th ACM International conf. on Multimedia*, Ottawa, Canada, pp. 365-371, 2001.

[7] Y. C. Lin, "A Study on Virtual Talking Head Animation by 2D Image Analysis and Voice Synchronization Techniques," *Master's Thesis*, Dept. of Computer and Information Science, National Chiao Tung Univ., Hsinchu, Taiwan, June 2002.

[8] C. J. Lai and W. H. Tsai, "A Study on Automatic Construction of Virtual Talking Faces and Applications," *Proc. of 2004 Conf. on Computer Vision, Graphics and Image Processing*, Hualien, Taiwan, Aug. 2004.