Feature-based Multi-style Cartoon System

Jianfeng Che, Jianhua Tao, Xingang Wang, Kaïhui Mu, Hongtao Li

1 Institute of Automation, Chinese Academy of Sciences
2 No.2 Power Plant, China Petroleum Jilin Petrochemical Company
jianfeng.che@ia.ac.cn

Abstract

In this paper, we present a feature-based cartoon system, which can generate a realistic cartoon face from an input picture. The cartoon face can be multi-style by using different cartoon templates. The whole process requires little user interaction. The system consists of four components, ASM-based facial feature locating, cartoon texture mapping, ears adding and hair adding. The advantages of method we propose are easy to accomplish, low time complexity and the good ability of retaining the real facial feature.

1. Introduction

Recently, with the development of computer technology, digital media is involved in people’s daily lives. One person can take a photograph of his friend or himself with mobile phone or digital camera as a souvenir. However, with the prosperity of cartoon book and cartoon movie around the world, the real photos can’t meet people’s demands. Pursuing the cartoons of real facial photos is becoming more and more popular. Then, how to turn real facial photos to cartoon ones is receiving common attentions.

In the facial cartoon creation field, lots of work has been done. In [1], the authors present a system that can generate exaggerated portrait. In [2], the famous face edit system, PICASSO, is put forward. It draws facial portrait according to face contour. In [3] [4], the authors use example-based learning method to produce line drawings. In [5], the presented system can generate specific cartoon style, which uses a cartoon photo and the positions of hand-marked facial feature points. In [6], the authors train real faces and cartoon ones in order to get the relationship between them. In [7], the authors describe the input face with several exaggerated prototypes. The final exaggerated cartoon face can be produced by a liner combination of above prototypes. In [8], line pair deformation is employed to make source image distorted by specific hand-marked lines. In [9], the authors present a feature-based cartoon creation system. The system counts facial features of one hundred men and one hundred women in order to get the normal range of facial feature. A new input face image will be judged to test whether need to be exaggerated. If the exaggeration is needed, one artist’s work will be modified to get an exaggerated effect.

Overall, most of the presented systems generate real facial portrait, which has simple lines and no cartoon texture. Besides, some systems need lots of training samples, which only specialized art designers can make. A large number of previous work should be prepared. In [9], the method just does shape exaggeration processing according to the new positions of feature points, so it’s not a true meaning method for turning a real face to a cartoon one. In this paper, we propose a novel cartoon face creation method. Facial feature points locating, texture mapping and line drawing are combined in the method. The system can produce a realistic-looking cartoon face that owns cartoon texture only by some cartoon templates.

The rest of the paper is organized as follows. Firstly, we present the whole framework of system and all components. Then, we introduce the key technologies of every component, followed by a set of output demos. Finally, we sum up the features and benefits of our system, and outline some future work.

2. System overview

Our system is developed to turn user’s real frontal face picture to cartoon one. It generates multi-style cartoon face according to different cartoon template with little user interaction. The cartoon face reflects faithfully the input image and doesn’t have any exaggeration. There are four major components, as shown in Figure 1:

(1) Facial feature locating. It locates the input frontal fifty-six facial feature points around cheek, eyebrow, eye, nose and mouth. The input real facial feature is reflected by these points.

(2) Cartoon texture mapping. It is the key component of the whole system. It generates cartoon face with texture. User can add different cartoon texture to the cartoon face by choosing different cartoon template.
(3) Ears adding. It adds ears to the cartoon that generated by the two front components.

(4) Hair adding. It adds hair to the cartoon that generated by the three front components. User can add different hair to cartoon face by choosing different hair style.

3. Facial feature locating

Facial feature locating is a frontal facial feature points locator that uses ASM (Active Shape Model) [10]. ASM is a facial points locating algorithm which is commonly used. It firstly sets up two models according to a set of hand-marked images: local texture model ($M_1$) and global shape model($M_2$), then searches in the input image to find local optimal positions of feature points according to $M_1$, meanwhile adjusts the positions constrained by $M_2$. Repeats until the best coordinates of feature points are obtained.

Input a frontal face image, the locating result is as shown in figure 2.

4. Cartoon texture mapping

Cartoon texture mapping innovatively makes use of triangle mesh to do cartoon texture mapping. It aims to map cartoon template’s texture to feature shape that is obtained by facial feature locator. In this way, the module can output one cartoon face which owns cartoon texture and retains real facial feature. The concrete principle is as follows:

In 2-D image, the real facial feature points are triangulated using Delaunay triangulation technique. A set of triangles $T = \{t_1, t_2, \ldots, t_m\}$ is obtained. Do the same thing to cartoon template’s feature points that are hand-marked. Cartoon template’s feature points are in the same position as the real ones. A set of triangles $T' = \{t'_1, t'_2, \ldots, t'_m\}$ are also obtained. Take one cartoon template [11] for example, the hand-marked feature points and the triangles are shown in figure 3 and figure 4.

Compared figure 3 with 2, it’s easy to find that the feature points have the same amount and positions, so $T$ and $T'$ have equal amount of triangles, supposed to be $m$. However, their shapes are different. $t_k$ is correlated with $t'_k (k=1,2, \ldots, m)$ and they are
composed of the points in the same positions of each image. Assuming that \( p \) is a point inside \( t_k \), its position can be presented by the triangle vertexes it belongs to.

\[
p = \alpha \cdot p_1 + \beta \cdot p_2 + \gamma \cdot p_3
\]  
(1)

Where \( p_1, p_2, p_3 \) are vertexes, \( \alpha, \beta, \gamma \) are weighted coefficients which satisfy \( \alpha + \beta + \gamma = 1 \), \( 0 \leq \alpha, \beta, \gamma \leq 1 \).

According to this formula, there’s one point \( P' \) inside \( t_k \), which can be computed by

\[
P' = \alpha \cdot P_1 + \beta \cdot P_2 + \gamma \cdot P_3
\]  
(2)

Where \( P_1, P_2, P_3 \) are vertexes of \( t_k \). The whole transformation is as shown in figure 5.

![Figure 5. Triangular mapping](image)

For every triangle \( t_n \) \((n=1, 2, \cdots, m)\) in \( T \), we do the same mapping to its all inside points and replace the points’ pixel value by the corresponding ones in \( T' \). The transformation can eliminate interferences caused by ration, stretching and different shapes.

According to above transformation, the module maps cartoon template’s texture to real facial feature shape. As shown in figure 6, the result obtains cartoon texture and retains the real facial feature. The module can output different result by different cartoon template that user chooses.

![Figure 6. The result of texture mapping](image)

5. Ears adding

Ears adding mainly adopts curve rendering and texture mapping in CG (Computer Graphics). The module has four steps. Firstly, it draws two curves \( C_1 \) and \( C_2 \) in the positions of given feature points (shown in figure 7) of the cheek. The curves have the same radian. By this, the curves and cheek compose two closed areas, \( S_1 \) and \( S_2 \), as shown in the step one of figure 8. Secondly, it maps cartoon texture to \( S_1 \) and \( S_2 \). Supposed that \( P \) is one point in \( S_1 \) or \( S_2 \), whose pixel value is \( P_c \) and \( P' \) is any point in cartoon skin, whose pixel value is \( P'_c \). The mapping target is for any \( P, P \in (S_1 \cup S_2) \), then \( P_c = P'_c \). The result is shown in the step two of figure 8. Thirdly, it draws curves inside of ears to make it more natural. The result is shown in the step three of figure 8. Because the result processed by above steps has serrated edges, lastly, the module smoothes the edges by improved median filtering algorithm. The filtering algorithm deals with the pixels near the edges directly and has a very good effect, as shown in the step four of figure 8.

![Figure 7. The given feature points](image)

![Step one](image)

![Step two](image)
6. Hair adding

The cartoon obtained by above treatments is lack of another important characteristic which is cartoon hair. Hair adding is to add multiple hair style to the cartoon. The hair styles are all saved in the hair template library. The module mainly utilizes image scaling and image mapping methods to map the hair to cartoon. This idea is first put forward in the cartoon creation field, whose advantage is that system can expand hair templates anytime without modification of the algorithm. The detail is as follows:

Firstly, it calculates the distances between the designated points in cartoon $I_1$ and hair template $I_2$.

$$l_1 = |P_1 - P'_1| \quad l_2 = |P_2 - P'_2|$$

(3)

Where $P_1$ and $P'_1$ are designated points in cartoon $I_1$, $P_2$ and $P'_2$ are designated points in hair template $I_2$, as shown in figure 9 and 10. $l_1$ is Euclidean distance between $P_1$ and $P'_1$ while $l_2$ is the one between $P_2$ and $P'_2$.

Secondly, according to $k$, which is the percent of $l_1$ and $l_2$ ($k = \frac{l_1}{l_2}$), it scales the image of hair template.

The scaled image is

$$I_2 = S(I_2, k)$$

(4)

Lastly, it translates $I'_2$ to the proper position of $I_1$ according to

$$I'_1 = T(I_1, I'_2, rp)$$

(5)

Where $T(\cdot)$ is translating function, $rp$ is relative position between $I_1$ and $I'_2$. The purpose of translation is to make the designated points of $I'_2$ coincided with the ones of $I_1$, that is to say $P_1 = P_2$, $P'_1 = P'_2$, and the pixel values of specific position in $I_1$ replaced by the ones of $I'_2$. The final result is as shown in figure 11.

7. Conclusion

The demos outputted by our system and PicToon [3] are shown in figure 12 and 13. Both results preserve facial features of the original images. However, our system can output cartoon face with cartoon texture. It is more suitable to cartoon’s needs.
Our system takes about 2 seconds on a Intel dual core 2.33Ghz PC to generate a cartoon face that retains the person’s facial characteristics, which is faster than most of presented systems. Besides, in theory, our system can generate infinite results if enough templates are prepared.

There are some future directions to further improve our system. First, our system can only deal with frontal face image with the limit of facial feature points automatic locating algorithm. In order to retain person’s facial characteristics, the algorithm must be improved. However, this problem is a very challenging one. The manual setting module should be developed to adjust the points that can’t be located accurately by automatic module. Second, our system currently processes the neutral face and the generated cartoon face is also a neutral one. The different expressions will be added to the cartoon face. Three, our system fully retains the real facial feature. With purpose of humor, characteristic exaggeration should be considered to append.

8. References