EasyToon: An Easy and Quick Tool to Personalize a Cartoon Storyboard Using Family Photo Album

Shifeng Chen
The Chinese University of Hong Kong
sfchen5@ie.cuhk.edu.hk

Yuandong Tian
Shanghai Jiaotong University, China
tydsh@sjtu.edu.cn

Fang Wen, Ying-Qing Xu
Microsoft Research Asia, Beijing, China
{fangwen, yqxu}@microsoft.com

Xiaoou Tang
The Chinese University of Hong Kong
xtang@ie.cuhk.edu.hk

ABSTRACT

A family photo album based cartoon personalization system, EasyToon, is proposed in this paper. Using state of the art computer vision and graphics technologies and effective UI design, the interactive tool can quickly generate a personalized cartoon storyboard, which naturally blends a real face chosen from the family photo album into a cartoon picture. The personalized cartoon image is easily and quickly obtained in two main steps. First, the best face candidate is selected from the album interactively. Then a personalized cartoon image is automatically synthesized by blending the selected face into the interesting cartoon image. Experiments show that most users express great interest in our system. Without any art background, they can make a personalized cartoon of high quality using the EasyToon within minutes.

Categories and Subject Descriptors

I.4.9 [**Image Processing and Computer Vision**]: Applications

General Terms

Algorithms, Experimentation

Keywords

Cartoon Personalization, Family Photo Album

1. INTRODUCTION

This work is inspired by recent progress of two different human-centered digital photo processing technologies: on-line visual art personalization [1, 2, 3] and digital family photo management [4, 5, 9, 15, 16, 17].

People love to personalize their identity in the digital world such as through the MSN avatar. Recently, a number of websites [1, 2, 3] start to provide services to help users making personalized cartoon works by choosing different templates and inserting personalized text. This kind of cartoon picture can be used as personalized avatars, greeting cards, or special gifts (print on T-shirt). Although these web sites provide a large number of choices for personalization, people can hardly say that the final products belong to themselves, because other users may select the same templates. In fact, a good way of personalization is to get people themselves involved in the final products. The most identifiable personal feature is of course the face of the person. Thus inserting face into the cartoon picture is clearly the most effective and satisfying way of cartoon personalization.

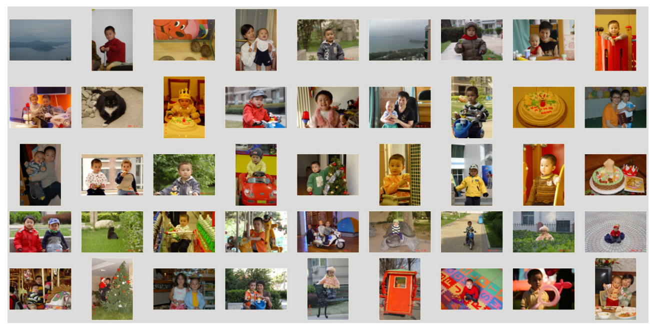

Figure 1: Some photos taken from a family album [5].

Fortunately, with the rapid popularization of digital cameras and mobile phone cameras, digital family photo albums grows explosively(Fig. 1). These family photo albums provide a huge pool of face photos that can be used for cartoon personalization [5]. In this work, we intend to bring the above two human-centered photo processing technologies together, to develop a unique cartoon personalization system.

Motivated by the observation that the most identifiable personal feature is one's face, our scheme for personalized cartoon is to extract a suitable face from the photo album and insert the real face into the cartoon pictures. Obviously, such composite cartoon image is really "personalized" compared with the result obtained by other methods.

Such face inserting cartoon personalization is a challenging task even for a professional artist. We asked an artist to demonstrate the processes of obtaining such a cartoon image. First, a suitable photo should be selected carefully from hundreds or thousands of images in the album. Then, the face should be cut out of the photo

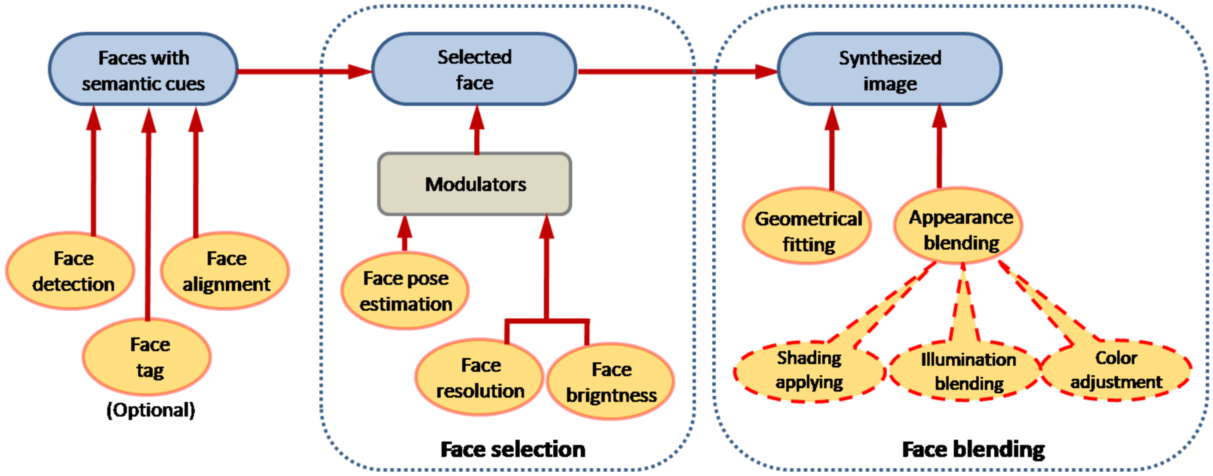

Figure 2: The framework of EasyToon. The two steps indicated by the dashed boxes are face selection and face blending, respectively.

carefully. Before seamlessly compositing the target cartoon image, the size, rotation, and colors of the face should be adjusted. Each step of this process is time consuming and the user is required to be familiar with some image editing tools such as Adobe Photoshop.

In this paper, we aim to design a system simulating the artist's action to generate the personalized cartoon in a fast and interactive manner. In addition to generating good results, the system should also be easy to use by users without any image processing or art background. It should also be fast enough for interactive processing. Fortunately, great progress has been made in computer vision and graphics making the design of such a system possible.

In recent years, remarkable progress has been made in face related technologies. The efficiency and performance of automatic Face Detection [10, 14], Face Alignment [19], and Face Recognition [11, 12, 18] have improved significantly. We are able to semantically extract human face from photos.

In addition, combining these technologies with a friendly user interface, several face annotation systems [4, 5, 9, 15, 16] have been developed to help user easily tag people names in the photos and use these name tags to explore the family photo album. These will greatly help our cartoon personalization system.

On the other hand, graphics technologies also progress rapidly in the last decade. Various image editing techniques, such as Poisson image editing [6, 7] and color transfer [8], enable a seamless image composition for synthesized images.

By combining above motioned techniques, a semi-automatic cartoon storyboard system, EasyToon, is designed to generate personalized cartoon pictures. The system includes two main steps. First, the face is selected in a simple interactive manner. This greatly reduces the time for face selection by avoiding browsing all photos. Based on the selected face, a personalized cartoon picture is generated automatically in seconds by blending the selected face into the cartoon template.

Besides the simple and quick algorithm, an easy to use UI is designed in EasyToon. For a user without any background in image processing and art, a few minutes are enough to get familiar with the system and to use the system to generate a personalized cartoon.

2. SYSTEM OVERVIEW

The goal of EasyToon is to provide a simple and quick solution to generate personalized cartoon. It is designed to work in a semi-automatic way. The framework of EasyToon in shown in Fig. 2.

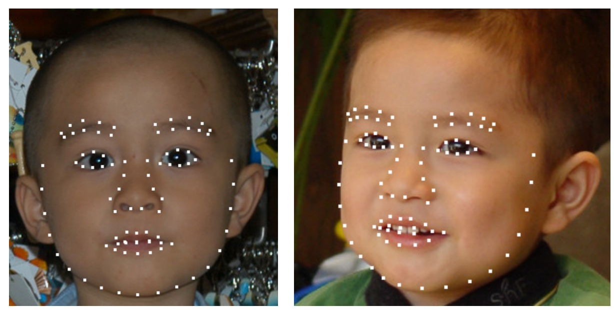

Figure 3: Examples of face alignment. Landmarks are labeled as white points. For each face, there are 87 landmarks, including 19 for profile, 10 for each brow, 8 for each eye, 12 for nose, 12 for outer lips, and 8 for inner lips.

Before starting the cartoon synthesis, the boosting-based face detection [14] and active shape model based face alignment [19] are performed to get 87 landmarks for each face in the albums (see Fig. 3), and each face is tagged by using EasyAlbum [5]. In order to reduce the running cost, this preprocessing is done off-line, i.e., for the same album, it is done once and the result is recorded.

Simulating human actions in cartoon personalization, EasyToon contains two key components:

1. Interactive Face Selection

   Since there are thousands of photos in the album, it is a time consuming job to select a face by browsing all of the photos. An interactive approach is designed in this system for face selection. Four key ideas are included in the face searching process.

   (a) Via face detection, photos without face can be filtered out so we can avoid browsing all photos in the album. After face detection, only regions that surround faces will be shown to the user.

   (b) Since we found that pose is the most important feature for face selection, an automatic real-time algorithm is used to select candidate faces via face pose estimation. The user can tune the "pose bar" in the UI to quickly get good face candidate via the preferred face orientation.

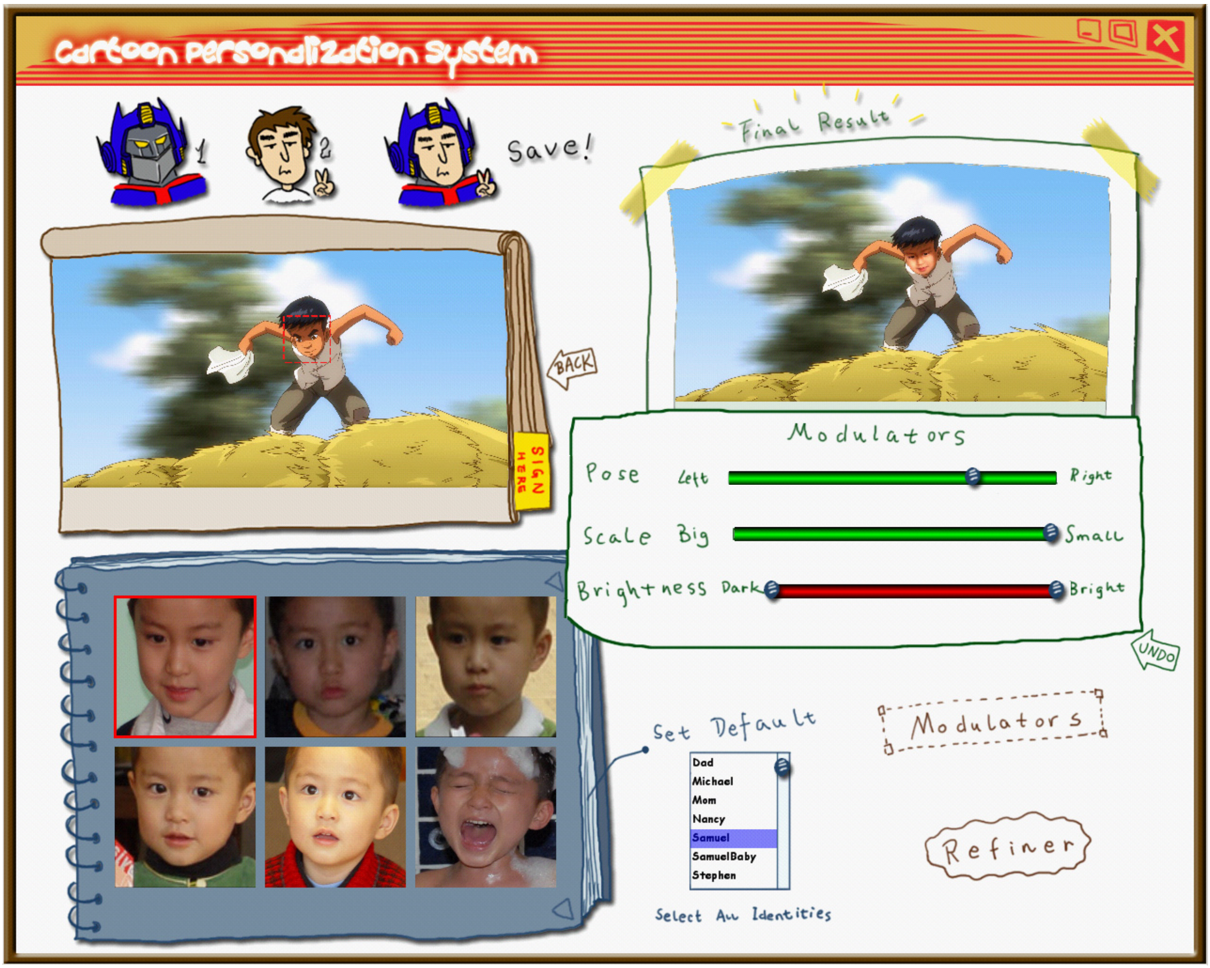

Figure 4: The UI of EasyToon.

This significantly reduces user labor in exploring the candidate faces.

(c) Since the sharply dark/bright faces may generate unsatisfactory results and the faces with size much smaller than that of the cartoon template may cause blurring artifacts, two face filters are designed to remove face candidates using the feature of brightness and resolution.

(d) An optional item is designed to make full use of tagged people name in the family photo album. If the user select to use this item, s/he can easily concentrate on faces of a specific person; otherwise, s/he chooses face from the candidates of all people.

2. Face Blending

   Given the selected face, the system will automatically blend it into the cartoon template. In this face blending step, the geometrical fitting and appearance blending process are performed to deal with cartoon synthesis in the aspect of geometry (position and shape) and appearance (shading, illumination, and color), respectively.

The UI of EasyToon is shown in Fig. 4. It is quite easy for users to use EasyToon and produce synthesized result. After family album importation, the user first picks a cartoon in the cartoon window (left-top) and picks a face in the cartoon. EasyToon will show all faces of the photo album in the face candidate window (left-bottom). The user can optionally select the name in the right of the face candidate window to concentrate on faces of a specific person. If the user chooses to select a name, the face candidate window will show only faces of the selected person. By clicking the "modulators" in the main frame, a window popped up. By tuning the "Pose" bar, all candidates are ranked based on the face orientation and by tuning the "Scale" and "Brightness" bars, too dark/bright candidates or those with too small size can be eliminated. Once the user chooses a candidate face, a synthesized result will appear in the result window (right-top) in seconds. User can easily try different candidates quickly.

3. FACE SELECTION

Although photos without faces can be filtered out via face detection, there are still a large number of photos left. Selecting a face from these photos is still a time consuming job. Based on features in face regions, we propose an interactive face selection method to avoid browsing all faces.

Face orientation is the most important feature in face selection. If the pose of the selected face is distinct from that of the cartoon image, the synthesized result will look unnatural (see the left ex-

Figure 5: Example of results for bad face candidates. Upper row: selected faces; bottom row: synthesized images. From left to right: artifact caused due to wrong pose, too small resolution, and serious darkness of the selected face.

ample in Fig. 5). Therefore, the key in our face selection approach is that pose orientations are estimated for all faces and good face candidates can be obtained based on an interactively given face orientation. On the other hand, faces with sharply dark/bright illumination may cause serious artificial effect (see the right example in Fig. 5), and faces with size much smaller than that of the cartoon template may cause blurring in the synthesis result (see the middle example in Fig. 5). To eliminate such images, two filters are designed based on the features of illumination and resolution.

3.1 Pose Estimation

To calculate the face orientation, we first estimate the symmetric axis in the 2D face plane. Suppose θ is the in-plane rotation of a face and C is the shift parameter, then θ and C can be estimated by optimizing the following symmetric criteria:

$$(\hat{\theta}, \hat{C}) = \underset{(\theta, C)}{\operatorname{argmin}} \sum_{i \in \mathcal{L}} |u^T p_i + u^T p_i'|^2, \quad \text{s.t.} \quad u = (\cos\theta, \sin\theta, C)^T, \tag{1}$$

where $\mathcal{L}$ is the set of landmarks in the left nose and mouth (refer to Fig. 3), $p_i = (x_i, y_i, 1)^T$ is a point in $\mathcal{L}$, and p_i' is the corresponding point of p_i in the right part of the face. The reason of using only points in the nose and mouth to estimate the symmetric axis is that points far away from the symmetric axis increase the amount of effect caused by 3D rotation.

After obtaining $(\hat{\theta}, \hat{C})$ from (1), the distance between the symmetric axis and points in the left and right face are calculated as:

$$d_l = \sum_{i \in \mathcal{L}'} \hat{u}^T p_i, \quad d_r = \sum_{i \in \mathcal{R}'} \hat{u}^T p_i, \tag{2}$$

where $\hat{u} = (\cos\hat{\theta}, \sin\hat{\theta}, \hat{C})^T$, and $\mathcal{L}'$ and $\mathcal{R}'$ are the set of points excluding nose and mouth in left and right respectively. The final face orientation is calculated as:

$$o = -\frac{d_r}{d_l}. \tag{3}$$

Theoretically, for the frontal face $o = 1$, for the right-viewed face $o > 1$, and for the left-viewed face $o < 1$.

If poses are calculated, good face candidates can be selected based on preferred face orientation (see Fig. 6). The "Pose" bar in the "Modulators" represents the face orientation. Different positions of the bar correspond different face orientations. When users drag the bar to the position corresponding to the desired face orientation, the candidate faces with preferred orientation will be shown in the left-bottom window.

3.2 Face Filters

Based on pose selection, one can quickly select face candidates. However, pose is not the only feature for face selection. An overly dark/bright face may generate unsatisfactory result. So a filter based on the face feature brightness is designed to eliminate some bad face images. On the other hand, because the face is scaled to have the same size as that of the cartoon template in the synthesis step, blurring occurs if the selected face is much smaller than the cartoon template. Therefore, another face filter is designed based on the feature of image resolution.

Denote l_i as the illumination of point i in the face. Then the mean value of illumination of all points in the face is regarded as l_m, the brightness of the face, which is expressed as:

$$l_m = \frac{1}{N} \sum_i l_i, \tag{4}$$

where N is the pixel number of the face. Then the brightness filter is designed as follows: given two threshold brightness values, l_{min} and l_{max}, if $l_{min} \leq l_m \leq l_{max}$, then the face is added as the face candidates; otherwise, the face is eliminated from the face candidates. In the system, the value of l_{min} and l_{max} are given by tuning two button on the "Brightness" bar.

Suppose that the width and height of the cartoon template are w_c and h_c, and the width and height of the face are w_f and h_f. For a given ratio s, the face filter based on resolution is designed as follows: if $w_f/w_c \geq s$ & $h_f/h_c \geq s$, then the face is accepted; otherwise, the face is filtered. Different positions of the "Scale" bar are used to control different ratio s in the UI.

4. FACE BLENDING

After selecting the face, the following step is used to blend the selected face into the cartoon template. This is a key step in EasyToon. Its main task is to geometrically fit the selected face to the given cartoon template and paste the face on the cartoon image naturally by changing the appearance of the face. Example results obtained in face blending process are shown in Fig. 7.

4.1 Geometrical Fitting

To blend the selected face into the cartoon template, two geometrical problems should be handled. The first one is to determine the place in which the face should be blended. The second is to estimate the transformation of the face (warp the face image to match the cartoon sketch labeled by the artist).

These two problems are simultaneously solved by estimate the affine transform matrix $\hat{A}$ via the following equation:

$$\hat{A} = \underset{A}{\operatorname{argmin}} \sum_{i \in M} |A p_i - p_i'|^2, \tag{5}$$

where A is a 3×3 affine matrix, M is the set of landmarks in the selected face (see Fig. 3), p_i is a landmark in the face, and p_i' is the corresponding point in the cartoon sketch. Here, p_i and p_i' are represented in homogenous coordinates system. After estimating the affine transformation, for a point p in the selected face, the corresponding coordinate in the cartoon image $p' = \hat{A}p$.

Theoretically, some other transformation models can warp the source image to the sketch more accurately. The reason of choosing the simple affine model is that the other models will distort the identity of the face, which is unacceptable in our system.

4.2 Appearance Blending

This step is to naturally blend the warped face in the cartoon template automatically. To conduct appearance blending, a face

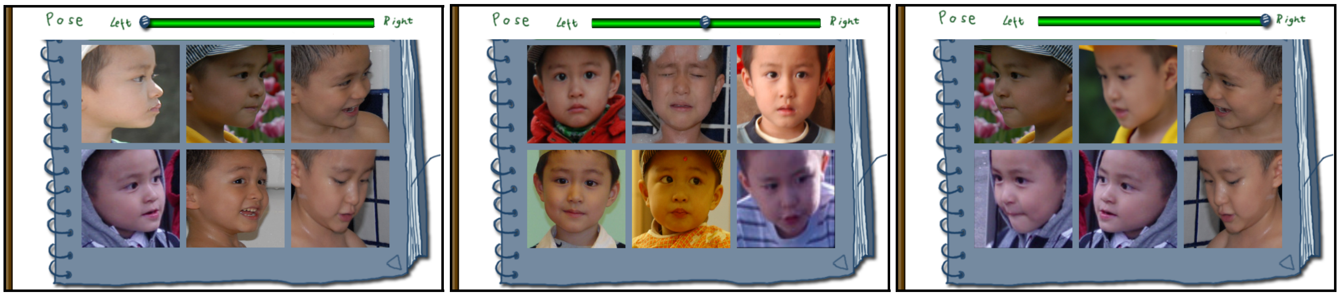

Figure 6: Demonstration of the "Pose" bar. Face ranking based on different preferred face orientations.

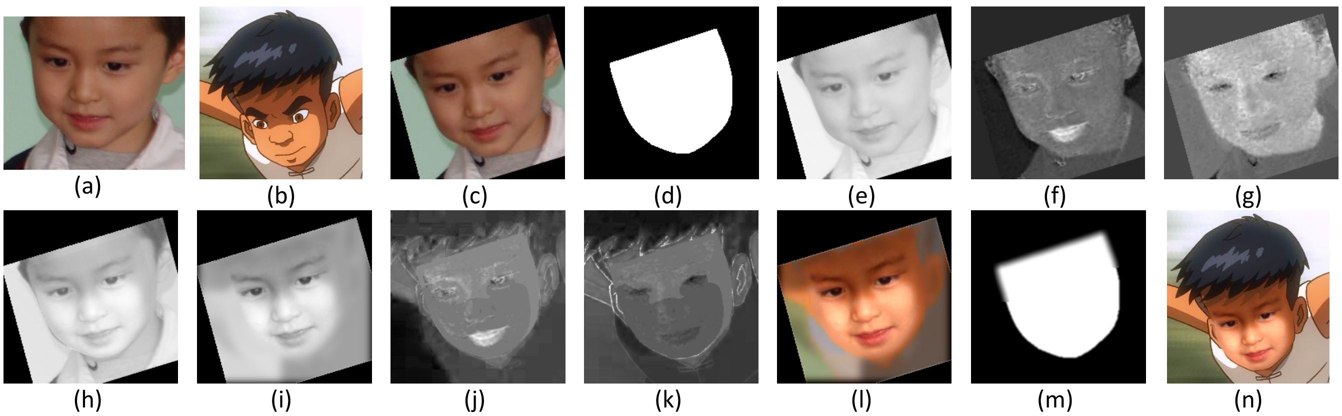

Figure 7: Results generated in automatic face blending process. (a): The original selected face image. (b): The selected cartoon. (c): Warped face image after geometrical fitting. (d): The face mask derived from the face image indicates the face region. (e)–(g): L, "a*" and "b*" components of the warped face image. (h): L component after shading applying. (i): L component after illumination blending. (j)–(k): "a*" and "b*" components after color adjustment. (l): The recombined image using components of (i)–(k). (m): Alpha map. (n): The synthesized cartoon.

mask indicating the region Ω where the face is blended should be calculated. The face mask is obtained by calculating the convex hull of the landmarks. Besides, object layers occluding the face in the cartoon and a cartoon skin color image implying shading information are given by the artist.

Besides, some objects occluding the face in the cartoon template, e.g., hair and glasses, are first labeled by the artist. Also, the shadow regions of the cartoon sketch is manually labeled by the artist such that they can be applied to the face.

Appearance blending includes three parts. The first part is shading applying, which applies shading in the synthesized image. The second one is illumination blending, which makes the illumination natural in the synthesized image. The third part is color adjustment, which adjusts the real face color to match the face color in the original cartoon image.

To perform appearance blending, the color RGB image is converted to L*a*b* color space. The "L" channel is the lumination channel, and "a*b*" are the color channels. In this system, shading applying and illumination blending are performed in the "L" channel and the color adjustment is performed in the "a*b*" channels.

4.2.1 Shading

Since real faces are taken in natural circumstances, they have different shading with cartoon faces. To make synthesized images look natural and have similar shading with the cartoon faces, the step of shading adds cartoon shading information in the composite result. Before performing this step, a cartoon skin image is provided by the artist indicating the skin color of the input cartoon face. The illumination of the cartoon skin image implies shading information. The regions with shading are darker than others. Our shading process is performed by utilizing the illumination information of the cartoon skin image.

Suppose that the cartoon skin image in the lumination channel is u^c. It includes K uniform color regions and the lumination value for each region k is $u^c(k)$. Denote the lumination image of the warped selected face as u and u_p is the lumination of point p. To combine shading, the first step is to find all points in the face whose corresponding points belong to the same region in the cartoon skin image. Suppose that the corresponding point of p belongs to region k. Then after shading, the new lumination value of p is set as

$$u'_p = \frac{u_p \cdot u^c(k)}{\overline{u}(k)}, \tag{6}$$

where $\overline{u}(k)$ is the mean lumination value of all points whose correspondences in the cartoon skin image belong to region k.

From Eq. (6) we can see that shading makes the lumination image of the selected face close to that of the cartoon face. Since shading information of cartoon is represented by its lumination image, it is applied in the real face image after shading. An example of the effect of shading is given in Fig. 8.

4.2.2 Illumination Blending

After shading, we get a new lumination image. Illumination blending puts the new lumination image u into the region Ω (face mask) naturally, keeping illumination smooth across $\partial\Omega$, the boundary of Ω. In EasyToon, illumination blending is performed vis Poisson Blending [7]. Mathematically, the Poisson approach formulates the blending as the following variational energy minimization problem:

$$J(u') = \min \int_{\Omega} |\nabla u' - v|^2 dxdy,$$
$$\text{s.t.} \quad u|_{\partial\Omega} = f, \tag{7}$$

where u' is the unknown function in Ω, v is the vector field in Ω, and f is the cartoon image on $\partial\Omega$. Usually, v is the gradient field of the image u, i.e., $v = \nabla u$. In our system, the minimization problem is discretized to be a sparse linear system and can be solved using Gauss-Seidel method in seconds [7].

Poisson Blending will degenerate if there are any image edges near the boundary of the mask. Unfortunately, it is the case in our system since the boundary of face mask is precisely on the edge of face. So the blending algorithm is slightly modified. Before doing blending, the face mask is contracted so that its boundary does not coincide with the face's.

An example to demonstrate the importance of illumination blending is shown in Fig. 8. From the figure we can see that the result without illumination blending looks unnatural and non-smooth.

4.2.3 Color Adjustment

The target of color adjustment is to find the skin color of the selected face and transfer the real face skin color to the cartoon-like skin color. This process is performed in "a*b*" color channels.

To estimate the skin color of the selected face, the face is clustered into K ($K = 6$ in our system) components via Gaussian Mixture Model (GMM) and each component is represented as (m_k, v_k, w_k), where m_k, v_k, and w_k are the mean, variance, and weight of the kth component. Intuitively, the largest component is regarded as the skin color component, denoted as (m_s, v_s, w_s), and the others are regarded as the non-skin components.

After getting the skin component, each point in the selected face can be labeled via a simple Bayesian inference. The posterior of point p belonging to the skin point is:

$$P(p \in \{\text{skin}\}) = \frac{G(m_s, v_s, p)w_s}{\sum_{k=1}^{K} G(m_k, v_k, p)w_k}, \tag{8}$$

where $G(m_k, v_k, p)$ is the likelihood of point p belonging to the kth GMM component. If $P(p \in \{\text{skin}\}) > 0.5$, then point p is regarded as a skin point; otherwise, it is a non-skin point. After labeling points, the following step is to modify the face color to cartoon-like color.

Suppose that p is a skin point, its original color is c_p (values in "a*b*" channels), and the modified color is c'_p. Then the following simple linear transform is used to map the center of the skin component to the center of the skin color of the cartoon sketch:

$$c'_p = c_p - m_s + c_c, \tag{9}$$

where m_s is the mean of skin component and c_c is the center of the skin color of the cartoon sketch. The colors of non-skin points are kept unchanged.

Refer to the last row in Fig. 8. The strange color of the face in the result without color adjustment indicates the necessity of the color adjustment step.

5. EXPERIMENT

To validate the usability and performance of EasyToon, we test it on a regular family album with 1040 photos [5]. In the cartoon

Figure 8: Comparative results of EasyToon with and without appearance blending steps. From top to bottom: effect of shading applying, illumination blending, and color adjustment. From left to right: cartoon images, results of EasyToon, and results of EasyToon with specific appearance blending step removed.

templates, layers (e.g., hair and glasses) and key points indicating facial parts (correspond to the landmarks in real faces) are provided by the artist. The experiment is conducted on an Intel Pentium 4, 3.0GHz computer with 1G memory.

At first, it takes about 30 minutes to find 1153 faces in the whole photo album via face detection and to find landmarks on all detected faces by face alignment. Considering the errors of the face alignment technique, the following step is used to eliminate the faces with bad alignment results. First all faces are scaled and warped in the frontal view. Then the mean value for each specific people is calculated based on all warped images. Then faces far away from the mean value are eliminated. After this step, 864 faces are kept and the rest are thrown away. The left-right flipped copies of the kept 864 faces are used to double the face data. Therefore, there are total 1728 faces used the the experiment. These preprocesses are performed off-line and all results are recorded for the later experiment.

Twenty volunteers take part in the experiment. They are university students with ages ranging from 21 to 25. None of them are from art-school or other art-related institutions. They are asked to test the system and provide their feedback.

5.1 System Performance

At the beginning of the experiment, the goal and the interface of EasyToon are introduced to participants. A navigator tells them the whole process of making synthesis images, and the main function of each UI component. Then after they got familiar with EasyToon (usually it takes about 10-15 minutes), each of them is asked to synthesize two cartoons based on two cartoon templates. The first cartoon template is appointed by us and the second cartoon template is selected by themselves from all given templates.

Figure 9: Some synthesized results of EasyToon. Top image: the given cartoon image. Row 2 – 4: column 1 and 3 show the selected face images and column 2 and 4 show the synthesized results.

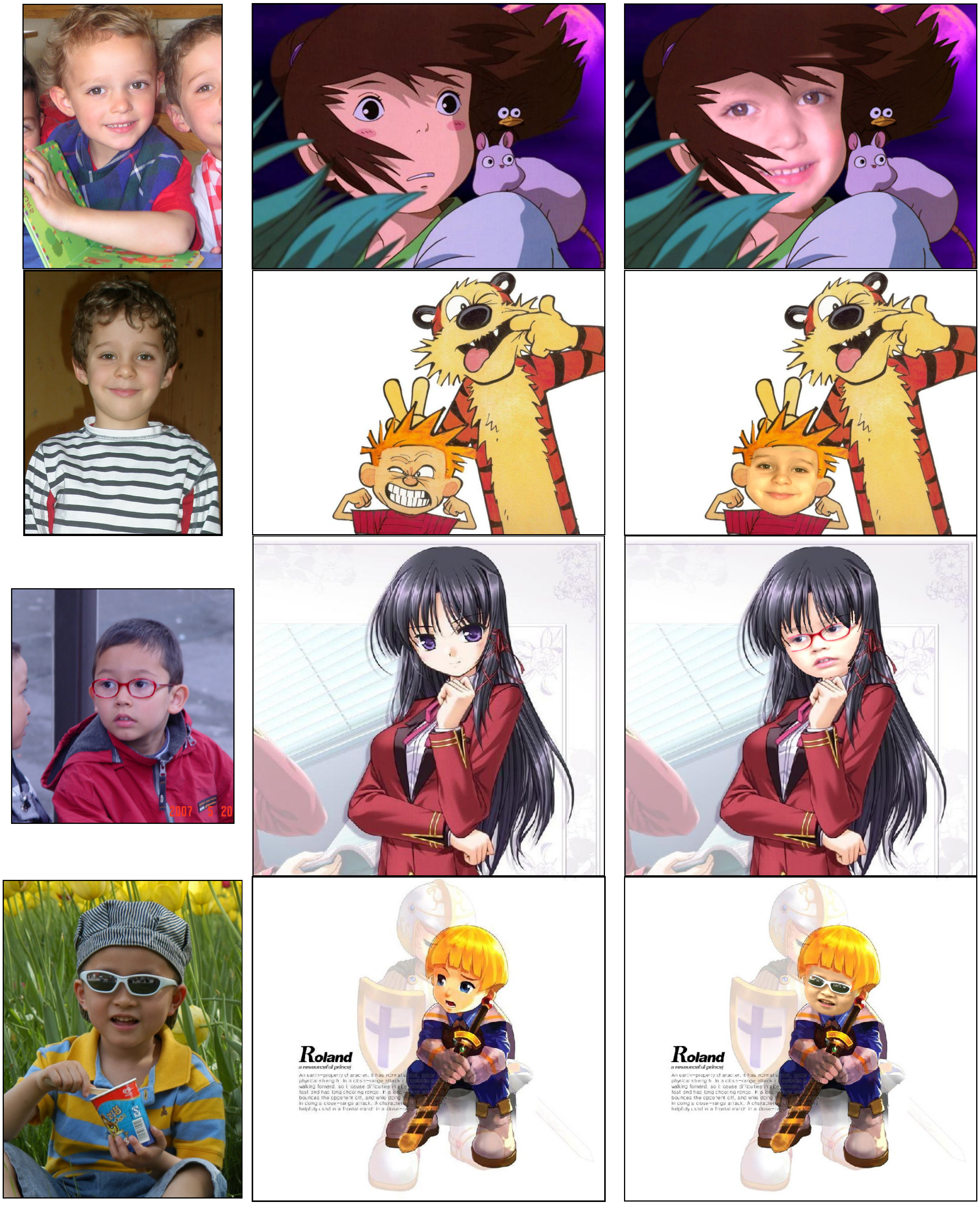

Figure 10: Some synthesized results of EasyToon. From left to right: the selected face images, the cartoon images, and the synthesized images.

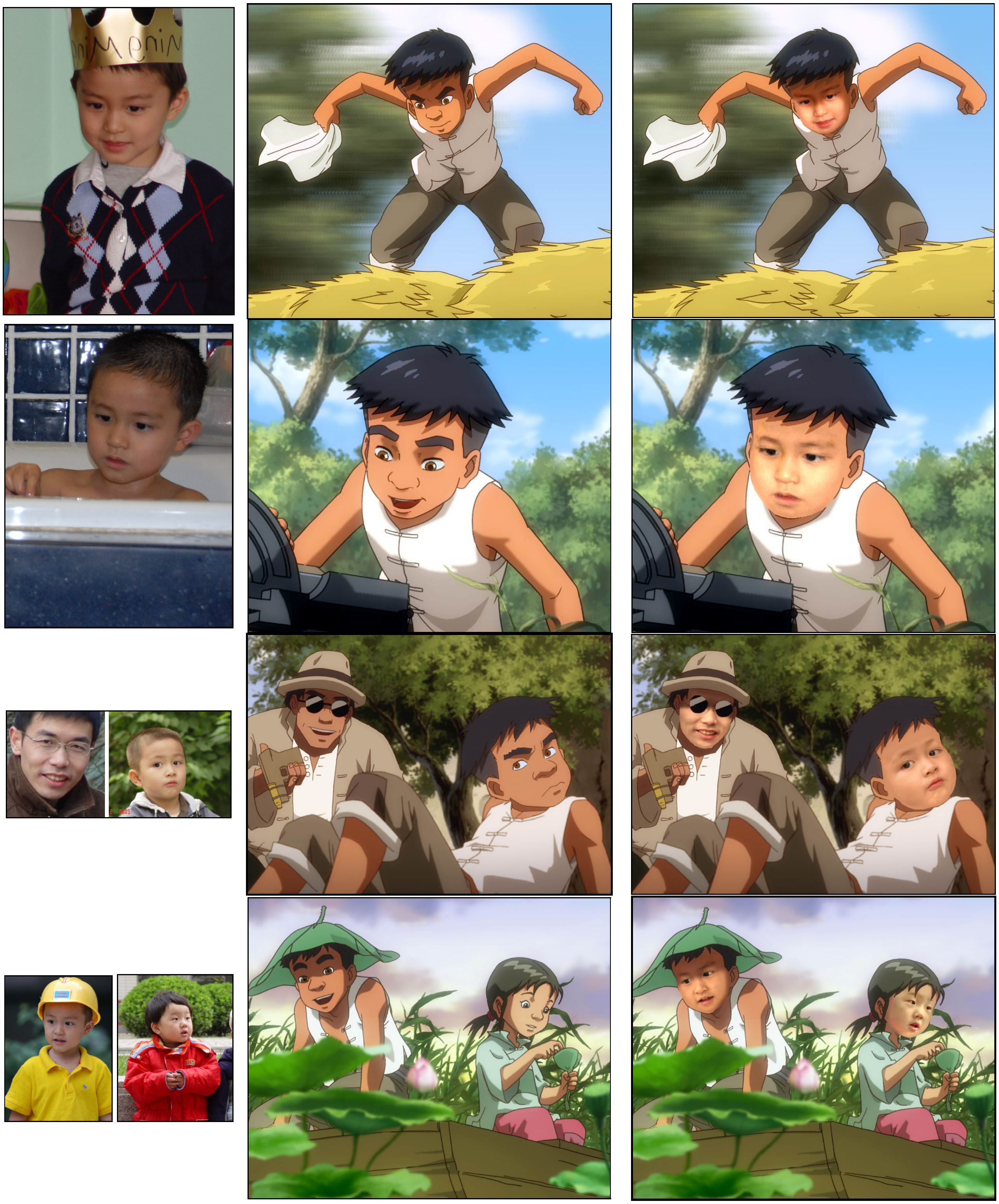

Figure 11: Some synthesized results of EasyToon. From left to right: the selected face images, the cartoon images, and the synthesized images.

Without any art background, they did a wonderful job in less than 5 minutes with 15 of them in less than 2 minutes. Compared with artists, who usually spends about 30-60 minutes to find a suitable face, adjust and paste it using advanced tools, EasyToon is very effective and efficient.

Some results obtained by different participants on the same appointed template is provided in Fig. 9. From the result we can see that for a given cartoon template, there may be more than one good results obtained.

Fig. 10 provides some results on different cartoon templates selected by different participants. From the result we can see that for different cartoon, EasyToon can obtain natural and satisfactory results.

One interesting application of EasyToon is that we can tell a story by using it. An example result is given in Fig. 11. All images in Fig. 11 are extracted from a strip cartoon. We replace the leading character by faces of the same boy. What is more, from the results we can see that EasyToon can replace more than one faces in a cartoon image.

5.2 User Study

After all participants used EasyToon, they are asked to finish a questionnaire. The questions in the questionnaire includes four aspects of EasyToon: *Is it interesting? Is it convenient to use? Is it fast? Is its result satisfactory?*

For the first question, 16 out of 20 (80%) participants answer "very interesting". They show great interest to EasyToon. This validates our work. It is useful and meets potential demands of customization. For the second question, most participants (15 of 20, 75%) feel that EasyToon is convenient to use (choose "very convenient" or "quite convenient"), which basically justifies the overall ease of use of EasyToon. About the speed, 17 of 20 (85%) participants choose "very good" or "quite good", and about the result, all participants answer "very good" or "quite good". This validates the efficiency and good performance of EasyToon.

6. CONCLUSION

In this paper, a family photo album based semi-automatic cartoon personalization system, EasyToon, is proposed to generate personalized cartoon picture by replacing the cartoon face using a real face in the album.

To simulate the action of the professional artist, EasyToon is designed in two stages. First, an interactive approach is proposed to select a good face candidate from thousands of faces. This process avoids browsing all photos and greatly reduces the time cost for face selection. Second, an automatic face blending process is designed to paste the selected face on the cartoon image. This process releases the user from many boring tasks, e.g., face cutout, face transformation, and color adjustment. And it allows the user to quickly try out different results.

With a friendly UI, EasyToon is simple and quick. A user without any artist background can use this system easily in a short time and synthesize personalized cartoon fast. For the actual demo system, refer to [13].

So far, EasyToon personalizes cartoon images by inserting faces. One possible future works is to generalize the system to more general image content replacement problem. Not only faces but also other contents, e.g., body and clothes, can be replaced.

7. REFERENCES

[1] `http://www.whimsies-online.com/`.

[2] `http://www.oktoon.com/`.

[3] `http://www.cowboychuck.com/`.

[4] Photoshop elements. `http://www.adobe.com/products/photoshopelwin`.

[5] J. Cui, F. Wen, R. Xiao, Y. Tian, and X. Tang. EasyAlbum: an interactive photo annotation system based on face clustering and re-ranking. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 367–376, 2007.

[6] J. Jia, J. Sun, C. Tang, and H. Shum. Drag-and-drop pasting. *ACM Transactions on Graphics (TOG)*, 25(3):631–637, 2006.

[7] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics (TOG)*, 22(3):313–318, 2003.

[8] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *Computer Graphics and Applications*, 21(5):34–41, 2001.

[9] Y. Tian, W. Liu, R. Xiao, F. Wen, and X. Tang. A face annotation framework with partial clustering and interactive labeling. In *Proceedings of Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[10] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[11] X. Wang and X. Tang. A unified framework for subspace face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1222–1228, 2004.

[12] X. Wang and X. Tang. Random Sampling for Subspace Face Recognition. *International Journal of Computer Vision*, 70(1):91–104, 2006.

[13] F. Wen, S. Chen, and X. Tang. Easytoon: Cartoon personalization using face photos. In *Proceedings of the 16th annual ACM international conference on Multimedia*, 2008.

[14] R. Xiao, H. Zhu, H. Sun, and X. Tang. Dynamic cascades for face detection. In *Proceedings of International Conference on Computer Vision*, 2007.

[15] L. Zhang, L. Chen, M. Li, and H. Zhang. Automated annotation of human faces in family albums. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 355–358. ACM Press New York, NY, USA, 2003.

[16] L. Zhang, Y. Hu, M. Li, W. Ma, and H. Zhang. Efficient propagation for face annotation in family albums. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 716–723. ACM Press New York, NY, USA, 2004.

[17] M. Zhao, Y. Teo, S. Liu, T. Chua, and R. Jain. Automatic person annotation of family photo album. In *Proceedings of International Conference on Image and Video Retrieval*, pages 163–172. Springer, 2006.

[18] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld. Face Recognition: A Literature Survey. *ACM Computing Surveys*, 35(4):399–458, 2003.

[19] Y. Zhou, L. Gu, and H. Zhang. Bayesian tangent shape model: estimating shape and pose parameters via Bayesian inference. In *Proceedings of Computer Vision and Pattern Recognition*, volume 1, 2003.