Multimed Tools Appl (2013) 63:729–756
DOI 10.1007/s11042-011-0891-1

Simulation of face/hairstyle swapping in photographs with skin texture synthesis

Jia-Kai Chou · Chuan-Kai Yang

Published online: 20 October 2011

Abstract The modern trend of diversification and personalization has encouraged people to boldly express their differentiation and uniqueness in many aspects, and one of the noticeable evidences is the wide variety of hairstyles that we could observe today. Given the needs for hairstyle customization, approaches or systems, ranging from 2D to 3D, or from automatic to manual, have been proposed or developed to digitally facilitate the choice of hairstyles. However, nearly all existing approaches suffer from providing realistic hairstyle synthesis results. By assuming the inputs to be 2D photos, the vividness of a hairstyle re-synthesis result relies heavily on the removal of the original hairstyle, because the co-existence of the original hairstyle and the newly re-synthesized hairstyle may lead to serious artifact on human perception. We resolve this issue by extending the *active shape model* to more precisely extract the entire facial contour, which can then be used to trim away the hair from the input photo. After hair removal, the facial skin of the revealed forehead needs to be recovered. Since the skin texture is non-stationary and there is little information left, the traditional texture synthesis and image inpainting approaches do not fit to solve this problem. Our proposed method yields a more desired facial skin patch by first interpolating a base skin patch, and followed by a non-stationary texture synthesis. In this paper, we also would like to reduce the user assistance during such a process as much as possible. We have devised a new and friendly facial contour and hairstyle adjusting mechanism that make it extremely easy to manipulate and fit a desired hairstyle onto a face. In addition, our system is

J.-K. Chou · C.-K. Yang (✉)
Department of Information Management, National Taiwan University of Science
and Technology, No. 43, Sec. 4, Keelung Road, Taipei, 106, Taiwan,
Republic of China
e-mail: ckyang@cs.ntust.edu.tw

J.-K. Chou
e-mail: A9409004@mail.ntust.edu.tw

also equipped with the functionality of extracting the hairstyle from a given photo, which makes our work more complete. Moreover, by extracting the face from the input photo, our system allows users to exchange faces as well. In the end of this paper, our re-synthesized results are shown, comparisons are made, and user studies are conducted as well to further demonstrate the usefulness of our system.

Keywords Active shape model • Skin texture synthesis • Hairstyle extraction

1 Introduction

We have come to an era, this diversification and personalization age, where everyone dares to pursue his/her life style that is not necessarily align with others. In this regard, hairstyle is one of the many manifestations that one may use to demonstrate his/her own characteristic or uniqueness. Furthermore, Perrett et al. [24] pointed out that different shapes of faces lead to different attractiveness, and in the meanwhile it is also well known that hairstyle is a very useful tool for illusively altering one's outlook. As a result, this trend has called for a variety of hairstyles to satisfy at least the following needs. First, it is not easy to foresee what one will look like before the making of a target hairstyle is completed, thus rendering the functionality of "previewing without making a real haircut" very attractive. Second, the preview mechanism would also be very beneficial and efficient for hair-cutters to quickly gain much experience on finding the suitable matches between faces and hairstyles. Third, it would be valuable to preserve existing representative hairstyles for stimulating new designs in the future.

Currently, there are solutions proposed for the aforementioned needs, and the involved approaches could be either 2D or 3D. However, most of them fail to provide realistic results due to the following reasons. First, a new hairstyle may not necessarily occlude the original one, thus making an unnatural coexistence of both hairstyles. Figure 1 shows some examples, where the original images, the selected hairstyles, the results generated from websites and some possibly more desired results are demonstrated, respectively. A straightforward approach is to remove the original hair before putting on the new one. However, the difficulty really lies in the proper dealing with the resulting empty regions from where the original hair gets removed. Note that to be able to accommodate a face to all possible hairstyles, we have to not only recover the *upper facial contour* containing the forehead, but also "inpaint a new skin patch" onto the newly revealed forehead region. To our knowledge, there exists little research efforts on synthesizing human facial skin patches after the removal of hair or facial features. Mohammed et al. [19] have recently proposed to solve a similar issue; however, their approach mainly focuses on generating novel facial images, instead of recovering the revealed facial regions. Second, the variety of shapes of facial contours and hairstyles have made the re-synthesis of a new hairstyle onto a given face very challenging. For example, a hairstyle that is vertically lengthy may not go well with a chubby face. As a consequence, in practice we have to scale the hairstyle along each dimension accordingly to match more closely with the input face. However, previous approaches rarely offer a flexible adjusting scheme that is both user-friendly and efficient for the purpose of manipulation.

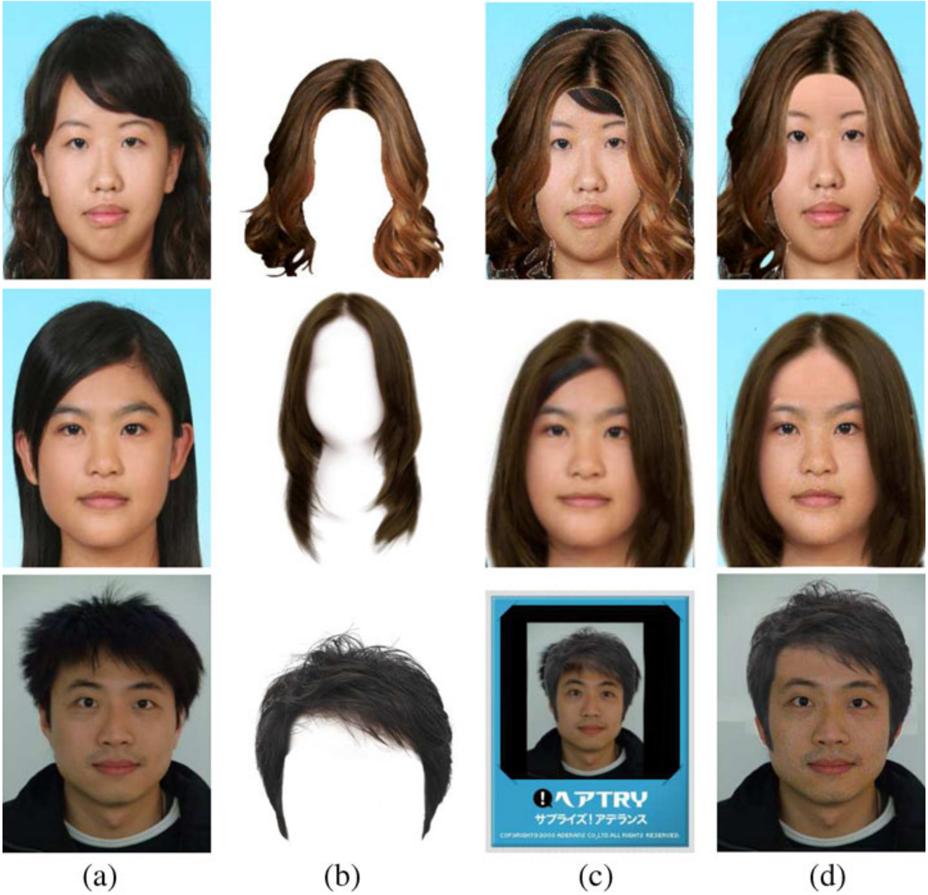


Fig. 1 Comparisons of our system with websites on hairstyle re-synthesis. The results are arranged in rows and the websites to be compared with are http://thehairstyler.com, http://f.hair-demo.jp and http://hairtry.jp, respectively. Note that the last 2 websites transform input images into 3D models, so the results may look a little bit different from the original images. **a** Original images. **b** Selected hairstyles. **c** Website results. **d** Our results

By assuming inputs to be 2D photos, our work successfully addresses the aforementioned issues to achieve the desired functionality of hairstyle extraction and re-synthesis while striking a good balance between the involved user assistance and desirable convincing results. Generally speaking, the contributions of this paper can be summarized as the following. First, we adopt the existing *active shape model* or ASM for short, to extract the facial contour from a given input photo. However, as ASM is not readily applicable for detecting the upper facial contour, we further extend its capability by fitting the missing facial contour portion with curves in concord with ASM's extracted part. Second, we present a novel skin texture synthesis scheme that could seamlessly fill in the missing skin details once the forehead region gets exposed after hair removal. Because the skin texture is non-stationary, traditional texture synthesis method cannot be readily adopted. And due to the lack of information in the revealed forehead region after hair removal, the approaches that simulate skin color/texture by composing different skin channels and inpainting algorithms are also undesirable. Our proposed method first generates an interpolated skin patch, and then we apply non-stationary texture synthesis to obtain a skin patch based on the interpolated skin patch and the texture information from the input human face. Therefore, a more realistic and desired result can be derived. Third, we strive to automate the whole process of hairstyle re-synthesis, including the identification of the entire facial contour, the recovery of revealed forehead region and the manipulation of fitting a hairstyle onto a given face. In addition, fine-tuning the automated results is also possible, while we have proposed a flexible, intuitive and efficient adjusting scheme, so that the effort of user assistance could be reduced. Finally, our work also offers a much desired functionality to perform the hair extraction from a photo, and the inclusion of such functionality not only completes our proposed framework, but also opens up the possibility of a database construction for various hairstyles. Results and user studies of our work are demonstrated to prove the effectiveness of our approaches.

The rest of the paper is organized as the following. Section 2 reviews previous literature related to this work. Section 3 describes our proposed algorithms on recovering a facial contour, synthesizing the forehead skin texture, extracting a hairstyle, and editing a hairstyle. Section 4 demonstrates the generated results of our work. Section 5 discusses the effectiveness of our proposed system with some statistical analysis and user studies. Section 6 concludes this paper, addresses some potential limitations, and envisions possible future directions.

2 Related work

There exist previous works related to this paper. Perhaps the most similar one is the patent by Blancato [4], where many similar functionalities were proposed, except that nearly most of them need manual assistance. Two interesting papers that can be viewed as “indirectly” related to our work are the ones proposed by Blanz et al. [5] and Bitouk et al. [3]. Their work could exchange faces in two images, and as a result two hairstyles get exchanged as well. However, strictly speaking, they achieve the purpose through swapping facial features but not the entire faces. In this work, we propose a scheme that allows a user to change his/her face or hairstyle in a user-friendly fashion.

Face detection plays a critical role in this work for finding the facial contour, as well as for extracting the hairstyle from an input image. Hsu et al. [15], Yang et al. [35], and Kjeldsen et al. [16] introduced face detection methods that are based on skin color detection. Several classic face detection algorithms have adopted machine learning methods [25, 26, 31]. However, in this paper, recovering the facial contour is more crucial than detecting the position of a human face. Cootes et al. proposed the ASM method to extract an object contour from an input image [8]. Their algorithm first collects the information from many contours drawn explicitly by people, finds out the rules or principles of object shapes based on the gathered statistics, and finally derives object outlooks under different viewing angles or deformations. They later extended ASM to focus particularly on the contour extraction of human faces and the identification of facial features [7]. It is worth mentioning that, due to the obstruction of hair, normally ASM only extracts the *lower facial contour*, that is, the portion of facial contour that is lower than the eyebrows. As a consequence, new techniques are needed to recover the *upper facial contour*, which is especially needed in this work.

A significant part of this paper is on the synthesis of forehead skin after hair removal. For this purpose, there are two types of approaches that could be adopted, *image inpainting* and *texture synthesis*. In terms of image inpainting, Bertalmio et al. [2] used the information of the known region to impose boundary conditions and derive the area to be inpainted by solving *partial differential equations*, or PDEs for short. Oliveira et al. [22] sped up the inpainting process through image convolutions, while at the same time providing results with comparable image quality. Pertaining to texture synthesis, Efros et al. [12] employed a *pixel-based* method, that is, every pixel in the target texture is selected from the pixel in the source texture that bears the most similar neighboring pixels. Wei et al. [34] later improved the texture synthesis performance through a *vector quantization* scheme. By making use of a *patch-based* method, Efros et al. [11] synthesizing textures via overlapped patches, where the transition among multiple patches are smoothed by a *graph-cut* algorithm. There are also some approaches that intrinsically combine both traits of image inpainting and texture synthesis, such as those methods proposed by Drori et al. [10] for image completion and Sun et al. [29] for image completion with structure propagation. More specifically, Tsumura et al. proposed a skin synthesis scheme by decomposing a skin into three channels: Melanin, Hemoglobin, and lighting, and a particular skin type could be derived by a weighted sum of the three channels [30]. Doi et al. applied a similar weighted scheme, which is, however, mainly targeted for the synthesis of hand skin [9]. We have to point out that, *image inpainting* techniques cannot be readily applied to solve our problem, i.e., the forehead skin synthesis, as they are mainly designed for the situations where the area to be inpainted are *locally small*. On the other hand, *texture synthesis* methods cannot be easily adopted either, as the skin texture behaves quite differently from *local* and *stationary* textures that have been properly dealt with. Moreover, the approaches in [30] and [9] synthesize a skin texture as a whole and therefore are not suitable for the cases where a partial skin synthesis is required, unless the corresponding channel weighting factors can be inversely and efficiently discovered at the run time. Recently Mohammed et al. [19] successfully tackled this issue in a different way. They collected a certain amount of facial images to learn a probabilistic model. By adapting a non-stationary image quilting method, the system could generate novel facial images which are not identical to any others in the training data, together with the functionality to edit facial images. Though the second functionality is more related to what is desired in this paper, the involved setup is nonetheless not very trivial.

Regarding hair extraction, in general it could be done through *matting*, including *blue screen matting* by Smith et al. [27], *Baysian matting* by Chuang et al. [6], *Poisson* *matting* by Sun et al. [28], *intelligent scissor* by Mortensen et al. [20], *soft scissor* by Wang et al. [32], and so on. However, to be able to apply these techniques, either special environment setup is mandatory, such as in [27], or an initial contour must be drawn, as in the rest of these methods. Instead, our approach requires much less user guidance for hair extraction, and resorts to a more automatic segmentation approach that makes use of the color information from both hair and skin.

3 Hair extraction and re-synthesis

3.1 System overview

Figure 2 demonstrates our system overview. At first a 2D photo is given as the input, and the associated facial contour is detected, while in the mean time the corresponding hairstyle is extracted as well. The detected facial contour is used to remove hair to obtain a 2D head without hair, termed *bare head* hereafter. The extracted hairstyle and bare head are added into the database. There are two modes that a user can choose. First, the *hair-less mode*, where he/she can select a desired hairstyle from the database to be composited with his/her bare head. The second

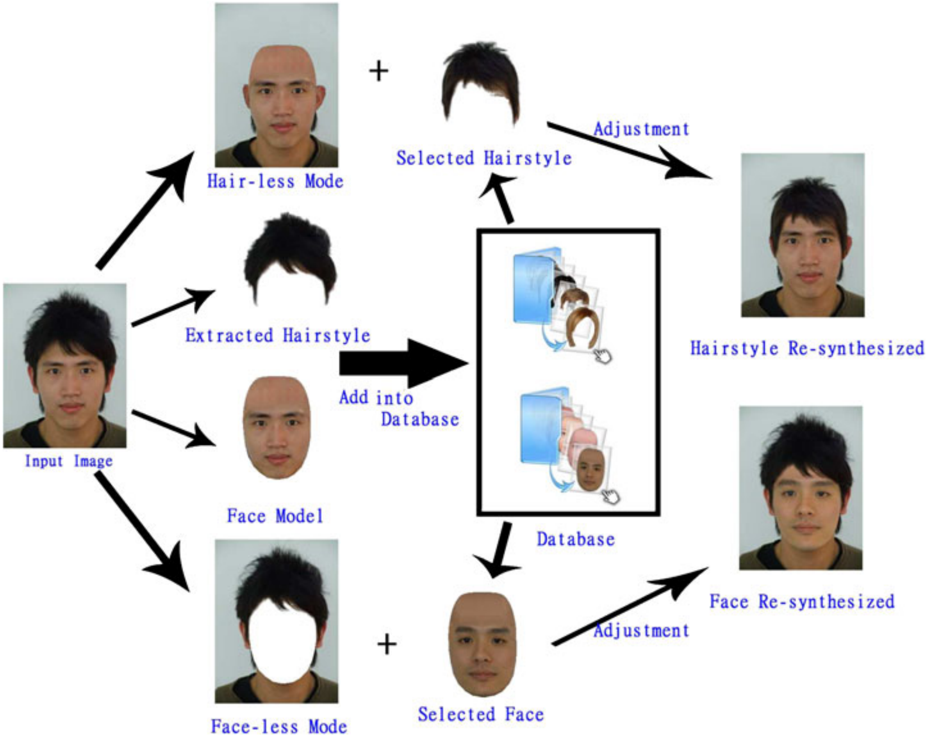


Fig. 2 System overview of this work

mode is *face-less* mode, where his/her face is removed and replaced by a selected face from the database. Note that during the composition process, the selected hair/head can be adjusted in terms of size and positions manually by the user or automatically by our system.

The details of the involved steps will be covered in the following sub-sections. Section 3.2 shows how we identify the entire facial contour by extending the ASM algorithm. Section 3.3 describes the process of skin patch generation, so that the revealed forehead region after hair removal can be recovered. Section 3.4 details our algorithm to extract the hairstyle from a input image. Section 3.5 demonstrates the abilities of our system to match an arbitrary hairstyle onto an arbitrary face automatically as well as a flexible and efficient adjusting mechanism to fine-tune the results if needed. Section 3.6 discusses the replacement of partially/completely occluded ears with intact but somehow fake ears, while it oftentimes yields much more convincing results.

3.2 Facial contour extraction

A facial contour consists of two parts, i.e., the *lower contour*, as shown in Fig. 3a, and the *upper contour*, as shown in Fig. 3b. For detecting the lower contour, we simply apply ASM [7], whose code can be downloaded from [1]. Figure 4 demonstrates the ASM process. Note that ASM not only extracts the lower contour, but also other facial features as well, such as eyes and mouth, etc. The identification of these facial features is helpful to our skin texture synthesis process, which will be described later.

As our task is to synthesize the revealed forehead skin after hair removal, it becomes necessary that we first recover the upper facial contour. In the initial state, as shown in Fig. 5a, let M, N correspond to the left and right topmost pixels of the lower contour as shown in Fig. 3a, and according to the ASM algorithm, normally these two pixels are vertically between the eyes and eyebrows. We next determine O and P such that $\overline{MO} = \overline{PN} = \frac{1}{4}\overline{MN}$, and thus $\overline{OP} = \frac{1}{2}\overline{MN}$. Note that the horizontal positions of I and K are the same as O and P, respectively. The vertical positions of I and J are set to be above the right eyebrow and the left eyebrow, through the help of ASM. According to [21], a human face could be divided vertically into three distinct thirds, which are the hairline to the eyebrows, the eyebrows to the base of nose and the base of nose to the bottom of the chin, respectively. We refer J as the

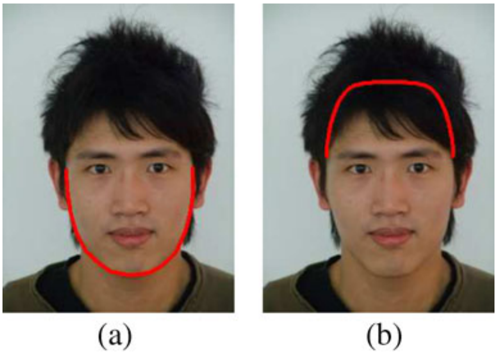


Fig. 3 **a** The *lower facial* contour detected by ASM. **b** The *upper facial* contour extracted by our proposed approach

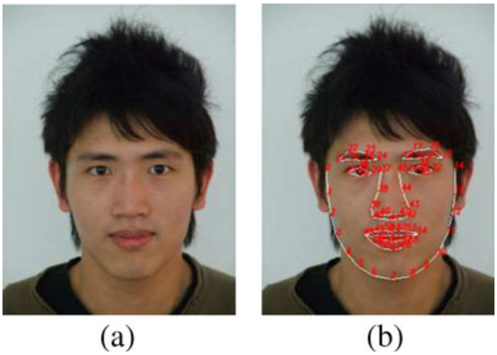


Fig. 4 The ASM process. **a** The original image. **b** The resulting image after ASM applies

hairline of a human face. Thus, we can then derive J's vertical position accordingly. And J's horizontal position is determined as the midpoint between O and P. By making use of the position of O, and treating $\overline{MO}$ and $\overline{OI}$ as the *minor radius and major radius* of an ellipse, a quarter arc on the circumference of the ellipse, denoted by A, could be derived, and so could arc C. On the other hand, arc B represents a half arc on the circumference of the ellipse that is determined by I, J and K. That is,

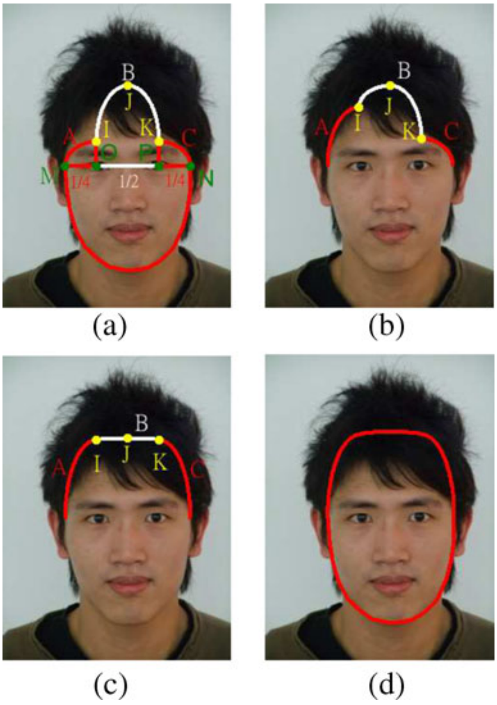
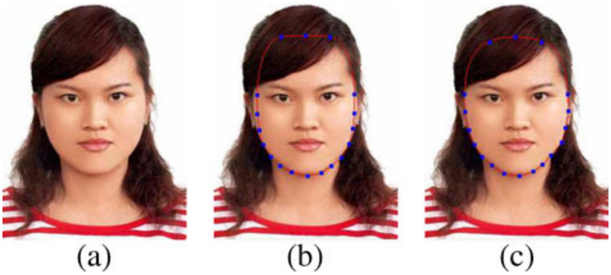


Fig. 5 **a** The initial state of the *upper facial* contour detection. **b** Refined *upper facial* contour concerning the revealed skin pixels on the forehead region of the input image. **c** The final result after smoothing the connecting points I and K. **d** The entire facial contour detected by our system

the minor radius and major radius of this ellipse are $\frac{1}{2}\overline{IK}$, and the distance between J and the line segment $\overline{IK}$, respectively. So the initial setup for the upper facial contour has been done, as shown in Fig. 5a. We now move I and K upwardly to finish the process of upper facial contour detection. For I, we start from every pixel lying on arc A, and we search upwardly until meeting the first non-skin pixel. Then the vertical position of I is set to be the highest vertical position found. The vertical position of K is determine in the same fashion but starts with arc C. However, as shown in Fig. 5b, as I and K are discovered independently, it is not necessary that the three arcs join smoothly at the connecting points I and K. We resolve this by the following procedures. We first enforce the constraint that the vertical position of J must not be lower than that of I or K, and if so, the vertical position of J is elevated to be the higher vertical position of I and K. On the other hand, assuming that both the vertical positions of I and K are lower than that of J, we move point I and K upwardly one pixel at a time. After each movement, we check whether the tangent directions of arc A and arc B at I (similarly arc B and arc C at K) align with each other or not, under the constraint that their vertical positions cannot be higher than that of J. The final result is shown in Fig. 5c. And the combined complete facial contour is shown in Fig. 5d.

There are two technical issues worth mentioning. First, the contour extraction process may not be always accurate. When ASM does not do a perfect job, user assistance may be required to adjust the detected lower facial contour. As can be seen from the previous contour detection process, the main purpose of the lower facial contour detection is for the extraction of facial contour, and in particular, the determination of the positions of M and N, and thus the derivation of other crucial points and curves. Second, the upper facial contour does not need to be very accurate, as in most of the situations the forehead of an input photo is blocked by hair anyway. Therefore as long as the shape of the proposed contour looks reasonable, it should not cause any serious problems. Most importantly, even in the case where a head without or with little hair is encountered, our proposed algorithm can still perform its job satisfactorily. In our implementation, an adjustment mechanism is provided to let the user to fine-tune the final result of the detected facial contour. As shown in Fig. 6b, the original detected facial contour does not fit the person's face very well. In this case, the user can adjust the facial contour by simply dragging any of the feature points (blue rectangles) to a more proper position, so that the adjusted facial contour might be desirable to the user. An adjusted and potentially more desired result can be seen in Fig. 6c.

Fig. 6 **a** The original image. **b** Facial contour detected by our system. **c** Facial contour fine-tuned by the user



3.3 Skin synthesis

Once a complete facial contour is extracted, the next step is to synthesize a patch of skin texture on the resulting hole region after hair removal. Due to the normally very limited available information, as oftentimes other skin portions may be blocked by hair, we first make a seemingly bold assumption that the forehead skin texture is very similar to the skin portion below the eyes and above the mouth, as can also be observed from the regions marked in red in Fig. 7a. And through the help of ASM, the search for such a rectangular area could be easily accomplished without difficulty. However, as region B includes the nose, thus resulting a lighting situation that may be different with other regions, such as regions A and C. To resolve this, we artificially regenerate the skin texture in region B as follows. First, we convert all the colors from the RGB color system into the $CIELAB$ color system, so that a more *decorrelated* chrominance and luminance distribution could be obtained. Note that the $CIELAB$ color system consists of L, a and b channels, where L is related to luminance and a, b are regarded as the information for chrominance. Second, we calculate the average a and b values for the pixels in regions A and C. These averaged

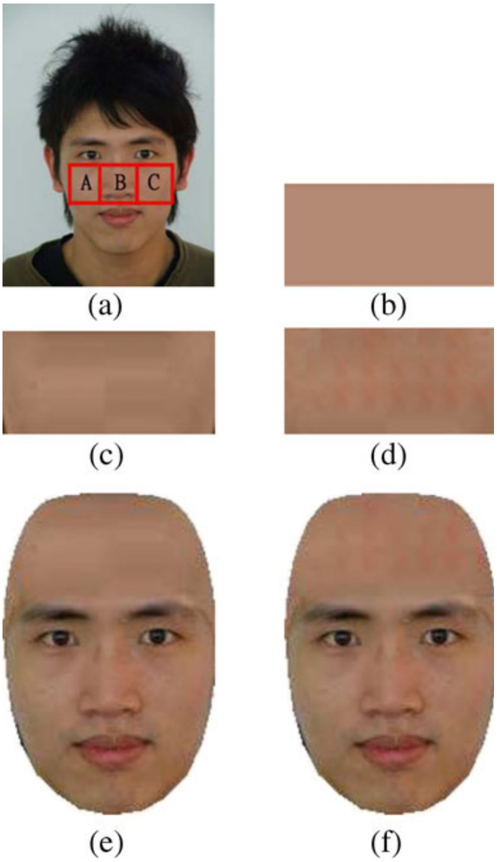


Fig. 7 Our skin synthesis process. **a** The skin texture, marked in *red*, that is to be extracted and used to synthesize the forehead skin. **b** The base color used for skin synthesis. **c** The linearly-interpolated skin. **d** The texture synthesized skin based on **c**. **e** The resulting head with hair removal and forehead skin recovered without texture synthesis. **f** The resulting head with hair removal and forehead skin recovered with texture synthesis

a and b values are combined to derive the base color for ensuing skin synthesis, as shown in Fig. 7b. Note that for displaying purpose, the L channel for the color shown in Fig. 7b is temporarily set to be the highest L value in regions A and C, as this skin color will eventually be replaced by the final synthesized skin color in the next step. The base skin color serves as the basic chrominance value for all the skin pixels to be synthesized within these three regions, and the L values, or the luminance values, for these pixels are then *linearly and horizontally interpolated* between the pixels on the (one pixel wide) *brightest column* in region B, and the pixels in regions A and C. Here the brightest column means the column containing the brightest pixel, where the nose tip usually resides. Third, divided by the brightest column in region B, the area to the left of the brightest column is linearly interpolated using the rightmost column of region A, while the area to the right is linearly interpolated with the leftmost column of region C. To smooth the potentially inconsistent transition on the brightest column, and also to counteract the effect of excessive lighting for originally being the nose region, we further apply the technique of *Poisson image editing* [23] around this region, and the result is shown in Fig. 7c. For each pixel within region A, B, and C, once all its three channel $CIELAB$ values are determined, its color representation could be converted back to the RGB color system. Finally, by making reference to region A and C in Fig. 7a, and using the region of the linearly interpolated result in Fig. 7c as the base texture, we perform the final texture synthesis process to derive the result shown in Fig. 7d. Note that to avoid odd transition between the synthesized forehead and the lower known facial region, *Poisson image editing* is again adopted during the texture synthesis process just mentioned. Figure 7e and f demonstrate the resulting *bare head* images without and with the aforementioned texture synthesis process, respectively.

3.4 Hairstyle extraction

We perform hairstyle extraction by making the following assumptions. First, the background color is simple, and should not interfere with the detection of skin, hair and clothes. Second, there should be enough hair for the purpose of detection, and there should be presumably some hair above the head. Third, the hair color should be distinguishable from the color of the clothes. If all these requirements are satisfied, we proceed with the hairstyle extraction by dividing an input image into four segmented regions, including skin, hair, background, and clothes by the following process. First, we define the polygon to represent the facial shape by using the set of points from the 0th to the 14th derived from ASM, as shown in Fig. 4b. Within the facial shape, we discard the pixels within the facial feature boundaries detected by ASM. For the remaining pixels, as the non-black area shown in Fig. 8, we calculate their statistical mean to obtain an average skin color estimation. Second, we calculate the average pixel color from the pixels in the topmost row on the image and treat it as the initial background color. Third, by finding the average color from the pixels in the area immediately above arc B in Fig. 5c, we could roughly get the hair color. Fourth, we compute the pixels in the last row, and their average is used to derive the base color for the clothes. Finally, once all four average colors are defined, we then use the four colors as the initial seeds and apply the *K-means* approach [18] to iteratively cluster the input image into the final four segmentations, as shown in Fig. 9b. Note that, the main purpose of applying the *K-means* algorithm

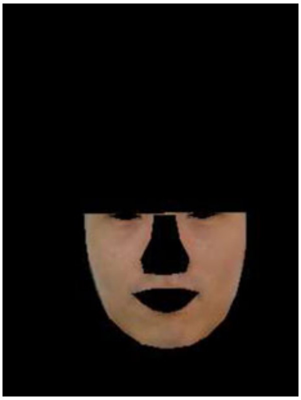


Fig. 8 The mask for calculating the skin color distribution

is to distinguish the color distribution of the hairstyle from others, so that we could make sure that no non-hair pixel is mis-classified into the hair cluster.

Next, for the ensuing matting purpose, we generate a *trimap* by the following process based on the aforementioned segmentation result. First, we find the *connected component* that intersects with arc *B* in Fig. 5c through a very efficient *run-based two-scan labeling algorithm* proposed by He et al. [14]. Second, we perform the *erosion* operations borrowed from *morphology* [13] on the component, where the involved mask is a 11 × 11 mask, to derive the white region shown in Fig. 9c.

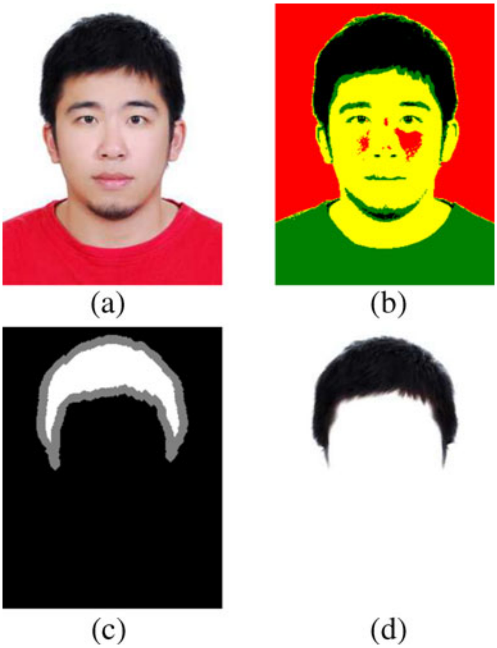


Fig. 9 The hair extraction process. **a** Original image. **b** The color segmentation result. **c** The generated trimap. **d** The extracted hairstyle

Note that, the sizes of most of the images in our database are 240 pixels in width and 320 pixels in height. Therefore, the size of the morphology mask should be adjusted depending on the size of the input image. Third, we perform another morphology operation, the *dilation* operation with the same mask also on the largest component to obtain the gray region shown in Fig. 9c. Note that due the properties of these two operations, the gray region must contain the white one, which for demonstration purpose is drawn on top the gray region. Forth, by subtracting the white region from the gray one, we could then generate a *trimap* to be sent to a *matting* process to extract the hairstyle, as shown in Fig. 9d. Here for matting we resort to Levin et al.'s *closed-form matting* [17] as it performs quite well and its implementation source code is available. Note that, for the ensuing operations, in addition to extracting the desired hairstyle from an image, we also need to generate a *bare head* image from the input photo. This is essentially the *background recovery process*, which is achieved by first removing the gray region from the input image, then texture synthesizing the resulting holes, and finally pasting back the previously synthesized bare head as shown in Fig. 7f.

3.5 Hairstyle editing

All the previously mentioned processings are for the preparation of the final stage, that is, the hairstyle re-synthesis. In other words, a user can freely select a desired hairstyle from the database and our system could be used to match a selected hairstyle with a bare head, obtained from previous steps. To allow maximal flexibility, our system offers two matching modes, automatic and manual. However, for both modes, our system first tries to locate the *reference points* on the hairstyle, which are later used to aligned with M, J and N in Fig. 5, so that the initial position of a given hairstyle can be determined. In the case where the hair extraction is done by our system, the reference points' location are already known. Notice that all the scaling operations mentioned hereafter are assumed to be with respect to the reference points.

For manual matching, our interface is shown in Fig. 10a. As can be seen in this figure, there are five control points, marked in blue, to scale a given hairstyle. The lower right control point is a *global scaler*, that is, its movement will cause

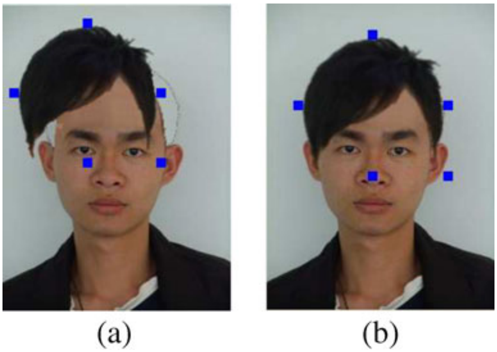


Fig. 10 **a** The original hairstyle with initial control points. **b** The scaled hairstyle with control points moved

the hairstyle to scale uniformly with respect to the reference point. On the other hand, the upper and lower two control points will only affect the upper and lower vertical scales, respectively, and similarly for the left and right control points to scale horizontally, as shown in Fig. 10b. Note that the result shown in Fig. 10b cannot be easily achieved by merely using the global scaler.

For automatic matching, the basic idea is to automatically scale a given hairstyle properly so that there exists no *holes* between hair and head. To meet these desired requirements, we define an energy function as the following:

$$E = E_{\text{hole}} \tag{1}$$

where E_{hole} represent the number of *hole pixels* that the hairstyle intersects or generates. Here the number of hole pixels is counted by the number of facial contour pixels above the mouth that are *adjacent to a hole*. These contour pixels are marked in red in Fig. 11, where the green line denotes the lower bound of facial contour that we put into consideration. A hole can be detected by checking if it is with the background color and gets enclosed by either skin or hair pixels. It is thus apparent that reducing these numbers, or equivalently the energy term, could lead to a better placement of the hairstyle.

With the energy function being defined, our system automatically and iteratively adjust the hairstyle as follows. In each iteration, we always check to see if any vertical or horizontal scaling operation can lower the total energy, and if so, scaling operation should be performed vertically or horizontally, depending on which scaling operation can lower the total energy more. Once a scaling operation is performed, the current iteration ends, and the next iteration starts. However, in the case where no scaling operation can lower the total energy, we move the hairstyle as a whole, and this indicates the mapping between the hairstyle's reference point and J (in Fig. 5) on the bare head should be modified as well, and the current iteration also finishes. Note that no matter for scaling or translation, the probe for lower energy is simply a trial on a position's four neighboring pixels: upper, lower, left, and right, and the position/operation that could further minimize the total energy is selected/performed. As shown in Fig. 12, our automatic adjusting scheme can generate results that are comparable with the ones that are done manually. For face swapping, the involved alignment becomes much easier as the corresponding ASM information is available.

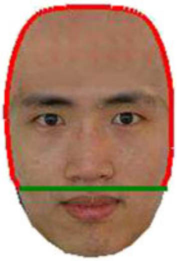


Fig. 11 The searching range for detecting undesired holes

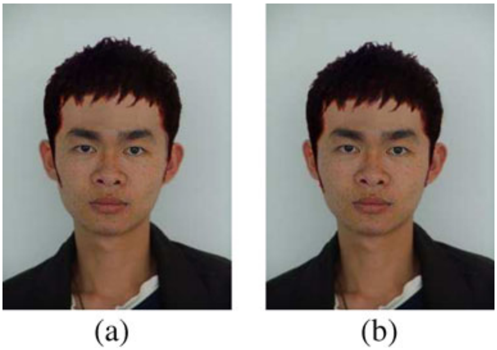


Fig. 12 **a** Automatic example. **b** Manual example

3.6 Ear synthesis

Through numerous experiments and user feedback, we have found that the synthesized results often look weird if the ears are removed. Or, if the remaining ear pixels are preserved after hair removal, the fragmented ears could be a distraction, or even sometimes make people taking more focus on the artifact near the ear region instead of the simulation result of swapping the hairstyle. Also, the users said that it is more perceptually acceptable if the fragmented ears are replaced with suitable ones. However, it is evident that like hair extraction, ear extraction is also not trivial especially when a long hairstyle is encountered. To cope with this, we manually extract the ears from some photos, as shown in Fig. 13. And for any given face, we pick one pair of the manually extracted ears which have similar luminance value to

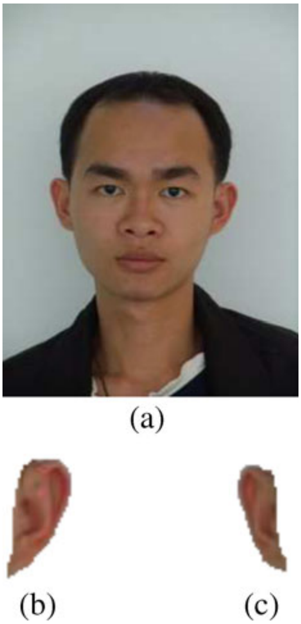


Fig. 13 The ear extraction process. **a** The original photo. **b** The extracted left ear. **c** The extracted right ear

the input face. For each pair of the manually extracted ears, we compute its distance to the input face in the L channel of the $CIELAB$ color space as the following:

$$D_{\text{ear,face}} = Z(face, left\_ear) + Z(face, right\_ear),$$
$$\text{where } Z(x, y) = \left|\frac{\mu_x - \mu_y}{\sigma_y}\right| + \left|\frac{\mu_y - \mu_x}{\sigma_x}\right| \tag{2}$$

The symbol μ computes the mean values, in L channel, of the extracted left ear, extracted right ear or the input face, while symbol σ refers to the standard deviation. $Z(x, y)$ sums up the distance, similar to the Z-value or standard score in statistics, between the mean value of the extracted ear and the distribution of the input face, and the distance between the mean value of the input face and the distribution of the extracted ear. After a pair of ears with the most similar luminance, with respect to the input face, have been chosen, the possibility of inconsistency in luminance between the ears and the face still exists. Therefore, we make a luminance adjustment on every pixel of the selected ears by simply transforming the ears' luminance distribution into the face's. The following equation shows the adjustment on one pixel:

$$ear_{\text{adjusted}} = (ear_{\text{original}} - \mu_{\text{ear}}) \cdot \left(\frac{\sigma_{\text{face}}}{\sigma_{\text{ear}}}\right) + \mu_{\text{face}} \tag{3}$$

The final stage is to synthesis the ears onto the input face. To facilitate the ensuing ear synthesis process, during our manual ear extraction process just mentioned, we also manually marked the two *contacting positions* that an ear connects to its original facial contour. Figure 14 demonstrates an example of such an alignment process. In Fig. 14a, the ears are to be aligned with a new facial contour, where we mark the connecting portions of the ears in red, and the connecting portions are the lines connecting the aforementioned *contacting positions*. On the other hand, the connecting portions on the new facial contour, marked blue in the same Figure, are determined by the following steps. First, according the bounding boxes of an ear's original facial contour and the new facial contour, the ear is first scaled in width and height. Second, we adjust an ear's position so that its boundingbox's midpoint aligns vertically with the boundingbox of the new facial contour. Third, we horizontally move an ear until its bottommost point touches the facial contour, as shown in Fig. 14a, and this also determines the lower point on one of the connecting portions on the facial contour. Fourth, the upper point on the facial contour is decided by horizontally projecting the top of the ear's boundingbox onto the facial contour. Finally, we calculate the angle between the two connecting portions, approximated

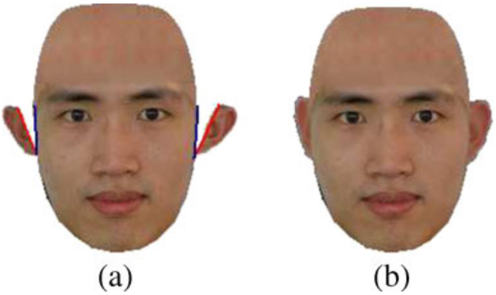


Fig. 14 The ear alignment process. **a** Before ear alignment and luminance adjustment. **b** After ear alignment and luminance adjustment

by lines and perform a rotation accordingly to finish the alignment process. Note that for the demonstration purpose, in Fig. 14a we in fact have intentionally rotated the original extracted ears from Fig. 13b and c to denote the cases where the formed angles between two connecting portions may not be small. Nevertheless, after a proper alignment process, the desired result can be achieved, as shown in Fig. 14b. In fact, the synthesis results after attaching "fake" ears have often become much more convincing than they were before, as will be shown in the Section 4.

4 Results

Our experiments are conducted on a machine with a Intel Core 2 1.86GHz CPU and 3GByte memory, running on MS Windows XP. The involved programming language is Visual C++, and the Poisson equation is solved by MATLAB 7.0. In terms of timing, it usually takes 3–5 s to derive a bare head model, and the automatic hairstyle adjusting scheme requires less than 3 s to converge on an average.

As shown in the Introduction section by the Fig. 1c, failure to remove the original hair may cause severely undesired effect. On the contrary, after proper hair removal, our system can provide a more pleasing result, as shown in Fig. 1d.

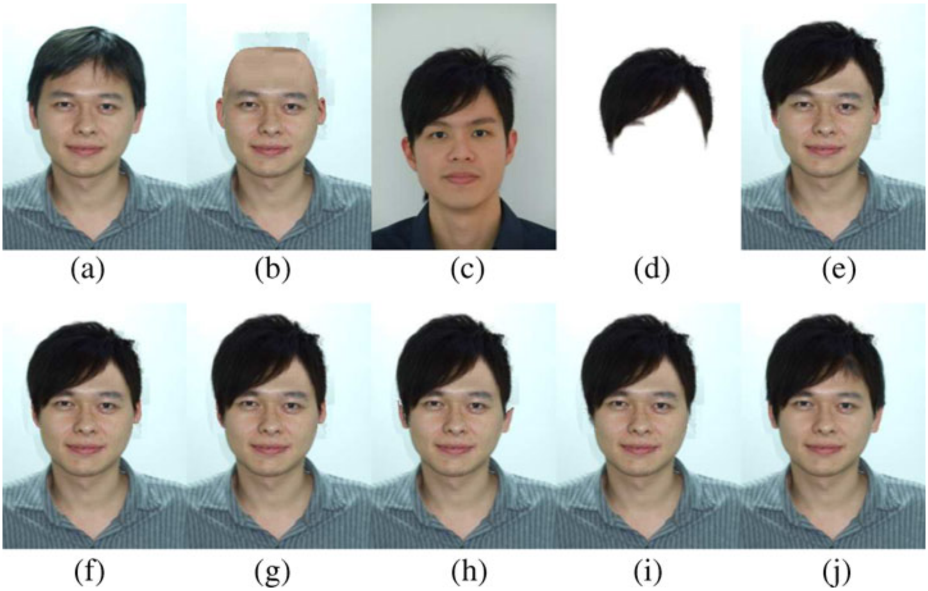


Fig. 15 An example on changing one's hairstyle into another with different algorithms. **a** Input image. **b** Bare head image generated by our system. **c** The hairstyle model. **d** The hairstyle extracted from the model. **e** Initial state of resynthesizing the hairstyle (**d**) onto the given face (**a**). **f** The re-synthesis result optimized by our system. **g** The re-synthesis result adjusted manually. **h** The re-synthesis result without dealing with the fragmented ears. **i** The re-synthesis result by removing the fragmented ears. **j** The re-synthesis result by just simply attach the new hairstyle onto the face without making any modifications on the forehead region and the ears. Images **e**–**g** will be used in our analysis and user studies in Section 5

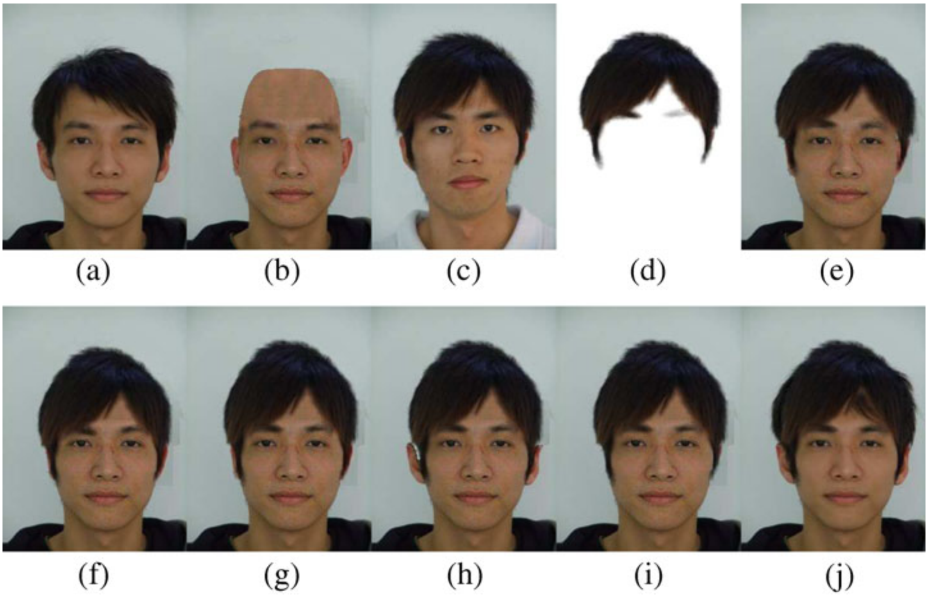


Fig. 16 The second example on changing one's hairstyle. The configuration is the same as Fig. 15

Figures 15, 16, 17, 18, 19 and 20 demonstrate the effectiveness of our system by testing many cases on face and hairstyle swapping. Note that due to various original skin colors or lighting conditions, the extracted hairstyle may not be very clean, as shown in Fig. 18, thus leading to the undesired *color bleeding* effects along the hair

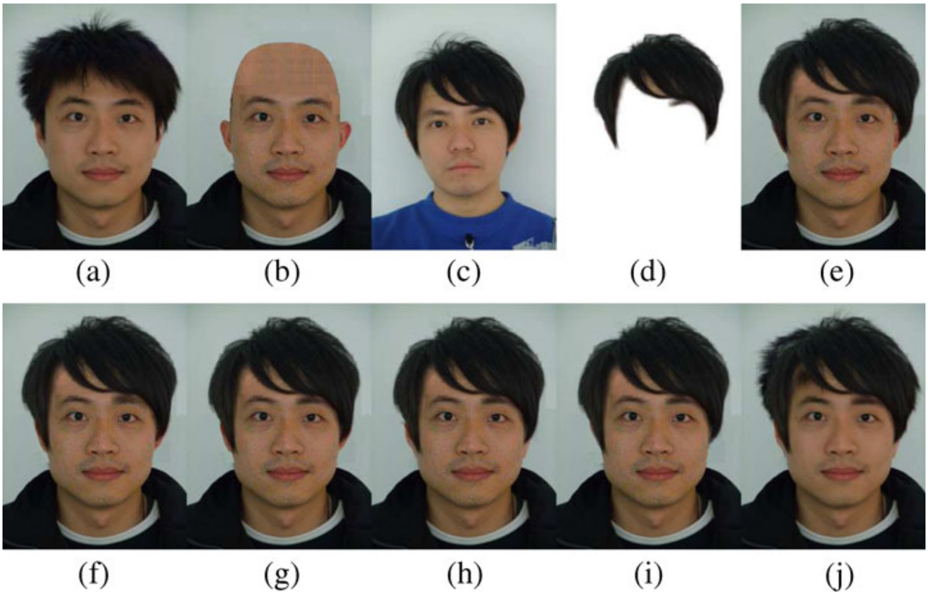


Fig. 17 The third example on changing one's hairstyle. The configuration is the same as Fig. 15

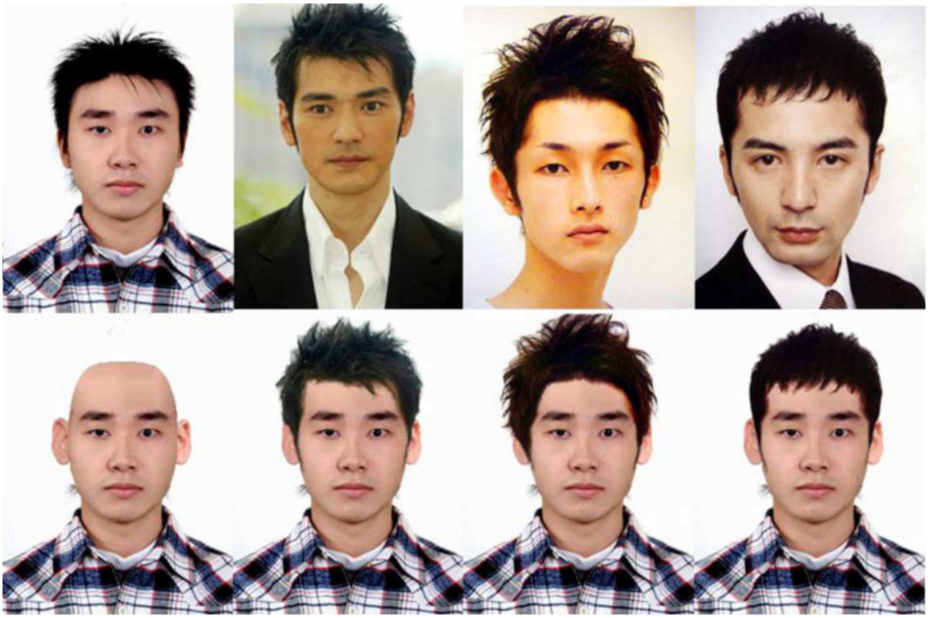


Fig. 18 More examples on changing hairstyles. The *leftmost column*, from *top* to *bottom*, shows the original input image and its corresponding *bare head* result. The other three images in the *first row* are the models whose hairstyles are extracted and then used to resynthesize with the input image. The other three images in the *second row* are the final hairstyle resynthesized results

boundary. A more uniform or controlled lighting setup would greatly help improve the results, as shown in Figs. 15–17.

Moreover, as we have mentioned previously, unlike the work proposed by Bitouk et al. [3], our results in Figs. 19 and 20 demonstrate our unique capability of altering not only the facial features, but the *entire facial contour* as well, to achieve sometimes more desirable *face-off* results. To show more concrete examples, we try to compare our results with those from Bitouk et al.'s work in Fig. 21. Figure 21a, b and c are a source photo and two target photos from Bitouk et al.'s work [3], respectively, while Fig. 21d and e are their face swapping results by replacing the source photo's face with the target photos' faces, respectively. On the other hand, Fig. 21f and g show our hairstyle swapping results, while Fig. 21h and i our face swapping results. As can be seen from these figures, our face swapping and hairstyle swapping capabilities are more like *dual* operations to each other, and we believe such pairing operations would be more useful for facial appearance alteration. However, We also have to point out that, for the cases where the involved models contain complex or ambiguous backgrounds, the hairstyle extraction process may be semi-automatic, that is, user assistance may be required. For example, In Fig. 21a, the hair and background are too similar to be automatically separated. Even though we manually generate the tri-map for the source image, some artifacts still exist in the extracted hairstyle result, thus making the synthesized images less natural. It is especially obvious in the background of Fig. 21g and in the upper right region of the left eyebrow in Fig. 21h, Also note that, in Fig. 21f and g the words *Rank 1* and *Rank 4* are occluded by the hair whereas they are not in Fig. 21d and e. This is due to fact

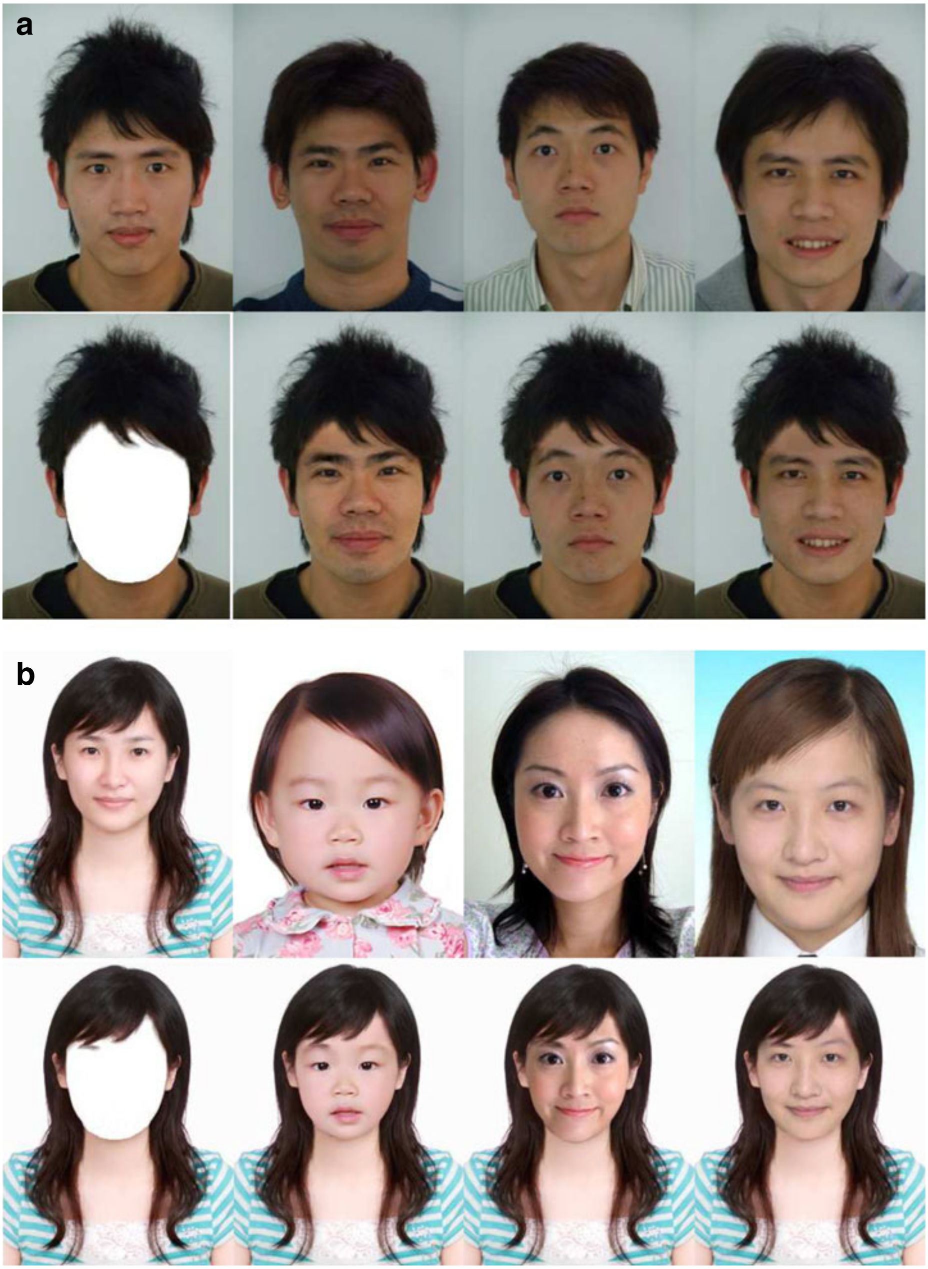


Fig. 19 Results of two sets of examples on exchanging faces. The configuration is similar to that in Fig. 18 but is for hairstyle replacement. **a** The first input set. **b** The second input set

the extracted hairstyle is from Fig. 21a and it is pasted onto the target photos to make part of the words of *Rank 1* and *Rank 4* invisible. Nevertheless, all the other hairstyle extraction results presented in this paper are done automatically.

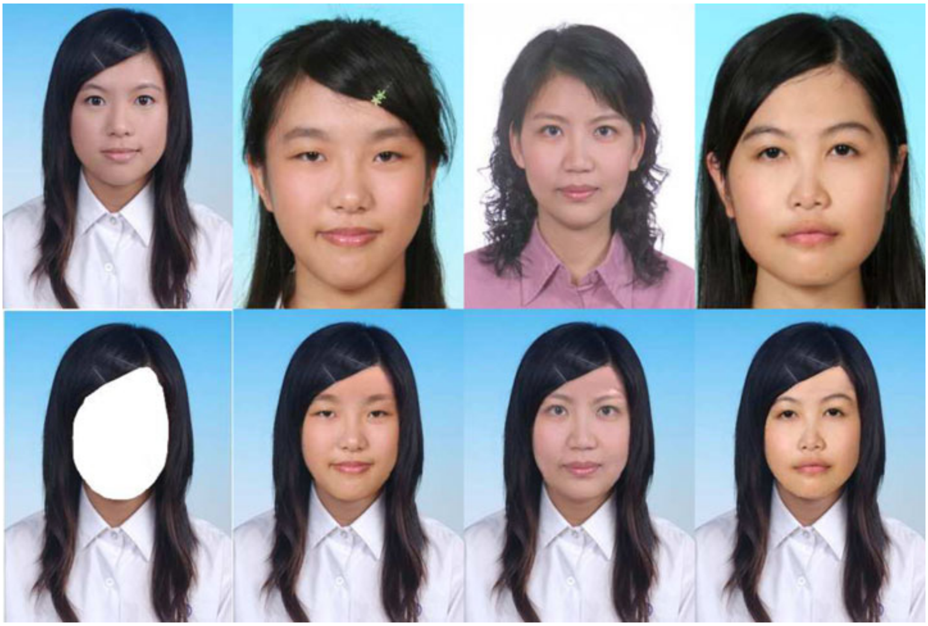


Fig. 20 The third input set are results of one more set of examples on exchanging faces. The configuration is similar to that in Fig. 18 but is for hairstyle replacement

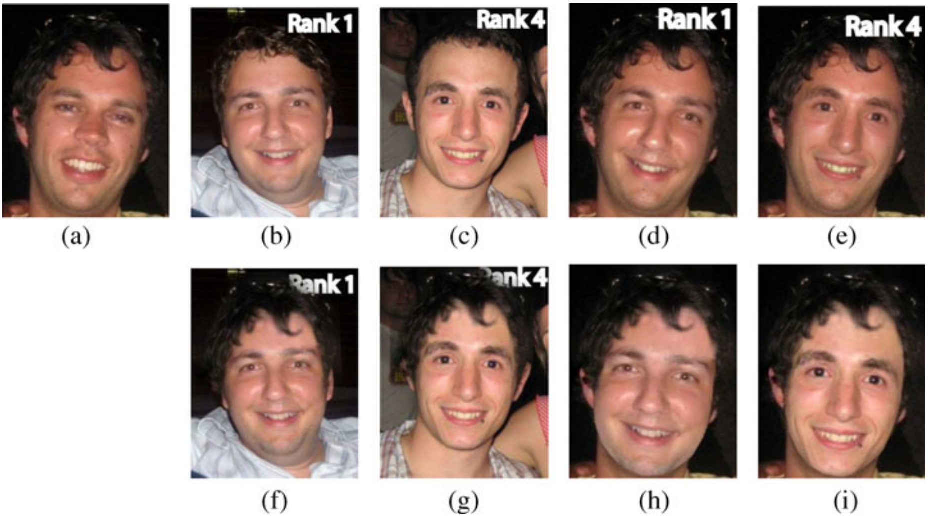


Fig. 21 The comparisons with Bitouk et al.'s work [3], while we have obtained the permission for using the figures from their paper. Our work not only is able to swap hairstyles, but also capable of exchanging faces. And we believe that instead of only replacing the facial features, swapping faces while keeping the entire facial shape at the same time sometimes may achieve more desirable results. **a** Source photo. **b** Bitouk et al.'s 1st target photo. **c** Bitouk et al.'s 2nd target photo. **d** Bitouk et al.'s first face swap. **e** Bitouk et al.'s second face swap **f** Our first hairstyle swap. **g** Our 2nd hairstyle swap. **h** Our 1st face swap. **i** Our 2nd face swap

As a final note, currently we do not compare our work with Mohammed et al.'s work [19] for the following two reasons. First, their work exchanges part of the facial features, while ours exchanges the entire head. Second, their algorithm arbitrarily chooses sample patches for the skin texture synthesis, so that the results may not be controllable.

5 Discussion

As one of the most important contributions of this work is to recover the skin of the revealed forehead region, so that the results might be more desirable after putting on new hairstyles. We wish to measure how this feature is being needed. In our current implementation, we synthesize a skin patch for replacing the entire forehead region no matter which pixels in the forehead region have been occluded by hair. After wearing on a newly selected hairstyle for the input image, we can divide the forehead region into three parts. The first part consists of the pixels occluded by the new hairstyle. Since these pixels do not affect the quality of the final result, we simply ignore them. The second part refers to the pixels that are considered hair pixels in the input image but are not covered by the new hairstyle. We should definitely take special care on theses pixels, or they may induce prominent artifact to the final result. Such examples can be seen in the third column of Fig. 1. Finally, the rest of the pixels are those who are considered skin pixels in the input image and are not occluded by either the original hair or the new hairstyle. We then analyze the statistical data from (f), the automatically optimized results, and (g), the manually manipulated results, shown in Figs. 15–17. The collected numbers are listed in Table 1. To ease the discussion, we use symbol A as the total number of revealed forehead pixels, which combines the pixels in the second part and the third part of the forehead region as we have mentioned above, and symbol B for the number of pixels in the second part of forehead region. Therefore B/A means the percentage of the hair pixels of the input image, which may induce serious artifact if we do not take care of, in the revealed forehead region. It is evident that the values of B and B/A could be referred as how much the feature of recovering the forehead region is being needed. As can be seen in Table 1, the percentages of the necessarily recovered pixels in our examples range from 60% to 80% and the number of the necessarily recovered pixels range

Table 1 The statistical data for analyzing how much the feature of recovering the pixels in the forehead region is being needed

	Number of revealed forehead pixels after putting on new hairstyle (A)	Hair pixels in the input image but are not covered by the new hairstyle (B)	Percentage (%) of pixels that may induce obvious artifact (B/A)
Figure 15f	688	522	76
Figure 15g	820	614	75
Figure 16f	418	319	76
Figure 16g	556	338	61
Figure 17f	475	381	80
Figure 17g	453	359	79

from 300 to 600, which implies the recovery of the revealed region is pretty much being needed. Furthermore, notice that, the result should has been evaluated case by case. For example, if a man with a hairstyle, which covers almost the whole forehead region. And he would like to get a haircut with a closely cropped hairstyle. In this situation, the functionality of recovering the revealed forehead region is definitely desired, because we can hardly neglect the uncorrected hair pixels after wearing on the new hairstyle.

We have also conducted a user study to verify that our proposed method indeed reduces the user assistance during the hairstyle swapping process. We invite 12 subjects to participate this study. And there are three testing sets, six images in total, in the user study, which are images (e) and (f) of Figs. 15–17. For each testing set, image (e) refers to the image that the hairstyle is initially located by the reference points, while image (f) represents the automatic hairstyle re-synthesis result. We ask the participants to check the images to see whether he/she thought that the hairstyles are being positioned at the right place or not. He/she can use our system to scale the hairstyles to the right scale or move to the right location. And then, we record the elapsed seconds that the users take to adjust the results. Table 2 shows the records where zeros indicate the cases where the adjustments are considered unnecessary. According to the user study, it is apparent that our proposed approach reduces user assistance in editing a hairstyle into an appropriate one. As the timings shown in Table 2, the numbers in the right column of each testing set, where we simply match the correspondences between the reference points of an extracted hairstyle and the reference points of a input face, it takes more than 15 s in average for the users to make their adjustments. And after our proposed automation mechanism, it saves nearly half of the time for adjustment in average. In the case of Fig. 15, the improvement is even more significant, while half of the subjects thought that the final results yielded from our system need no further adjustment. An interesting phenomenon is also observed in this user study as well. Once a user decides to

Table 2 The timing results, measured in seconds, for the first user study

User	Fig. 15e-auto	Fig. 15f-manual	Fig. 16e-auto	Fig. 16f-manual	Fig. 17e-auto	Fig. 17f-manual
1	0	18	12	16	13	30
2	0	5	0	7	15	15
3	0	17	12	14	11	17
4	5	14	8	8	5	16
5	0	26	0	13	12	20
6	12	6	7	8	4	7
7	15	20	15	25	16	19
8	0	6	0	12	0	8
9	13	23	23	18	15	24
10	0	5	8	8	8	18
11	17	27	19	38	13	32
12	7	19	0	16	11	14
Average	5.75	15.5	8.67	15.25	10.25	18.33

In each testing set, e.g. Figs. 15, 16 or 17, the *left column* records the adjusting times for (e), the cases of the automatically matched hairstyles, and the *right column* records the adjusting times for (f), the cases of the initially located hairstyles. The numbers in the *last row* show the average adjusting time for each testing set move a hairstyle's position or to change a hairstyle's size, there are oftentimes two notable outcomes. First, a user may just try to probe for a better configuration for the hairstyle, but suddenly found that the original setup is quite acceptable for him/her. Thus, he/she would stop the adjusting process immediately. Such cases are the ones that the timings are under 5 s (included) in Table 2. Second, a user may end up spending lots of time on fine-tuning, even though the change is not easily perceivable. Sometimes, the fine-tuning process could take up to more than 10 s.

Knowing that recovering the revealed forehead region a being needed feature and our proposed mechanism indeed reduces the user assistances to some extent. We would like to further measure the quality of our simulated results, i.e. to compare our results with the ones that do not take good care of the revealing hair pixels, and to evaluate the users' acceptance level of our hairstyle re-synthesis results. As mentioned in Section 3.6, distractions may happen if the regions near the ears are not handled properly. Hence, in this evaluation, we include four kinds of simulated results for each hairstyle re-synthesis comparison, as shown from (g) to (j) in Figs. 15–17. There are 46 subjects invited to provide their opinions, and three testing sets of images are demonstrated to each participant. Each testing set contains an original image, a hairstyle image that is to be exchanged and four hairstyle re-synthesis outcomes, which are (g) the result with fake ears and synthesized forehead, (h) the result with synthesized forehead and preservation of the original but fragmented ears, (i) the result with synthesized forehead but removal of the ears, and (j) the result without making any modifications to the original image. The four hairstyle re-synthesis outcomes are shown to the participants in random orders. And the following introductions are given: *We have provided four methods to achieve the goal of replacing one's current hairstyle with another in a single image. However, there exist different kinds of artifacts in each of them due to various reasons. We want to know which of the artifact makes the least impact on distracting people from perceiving the result of a hairstyle replacement. Or, regarding them as simulated hairstyle swapping results, how do the artifacts affect your imagination on the hairstyle exchanging results.* Then, the users are inquired to give every result a score of a seven-level (seven-point) Likert scale, while lower points mean bigger distractions or lower acceptance levels. The scorings are collected and shown in Fig. 22, as different colored bars represent

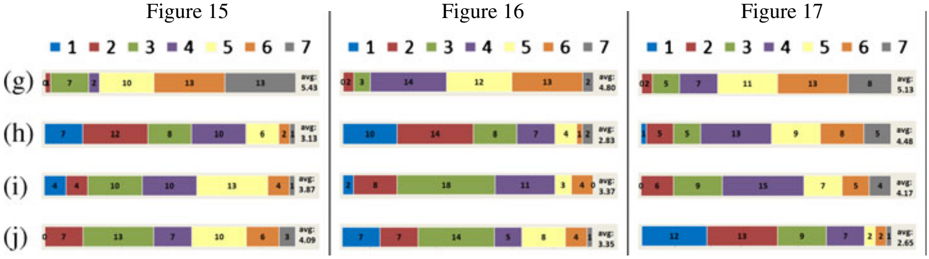


Fig. 22 The collected scores of the acceptance levels that the users gave to the four different hairstyle re-synthesis results, while higher scores can be referred as higher acceptance levels. Row **g** presents the scores given to the results with fake ears and synthesized forehead. Row **h** presents the scores given to the results with synthesized forehead and preservation of the original but fragmented ears. Row **i** presents the scores given to the results with synthesized forehead but removal of the ears. Row **j** presents the scores given to the results without making any modifications to the original images

different scores that the users gave. It can be easily observed that the scorings of our results, with fake ears and synthesized forehead, shown in row (g), apparently outperform the others. In each testing set, its averaged rating is significantly higher than any of the others. And if we set 5 point as a dividing boundary, it has been given 5 point or above by more than half of the participants in each testing set, while the results of the other three methods do not perform so well under this standard. Moreover, we cannot comment that which of the other three methods is better since their scoring distributions are unstable through the three testing sets. For example, in the first testing set, i.e. Fig. 15, (j) is considered having higher quality than (h), but it turns out to be the opposite situation in the third testing set, i.e. Fig. 17.

6 Conclusions and future work

We have proposed methods to automatically perform hair removal, hairstyle extraction, and face/hairstyle swapping for 2D images. Our system has its strengths on successfully fitting a reasonable forehead shape and on re-synthesizing the skin texture for the revealed forehead after hair removal. A hairstyle extraction algorithm is developed, and a user-friendly head/hairstyle adjusting scheme is proposed to facilitate some possible manipulations for hairstyle cloning. An automatic adjusting scheme is also introduced to further increase the efficiency for the cloning on different hairstyles. Some analysis and user studies have been conducted, and the statistical numbers of the analysis and the user studies demonstrate the effectiveness of our system on reducing the user assistances in hairstyle adjustment as well as providing more convincing hairstyle re-synthesis results.

There are, however, several limitations in our current system. First, we are not able to recover the texture or shape of the clothes or human neck which are occluded by the hair. An example is shown in Fig. 23a. Second, our skin synthesis algorithm is

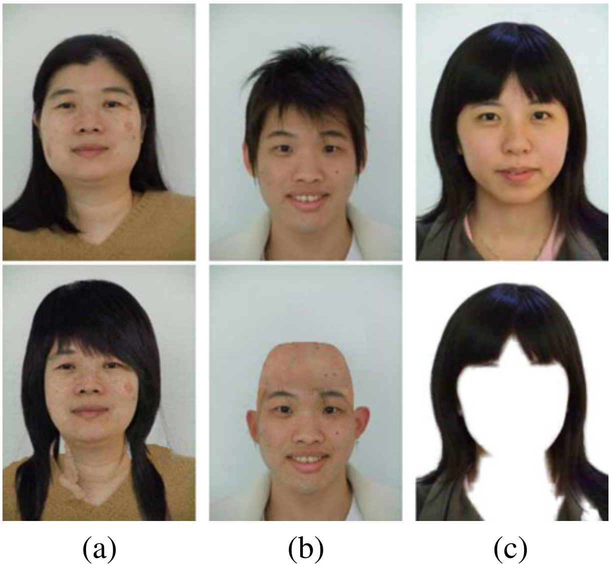


Fig. 23 Examples of erroneous results yielded from our system. The images in the *first row* are input images, while those in the *second row* are results. In **a**, parts of the clothes and the neck are occluded by hair, which cannot be handled properly. The non-smoothness of the facial skin causes a non-smooth forehead skin patch, as shown in **b**. **c** shows an example where the color of the clothes is too close to that of the hair, thus yielding an erroneous segmentation result

based on the patch in the lower known facial region, so the smoothness of the lower known facial region affects the skin synthesis result. Also, for the cases where the hair is long enough to occlude the eyebrows or even the eyes cannot be handled properly in this paper, as shown in Fig. 23b. Third, if the background contains complex color patterns or the hair color is close to that of the clothes, segmentation errors may occur, thus leading to erroneous results as can be seen in Fig. 23c.

In the future, in addition to addressing aforementioned limitations, there are several directions that we want to pursue. First, more fine-grained hairstyle editing functionalities, other than just scaling and translation, are desired. A good example of such is the approach proposed by Wang et al. [33], but it is mainly used in 3D cases. Second, we currently have not performed proper ear extractions but resort to a compromised version by combining results with "fake" ears. The results look reasonably well but still not perfect. Finally, the aesthetic factor may be put into concern. For example, given a face, a system could automatically select a hairstyle from the database that forms the best match, in terms of attractiveness.

References

1. Active Shape Models with Stasm (2008) http://www.milbo.users.sonic.net/stasm/. Accessed 19 Jan 2009
2. Bertalmio M, Sapiro G, Caselles V, Ballester C (2000) Image inpainting. In: SIGGRAPH '2000
3. Bitouk D, Kumar N, Dhillon S, Belhumeur P, Nayar SK (2008) Face swapping: automatically replacing faces in photographs. ACM Trans Graph (Also Proc. of ACM SIGGRAPH '2008) 27(3):1–8
4. Blancato V (1988) Method and apparatus for displaying hairstyles. US Patent 4731743
5. Blanz V, Scherbaum K, Vetter T, Seidel H (2004) Exchanging faces in images. In: Eurographics '2004
6. Chuang YY, Curless B, Salesin DH, Szeliski R (2001) A Bayesian approach to digital matting. In: Proceedings of CVPR 2001, pp 264–271
7. Cootes TF, Taylor CJ (2001) Statistical models of appearance for medical image analysis and computer vision. In: SPIE medical imaging
8. Cootes TF, Taylor CJ, Cooper DH, Graham J (1995) Active shape models—their training and application. Comput Vis Image Underst 61(1):38–59
9. Doi M, Tominaga S (2006) Image analysis and synthesis of skin color textures by wavelet transform. In: Image analysis and interpretation '2006, pp 193–197
10. Drori I, Cohen-Or D, Yeshurun H (2003) Fragment-based image completion. In: SIGGRAPH '2003, pp 303–312
11. Efros AA, Freeman WT (2001) Image quilting for texture synthesis and transfer. In: SIGGRAPH '2001, pp 341–346
12. Efros AA, Leung T (1999) Texture synthesis by non-parametric sampling. In: International conference on computer vision, pp 1033–1038
13. Gonzalez RC, Woods RE (2002) Digital image processing, 2nd edn. Prentice-Hall, New York
14. He L, Chao Y, Suzuki K (2008) A run-based two-scan labeling algorithm. IEEE Trans Image Process 17(5):749–756
15. Hsu RL, Abdel-Mottaleb M, Jain AK (2002) Face detection in color images. IEEE Trans Pattern Anal Mach Intell 24(5):696–706
16. Kjeldsen R, Kender J (1996) Finding skin in color images. In: FG' 96 (2nd international conference on automatic face and gesture recognition)
17. Levin A, Lischinski D, Weiss Y (2008) A closed-form solution to natural image matting. IEEE Trans Pattern Anal Mach Intell 30(2):228–242
18. MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley symposium on mathematical statistics and probability, pp 281–297
19. Mohammed U, Prince S, Kautz J (2009) Visio-lization generating novel facial images. In: SIGGRAPH '2009

20. Mortensen EN, Barrett WA (1995) Intelligent scissors for image composition. In: SIGGRAPH ‘1995, pp 191–198
21. Nainia FB, Mossb JP, Gillc DS (2006) The enigma of facial beauty: esthetics, proportions, deformity, and controversy. Am J Orthod Dentofac Orthop 130(3):277–282
22. Oliveira MM, Bowen B, McKenna R, Chang Y (2001) Fast digital image inpainting. In: VIIP (international conference on Visualization, Imaging and Image Processing) ‘2001
23. Perez P, Gangnet M, Blake A (2003) Poisson image editing. In: SIGGRAPH ‘2003, pp 313–318
24. Perret DI, May KA, Yoshikawa S (1994) Facial shape and judgments of female attractiveness. Nature 368(6468):239–242
25. Rowley HA, Baluja S, Kanade T (1998) Neural network-based face detection. IEEE Trans Pattern Anal Mach Intell 20(1):23–38
26. Rowley HA, Baluja S, Kanade T (1998) Rotation invariant neural network-based face detection. In: CVPR ‘1998, pp 38–44
27. Smith AR, Blinn JF (1996) Blue screen matting. In: SIGGRAPH ‘1996, pp 259–268
28. Sun J, Jia J, Tang CK, Shum HY (2004) Poisson matting. ACM Trans Graph 23(3):315–321
29. Sun J, Yuan L, Jia J, Shum H (2005) Image completion with structural propagation. In: SIGGRAPH ‘2005, pp 861–868
30. Tsumura N, Ojima N, Sato K, Shimizu H, Nabeshima H, Akazaki S, Hori K, Miyake Y (2003) Image-based skin color and texture analysis/synthesis by extracting hemoglobin and melanin information in the skin. In: Proceedings of the 30th annual conference on computer graphics and interactive techniques SIGGRAPH 2003
31. Viola P (2004) Robust real-time face detection. Int J Comput Vis 57(2):137–154
32. Wang J, Agrawala M, Cohen M (2007) Soft scissors: an interactive tool for realtime high quality matting. In: SIGGRAPH ‘2007
33. Wang L, Yu Y, Zhou K, Guo B (2009) Example-based hair geometry synthesis. In: SIGGRAPH ‘2009
34. Wei L, Levoy M (2000) Fast texture synthesis using tree-structured vector quantization. In: SIGGRAPH ‘2000, pp 479–488
35. Yang MH, Ahuja N (1998) Detecting human faces in color images. In: International conference on image processing, pp 127–130

Jia-Kai Chou received his Bachelor degree and Master degree in information management at National Taiwan University of Science and Technology in 2007 and 2009, respectively. He is now a Ph.D. student in the department of information management at National Taiwan University of Science and Technology. He is interested in computer graphics, multimedia systems, and information visualization.

Chuan-Kai Yang received his Ph.D. degree in computer science from Stony Brook University, USA, in 2002, and his M.S. and B.S. degree in computer science and in mathematics from National Taiwan University in 1993 and 1991, respectively. He is currently an Associate Processor of the information management department at National Taiwan University of Science and Technology. His research interests include computer graphics, scientific visualization, multimedia systems, and computational geometry.