Face-off: Automatic Alteration of Facial Features

Jia-Kai Chou and Chuan-Kai Yang and Sing-Dong Gong
Department of Information Management
National Taiwan University of Science and Technology
No. 43, Sec. 4, Keelung Road
Taipei, 106, Taiwan, ROC
A9409004@mail.ntust.edu.tw,ckyang@cs.ntust.edu.tw,hgznrn@uj.com.tw

ABSTRACT

Pursuing or maintaining beautifulness nowadays has become a trend in modern society, especially among the celebrity community. In some cases, one may choose to adopt drastic procedures to alter his or her facial or body features to achieve the desired beauty, thus the blossom of industry on cosmetic plastic surgeries. In addition, people whose faces got damaged due to accidental burns or wounds may also find these surgeries necessary. However, as performing the related surgeries are still considered intrusive and costly, it is better to "preview" the result before a surgery is actually carried out. As many believe that facial appearance matters most, we have developed a system that allows a user to input a photo and changes the associated individual facial feature in an automatic and user-friendly manner. Overall speaking, our system makes contributions in the following four aspects. First, our system not only offers the previewing functionality, but also allows users to interactively fine-tune the desired results, thus making it a useful companion tool for facial cosmetic surgeries. Second, instead of exchanging the overall look of a face, as being done by some existing approaches, our system offers much finer granularity by allowing each and every facial feature to be changed individually and independently, thus achieving higher face-off flexibility. Third, while existing tools generally entail manual effort to locate or align facial features, our system, through the help of *Active Shape Model* or ASM for short, characterized by a scheme of *automatic feature extraction*, eliminates most of the needs of user assistance. Finally, for convenience, we have constructed a database of facial features to facilitate the facial feature alteration process. To justify our claims, we have rendered results and compared them with those from existing approaches to demonstrate the effectiveness of our system. We have also conducted a user study to further confirm the usefulness of such a system.

Categories and Subject Descriptors

I.3.4 [**Computer Graphics**]: Graphics Utilities; I.4.8 [**Image Processing And Computer Vision**]: Scene Analysis

Keywords

Poisson Image Editing, Feature Extraction, Active Shape Model, Facial Feature Alteration

1. INTRODUCTION

The rapid advances on modern technology has turned many impossible things into reality, including the alteration of body or facial appearance of a person. A person may request such a change for enhancing his/her beautifulness or recovering from a damage due to some accident. Though viewed as minor surgeries now, the involved procedures for appearance alteration are still considered invasive and costly, therefore a vivid preview of the result would be very helpful. Empirical studies suggest that facial appearance in general significantly influences the first impression of a person, we therefore employ all our endeavor to develop a system that allows a user to rapidly and conveniently foresee what he/she will be looked like if the desired procedure is carried out. For convenience, hereafter we will refer the process of altering one's facial look as *face-off*.

In general, there are some existing tools/approaches that might help the preview process, at least to a certain degree. For example, the work of *interactive digital photomontage* by Agarwala *et al.* [2] and the work of *drag-and-drop pasting* by Jia *et al.* [13]; however, the involved manual effort, i.e., to identify the facial feature contour from the source image, and to locate the corresponding region in the target image to be changed, still remains, thus making it a relatively tedious and inconvenient process. In addition, one subtle and un-addressed issue is the dealing with cases where a source facial feature is smaller than the target one, which makes a direct application of previous approaches inapplicable. Moreover, it is quite possible that a user might have to try numerous combinations before he/she could make the final decision, and therefore it justifies the need of an efficient and friendly user interface.

The main contribution of this paper is fourfold. First, based on the technique of *Active Shape Model* or ASM for short [9, 10], we have devised an automatic facial feature identification scheme that is especially tailored to the need of the proposed face-off process, thus eliminating the need of any user involvement. Second, unlike most existing approaches to allow only a face change in its entirety, our system provides finer granularity by permitting the change of each and every facial feature, and a result, a higher degree of face-off flexibility can be achieved. Third, we have managed to deal with nearly all the cases where source and target facial features may present a huge variety in terms of size, shape and color, etc. Finally, to facilitate the whole face-off process, we have built a database containing a variety of facial features to choose from, thus greatly simplifying the involved procedures. By coupling our proposed *au-*

tomatic facial feature detection scheme with the well-known *Poisson image editing* [23] technique, we have built a system that fulfills our claims. We have rendered results and compared them with those from existing approaches when applicable to prove the usefulness of our system. A user study is also conducted to further complete this study. In addition to face-off, our developed system could be adopted in other applications as well. For example, many people now choose *avatars* to represent themselves, either for concealing their identities or appearance imperfections, in the virtual world communications with others. In this regard, our system is thus an ideal tool for composing a desired image. Another example is for the police to compose a suspect photo, as was also done by [17], and our system is better equipped to produce more pleasing results.

The rest of the paper is organized as follows. Section 2 reviews related literature regarding this work. Section 3 describes the face-off approach that we adopt to achieve the desired purpose. Section 4 details the algorithm that we employ to automatically and rapidly locate the facial features in a given photo. Section 5 demonstrates the rendered results of this work to proof the effectiveness of our approach. Section 6 concludes our work, discusses its potential limitations, and hints for its possible future extensions.

2. RELATED WORK

To extract the source feature and paint it on the target face, three techniques will be involved. First, a seamless *image stitching* is required so that the target face does not look obtrusive. Second, when a smaller source feature is to replace a larger feature on the target face, an *image inpainting* is needed to remove the underlying larger feature before the source feature being placed to avoid potential artifacts. Finally, to eliminate the manual effort for locating and aligning source and target features, an automatic *facial feature identification* scheme is better provided. As a result, we review the related work for each of the three mentioned techniques in turn.

Regarding image stitching, there are basically three approaches. The first and presumably the most naive approach is to smooth the transition or perform blending between two images, such as the technique of *Feathering* or *alpha blending* in Peleg's work [22], the method of *pyramid blending* in Adelson *et al*.'s work [1], and the addition of a smooth function to make consistent the difference along the joining seam of two images, as was done by Uyttendaele *et al*.'s work [25]. The second approach is to find the best *cut* or *seam* to merge two images, as the work by Kwatra *et al.* [15] for stitching images or synthesizing textures, which is based on the *graphcut* algorithm proposed by Boykov *et al.* [8]. The third approach involves manipulation in the *gradient domain*, that is, image stitching is achieved by modifying the gradients in the source image, under the boundary constraint determined by the seam where two images are to be combined, as in the work by Perez *et al.* [23], and Levin *et al.* [17]. We opt for Poisson image editing [23] as it provides satisfactory results, both in quality and performance.

There are also a number of image inpainting algorithms available. Bertalmio *et al.* proposed to perform the inpainting task by solving a *partial differential equation* or PDE for short, and in terms of implementation they applied iterative diffusion to propagate from the border of the region to be inpainted towards the interior [3]. Though such a method could provide reasonable results, it is very computationally expensive. Oliveira *et al.* later addressed this issue by devising a simpler *convolution* algorithm to significantly speed up the whole inpainting process [21]. The idea of *pixel-based texture synthesis*, originally proposed by Efros *et al.* [11] and later improved by Wei *et al.* [27], was adopted by Bornard *et al.* for image inpainting [7]. Generally speaking, in their approaches, every pixel to be inpainted gets the color of the most similar pixel belonging to the un-inpainted area, where the similarity is defined based on pixel's neighborhood statistics. Sun *et al.* proposed to first distinguish structures and textures from an image. Structures are handled with user assistance and *dynamic programming*, while the rest with a *patch-based texture synthesis* technique, which copies and stitches patches from the original textures to complete an image [24]. As many of these inpainting algorithms involve non-negligible searching or iteration overhead, we circumvent the case where inpainting is needed by using Poisson image editing in a clever way, as will be described in Section 3.

For facial feature identification, a straightforward way is to manually select the desired features, or through the "semi-automatic" image segmentation tools such as *intelligent scissor* by Mortensen *et al.* [20] or *soft scissor* by Wang *et al.* [26]. However, as we claimed in the Introduction Section, we want to automate the procedures of feature segmentation, and therefore the techniques for automatic extraction of facial features become relevant. According to the survey done by Yang *et al.* [29], there are basically *feature-based*, *knowledge-based*, *template-based*, and *color-based approaches*, where the last one, also the one most similar to our approach, is to perform the judgment based on colors, which could be RGB, normalized RGB, HSV, and YC_bC_r, etc. Kjeldsen *et al.* applied the HSV color system to extract human skin from the background of an image [14], and such an idea is also adopted in our current implementation. By assuming that a face usually appears in the middle of a photo, Lin *et al.* made use of the symmetry of facial features and their relative sizes and locations for feature extraction. The whole process was later sped up by employing a genetic algorithm. Gu *et al.* applied the SUSAN (Smallest Univalue Segment Assimilating Nucleus) algorithm to identify edge and corner points from an input photo, and then the feature points are located [12]. In this work, we borrow the idea from [14] for facial skin detection. Cootes et al. proposed the *Active Shape Model*, or ASM for short, to extract an object contour from an input image [9]. 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 [10]. Based on ASM, we further devise our own facial feature detection algorithm to fit the needs for facial feature alteration. Our new algorithm not only suits our need for the ensuing face-off process, but also is much simpler and thus efficient to provide interactive response.

In terms of interface, we think that *drag-and-drop pasting* by Jia *et al.* [13] provides a very intuitive way of manipulation for image editing; however, to make things even simpler and more automatic, our system offers a *choose-and-adjust pasting* interface. That is, a user just needs to select from a given pool of facial features with a mouse click, and then the picked feature will be placed on the target face in a seamless fashion. He/She can also drag the bounding box of the source feature to further adjust its size, location, and orientation, should he/she be not satisfied with the result. Lalonde *et al.*'s *photo clip art* [16] provided a database of objects segmented from real images to facilitate the composition of authentic scenes. Our system is similar in this regard, except that the involved objects are facial features.

So far perhaps the most similar work to ours, Bitouk *et al.* [4] performed *face swapping* by matching the input image with the ones in the database in terms of appearance and pose, adjusting automatically the related parameters of the most matched ones from the database, and finally replaced the input face with the candidate faces in a ranked fashion. While our system lacks such an automatic pose adjustment functionality, its ability to selectively replace individual facial feature is comparatively advantageous and flexible in terms of practical applications.

To sum up, facial appearance alteration has been tried at least on the following work,[2] by Agarwala *et al.*, [17] by Levin *et al.*, [23] by Perez *et al.*, [5] by Blanz *et al.*, [18, 19] by Leyvand *et al.*, and [4] by Bitouk *et al.*. Nevertheless, it should be fair to say that our system addresses this issue in a more systematical way, whereas explicit manual effort is generally required in aforementioned approaches.

3. FACE-OFF WITH POISSON IMAGE EDITING

In this section, we describe how a face-off is achieved through the use of Poisson image editing in great details. We start by giving a system overview. A brief introduction on Poisson image editing follows, and then we show how it is applied on the proposed face-off process. Next we present some of our system interfaces, and discuss several implementation-related issues.

3.1 System Overview

Figure 1 depicts the system overview of this work. A user first loads a facial image that is to be altered, and our system then automatically detects the regions of facial features. The next step is to select the feature (i.e., eyes, nose, and mouth) to be modified, where the feature could be either from the *feature database*, or loaded again by the user. Once the input image and source feature are ready, face-off could be initiated. If the user is not satisfied with the result, he/she could adjust the location, scale, and orientation of the source feature, as will be explained in Section 3.3, and perform the face-off again, until a desired result is obtained; otherwise he/she could load another source feature to re-run the whole process. The same facial feature detection scheme can also be applied to construct our feature database, where we automatically extract the facial features from each input image, and this database could be queried to supply desired features with a wide variety.

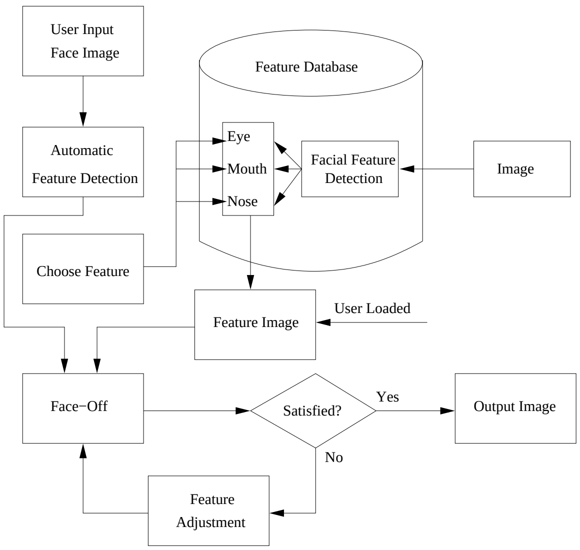


Figure 1: System overview.

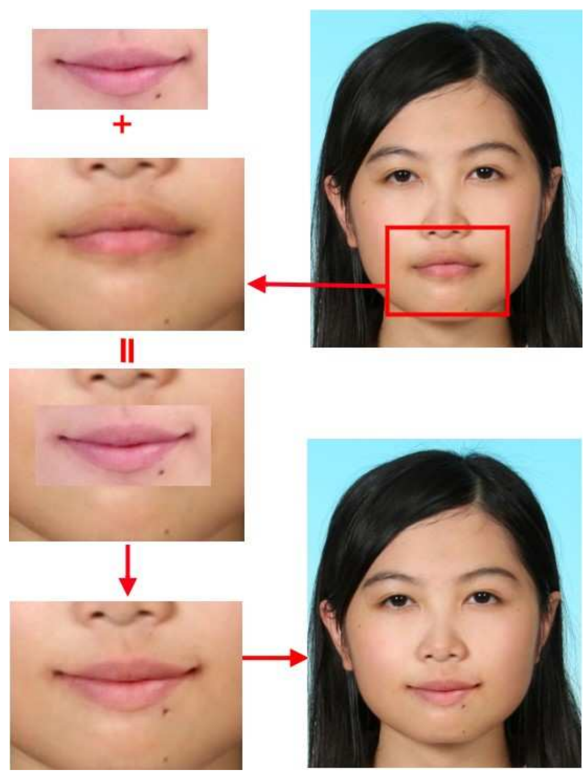
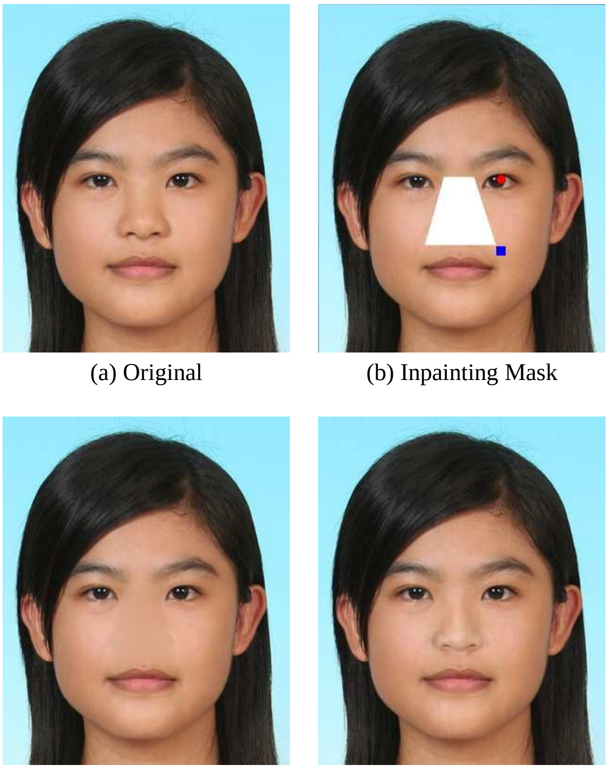


Figure 2: Example of Poisson image editing.

3.2 Poisson Image Editing

We adopt the idea of *Poisson image editing* by Perez *et al.* [23] to insert source features onto the target image. Note that involved Poisson equation can be solved by the *Gauss-Seidel* iteration with *successive over-relaxation*, and for a typical size of Ω, it normally takes less than one second to find the solution. This is also why even for the job of inpainting, to be described next, we also resort to Poisson image editing to speed up the whole face-off process. Figure 2 demonstrates the Poisson image editing process by an example.

However, as shown in Figure 3, it is possible that sometimes the feature to be painted (the nose shown in Figure 3(d)) on the target face is smaller than the original feature (the nose shown in Figure 3(a)), and a direct application of Poisson image editing will lead to an erroneous result. Figure 3(b),(c) demonstrate how we



(a) Original (b) Inpainting Mask

(c) Poisson Inpainting (d) Final Result

Figure 3: The approach that we used for replacing a larger feature with a smaller one.

achieve the inpainting via Poisson image editing, where a white image mask is first applied to alter the underlying image, before the real feature to be added.

3.3 System Interface

As mentioned at the beginning of this Section, a user first loads an input photo to perform the face-off task. Once the loading is done, the user could then select *Face Detection On* to automatically identify the facial features, as shown in Figure 4. *Face Detection Off* is used to turn off the display of the markings of the facial features, should these markings become distracting. The next step is to select a feature to alter. Figure 4 also demonstrates how a user could select a feature type, i.e., left eye, right eye, nose, and mouth, and then select a desired feature of that type with a simple mouse click. Note that the corresponding face, from which the selected feature was extracted, is also displayed for user's reference. Alternatively, a user could load a feature from an external file, as shown in Figure 5.

Once the input photo and feature to alter are ready, *Face Off* can be clicked and performed, as shown in Figure 4. However, there are cases where either the feature detection is not perfect, or the loaded feature is not placed properly, thus leading to undesired results. To address this, our system offers a fine-tune mechanism that allows a user to adjust the position, scale, and orientation of the selected feature, before/after the face-off is performed, thus making the face-off and adjustment altogether an iterative process, until the desired result is achieved. On the other hand, even when our system's automatic feature placement is already perfect, users can also apply such a fine-tune mechanism to intentionally create

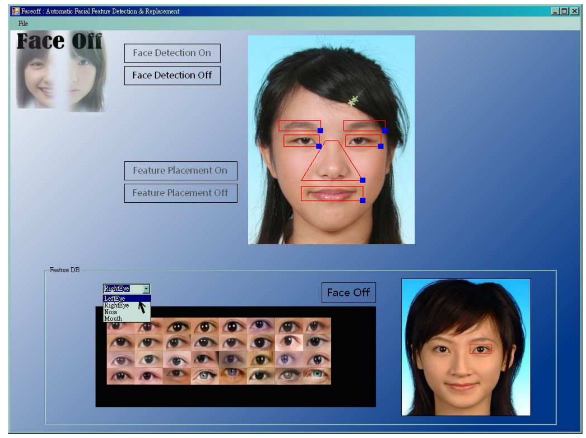


Figure 4: Example of loading a feature from system's feature database.

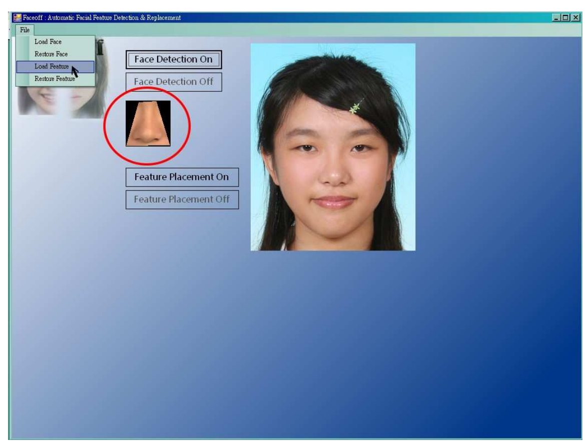


Figure 5: Example of loading a feature from an external file.

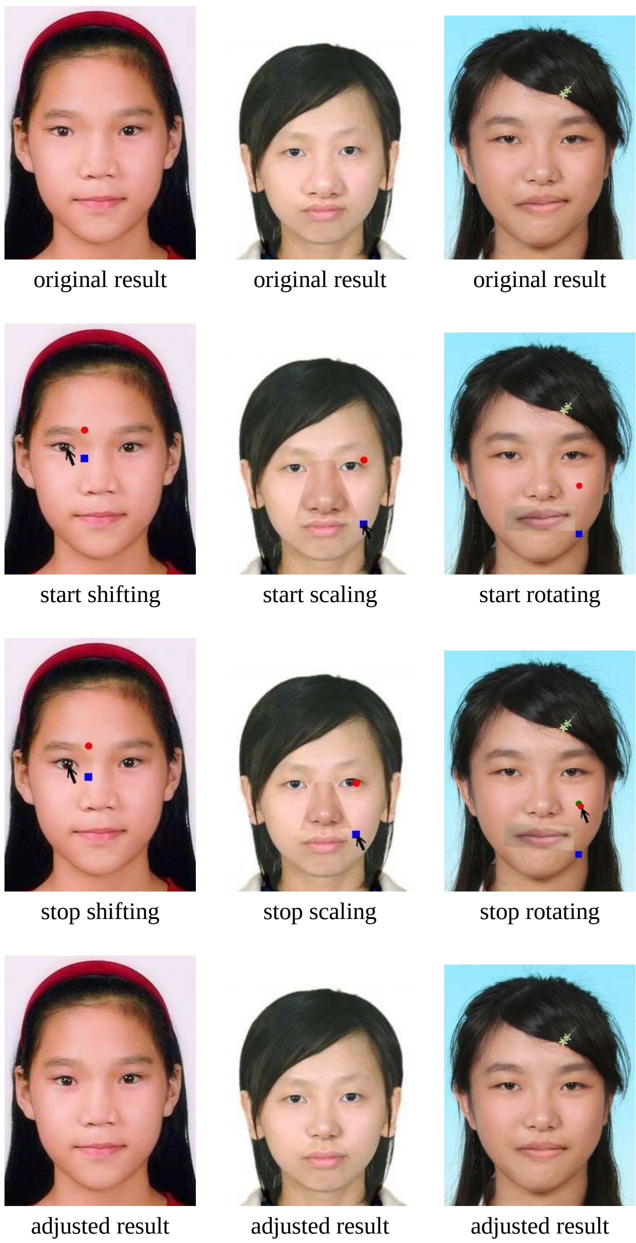
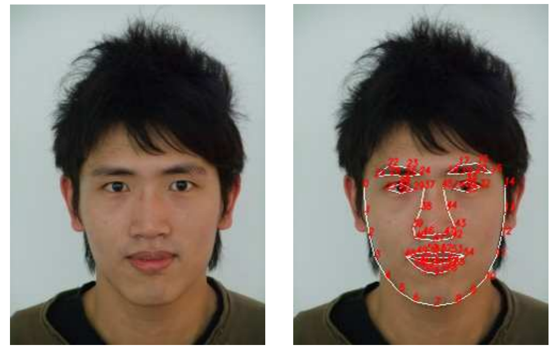


Figure 6: The adjustment operations that are supported in our system. Left column: shifting. Central column: scaling. Right column: rotation.



(a) (b)

Figure 7: The ASM process. (a) The original image. (b) The resulting image after ASM applies.

some "weird looking" face-off results to impress others. Figures 6 demonstrates three examples, and each of them is represented as a column of four images, where a user adjusts the involved features in three different ways.

3.4 Implementation

In terms of implementation, we opt for combining C# and MATLAB into one system, as C# provides better interfacing capability, while MATLAB offers superb numerical computation support. By making the MATLAB code a *dynamic linking library* or *DLL* for short, it can be called from the C# code, thus making the combination possible and efficient.

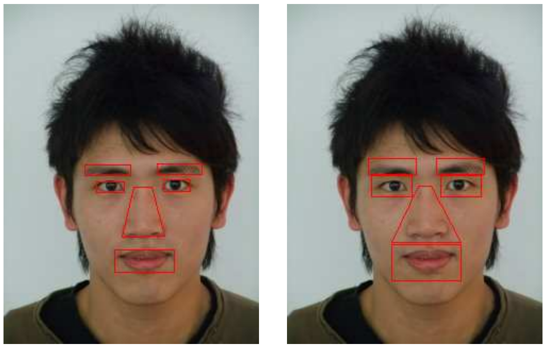
4. AUTOMATIC DETECTION OF FACIAL FEATURES

The automatic facial feature detection algorithm, which greatly enhances the usability of our system, is described in this section. To enhance the robustness, we build our feature detection scheme on top of the *Active Shape Model*, or ASM for short, to to locate the facial feature points. ASM outputs 63 feature points ($0 \sim 62$) for each person detected from the input image, and an example of such is shown in Figure 7.

From this ASM result, we could construct the bounding box for each facial feature. For example, the right eye consists of feature points $27 \sim 30$, a tight bounding box therefore can be constructed accordingly, and similarly for other facial features, as shown in Figure 8(a). However, for the purpose of altering the facial features automatically, we have to find larger bounding boxes to enclose the facial feature regions so that during the aforementioned Poisson image editing process, the blending procedure will not be undesirably "contaminated" by the edges of the detected facial features. Such adjustments can be seen in Figure 8(b), and how these adjustments are done are to be described in the ensuing sub-sections.

4.1 Eye Region Adjustment

For the ease of discussion, we first define what *skin pixels* are. 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 Figure 7(b). Second, within the facial shape, we discard the pixels within the facial feature boundaries detected by ASM. Third, for the remaining pixels, as shown in Figure 9 as the non-black area, we calculate their statistical mean and standard deviation to derive the skin color distribution. Through experiments, the skin colors



(a) (b)

Figure 8: The bounding box adjustment process. (a) The original bounding boxes derived from the ASM process. (b) The resulting bounding boxes after proposed adjustments.

Figure 9: The mask for calculating the skin color distribution.

are currently set to be within 2 standard deviations from the mean color. Finally, we can determine if a pixel is of a skin color or not for all the pixels in the image accordingly, and the result is shown in Figure 10(a) in non-black colors. Next, through the application of *Sobel edge detection*, we could identify the *edge pixels*, as shown in Figure 10(b). Now starting with the original eye bounding boxes derived from ASM, we vertically expand the bounding boxes until there is no pixel, which is a non-skin edge pixel, as shown in Figure 10(c), that lies on the upper and lower bounding box boundaries. We then do the same thing horizontally, and as a result the final adjusted eye bounding boxes, as shown in Figure 8(b), can be derived.

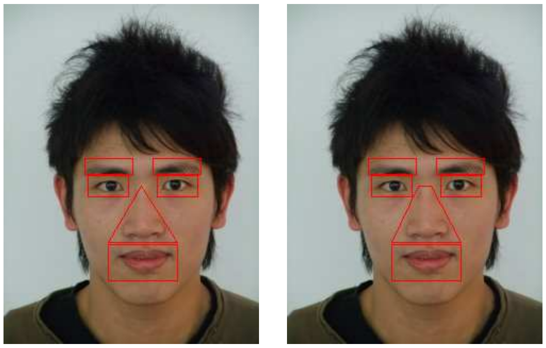
4.2 Eyebrow Region Adjustment

(a) (b) (c)

Figure 10: The process of adjusting the eye region. (a) The skin pixels. (b) The edge pixels. (c) The resulting pixels that are non-skin edge pixels.

Figure 11: The detected eyebrow regions.

Note that if a person's eyebrows are occluded by hair, then even we can mark the eyebrow regions correctly, the extracted eyebrows are not very useful in terms of later processing. For eyebrow region detection, we proceed as follows. First, similar to the skin detection process mentioned previously, we could also compute the eyebrow color distribution. Again we make use of ASM to form the polygons for eyebrows. More specifically, points from the 15th to the 20th can be used to enclose the left eyebrow, and thus the distribution of eyebrows can be calculated. Second, for each pixel in the following probing process, we compute its Euclidean distances to the skin color distribution and to the eyebrow color distribution. It is set to be a skin pixel, if it closer to the skin color distribution; otherwise, it is set as an eyebrow pixel. To decide the height of the eyebrows bounding box, we start from the upper boundary of the eye regions. For every point on the boundary, we go upward, until the first eyebrow pixel has been met. The vertical position of the point is then set to be a candidate for the lower bound of the eyebrow, as marked in green in Figure 11. After that, we go upward again until the first skin pixel has been met, and it is set to be a candidate for the upper bound of the eyebrow, as marked in blue in Figure 11. Next, we compute the statistical mean and standard deviation of the lower and the upper bound candidates, respectively. We discard the candidate points whose vertical positions lies outside of two standard deviations. The lower bound and the upper bound of the eyebrow bounding box are assigned to be the lowest and the highest vertical positions from the remaining candidates. After that, we have to determine the width of the eyebrow bounding box. The original width of the eyebrow bounding box is set to be the same with the eye bounding box. And we separately expand the boundary rightward and leftward one pixel at a time. While expanding the bounding box horizontally, we simultaneously check the newly expanded column for its lower bound candidate as described above. If the lower bound of the newly expanded column is higher than the upper bound of the eyebrow bounding box, the expansion process terminates. Finally, when both sides of the expansion process terminate, the width of the eyebrow bounding box is decided. There are two more things that should be pay attention to. First, the maximum height of the eyebrow bounding box can never be larger than the height of the eye bounding box. Second, the expansion process should have stopped earlier if the two eyebrow bounding boxes are going to collide with each other, or the horizontal boundaries are going to cross the facial shape derived from ASM.



(a) (b)

Figure 12: The process of nose detection. (a) The original triangular shape region. (b) The final bounding boxes, including the modified trapezoidal nose region.

4.3 Mouth Region Adjustment

To adjust the mouth region, the horizontal boundaries of ASM mouth bounding box are first set to be the horizontal positions of the 31st and 36th points (pupils of the eyes) derived from ASM. Then, the upper bound of the mouth bounding box is calculated as the vertical position of the midpoint of the 41st and 51st points (the base of nose and the upper lip, respectively) from ASM. Lastly, the height of the mouth bounding box is estimated as two times the height between the upper bound and the 61st point (the middle of lips) from ASM.

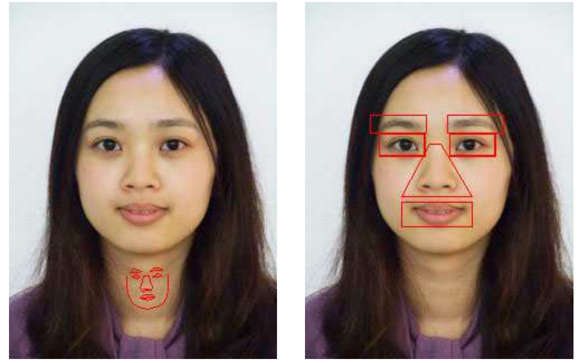
4.4 Nose Region Adjustment

Adjusting the nose region involves the following steps. First, we mark a *triangle* that will enclose the nose. The topmost vertex of the triangle is the mid point of two eye bounding boxes. The horizontal span of the other two vertices of the triangle is the same as the bounding box of the mouth, while the lower bound of the nose is set to be 2 pixels upward with respect to the upper bound of the mouth region determined previously. Figure 12(a) shows the triangle, marked in red, that is constructed this way. Finally, to accommodate the cases where people have wide nose wings, we enlarge the triangle to become a *trapezoid*, by expanding the topmost vertex of the triangle into a line segment, with the same vertical position, while its horizontal span is equal to half of the distance between the two eye bounding boxes, as shown Figure 12(b), which also demonstrates the aggregated final result of the facial feature detection.

Note that due to some undesirable lighting conditions or skin colors, the ASM process may fail, as shown in Figure 13(a). In that case, users are able to refine the final detection result by some intuitive manipulations such as translation or scaling, etc., to obtain the final desired bounding boxes, as shown in Figure 13(b). Once the boundaries of the facial feature bounding boxes are determined, the facial features are extracted and to be stored in the database, as shown in Figure 14.

4.5 Database of Facial Features

We have collected 161 facial photos for our testing purpose, and their features are automatically detected and placed into our feature database, where part of the database can be seen in Figure 4. Out of these 161 people, there are 61 occidental people, with 20 females and 41 males. The rest 100 people are oriental people, with 50 females and 50 males. Note that as our facial feature detection



(a) (b)

Figure 13: (a) Mis-detection of ASM. (b) The adjusted bounding boxes.

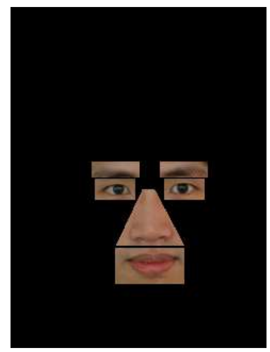


Figure 14: The extracted features through the aforementioned bounding boxes.

algorithm is based on ASM, the input images should be *passport similar*, otherwise the output features points from ASM may not be accurate, thus affecting ensuing feature alteration process. In addition, lighting condition and skin colors may also influence the accuracy of ASM. As a result, out of these 161 facial photos, our algorithm can perfectly detect the facial features for 125 of them through the help of ASM, thus leaving 36 photos that need further manual adjustment. Out of these 36 incorrect results, 17 of them are due to exclusively mis-detection of eyebrows, and 13 of them are due to the errors of other facial features. The rest 6 cases are severe ASM errors, as shown in Figure 13(a), where the detected facial shape is completely outside of the desired one. We should nevertheless admit that, as a prototype system, there is still room for further improvement, at least in terms of completeness of the feature database.

5. RESULTS

We have conducted our experiments on a Pentium IV 2.8GHz machine with 768 MBytes memory, running on the Windows XP operating system. The tool for development is Visual Studio .Net 2005 Visual C#, with a back-end MATLAB 7.0 engine for solving the Poisson equation. In terms of timing, the face-off time is roughly 1 second, while the automatic facial feature detection is even much faster.

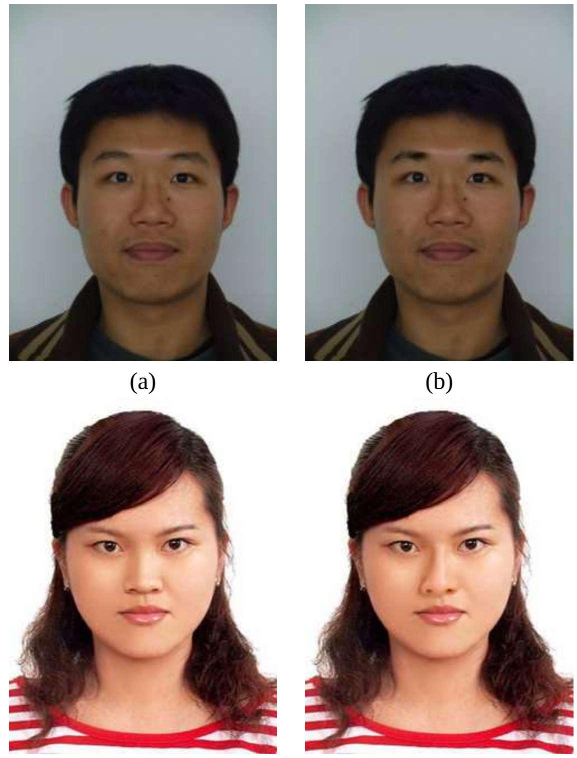
Figure 15 demonstrates the results of our system. We believe that the successfulness could be observed from this set of trials, which even boldly merge facial features coming from different ethnic backgrounds. A more vivid demo of our system can be seen from the accompanying video that can be downloaded from http://www.cs.ntust.edu.tw/ckyang/faceoff2.avi.

To further demonstrate the effectiveness of our approach, we compare the results generated from our system with those from Bitouk *et al.*, as shown in Figure 16 and Figure 17. More specifically, instead of replacing the entire face at one time, we replace facial features altogether to achieve the similar desired effect. As can be seen from this figure, except for the first result in Figure 17, where the imperfection of our result is due to the lack of further parameter adjustment, such as lighting and skin color, nearly all of our results show little noticeable difference. However, as claimed aforementionedly, our system enjoys the mechanism of a finer granularity that allows users to selectively replace each and every facial feature individually and independently, as shown in Figure 18. And such flexibility would make our system more useful in practice.

To have a more objective evaluation of our system, we have also conducted a user study, and part of the results are shown in Figure 19 and Figure 20. For this user study, we have collected 11 test sets. Each set contains an original image photo and 3 to 5 face-off results generated by our system, and there are 84 people participated in our study. We ask each participant to give ranks for each set of the images (ties are allowed) without telling them which the original ones are. The images shown here are the best results in each of these 11 test sets. In this figure, we also show the number of people, out of these 84 participants, that find the "face-offed" results are more appealing than the originals. Note that for all the results in these two Figures are automatically generated by our system without any user assistance, except the image labeled 60, where we changed the subject's nose. The reason for such manual adjustment is due to the imperfection of nose detection, which is affected by the wrinkles around the nose. It should be evident from this study that by offering the ability to selectively change some facial features, there can always be more than half of the participants find the properly face-offed results more attractive, thus proving our claims.

After analyzing the result of the rankings, we have some findings. Changing one's eyes may give others a very different impression. Especially a pair of more catching eyes usually yields a better ranking, as shown in the sixth and seventh test sets in our user study, as shown in Figure 19 and Figure 20. A smiling mouth could make a person look kinder, thus a higher ranking, as shown in the fourth and tenth test sets in Figure 19 and Figure 20. Furthermore, we have found that altering one's eyebrows or nose seems not to make significant impacts to humans visual perception, as demonstrated in Figure 21.

In addition to making the adjusted results more attractive, we also want to make sure the face-off results generated by our systems do not look fake. To validate this, we conducted the second user study by using the same 11 test sets, but with each of test set containing one original image and the most favored face-off image chosen from the first user study. The second user study was performed as follows, and the statistics are shown in Table 1. For each of the 11 test sets, we asked each of the 84 participating people whether the original image or the face-off image looks fake, without telling them which one is the original image beforehand. The second column in this table denotes the number of people, out of 84 people, who think that the original image looks fake, while the third column the number of people who consider the face-off result a fake



(a) (b) (c) (d)

Figure 21: (a) and (c) The original images. (b) Eyebrows changed from (a). (d) Nose changed from (c).

one. The last column in this table calculates the differences by subtracting the numbers in second column from those in the third one. To prove that our results look real, we set two hypotheses as the following.

$$\begin{aligned} H_0: &\quad D = 0 \\ H_1: &\quad D \neq 0 \end{aligned} \tag{1}$$

where D represents the differences between the two observations. By using the *paired t-test*, the *t value* we derive is 1.429, and it is less than the two-sided *p value*, which is 2.201, for 0.05 level of significance. Therefore, we cannot reject H_0, or equivalently, the result images generated by our system cannot be easily recognized as artificial ones. As a result, this consolidates our belief that our generated results look natural.

However, we must point out that some results still look weird because the newly face-offed facial features are just not fit to the person, as shown in Figure 22(a) and (b), or the combination of the exchanged facial features just appear strange, as shown in Figure 22(c) and (d).

6. CONCLUSIONS AND FUTURE WORK

We have developed a system that could alter facial appearance automatically, gracefully, and interactively. The two enabling techniques that underlie this work are *Poisson image editing* and *automatic extraction of facial features*, where the later one is built on top of *Active Shape Model* to increase the robustness of facial feature detection. Results are demonstrated to prove the effectiveness of our proposed approach, and compared with existing approach when applicable. A user study is also conducted to verify

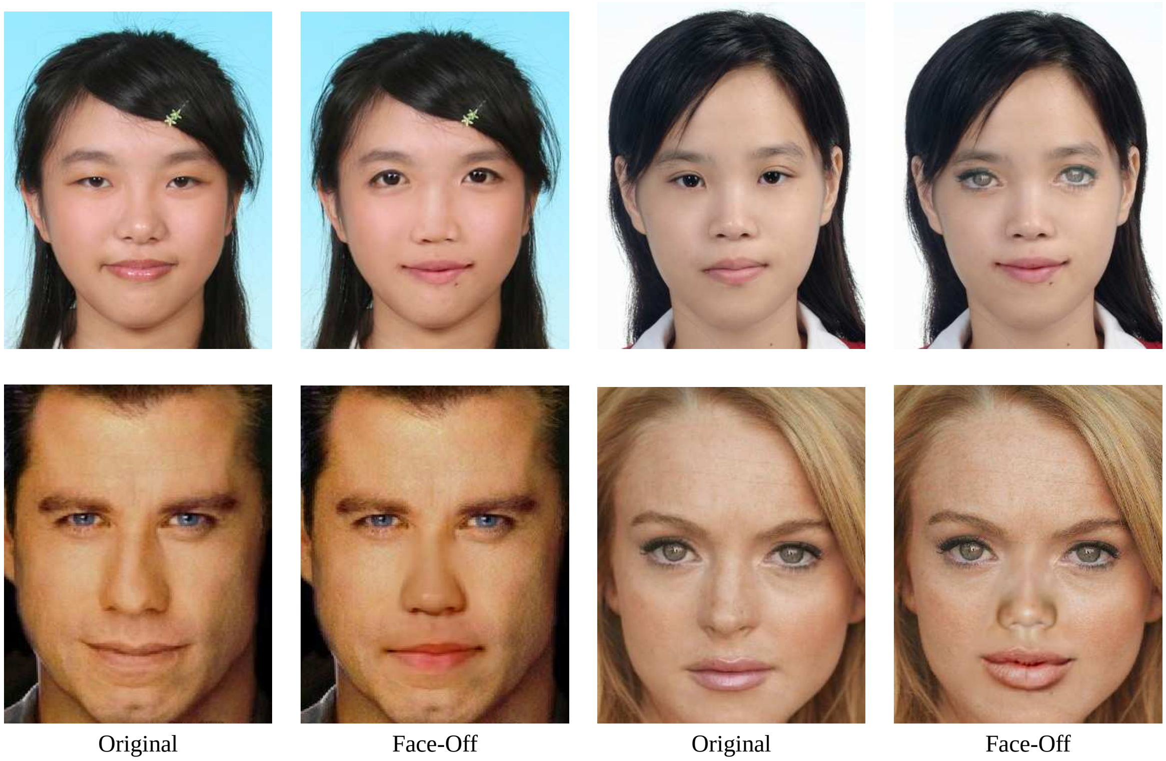


Figure 15: The face-off results of our system.

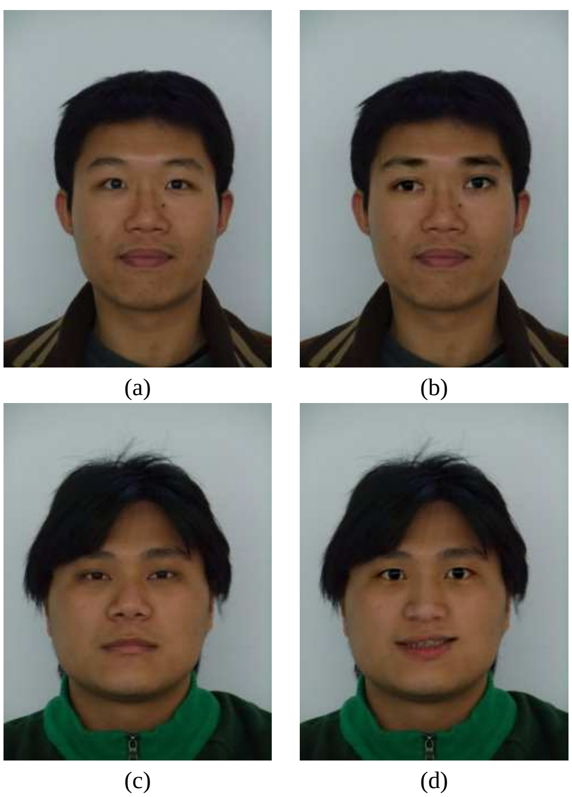


Figure 22: (a) and (c) The original images. (b) and (d) The weird face-off results.

Table 1: The comparisons of our system's face-off results against the original images for authenticity.

Model	Original	Face-off	Difference
1	5	4	-1
2	0	17	17
3	1	0	-1
4	0	0	0
5	1	12	11
6	9	7	-2
7	4	8	4
8	5	2	-3
9	5	9	4
10	1	5	4
11	5	2	-3

the usefulness of our system. In addition to facial alteration, our system could also be employed in numerous applications, such as the generation of *avatars* or *photo of suspects*, while at the same time offering superior efficiency and quality. However, our current system still has its limitations. For example, our system does not deal with eyebrows and ears presently as they are often occluded by hair. Similarly, due to the interference of hair, alteration of facial contours is not handled currently. Errors could also be resulted if one's beard or mustache is too dense or too dark, which could potentially mislead the nose and mouth detection processes. In addition, for the automatic facial feature detection to be correct, the lighting condition cannot be too extreme. For example, if the left face is too bright and the right face is too dark, erroneous detections

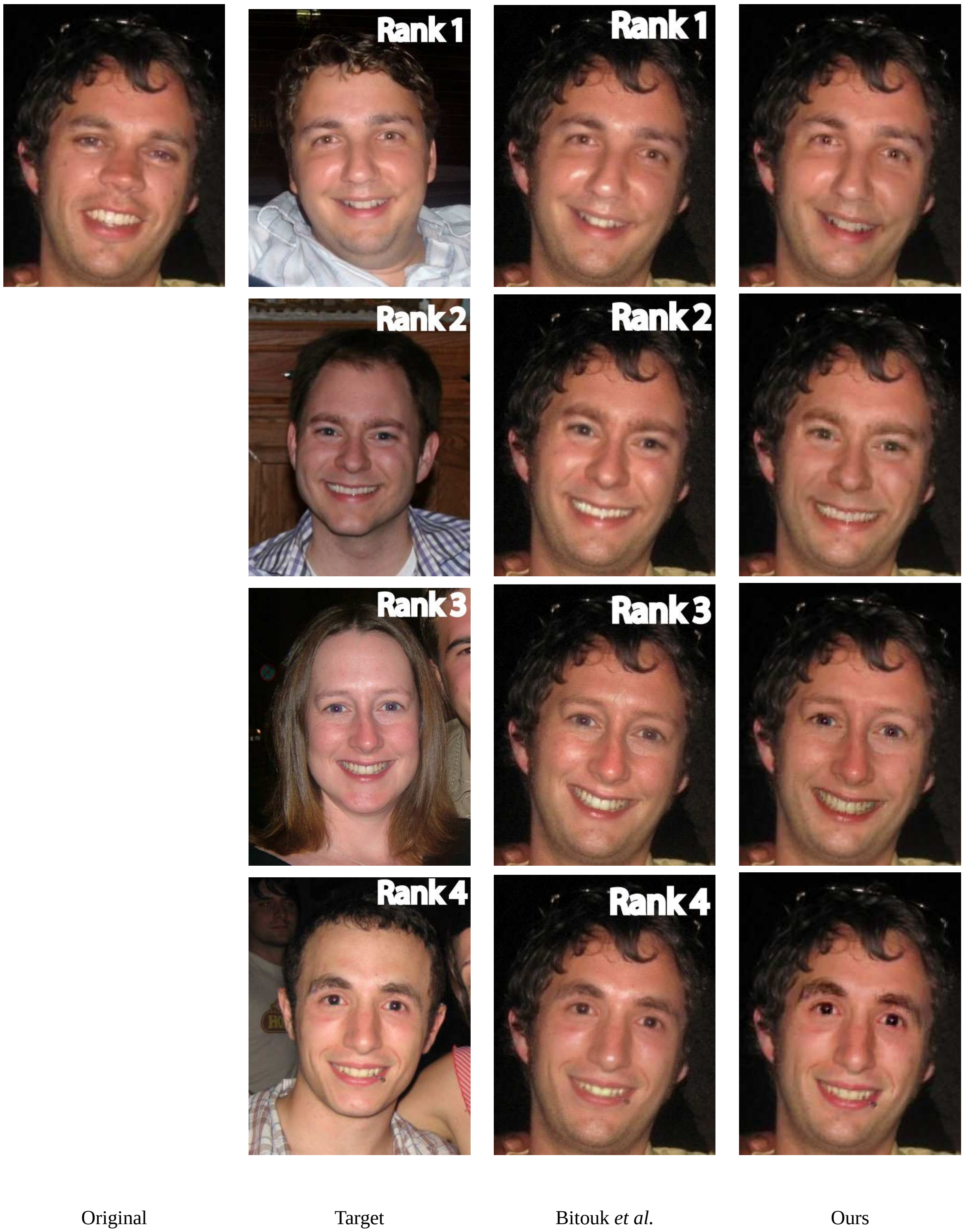
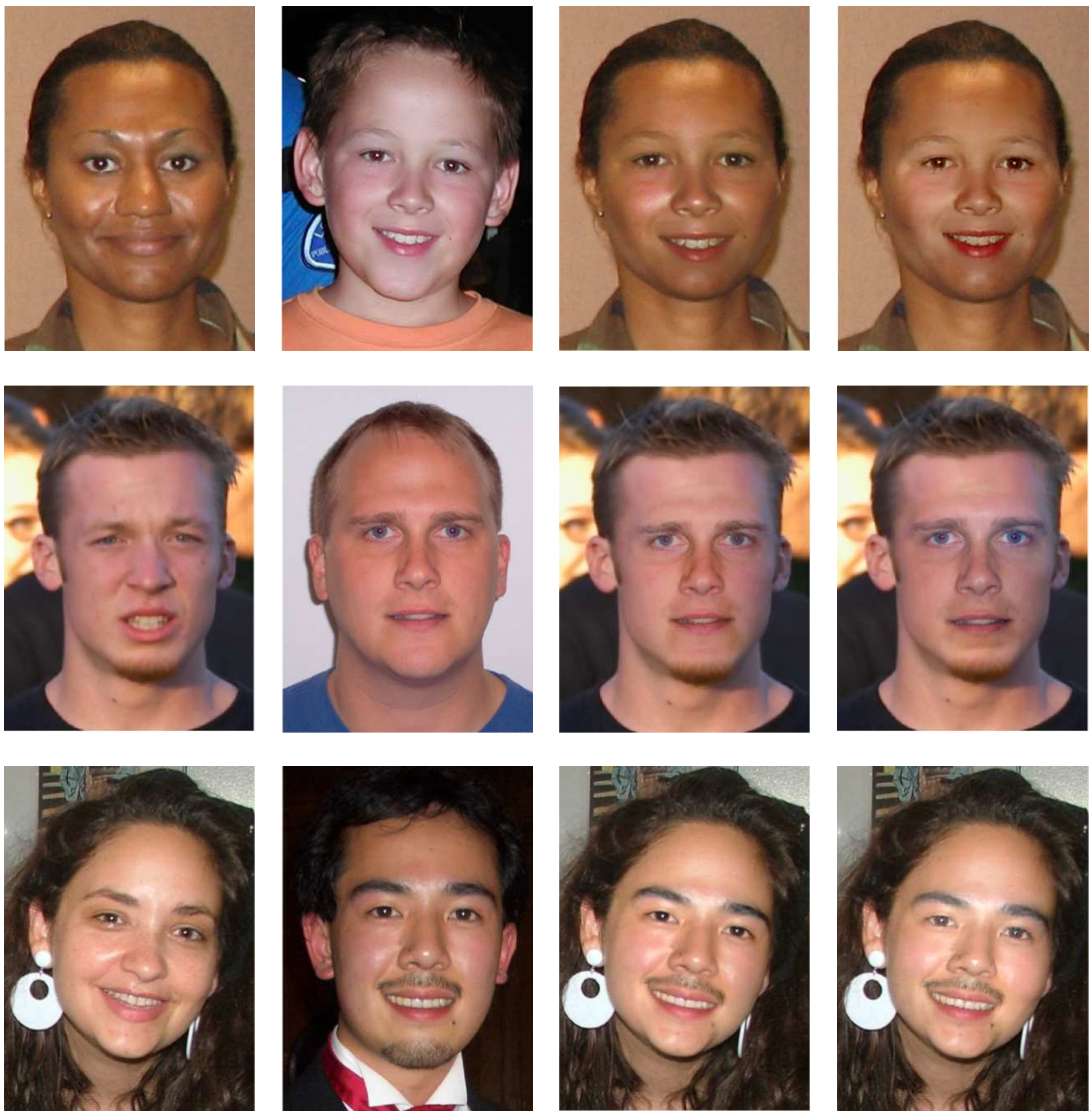


Figure 16: Comparisons between Bitouk *et al.’s* results and ours.



Original Target Bitouk *et al.* Ours

Figure 17: More comparisons between Bitouk *et al.'s* results and ours.

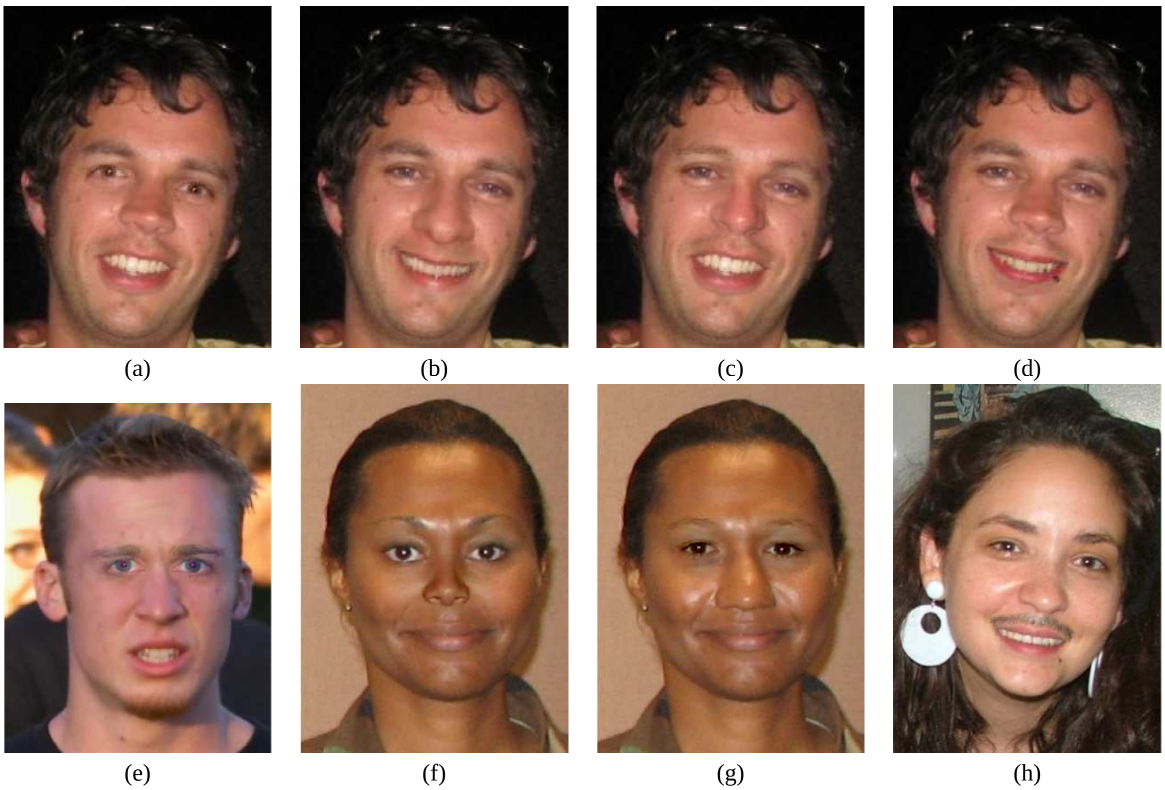


Figure 18: Some face-off results of our system, where the original images are from Figure 16 and Figure 17. More specifically, we replaced the eyes in (a), nose and mouth in (b), eyebrows and nose in (c), mouth in (d), eyes and eyebrows in (e), nose in (f), eyes and eyebrows in (g), and mouth in (h), respectively.

Figure 19: Some face-off results from the user study of our system. The odd columns are the original images, from the first set to the eleventh set, while the even columns the face-off results. Note that the numbers shown in even columns are the number of people, out of 84 people, that find the face-offed results more attractive than the original images; the larger the number, the more attractive a face-offed resulting image.

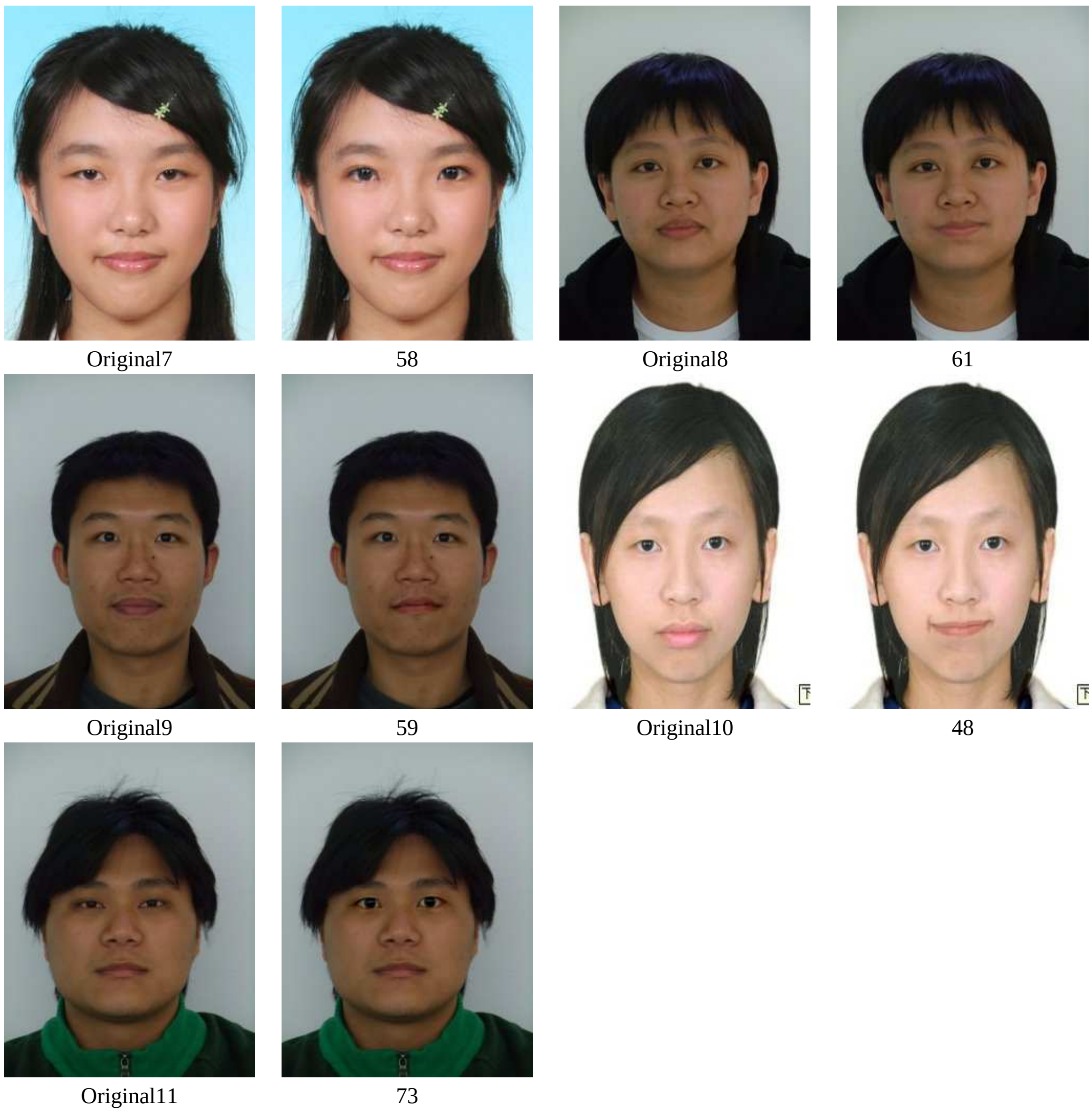


Figure 20: More results from the user study.

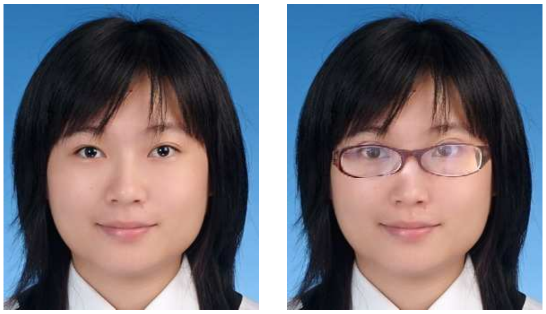


Figure 23: Example of adding glasses for a person.

may occur. Moreover, if there is a significant difference in resolution between the original image and the target image or where the facial features come from, problems may occur. For example, when replacing a high-resolution facial feature with a facial feature with a lower resolution, the blending, as well the final outlook may contain undesired artifacts.

In the future, we will not only address the aforementioned issues, but also strive for providing more interesting functionalities, such as the addition of eye-glasses (a preliminary trial result is shown in Figure 23), ear-rings, necklaces, etc., or even the change of hairstyle. Another issue is regarding the completeness of our facial feature database. Currently it is still hard to tell if our manually collected features are complete or not, i.e., whether the set of features have covered all possibilities to a reasonable degree. One solution may be to crawl the web to look for more different facial features, and only the features that are different enough from the existing ones in the database are retained, where the comparison is based on a pre-defined metric. All features collected as such could be classified into a fix number of categories, while within each category the most representative feature will be selected and displayed along with the system for the face-off purpose. The issue of automatically adjusting lighting and skin color parameters to enhance the blending effects, as done by Bitouk *et al.* [4] should also be addressed. One more interesting direction to pursue is to automatically determine the best face-off strategy, i.e., the combination of specific facial features from the database, to achieve the most ideal or pleasing result, where the aesthetic standards could be trained and learned by employing a *machine learning* paradigm. It would also be interesting to extend this framework to videos or 3D surface models, such as Blanz *et al.* did [6]. Finally, the synthesis of different expressions could also be provided, such as the work done by Yang *et al.* [28], to further enrich the capabilities of our system.

7. REFERENCES

[1] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid Mothod in Image Processing. *RCA Engineer*, 29(6):33–41, 1984.

[2] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive Digital Photomontage. In *SIGGRAPH '2004*, pages 294–302, 2004.

[3] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image Inpainting. In *SIGGRAPH '2000*, 2000.

[4] D. Bitouk, N. Kumar, S. Dhilon, P. Belhumeur, and S. K. Nayar. Face Swapping: Automatically Replacing Faces in Photographs. In *SIGGRAPH '2008*, 2008.

[5] V. Blanz, K. Scherbaum, T. Vetter, and H. Seidel. Exchanging Faces in Images. In *Eurographics '2004*, 2004.

[6] V. Blanz and T. Vetter. A Morphable Model for the Synthesis of 3D Faces. In *SIGGRAPH '1999*, 1999.

[7] R. Bornard, E. Lecan, L. Laborelli, and J. Chenot. Missing Data Correction in Still Images and Image Sequences. In *ACM Multimedia '2002*, pages 355–361, 2002.

[8] Y. Boykov and M. P. Jolly. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. In *ICCV '2001*, 2001.

[9] T. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models – their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[10] T. F. Cootes and C. J. Taylor. Statistical Models of Appearance for Medical Image Analysis and Computer Vision. In *In Proc. SPIE Medical Imaging*, pages 236–248, 2001.

[11] A. A. Efros and T. K. Leung. Texture Synthesis by Non-Parametric Sampling. In *ICCV '99*, 1999.

[12] H. Gu, G. Su, and C. Du. Feature Points Extraction from Faces. In *Image and Vision Computing NZ*, 2003.

[13] J. Jia, J. Sun, C. Tang, and H. Shum. Drag-and-Drop Pasting. In *SIGGRAPH '2006*, pages 631–637, 2006.

[14] R. Kjeldsen and J. Kender. Finding Skin in Color Images. In *FG' 96 (2nd International Conference on Automatic Face and Gesture Recognition)*, 1996.

[15] V. Kwatra, A. Schodl, I. Essa, and A. Bobick. Graphcut Textures: Image and Video Synthesis Using Graph Cuts. In *SIGGRAPH '2003*, 2003.

[16] J. Lalonde, D. Hoiem, A. a. Efros, C. Rother, J. Winn, and A. Criminisi. Photo Clip Art. In *SIGGRAPH '2007*, 2007.

[17] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless Image Stitching in the Gradient Domain. In *ECCV '2004*, pages 377–389, 2004.

[18] T. Leyvand, D. Cohen-Or, G. Dror, and D. Lischinski. Digital Face Beautification. In *SIGGRAPH 2006 Technical Sketch*, 2006.

[19] T. Leyvand, D. Cohen-Or, G. Dror, and D. Lischinski. Data-Driven Enhancement of Facial Attractiveness. In *SIGGRAPH '2008*, 2008.

[20] E. N. Mortensen and W. A. Barrett. Intelligent Scissors for Image Composition. In *SIGGRAPH '1995*, pages 191–198, 1995.

[21] M. M. Oliveira, B. Bowen, R. McKenna, and Y. Chang. Fast Digital Image Inpainting. In *VIIP (International Conference on Visualization, Imaging and Image Processing) '2001*, 2001.

[22] S. Peleg. Elimination Seams from Photomosaics. In *Conference on Pattern Recognition and Image Processing*, pages 426–429, 1981.

[23] P. Perez, M. Gangnet, and A. Blake. Poisson Image Editing. In *SIGGRAPH '2003*, pages 313–318, 2003.

[24] J. Sun, L. Yuan, J. Jia, and H. Shum. Image Completion with Structural Propagation. In *SIGGRAPH '2005*, pages 861–868, 2005.

[25] M. Uyttendaele, A. Eden, and R. Szeliski. Eliminating Ghosting and Exposure Artifacts in Image Mosaics. In *CVPR '2001*, pages 509–516, 2001.

[26] J. Wang, M. Agrawala, and M. Cohen. Soft Scissors: An Interactive Tool for Realtime High Quality Matting. In *SIGGRAPH '2007*, 2007.

[27] L. Wei and M. Levoy. Fast Texture Synthesis Using Tree-Structured Vector Quantization. In *SIGGRAPH '2000*, pages 479–488, 2000.

[28] C. Yang and W. Chiang. An Interactive Facial Expression Generation system. *Multimedia Tools and Applications*, 2008.

[29] M. Yang, D. J. Kriegman, and N. Ahuja. Detecting Faces in Images: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.