Digital Face Replacement in Photographs

Shahzad Malik
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
smalik@cs.toronto.edu

Abstract

In this paper, we describe a simple method for digitally replacing an individual's face in a single photograph with that of another person. Our work focuses on the illumination aspect of face replacement, whereby the lighting conditions across a human face in an arbitrary image are extracted and then applied to a target replacement face. We also describe a method to seamlessly merge this newly lit face into the original photograph. The face replacement images are shown to produce convincing results in many cases, even with large variations in skin tones.

1. Introduction

The ability to automatically replace a face in a photograph with that of another person has huge implications in the entertainment and special effects industries. For example, consider a Hollywood stunt double performing a dangerous routine while remaining in full-view of the camera, and then having a post-processing step that automatically replaces each instance of the stunt double's face with that of the desired actor. Or imagine watching your favorite movie where the lead actor's role is seamlessly replaced with your own face. While a few recent films have achieved good results when performing face replacement on stunt doubles, the techniques require the stunt double to wear a special custom-fit mask with reflective markers for tracking, as well as requiring controlled illumination conditions in the environment being filmed [5].

There are a number of related but separate components that are required in order to accurately replace a face in a photograph or frame of video. The first is a system to robustly detect and track the desired face (the source face) in a video sequence, and accurately estimate its pose. The second is a system to estimate the illumination conditions across the source face in the image, and then accurately applying this lighting to a new face (the target face). Next we need a system to extract facial expressions from the source face and then apply them to the target face. Finally, a system to accurately merge the target face over the source face in the original photograph is required. Ideally, this last component should recognize partial occlusions of the source face so that they are still present over top of the target face.

The focus of this paper is on a system to capture the illumination condition across the source face in a single 2D image, and then applying this lighting onto some target face. Additionally, we describe a method to seamlessly merge this newly lit target face into the original photograph.

The remainder of this paper is organized as follows. We first describe related work in the next section. Section 3 then describes a technique to extract the lighting conditions across a human face in an arbitrarily lit image. Section 4 and Section 5 then describe how to relight a new face using the extracted lighting information, and then seamlessly merge it into the original image. Section 6 presents the results from our work, and Section 7 discusses areas for future research. We conclude the paper in Section 8.

2. Related Work

There is plenty of existing work regarding lighting estimation and re-illumination of a face in the current literature, but none directly addresses the issue of face replacement. A method to estimate the reflectance field of a human face is described in [3], which is then used to re-illuminate the face under different lighting conditions and in various environments. Clearly this method is applicable to the problem of digital face replacement, but it first requires capturing the reflectance field of the target face using a complicated *light stage* apparatus. The morphable model algorithm described in [1] could potentially be used to replace a face in a photograph, but the proposed system is not shown to handle dramatic lighting conditions or different skin tones. Marschner presented a general inverse illumination algorithm in [8] that requires a 3D model of the environment and a set of basis lights situated on a sphere around the model. A class-based face re-rendering and recognition algorithm is presented in [11, 12] that can be used to re-illuminate front-facing heads in arbitrary photographs, but it cannot handle different head poses since it is completely image-based.

3. Lighting Estimation

Our lighting estimation algorithm combines the model-based techniques presented in [8] with the image-based techniques presented in [11, 12]. Let I_s represent the source image or photograph containing the face we would like to replace, and let I_T represent an image of the target face that

should replace the source face. We make the following assumptions:

- The lighting conditions and head pose in I_S are arbitrary and unknown.
- The general skin tone of the face in I_S is known.
- The head pose in I_T is front-facing.
- The face in I_T is illuminated fully and evenly.

3.1 Generating the basis images

Using a Lambertian model of reflectance, each pixel p on the surface of a face in an image I can be represented by the following equation

$$I(p) = \rho \, \mathrm{n}_p \cdot \mathrm{s} \tag{1}$$

where n represents the surface normal at pixel p, s represents the direction to some point light source, and ρ is a scalar that represents a mixture of the surface albedo and spectral composition of the light source. Assuming that our light source intensity is represented as an (*R*,*G*,*B*) triple, we would have a different equation for each (*R*,*G*,*B*) component of the pixel p, where the ρ would vary based on the color component.

An interesting proposition that is presented in [12] states that an image of an object under the above Lambertian model can be represented as a linear combination of a fixed set of three images of the object, where the three images are illuminated from three linearly independent lighting directions. In other words, our image I_S can be represented as

$$I_S(p) = \rho \, \mathrm{n}_p [\alpha_1 \mathrm{s}_1 + \alpha_2 \mathrm{s}_2 + \alpha_3 \mathrm{s}_3] \tag{2}$$

which gives us

$$I_S(p) = \alpha_1 I_1(p) + \alpha_2 I_2(p) + \alpha_3 I_3(p) \tag{3}$$

where I_1, I_2, I_3 are the three *basis images* of the face in I_S taken under three linearly independent lighting conditions, s_1, s_2, s_3 represent vectors from p to each light source, and α_1, α_2, α_3 represent some coefficients for each basis image. Once again it is worth noting that we can have a different set of three components for each (*R*,*G*,*B*) color channel.

Note that we cannot assume we have access to three uniquely illuminated images of the face in I_S when dealing with an arbitrary photograph. In order to overcome this problem, we settle for an approximation whereby we manually select a skin tone that roughly matches the skin tone of the face in I_S that we wish to replace. The skin tone is chosen from a small database of nine representative faces that are front-facing and fully-lit, as depicted in Figure 1. Using the selected skin tone face, we manually fit a generic 3D face model to it using an interactive tool. The tool allows a user to manipulate the 3D mesh with six degrees of freedom. Once the mesh is aligned, a texture map can be lifted from the skin-tone image by performing a simple planar projection of the image onto the model vertices. Let M_A represent this texture-mapped 3D face model. A similar model-fitting approach is performed on both the source face in I_S and the target face in I_T. Let M_S and M_T represent the source texture-mapped 3D model and target texture-mapped 3D model respectively. Figure 2a shows the generic 3D model we are using after manually fitting it to a face from the skin-tone database. Figure 2b shows the texture-mapped model from some arbitrary viewpoint.

Figure 1 – Skin tone database

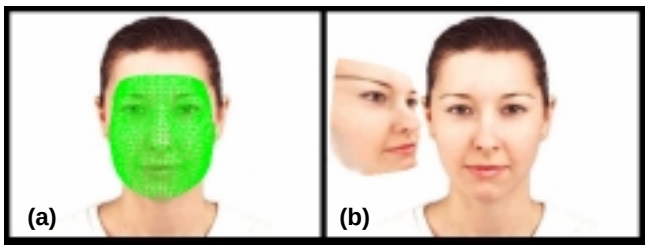

Figure 2 – (a) Generic 3D model fitted to a 2D face image; (b) Texture-mapped model after planar image projection, in some arbitrary pose

Let L_1, L_2, L_3 represent three white point light sources located at three linearly independent positions from the origin of the generic 3D model, as depicted in Figure 3. Given M_A and M_S, we then generate our approximate basis images I_1, I_2, I_3 by transforming M_A and the three light sources by the head pose of M_S, and then rendering M_A three times (once with each light source enabled and the other two disabled). Figure 4 shows the process in action. Figure 4a shows the original image that we wish to perform a face replacement on. Figure 4b shows the generic model being fit to the face in original image. Figure 4c depicts the selected skin tone model M_A under M_S's head pose. Finally, Figures 4d, 4e, and 4f show the basis images that were rendered under the L_1, L_2, and L_3.

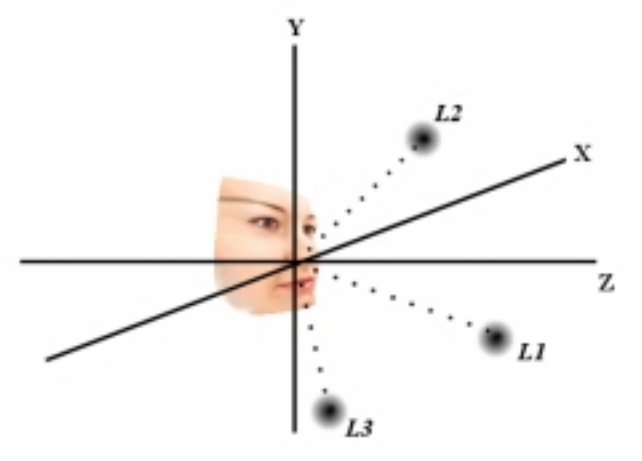

Figure 3 – Three linearly independent light sources around generic 3D model

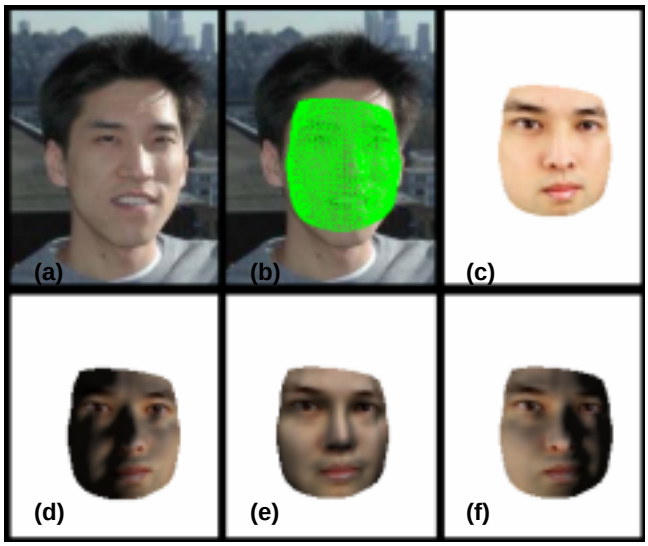

Figure 4 – (a) Original image; (b) Original image after generic model fitting; (c) Texture-mapped skin tone model M_A in M_S head pose; (d), (e), (f) M_A under illumination from L_1, L_2, and L_3 respectively.

3.2 Determining the coefficients

As mentioned earlier, with the basis images in place we can then solve for the coefficients. For each (*R*,*G*,*B*) color channel, we solve Equation 3 after posing it as the familiar *A*x=b matrix equation, where *A* is an *N*x3 matrix with each *i*-th column consisting of the pixels in the *i*-th basis image, b is an *N*-vector consisting of the pixels in the original image, and x is a 3-vector containing our coefficients. Let *N* denote the number of pixels in the rendered image of the M_S model. In other words we need to solve the following matrix equation

$$\begin{bmatrix} \vdots \\ I_S(p) \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots \\ I_1(p) & I_2(p) & I_3(p) \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} \quad (4)$$

We compute a separate least squares solution for the above matrix equation for each of the (*R*,*G*,*B*) color components, resulting in nine total coefficients as described earlier. For each of the *i* basis images, let us represent the (*R*,*G*,*B*) coefficients as $\alpha_{i,r}$, $\alpha_{i,g}$, and $\alpha_{i,b}$.

4. Illuminating the Target Face

Now that we have computed the coefficients for the illumination in the original image, we can attempt to relight our texture-mapped M_T model. This turns out to be a relatively simple task. We first transform the M_T model by the pose of the M_S model in order to place it in the proper location in the original image. We then reconfigure the basis lights so that they are no longer pure white. Instead we assign their intensities to be the coefficients of the basis images. In other words we have the intensity of L_i=($\alpha_{i,r}$, $\alpha_{i,g}$, $\alpha_{i,b}$), which follows from Equation 3. With each of the lights enabled simultaneously now, we render M_T. Figure 5 shows an example of relighting a target face with the lighting conditions extracted from Figure 4.

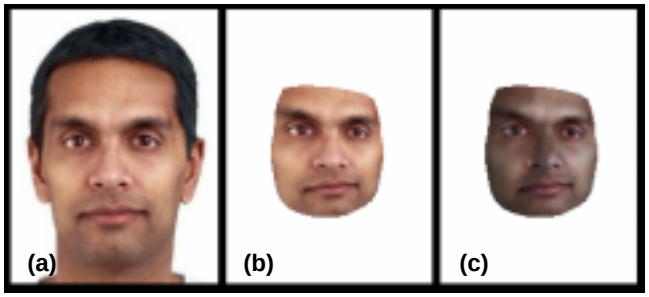

Figure 5 – (a) A target face; (b) The target model M_T in M_S's head pose from Figure 4c; (c) The target model after relighting with illumination conditions from Figure 4a.

5. Merging the Target Face

With the newly lit M_T, we can automatically merge it into the original image I_S, effectively replacing the original face contained in the area spanned by M_S. But this is not enough, since the generic 3D model does not fully capture all of the flesh areas of the source face. For example, in Figure 4b, we see that the generic model does not fully capture the ears, forehead, chin, and neck. If we were to simply paste the newly lit image from Figure 5c into Figure 4a, the replacement would not be very convincing.

To overcome this problem, we propose a method to replace any detected flesh pixels in the original image with similar flesh colors from the newly lit image. The approach is outlined in the following sections.

5.1 Flesh pixel detection

Before we can attempt to replace the flesh pixels in I_S, we must first detect them. Flesh pixel detection has been an active research area for the purposes of face recognition and tracking systems, and a number of fairly reliable detectors have been proposed [2, 4]. The majority of detectors rely on statistical models that make use of large training sets consisting of manually labeled skin and non-skin pixels. Given skin and non-skin histograms, these systems then assign a probability to each pixel in order to classify a particular color as either skin or non-skin. We will use a slightly simpler approach where our skin training set will only consist of the pixels of I_S that are contained within the projected boundary of M_S. In HLS color space, it is known that human skin has roughly the same skin hue [2]; skin tone is mainly defined by adjusting lightness and saturation. Given our skin training pixels, we then expect to find roughly Gaussian distributions for both our lightness and saturation channels. These distributions will then be used to classify pixels as either skin or non-skin, by simply determining whether the pixel's color falls within the appropriate bell curve. Figures 6a and 6b show the lightness and saturation distributions for the pixels contained inside the generic model in Figure 4b. Figure 6c shows the corresponding flesh pixels that were detected. Note that a morphological closing operation was performed on the flesh mask image to remove noisy pixels, followed by a flood-fill operation to detect only connected flesh areas around the projected area of M_S.

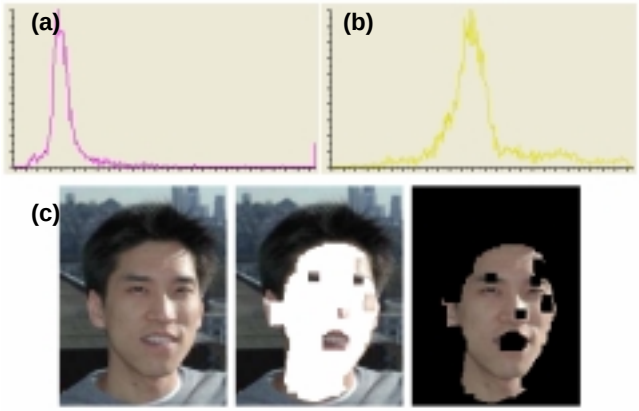

Figure 6 – (a) Lightness histogram for flesh pixels from Figure 4b; (b) Saturation histogram for flesh pixels from Figure 4b; (c) Detected flesh pixels from Figure 4b.

5.2 Histogram matching

Now that we have our flesh pixels, we must convert these into colors that more closely resemble the skin tone of our target face. We first compute a new set of HLS histograms for the newly lit M_T model. We then attempt to adjust the flesh pixels in I_S by matching the color histograms of M_S to closely fit the histograms of M_T. In other words, for each flesh pixel detected in the original I_S image, we map it to a color in the rendered M_T image that has a similar probability and location along the Gaussian distribution curve. Figure 7a shows the results of histogram matching for the face in Figure 5.

Figure 7 – (a) Result after histogram matching; (b) Result after color blending

5.3 Color blending

While the skin color matching algorithm from the previous section approximates the new skin tone fairly well, sharp edges will still be present in the image due to unavoidable inaccuracies (see Figure 7a for example). A simple color blending algorithm can be applied to reduce discontinuities where the rendered M_T model meets the histogram-adjusted pixels. We first define a blending size B_f, which represents how many pixels around the edge of M_T we wish to blend. A simple heuristic that works quite well is to choose B_f such that it is $1/8^{th}$ the radius of the projected area of M_S in the image. Thus for each flesh pixel p_f in the image we determine its distance d_f to the closest edge pixel p_e of M_T, and if this distance is less than B_f we weight the color inversely with respect to d_f. In other words

$$p_f = (1.0 - \frac{d_f}{B_f}) p_e + \frac{d_f}{B_f} p_f$$

A similar blending is then applied to each flesh pixel or pixel of M_T that borders a non-flesh pixel. For this second blending, however, we choose a smaller blending size B_e that is currently fixed at four pixels. This effectively blends the new face into the original image by reducing aliased edges that result from the polygon rendering engine and from the flesh detection algorithm. Figure 7b shows the final replaced face from Figure 7a after blending is applied.

6. Results

In this section we show some test results. Figure 8 is using the same source image as in Figure 7, but with a new target face applied. The quality of the final image is quite good, with the lighting conditions being accurately recre-

ated. One thing worth noting are the bright spots located in the right part of the forehead and neck area, as well as the ear at the left. In these locations, the flesh detector failed to classify these pixels as skin, and thus the color from the original photograph is being seen here. Blending reduces these artifacts somewhat, but the result is still not as seamless as we would like it to be. A more robust flesh detector would clearly help in this case. One other thing worth noting in Figure 8 is the apparent change of face shape. The target face is that of a woman with a soft rounded chin, while the source face is of a man with a sharp and elongated chin. Since our face replacement technique simply replaces flesh pixels outside the generic mesh area, the replaced face will take on physical features of the source face around the jaw line, ears, and forehead. A more advanced algorithm that can maintain the target face features and then properly rebuild the shape of the face using surrounding pixels would be an interesting area for future research.

Figure 8 – Face replacement result

Figure 9a shows a more complicated example where the pose of the source face is not completely front-facing. Manually aligning the generic mesh into such a pose can be difficult, motivating the need for a more automatic model fitting algorithm as described in [1]. Another thing to note in this result is the shape of the nose. The nose for the source face is long and wide, while the nose of the target face is short and thin (Figure 9c). Since the target model is simply transformed into the source head pose, the nose of the target subject will seem to grow (Figure 9b). While this is not a major problem for relatively front-facing scenes, more extreme head poses as depicted in Figure 9 further highlight the need for more precise model-fitting algorithms. Finally, Figure 9 shows the disadvantage in using a simple Lambertian reflectance model. For example, the specular highlights on the forehead of the source face are not recreated in the target face. Additionally, the shadows in the jaw area of the source face are not being cast, since in the original image the shadows appear to be a result of a light source located behind and to the right of the head.

Figure 9 – Face replacement result with non-frontal pose; (a) Original image; (b) Face replaced; (c) Target face

Figure 10 shows what happens when the flesh detector finds a large number of false positives that are connected to the true facial flesh area. As can be seen, some of the background and hair pixels are being classified as flesh pixels, and since they are connected to the facial flesh area they are incorrectly being converted into the darker skin tone of the target face. A simple heuristic approach that assumes the flesh pixels should fall within the elliptical face region may help to reduce such problems.

Figure 10 – Effect of false positive flesh pixels

Since our illumination estimation algorithm recovers intensities for three plausible light sources, we attempted to attach some virtual objects around the original face M_S instead of replacing it. Figure 11 shows the result of augmenting a virtual hat into the scene. As can be seen, the hat is illuminated in a convincing manner to match the illumination in the photograph. Note however that the hat does not currently cast any shadows into the scene. Additionally, the lighting can only be considered accurate in areas close to the face since the face is the only surface considered in our least squared lighting solution.

Figure 11 – Augmentation of a virtual hat

7. Discussion and Future Work

While the results we achieved were quite good in most cases, there are a number of areas that could be significantly improved. As mentioned earlier, a more advanced lighting model should be considered in order to handle specular reflections, attached and cast shadows, and sub-skin light scattering, similar to the work described in [3, 8].

The work in this paper focused on the relighting and merging of a target face over top of a source face in a single image. However, as described earlier, there are a number of areas that still need to be addressed in order to realize a complete digital face replacement system for video sequences. The first component that needs to be integrated into the existing system would be a head pose tracking algorithm. Since this is an active area of research, many algorithms exist that can be used for our face replacement tasks [2, 14]. Unfortunately, many of them are not robust to arbitrary illumination conditions or occlusions, so there is still much work to be done.

Realistically transferring facial expressions from the source face into the target face would also be an interesting component to add into the face replacement system. Much work has been done in this area recently [1, 6, 10], but issues related to illumination robustness, automatic operation, and reliability in a single image still need to be addressed.

The use of a generic head model is advantageous since it simplifies the mapping between our source and target faces. The major disadvantage, however, is that it is difficult to precisely align the mesh onto faces in non-frontal poses. Additionally, our interactive alignment tool currently does not attempt to recover the camera focal length in the original image, so the user is required to make some rough estimates. In most cases, however, the approximations are sufficient since the user is given full freedom to shrink or stretch the generic mesh as desired. Some advanced model-fitting algorithms as described in [1, 7, 14] would again be useful in order to automate the head pose determination process, as well as recover more accurate camera parameters.

8. Conclusion

In this paper we have presented a technique for replacing a human face in a single 2D image with the face of another individual. The method estimates the lighting conditions in the original image using a generic 3D face model, and applies the illumination to the same generic model that has been fitted over the replacement face. The newly lit face is then merged into the image over top of the original face, while a skin-color correction algorithm adjusts original face pixels not contained within the generic model area to match the skin tone of the new face. The techniques described here suggest several areas for future research, such as combining this work with expression synthesis techniques and robust head pose tracking, in order to fully realize a complete face replacement system for full-motion video sequences.

Acknowledgments

Thanks goes to Approach Infinity Media for allowing us to use some of their model images in our skin tone database. Aleix Martinez deserves a thank you for giving us access to the AR face database [9]. Finally, Kyros Kutulakos deserves a thank you for teaching an excellent Visual Modeling course that fuses the fields of computer graphics and computer vision.

References

[1] V. Blanz, T. Vetter. "A Morphable Model for the Synthesis of 3D Faces". In Proceedings of ACM SIGGRAPH, pp. 187-194 (1999).

[2] G. Bradski. "Computer Vision Face Tracking For Use in a Perceptual User Interface". In Intel Technology Journal Q2 (1998).

[3] P. Debevec, et al. "Acquiring the Reflectance Field of a Human Face". In Proceedings of ACM SIGGRAPH, pp. 145-156 (2000).

[4] M. Jones, J. Rehg. "Statistical Color Models with Application to Skin Detection". Technical Report CRL 98/11, Cambridge Research Laboratory (1998).

[5] J. Kleiser. "Kleiser-Walczak on *The One*". Available at: http://www.kwcc.com/works/ff/the_one.html

[6] Z. Liu, Y. Shan, Z. Zhang. "Expressive Expression Mapping with Ratio Images". In Proceedings of ACM SIGGRAPH, pp. 271-276 (2001).

[7] Z. Liu, Z. Zhang, C. Jacobs, M. Cohen. "Rapid Modeling of Animated Faces From Video". Technical Report MSR-TR-2000-11, Microsoft Research (2000).

[8] S. Marschner. "Inverse Rendering for Computer Graphics". PhD Thesis, Cornell University (1998).

[9] A.M. Martinez and R. Benavente. *The AR Face Database*. CVC Technical Report #24, June 1998.

[10] F. Pighin, et al. "Synthesizing Realistic Facial Expressions from Photographs". In Proceedings of ACM SIGGRAPH, pp. 75-84 (1998).

[11] T. Riklin-Raviv, A. Shashua. "The Quotient Image: Class-Based Re-rendering and Recognition with Vary-

ing Illuminations". In IEEE Conference on Computer Vision and Pattern Recognition, pp. 566-571 (1999).

[12] A. Shashua. "On Photometric Issues in 3D Visual Recognition from a Single 2D Image". In International Journal of Computer Vision 21 (1/2), pp. 99-122 (1997).

[13] I. Shimizu, Z. Zhang, S. Akamatsu, K. Deguchi. "Head Pose Determination from One Image Using a Generic Model". In Proceedings of Third IEEE International Conference on Automatic Face and Gesture Recognition, pp. 100-105 (1998).

[14] R. Yang, Z. Zhang. "Model-based Head Pose Tracking with Stereovision". In Proceedings of Fifth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 255-260 (2002).