Face Swapping

Yifei Feng
Microsoft Corporation
Mountain View, CA

Wenxun Huang
Microsoft Corporation
Mountain View, CA

Tony Wu
Department of Electrical Engineering
Stanford University, Stanford, CA

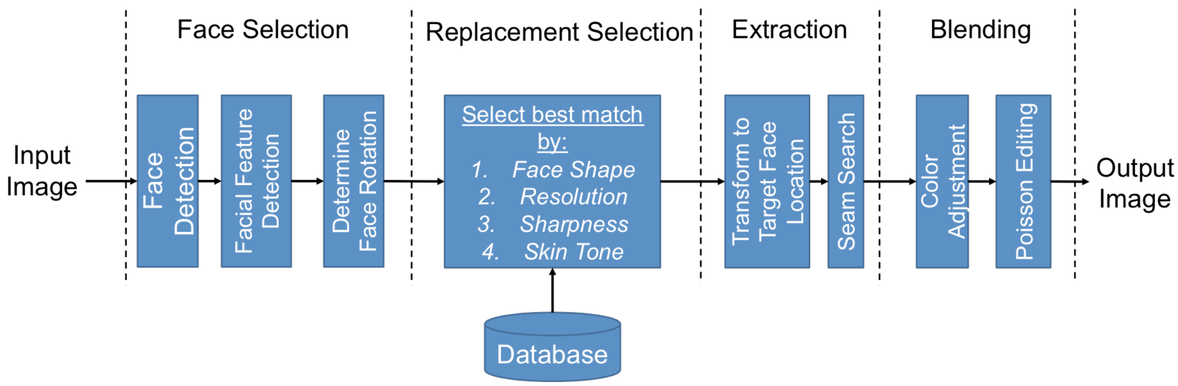

Figure 1. Automatic face swapping algorithm.

***Abstract*—Online privacy is a growing concern in today's world. To alleviate privacy concerns, we present an automatic face swapping algorithm that searches and selects faces within an image and replaces them with another face with realistic results. Our algorithm identifies and uses the most suitable faces using a selection process, face extraction using optimal seam search, and blending using linear RGB channel scaling and Poisson Editing.**

Keywords—facial recognition; image blending;

I. Introduction

Advances in digital photography and online search has made it possible to find photographs with great ease. While the availability of large amounts of these images can be hailed as a breakthrough for convenience, there is a growing concern about online privacy associated with image search. Online systems such as Google Street View allow users to browse photos of public images. Many of these photos contain people who have not consented to be photographed, much less for commercial purposes. While the current practice of obfuscating face regions using blurring or pixelation, it can often decrease the visual appeal of the image. One solution to this problem is to replace every face in the image with stock faces.

Our face swapping algorithm automatically selects front-facing faces and replaces them with stock faces, similar to [1]. However, our algorithm does not use illumination estimation and correction and pose estimation and correciton. The algorithm includes four major steps to generate a realistic looking composite image (Figure 1):

1. Face selection
2. Replacement selection
3. Extraction
4. Blending

Section II discusses the database creation method we used for our experiments. Section III through Section V discusses our algorithm in detail. Section VI provides a demonstration of results from our algorithm and presents some of its limitations. We conclude in Section VII with future work to be done.

II. Database Creation

Suitable face replacements for a query image were taken from a database of images consisting of well known faces. We hand selected 264 frontal faces from the LFW database [2]. Each image was selected based on a few loose requirements: 1. No objects obstructing the face (including glasses). 2. Faces must be front facing.

After the images were curated, the location of each face in the form of a bounding box was detected for each image using the Viola-Jones face detection algorithm [3] (Figure 2a, Figure 2b). Since the algorithm often detects many faces, the detected faces are ranked by the area of their associated bounding box and thefface with the largest bounding box is taken as the face for that database image.

Then, facial landmark features were extracted using the flandmark facial landmark detector [4]. This extracted the inside and outside corners of each eye, the nose, the corners of the mouth, and the center of the face (Figure 2b). The centers of each eye was calculated as the midpoint of the two eye corners. The angle of the face, defined below, was calculated from the locations of the left and right eyes. These locations were also stored in the database along with the output of the facial landmark detector.

$$\theta = \arcsin\left(\frac{y_{left} - y_{right}}{x_{left} - x_{right}}\right)$$

Additionally, a histogram of the magnitude of the gradient of the eye region was also kept. To calculate the histogram, a rectangular region of the eye region was cropped from the face and converted to grayscale. The length of the crop region was determined by the distance between the outer corners of the eyes. The width of the crop region is taken as half of that distance. The grayscale image is then normalized by subtracting the mean value and dividing by the standard deviation. This was to account for illumination variations across images. The gradient of the normalized image was taken and a histogram of 256 bins was taken from its magnitude and stored in the database.

III. Replacement Selection

For each query image, we first detect the locations of the faces with the Viola-Jones face detection algorithm. Then, the facial landmark features were calculated for each face using flandmark. The angle for each face was also calculated as described in the previous section.

For each face (destination face), the database is searched for a suitable face for replacement based on 4 criteria: face shape, image resolution, sharpness, and skin tone.

A. Face Shape

We apply an Affine transformation to the facial landmarks of each database face to scale, rotate, and translate them onto the destination face. The mean squared error (MSE) is calculated with the transformed facial landmarks of the database face and those of the destination face. This value is then multiplied by the nose weight. The nose weight is a Boolean value derived from comparing the yaw rotation of the faces as described below:

$$r = \frac{x_{nose} - x_{center}}{\Delta x_{eye}}$$

$$\Delta r = |r_{destination} - r_{database}|$$

$$W_{nose} = \begin{cases} 1 & \Delta r < R_T \\ 0 & \Delta r \geq R_T \end{cases}$$

where x_{nose} and x_{center} are the x coordinates of the nose and the center of the face, respectively, and Δx_{eye} is the distance between the outer corners of the eyes. If Δr is less than a certain threshold, R_T, the weight is 1. Otherwise, it is 0. The product of the MSE and the nose weights are taken and sorted in increasing order. We keep the top 20 images from the list.

B. Resolution

Next, we filter through the sorted list of database images according to resolution. We take the number of pixels in the bounding box of the face as the resolution. If the resolution of the database face is at least 50% of the resolution of destination face, the database face is kept. Otherwise, it is removed from the list.

C. Sharpness

Then, we filter by the sharpness of the eye region. The histogram of the magnitude gradient of the eye region is first calculated for the destintation face. Then, overlapping area of the histograms of the database and destination faces is calculated. The list of remaining suitable database faces is then sorted according to large area overlap.

D. Skin Tone

The final step of the database image selection process involves filtering out faces with skin tones that area too far

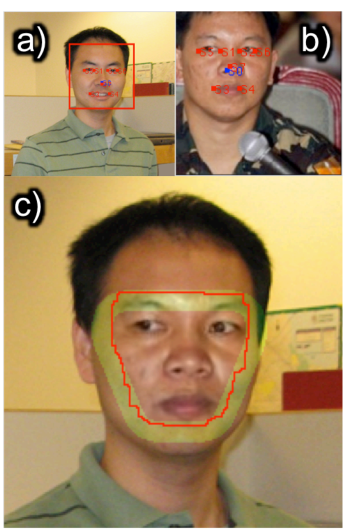

Figure 2. a) Viola-Jones face recognition and flandmark feature detection. b) Feature detection of source image. c) Optimal seam selection.

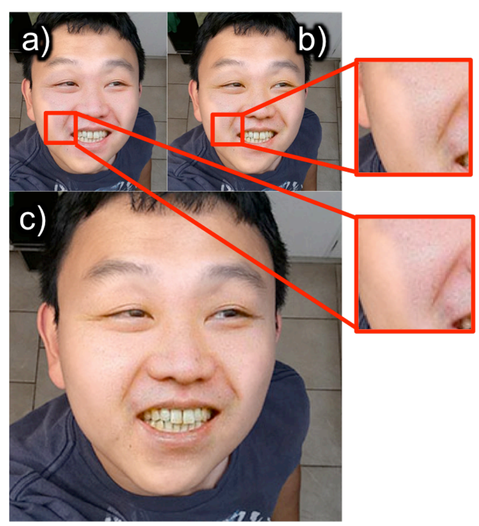

Figure 3. Image blending. a) No blending. b) After linear RGB transformation. The resulting image is less red. c) After Poisson image editing. The seam becomes unnoticeable.

from that of the destination face. Skin tone selection is based on the hue and saturation of the faces. The hue and saturation ranges for the skin tones are described in [5]. If both the difference in hue and saturation of the two faces is below a threshold, we keep the database face in the selection. Otherwise, it is removed. The first image in the resulting list is taken as the source face for the face swapping.

IV. Optimal Seam Search

Next, we determine the optimal area to extract from the source face. First, we must transform it to the correct location. An affine transformation is performed on the source image to transform the face to the location of the destination face. To avoid discontinuity from cutting through important facial features such as eyebrows, wrinkles, or specular highlights, a seam search algorithm is performed on the aligned images. In this step, we convert the images to CIELAB color space to better correlate visual differences with pixel value differences. The gradient of each image is calculated and normalized to be a value between 0 and 1 in order to reduce the effects of different illumination. The sum of the gradient values will be used as a cost function.

The optimal seam search is formulated as a shortest path problem. In contrast to [6]'s work, here we need to find a closed contour around the face region instead of a line from edge to edge that minimizes the sum of the gradient values on the path. To reduce the amount of computation, we constrain the search to a permissible path region based on the facial landmarks (as indicated with green color in Figure 2c). An adjacency matrix is built based on 4-neighborhood connectivity of pixels within the path region. The cost of adding a pixel on the path is set to be the magnitude of gradient at that location. In order to enforce a circular path that encloses the face, a cut is made to divide the ring shaped path region. Along the cut, there is no path crossing from left to right. If optimal solution is desired, one only needs to exhaustively search all shortest paths starting from the pixels along the cut and ending at the pixel to the immediate right of the starting pixel. However, this is obviously quite computationally expensive. We believe a possible compromise is to position the cut through the forehead region, which is quite often relatively flat, and sample a few points (20 points used for this work) along the cut as start/end position pairs. We find the shortest path with a MATLAB

Figure 4. Face swapping results. Each row contains (from left to right) the original photograph, a candidate face selected from the database, and the composite face.

Dijkstra implementation from [7].

V. Color Blending

After an optimal seam has been found, we run two different color blending methods to create a realistic composite, Figure 3.

A. *Color Adjustment*

The first step of the color blending process is a linear scaling of the R, G, and B channels. To find the scaling factor for each channel, the boundary pixels from the seam are extracted for both faces and a linear transformation is found by minimizing the L2 norm of the pixel values. The scaling factors for each channel are found separately so no channel mixing is allowed. The transformation is then applied to all of the pixels in source face. This prepares the image for Poisson Image Editing as the algorithm assumes the images are of the same general tone (Figure 3b).

B. *Poisson Image Editing*

To create an even more realistic composite, we perform Poisson Image Editing on the [8] composite face. First, a composite image is created by superimposing the color adjusted source face on the distance image. A mask is made to denote the source face from the distance image. The algorithm reconstructs the source face from its gradient based while using the boundary values from the destination face. The result is a realistic looking composite with no visible seam (Figure 3c).

VI. Results

We tested our algorithm on a subset of MUCT face database [9] with moderate lighting condition and front facing subjects without glasses. Examples of typical results obtained using our system are shown in Figure 4. Each example shows, in order from left to right, the query face, the selected face from database, and the composite face. Note that with our 4-step face selection process, the chosen database faces are similar to the pose and skin tone of the query faces. Seam search allows a relative clear part of the face to be replaced. For example, in Figure 4h, our system keeps the mustache from the database image, but avoids the beard. In the end, with color adjustment and Poisson image editing, the composite faces look realistic, despite the difference in lighting and skin tone between query image and database image. This set of results also show that our system work well across different race, gender and age.

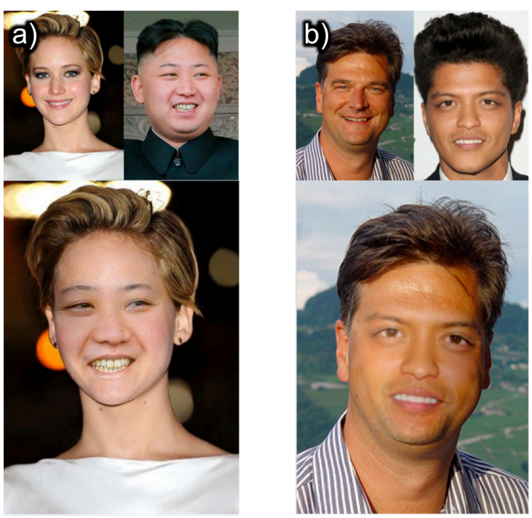

Figure 5. Limitations. a) When two faces are facing opposite directions, the composite image can look unrealistic. b) When illumination direction is drastically different in the two faces, the Poisson Image Editing algorithm results in washed out areas where the shadows must be brightened into highlights.

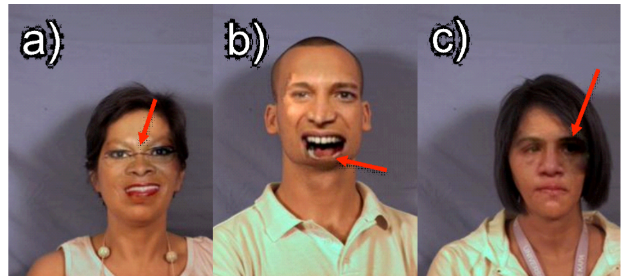

Figure 6. More limitations. a) Artifacts due to subject's glasses. b) Incorrect facial feature size due to open mouth. c) Color bleeding after Poisson blending when the seam (inevitably in this case) crosses non-skin region.

VII. Future Work

A. *Pose Estimation and Correction*

For our project, we only considered front facing faces with in-plane rotation. To correct for yaw and pitch of the face, a 3D model would need to be considered. An example of the limitations of our method for pose differences can be seen in Figure 5a.

B. *Illumination Direction Correction*

We also noticed that image blending when illumination directions were drastically different yielded unrealistic results, Figure 5b. For future work, we would need to correct for illumination effects, possibly using a linear combination of spherical harmonics to approximate the light sources [10].

C. *Artifacts Removal*

Artifact in Figure 6a can potentially be fixed by concealment method described in [8]. Color bleeding in Figure 6c can potentially be fixed by mixed gradient described in [8].

D. *Facial Expression Correction*

As shown in Figure 6b, the jaw of result face becomes disproportionately small since the mouth is wide open in the source image. This is because our landmark detector only detects corners of the mouth. In future work, the face shape filter should include the mouth when calculating the face aspect ratio.

VIII. Conclusion

An automatic image swapping algorithm has been developed without the need for 3D models. The algorithm utilizes selection criteria to select the most suitable image, computes the optimal seam for extraction, and blends the faces using linear RGB transformations and Poisson Image Editing to produce realistic composite faces.

Acknowledgment

We would like to thank Professor Bernd Girod of Stanford University for his teachings this quarter. We would also like thank Huizhong Chen for his guidance on our project.

Work Breakdown

Yifei Feng: Wrote code for database image selection and curated database.

Wenxun Huang: Wrote code for seam selection and color blending

Tony Wu: Wrote code for face detection and integrated parts of code.

References

[1] D. Bitouk, et al. "Face swapping: automatically replacing faces in photographs." *ACM Transactions on Graphics (TOG)*. Vol. 27. No. 3. ACM, 2008.

[2] G. B.Huang, et al. “Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained” Environments. *University of Massachusetts, Amherst, Technical Report 07-49*, October, 2007.

[3] P. Viola, and M. Jones. "Rapid object detection using a boosted cascade of simple features." *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Vol. 1. IEEE, 2001.

[4] M. Uřičář, V. Franc, and V. Hlaváč. "Detector of facial landmarks learned by the structured output SVM." *VISAPP* (2012): 547-556.

[5] C. Prema, and D. Manimegalai. "Survey on Skin Tone Detection using Color Spaces." *International Journal of Applied* 2.2 (2012): 18-26R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] S. Avidan, and A. Shamir. "Seam carving for content-aware image resizing." ACM Transactions on graphics (TOG). Vol. 26. No. 3. ACM, 2007.

[7] “MATLAB File Exchange: Dijkstra Algorithm”, http://www.mathworks.com/matlabcentral/fileexchange/36140-dijkstra-algorithm

[8] P. P´ erez, M. Gangnet, and A. Blake. "Poisson image editing."*ACM Transactions on Graphics (TOG)*. Vol. 22. No. 3. ACM, 2003.

[9] Milborrow, Stephen, John Morkel, and Fred Nicolls. "The MUCT landmarked face database." *Pattern Recognition Association of South Africa* 1 (2010): 4.

[10] R. Basri, and D. W. Jacobs. "Lambertian reflectance and linear subspaces." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*25.2 (2003): 218-233.

[11] V. Blanz, et al. "Exchanging faces in images." *Computer Graphics Forum*. Vol. 23. No. 3. Blackwell Publishing, Inc, 2004.

[12] A. Levin, et al. "Seamless image stitching in the gradient domain." *Computer Vision-ECCV 2004*. Springer Berlin Heidelberg, 2004. 377-389.