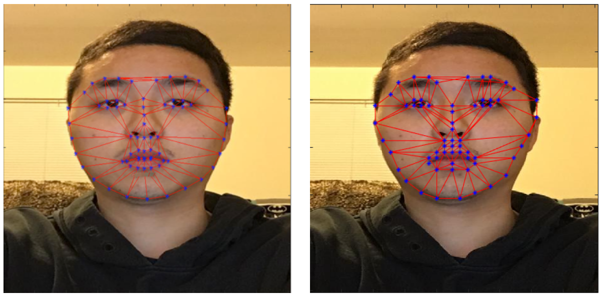
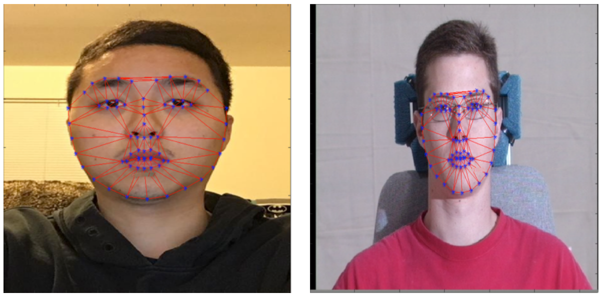
Computer Processing of Pictorial Information Project 2

Huai-Jen Liang
Department of Electrical and Computer Engineering
University of Maryland
College Park, Maryland 20740
Email: r81810g.umd@gmail.com



(a) DLib (b) X. Zhu

Fig. 1: Fiducial detection.



(a) Triangulation of source face (b) Triangulation of target face

Fig. 2: Triangulation.

I. Introduction

This project contains an end-to-end pipeline to do face swapping. To form a more interesting problem, we applied this procedure on a video. The results are presented in Fig.5.

II. Method

In this section, we describe the pipeline of face swapping. At first, we assume we are going to swap faces in a pair of images. Then, we describe the way we handled a video.

A. Face Detection with Landmarks

In order to determine the matches between two faces, we need to find fiducial points as features first. In our method, we used DLib[1] or X. Zhu's code[2] to extract 18 fiducial points. The result is shown in Fig.1. We can notice that in Fig.1b, the corner of the left eye is wrong, so we choose DLib[1] as our face detection.

B. Face Wrapping

After extracting features, we have the point correspondences between two images. Using these correspondences, we can either utilize $Triangulation$ or $ThinPlateSpline$ to find the transformation.

1) Triangulation: We use the point correspondences to create the triangulation by **Delaunay triangulation**. One should notice that we only create the triangulation from destination face and apply the result to the source face. The triangulation results are shown in Fig.2. After having triangulation, we calculate inverse mapping by the following procedure:

(a) For each in the destination face, determine which triangle it falls inside.

(b) Calculate the barycentric coordinate for each pixel in the corresponding triangle by solving the equation:

$$\begin{bmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

where a, b, c are the three points of a triangle, (x, y) is the pixel position and (α, β, γ) is a barycentric coordinate.

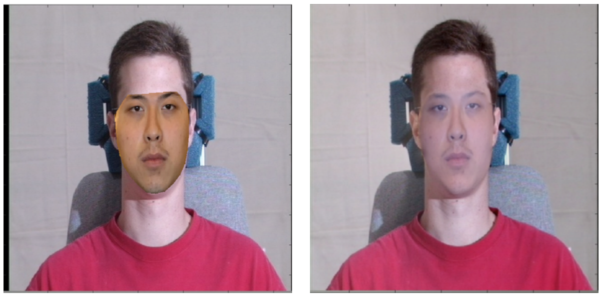
(c) Compute the corresponding pixel position in the source image (x^s, y^s) by

$$\begin{bmatrix} a_x^s & b_x^s & c_x^s \\ a_y^s & b_y^s & c_y^s \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} x^s \\ y^s \\ z^s \end{bmatrix} \tag{2}$$

where a^s, b^s, c^s are the three points of a triangle in the source image and (α, β, γ) is the barycentric coordinate from Eq.2. Notice that we need to normalize the pixel position, that is, $x^s = x^s/z^s$ and $y^s = y^s/z^s$.

(d) Copy the pixel from (x^s, y^s) to (x, y) using linear interpolation.

2) Thin Plate Spline: Since a Thin Plate Spline(TPS) can model arbitrary complex plane, it might be better to use Thin Plate Spline instead of triangulation. Recall the form of a Thin Plate Spline:



(a) Without blending (b) With blending

Fig. 3: One can easily observes that we can get a better result if we blend the source face on the target image.

$$f(x,y) = a_1 + a_x x + a_y y + \sum_{i=1}^{p} w_i U(\|(x_i, y_i) - (x,y)\|) \quad (3)$$

where $U(r) = -r^2 log(r^2)$. Given the corresponding points, the unknown will be $W = a_1, a_x, a_y, w_1, ..., w_p$. We can solve it by

$$\begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_p \\ a_x \\ a_y \\ 1 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_p \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4)$$

where $K_{ij} = U(\|(x_i, y_i) - (x_j, y_j)\|)$, $v_i = f(x_i, y_i)$, and the i_{th} row of P is $(x_i, y_i, 1)$. We need to solve two Thin Plate Spline, $f_x(x,y)$ and $f_y(x,y)$. After that, we can copy pixel from $(f_x(x,y), f_y(x,y))$ to (x,y) using linear interpolation.

C. Blending two images

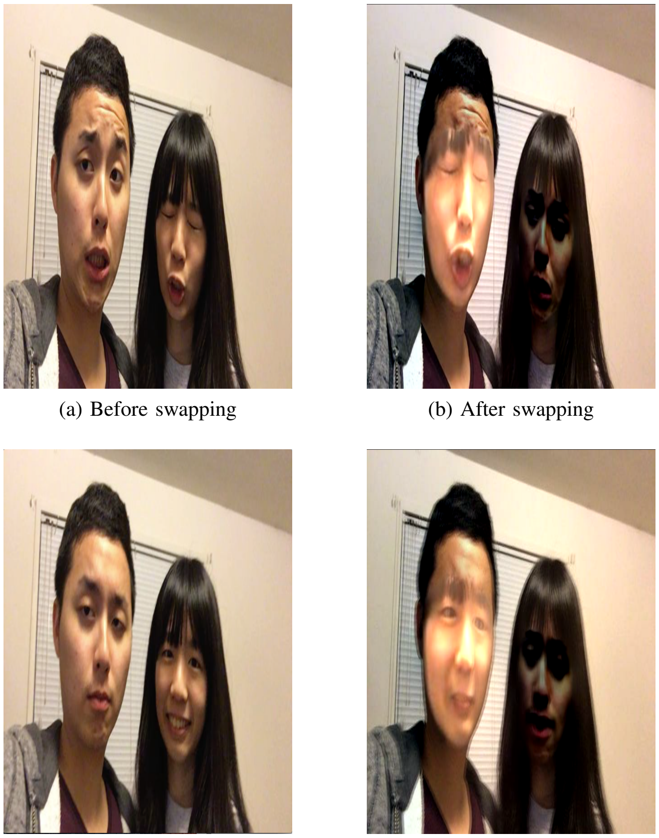
In order to combine source's face onto target image2 without strange artifacts, we have to blend two images. We applied Possion Blending[3] and the results are shown in Fig.3.

D. Motion Filtering

In order to handle vast motion, we simply interpolate k frames between the previous frame and the current frame. We first calculate the difference between these two frames and divide it to $k+1$ portions. At each step we add one portion to the previous frame and insert this frame as the previous frame for the next step.

III. Results

We show the results in Fig.5. We can observe that using Thin Plate Spline or Triangulation can get a similar result. However, in Fig.5b, TPS doesn't do well around the nose. The reason is that TPS thinks the features do not lie on a plane, and it fit the best model to the features. After wrapping the face, the shadow becomes strange, the nose is too sharp and the corner of an eye is stretched. From Fig.5d, we can also observe that if face detection fails, the wrapping result will be disaster. However, in Fig.5d, TPS wrapping gives a better result. Because the infant's face is twisted, the triangle is no longer a plane and TPS can have a better model. For swapping two faces in the same images, as in Fig.4, we have a reasonable result. However, if there is no corresponding pixel in the source image, it will give a bad result. One can observe that the mouth in Fig.4d becomes creepy. Another issue is luminance. If the luminance of two faces varies a lot, the result will be strange. The results of the test set are shown in Fig.6. For the third test set, we cannot do well. Since there is flash light, we cannot distinguish a face when the luminance is unnormal.



(a) Before swapping (b) After swapping

(c) Before swapping (Fail) (d) Before swapping (Fail)

Fig. 4: One can easily observes that we get a bad result if we didn't have corresponging pixel in the source image.

IV. Acknowledgement

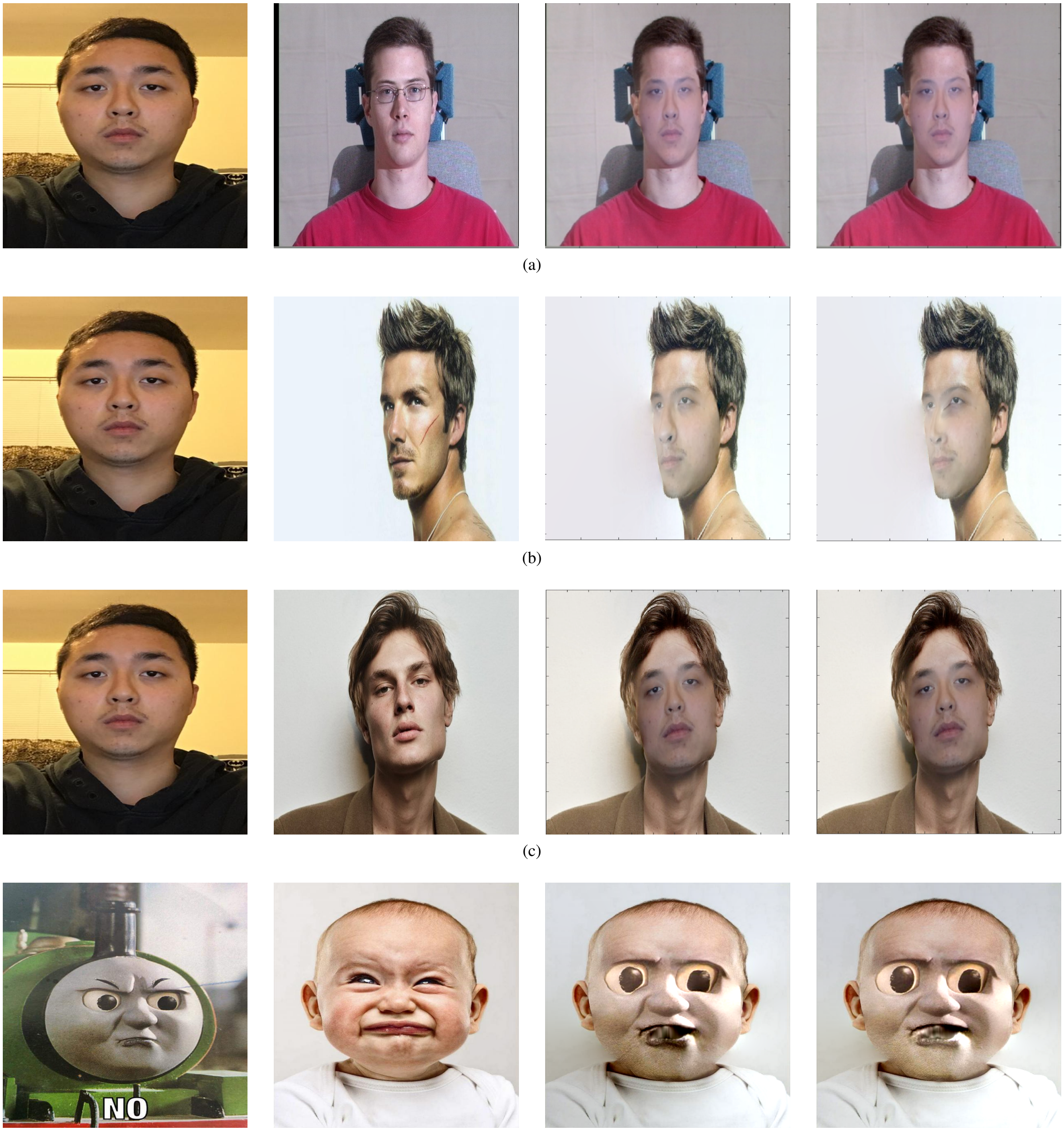
Special thanks Wei-An Lin for discussion.

References

[1] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[2] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012.

[3] Masayuki Tanaka, Ryo Kamio, and Masatoshi Okutomi. Seamless image cloning by a closed form solution of a modified poisson problem. In *SIGGRAPH Asia 2012 Posters*, SA '12. ACM, 2012.



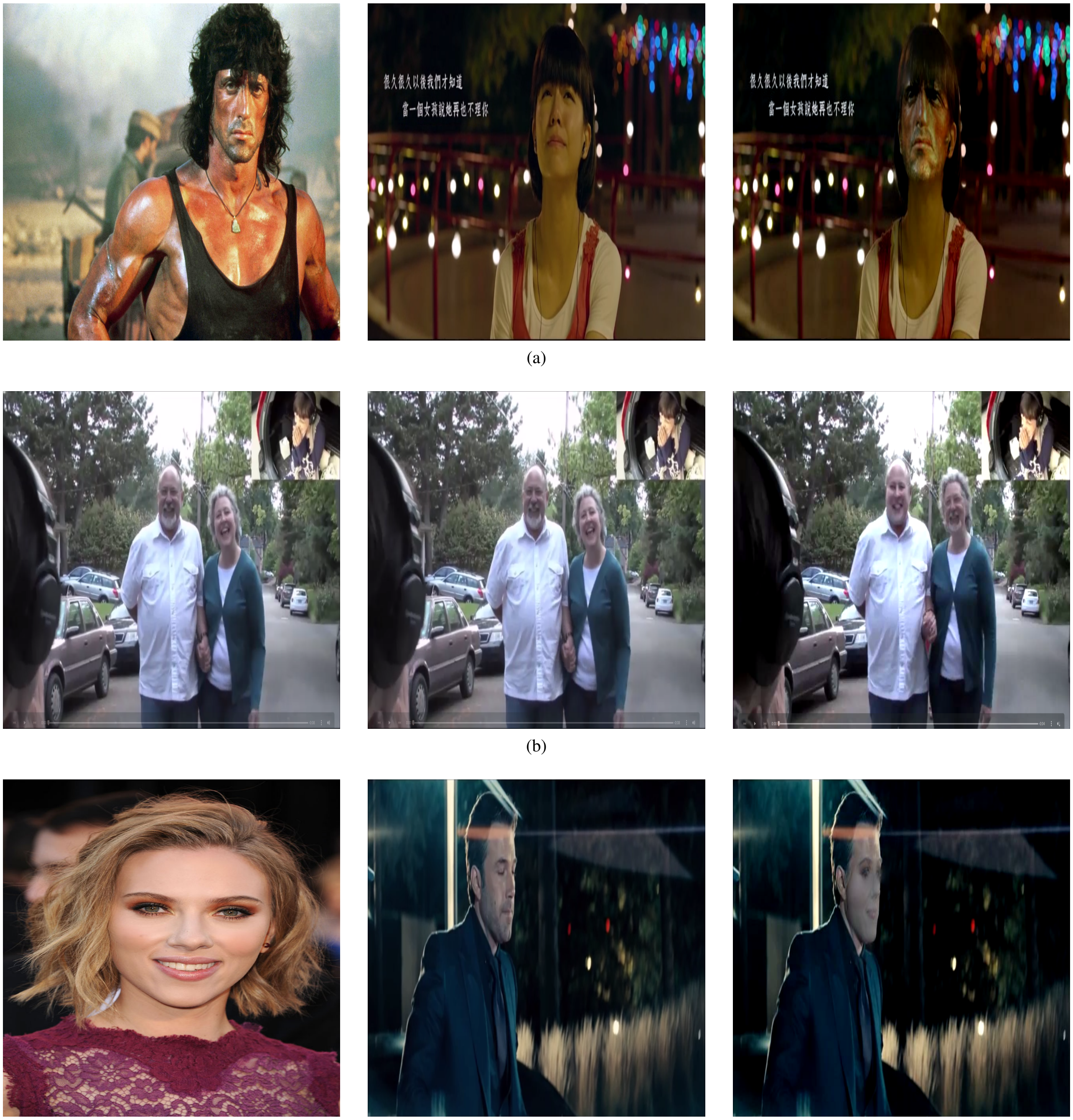
(a)

(b)

(c)

(d)

Fig. 5: Face swap. From left to right: Source, Target, Triangulation wrapping, TPS wrapping



(a)

(b)

(c)

Fig. 6: Face swap. From left to right: Source, Target, Wrapping