Project 2: FaceSwap
Using Two (2) Late Days

Nitin Suresh
School of Electrical and
Computer Engineering
University of Maryland - College Park
UID: 113638855

Jo Shoemaker
Department of Computer Science
University of Maryland - College Park
UID: 115506787

I. Our Data

II. Phase 1: Traditional Approach

In this section of the project, we implemented face swapping using two traditional geometric approaches - Delaunay triangulation, and Thin Plate Splines (TPS). Face bounding boxes and landmarks positions were first detected using the dlib library. These points were returned with reasonably good accuracy.

A. Face Warping using Thin Plate Splines

The general equation as provided in the problem statement was used for generating TPS equations for mapping to both the x-coordinate and the y-coordinate. We obtain the points representing face fiducials from the video frames and the image to replace the face. The convex hull of the source face is used as a mask to extract the face region that should be swapped. Using this convex hull eliminates the box effect that otherwise appears on using just a bounding box. Inverse mapping is then carried out in order to avoid gaps or holes in the warped image. After warping, the seamlessClone function was used to blend the two images together such as to minimize the boundary effect.

Video flicker was reduced using two techniques - image filtering and frame simulation. After warping and blending the image, each frame was filtered using a bilateral filter in order to smooth out edge transitions and jitter. Secondly, intermediate frames were simulated and placed between original frames in the initial video, using mean facial feature translation vectors.

Sample output images are displayed in Fig. 1 – Fig. 6. In all cases, the left image is an original frame, while the right image is the swapped frame.

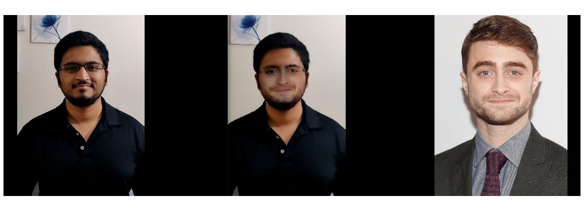

Fig. 1. Single face swap

Fig. 2. Double face swap

Fig. 3. Test Video 1

Fig. 4. Test Video 1

Fig. 5. Test Video 2

Fig. 6. Test Video 3

Fig. 7. Test Video 3

B. Face Warping using Triangulation

Using the same facial landmarks identified by `dlib` as for TPS, we found a matching Delaunay Triangulation for our source and target faces. Each source triangle's values could then be warped to fill in each corresponding target triangle. The resulting warped face "patch" was then blended into the target image using `seamlessClone`. Fig. 8–Fig. 12 show frames from each of our sample videos. In the interest of time, videos were processed with facial landmarks downsampled to 17.

Fig. 8. Single face swap

Fig. 9. Double face swap

Fig. 10. Test Video 1

Fig. 11. Test Video 2

Fig. 12. Test Video 3

III. Phase 2: Deep Learning Approach

We used the pretrained model from [1] to obtain a 3-dimensional representation of the locations of facial landmarks. We then warped faces following the same technique as used in that paper. Bilateral filtering was used to minimize the flicker between frames. Fig. 13–Fig. 17 show frames from each of our sample videos.

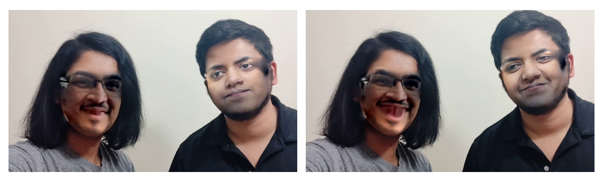

Fig. 13. Single face swap

Fig. 14. Double face swap

Fig. 15. Test Video 1

Fig. 16. Test Video 2

Fig. 17. Test Video 3

IV. Failure cases

In some cases, such as dark image frames, faces were not detected. This could be due to insufficient matches obtained by the dlib library. Also, the TPS approach works best when the faces are pointed towards the camera. Slightly angled faces also give reasonably good results, but when the angle of the face is too large, warped image quality decreases. Some example failure cases are as shown in Fig. 18 and Fig. 19. Our implementation skips over frames where the requisite number of faces are not detected.

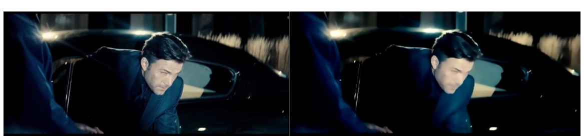

Fig. 18. Test Video 3 fail case

Fig. 19. Test Video 2 fail case

References

[1] Feng, Y., Wu, F., Shao, X., Wang, Y., and Zhou, X. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV* (2018).