CMSC733: Project 2, FaceSwap!

Naveen, Mangla
nmangla@umd.edu

Bhargav Kumar Soothram
bsoothra@umd.edu

I. Phase 1

A. Algorithm

This phase include the Traditional way of swapping faces. We will provide details of our approach for two faces in a single frame and two different faces in two different frames. We then extrapolated our results to videos by computing them frame by frame. The steps are as follows

1) Extracting Landmarks.
2) Warping
   a) Delaunay Triangles
   b) Thin Plate Spline
3) Blending

Steps are explained as follows:

B. Extracting Landmarks

For this section. We computed 68 landmarks for each face provided by dlib library.

Fig. 1: Landmarks for Two faces in Single frames

Fig. 2: Landmarks for Two faces in two different frames

C. Warping

1) Triangulation: In this section. We will use the detected facial landmarks to create the Delaunay Triangles. This algorithm tries to maximize the smallest angle in each triangle. In this way we can find correspondence between 2 faces. Since indexes of landmarks remains same for every face. We used these indexes to sort triangles in target face according to source face.

Fig. 3: Triangulation for Two faces in Single frames

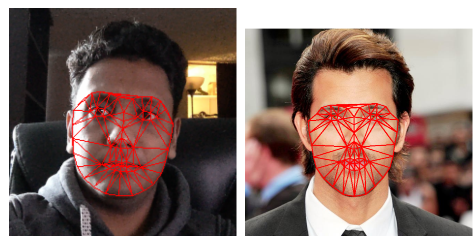

Fig. 4: Triangles for Two faces in Different frames

In order to Swap faces from source (face) A to target (face) B , we used Barycentric Coordinates using following equations.

$$\begin{pmatrix} B_{a,x} & B_{b,x} & B_{c,x} \\ B_{a,y} & B_{b,y} & B_{c,y} \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

for pixel (x,y) to be in triangle (a,b,c)

1) $\alpha\, \beta\, \gamma$ will be in set [0,1]
2) $0 < \alpha + \beta + \gamma <= 1$

Using Values of $\alpha\ \beta\ \gamma$, corresponding values of (x,y) was computed in face *A*. Bilinear Interpolation was used to get rid of the holes.

2) Thin Plate Spline : Since, Triangulation gives us decent results but are not much appealing as it looks as some planner warming, on the other hand thin Plate spline algorithm gives us ore promising results.

D. Blending

In order to blend images, we used seamless cloning. Results can be seen below

Fig. 5: Swapped Faces Masks using Delaunay Triangles

Fig. 6: Swapped Faces Masks using Thin Plate Spline

II. Phase 2: Deep Learning Approach

In this phase, we set out to get the 3D face fiducials using an off the shelf model proposed by Feng et al. in their work "Joint 3D Face Reconstruction and Dense Alignment with Position Map Regression Network". In this work, a 2D representation called UV position map is designed, which records the 3D shape of a complete face in UV space, then train a simple Convolutional Neural Network to regress it from a single 2D image. The results obtained using this approach are given in figure. We found satisfactory results in swapping faces for most parts, except for a few frames in the test set 2, where we used the original frame from the video.

III. Inferences and Conclusion

We have observed that the results obtained using the deep learning approach of the PRNet were better compared to the ones obtained using the two traditional approaches. This is because the deep learning approach obtains a full 3D mesh of the face and then warps it onto the face (TPS only "approximates a 3D surface" rather than create one).
There are also a few edge cases where we get some strange results. For example in figure 11, since the spectacles are regarded as part of the face texture when the subject is facing the camera, the entire face is warped over, but when the subject faces sideways, the part of the spectacles that is not

Fig. 7: Delaunay Swapping Result

Fig. 8: Thin Plate Spline Swapping Result

Fig. 9: PRN Swapping

Fig. 10: Denaulay vs Thin Plate Spline Swapping

on the face is shown too. To avoid artefacts like this, we have explored the use of semantic segmentation (clustering) to separate foreground from the background. However, due to time constraints, the results were not satisfactory enough to be shown here.

Fig. 11: Top: Subject facing the camera, Bottom: Subject sideways

IV. Test Results

Fig. 12: Test1 Using Delaunay Triangles

Fig. 13: Test1 Using Thin Plate Spline

Fig. 14: Test1 Using PRNet

Fig. 15: Test2 Using Delaunay Triangles

Fig. 16: Test2 Using Thin Plate Spline

Fig. 17: Test2 Using PRNet

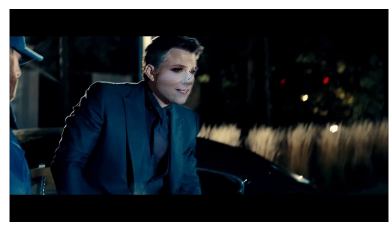

Fig. 18: Test3 Using Delaunay Triangles

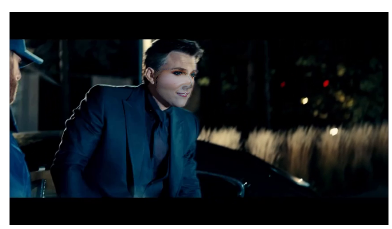

Fig. 19: Test3 Using Thin Plate Spline

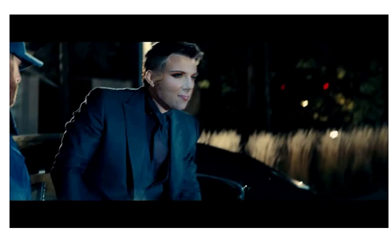

Fig. 20: Test3 Using PRNet