CIS 581 Project Proposal

Shikhar Brajesh, Abhijeet Singh
{shikharb, abhsingh}@seas.upenn.edu

We will be working on the **Track2 - Face Swapping** for the final project.

Face swapping is the process of detecting faces from two videos and swapping these faces - while making it look as natural as possible.

Proposed pipeline

We plan to implement the following variants for the face swapping pipeline -

1. Swap a face from one video to another using the following steps -
    a. Iterate over each frame of the source and target
    b. Facial landmark detection (like nose, eyes, chin etc) in both the source and target frame
    c. Creating a convex hull of the face from the detected features
    d. Making a Delaunay triangulation from the face features in the target frame
    e. Warping each region (region = a single Delaunay Triangle) of the source frame to the target frame.
    f. Blend the resulting image into the target body
2. Improve upon the above proposed pipeline by incorporating optical flow. In the existing pipeline, we will be swapping each source frame with the target frame which might lead to a lot of jitter. We will run the face swapping algorithm (as described in pipeline 1) every i^{th} frame (say *i=5*). For the rest of the frames, we will keep the previous swapped frame and use *optical flow* to transform the face to the target frame's face pose.
3. We also plan to look into using the information from multiple frames of the source video instead of swapping faces frame by frame into the target video. For example, we could look at the closest frame in the source video that matches each of the target frames in the target video.

Open source libraries

We plan to use the following open source libraries -

1. Face Recognition - https://face-recognition.readthedocs.io/en/latest/readme.html
2. Numpy, Scipy, OpenCV and the useful functions that they provide like *cv2.convexHull* (for calculating convex hull from a set of points), *cv2.calcOpticalFlowPyrLK* (for calculating optical flow), *scipy.spatial.Delaunay* (for calculating the Delaunay triangulation used for morphing), *cv2.seamlessClone* (for blending the source frame into the target frame) etc.