CS663 Project
Face Swapping

Team Members:
180050071 - Parth Laturia
180050101 - Shubham Gupta
180050106 - Sukhmanjit Singh Mann

Work Allocation

Sukhman - landmark detection, Affine transform for image alignment
Shubham - manual mask feature, optimizing Poisson blending
Parth - Poisson blending algorithm, helped in the mask feature

Problem Statement

Given a ‘source’ image and a ‘destination’ image, paste the face of the person in the source image on the destination image so that it looks natural.

Results

Fig. 1: Source and destination Images

Fig. 2: Swapped Images for Fig. 1

Fig. 3: Source and Destination Images

Fig. 4: Swapped Images for Fig. 3

Fig. 5: Source and Destination Images

Fig. 6: Swapped Images for Fig. 5

Fig. 7: Source and Destination Images

Fig. 8: Swapped Images for Fig. 7

Algorithm

1) Detect faces of persons in source and destination images using Viola-Jones algorithm (implemented in the vision library in MATLAB) and get facial landmarks (using open-source implementation).
2) Rotate the extracted face from the source image so that the eyes of the source and destination image are at the same angle.
3) Scale the rotated image so that distance between the eyes of the face in the destination image and the source image are at the same distance.
4) Translate the scaled image so that the eyes of the extracted source face and destination image align.
5) Ask the user to mark a mask on the transformed source image indicating the region of the face to be pasted.
6) Poisson blend each channel (R, G, B) of the source and destination image using the user-specified mask

Observations

1) We are using a sparse matrix for storing the coefficients in the linear equation system for Poisson blending memory conservation, and solving the linear equation for Poisson blending is faster using this construction.

2) After rotating and scaling the extracted face image, the size of the rotated image increases as imwarp in MATLAB does not crop the image to original size. So instead of running landmark feature detection again (which sometimes fails to detect features on the rotated image), we transform the landmark coordinates in the original image to increase efficiency and accuracy.

3) For implementing Poisson blending using sparse matrix, initially we used indexing to fill up the values one by one and it was a slow process, but when given all the values simultaneously speed drastically increased.

4) The facial landmark detector in the open-source implementation is not robust. We tried different images which were not in the dataset provided in the description and even in clear images it sometimes was not able to detect the facial landmarks. We had problems compiling the flandmark facial landmark detection library, which would have given better results.

5) If the difference in lighting and skin tone is low, Poisson blending makes the boundaries almost indistinguishable. It is very effective in maintaining the destination image within the mask and the source image outside while mixing the two images at the boundary.

6) If the source and destination images have pose variations, then affine transformation is not able to achieve acceptable results as the distance and alignment of eyes will be very different. Also, if the lighting conditions and skin tone are different, the resultant image looks very artificial, which is not desirable. So the choice of source and destination has a huge impact on the result. Some poor results because of pose differences:

Fig. 9: Source and Destination Images

Fig. 10: Swapped Images for Fig. 9

Improvements

1) A more generalized problem is given a source image and a set of destination images, find the most suitable destination image and swap the face of the source image. So first we have to filter the destination images, and we can do this based on the Face Shape, Resolution, Sharpness and hue and saturation (skin tone, lighting, etc.). This automatically selects better destination images so that the results look more realistic.

2) Instead of asking the user to specify the mask, it can be detected automatically by taking image gradients and then finding the contour surrounding facial landmarks which minimizes the sum of the gradient magnitudes. So it is a shortest path problem and the search space can be

constrained to a permissible space around the facial landmarks to increase the efficiency.

3) Using a better facial landmark detector will allow for more candidate images for our implementation.

References

1) https://stacks.stanford.edu/file/druid:bf950qp8995/Feng_Huang_Wu.pdf for Face Swapping reading
2) https://github.com/mahmoudnafifi/Poisson-image-editing/blob/main/ReadME.pdf for Poisson blending reading
3) https://in.mathworks.com/matlabcentral/fileexchange/47713-facial-landmarks?requestedDomain= for open source landmark detection.
4) http://vis-www.cs.umass.edu/lfw/ for face images