A Live Face Swapper

Shengtao Xiao [1,2] *
xiao_shengtao@u.nus.edu

Luoqi Liu [2]
liuluoqi@360.cn

Xuecheng Nie [1,2]
niexuecheng@gmail.com

Jiashi Feng [1]
elefjia@nus.edu.sg

Ashraf A. Kassim [1]
ashraf@nus.edu.sg

Shuicheng Yan [2,1]
eleyans@nus.edu.sg

[1] Department of Electrical and Computer Engineering, National University of Singapore
[2] Artificial Intelligence Institute, 360

ABSTRACT

In this technical demonstration, we propose a face swapping framework, which is able to interactively change the appearance of a face in the wild to a different person/creature's face in real time on a mobile device. To realize this objective, we develop a deep learning-based face detector which is able to accurately detect faces in the wild. Our face feature points tracking system based on progressive initialization ensures accurate and robust localization of facial landmarks under extreme poses and expressions in real time. Relying on the advances of our face detector and face feature points tracker, we construct the Face Swapper which can smoothly replace the face appearance of a user in real time.

Categories and Subject Descriptors

J.5 [**Arts and Humanities**]: Arts, fine and performing

Keywords

Face replacement; Facial animation

1. INTRODUCTION

Facial animation is one of the key components in the entertainment industries. To transform an actor/actress's facial appearance to that of a different person/creature, in most computer aided imagery movies, e.g. Planet of the Apes and Avatar, an actor/actress is required to wear a facial expression capturing device which usually consists of markers on the face and a wearable camera. The markers aim to provide accurate tracking of facial landmarks. With recent progress of face detection and facial points tracking, we believe that these devices are indeed unnecessary. In this demonstration, we show that without these expensive devices, face replacement can be achieved practically on a mobile device.

*This work was performed when Mr. Shengtao Xiao was an intern at Artificial Intelligence Institute, 360

MM '16 October 15-19, 2016, Amsterdam, Netherlands

ACM ISBN 978-1-4503-3603-1/16/10.

DOI: http://dx.doi.org/10.1145/2964284.2973808

Our demonstration consists of three key tasks: face detection, facial landmarks tracking and face swapping. For a given image from a video stream, our application first detects the locations of possible face images. Based on the detected face bounding box, we first localize the 95 predefined face feature points. Our feature points tracker then smoothly track all feature points in a few milliseconds. Face swapper is then applied to the input face with a selected target face mask. All tasks can be performed in real time on a recent mainstream mobile device.

2. SYSTEM

2.1 System Overview

Fig. 1 shows the flowchart of the entire system. The face detection module is triggered periodically to save computational resources and ensure detection of new faces. Detection process is also turned on immediately when the landmark tracking fails. When a new face is detected, landmarks will be first localized. For coming frames, landmark tracking process will utilize the location information of landmarks detected/tracked in the previous frame. This tracking performance is automatically evaluated to ensure the accuracy of landmark locations. The detected/tracked landmarks are passed to the face swapper, by which the correspondence between the source face and the target face can be established. Image manipulation is then performed between the input face and the target face.

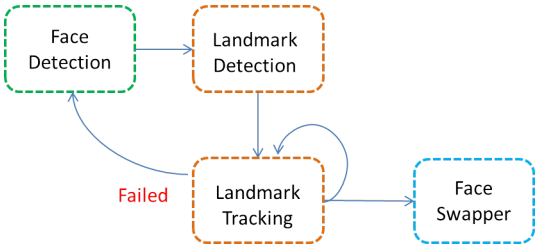

Figure 1: System flowchart.

2.2 Face Detection

We train our face detection model on the AFLW and Wider Face datasets with a framework similar to Faster R-CNN [2]. However, images from both data sets are from the

Internet, mostly with good lighting conditions. The training set is expanded by adding extra face images collected by ourselves based on application requirements. We train our detection model on the Caffe [1] platform. A deeply optimized version of key convolutional neural networks components, e.g. convolutional layer, pooling layer, fully connected layer, is realized for testing on the mobile devices. Our face detection model achieves 80% recall rate with 50 false positives on the FDDB dataset. It runs at around 20 FPS on an iPhone 6s Plus.

2.3 Facial Landmark Tracker

Accurate landmark locations and smooth tracking of the detected landmarks are essential for a good face swapper application. We follow the general framework proposed by [3] for facial landmark detection and tracking. The shape regressors are trained on a much larger training set. Each face image is annotated with 95 key points including face contour, eyebrows, eyes, nose and mouth lips. Xiao et al [3] used the variance of multiple regressed shapes as a measure of accuracy for the landmarks localized in the tracking process. We find that it is not very robust as it may easily reject face images with large poses. It also requires manually tuning the variance threshold which is not user friendly. We revised this process by introducing an adaption mechanism. This makes the landmark tracking process more robust to face images with large poses. Our optimized and revised version of [3] runs at the speed of 125 FPS on an iPhone 6s Plus, which shows dramatic enhancement on robustness, accuracy and speed. Fig. 2 shows that our framework can accurately and robustly track the facial points even under challenging conditions. Please refer to [3] for details of facial landmarks tracking.

Figure 2: Facial landmark tracking under challenging simulations, e.g. large poses, backlit, motion and half face. The green box indicates the face bounding box tracked with our framework and white dots are the localized landmarks.

2.4 Face Swapper

Based on the facial landmarks, a triangular mesh can be easily constructed with the Delaunay triangulation algorithm. In our framework, additional five points are used to create a face bounding box and a forehead reference point for morphing. Fig. 3 shows how the five additional points (in red) and the triangular mesh are generated. Now we can directly apply the triangular mesh based face morphing method. However, directly transforming the target face to a source face may distort the shape of the target face especially when the two faces have different face sizes. To maintain the target face's appearance, i.e. dimension, an additional step of warping the source face to a similar size of the target face is performed before we transform the source face to the target face. Face Swapper runs at a speed of 160 FPS on the iPhone 6s Plus.

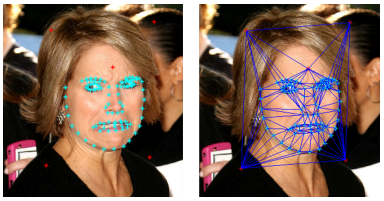

Figure 3: Key points (red and cyan dots) and triangular mesh (blue lines) used for face morphing.

3. INTERFACE

Fig. 4 shows the user interface and some selected results of the swapped face image. The user is able to select a face mask to change his/her facial appearance. This application can also show the facial landmarks tracked. Fig. 4 demonstrates that our application can seamlessly swap the face of the input image to a target face even under large poses and expressions. More results can be found in the supplementary material.

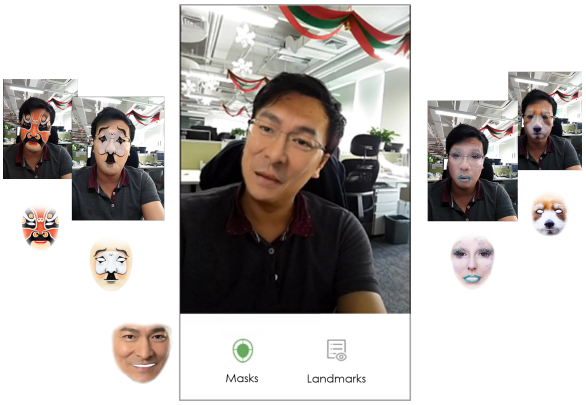

Figure 4: Interface of demonstration. A few masks that have been used in this demonstration. The visual effects after face swapping for corresponding face masks are shown here.

4. CONCLUSION

In this demonstration, we showcase an interactive face swapper framework which runs on a mobile device in real time. Based on our robust, accurate and computationally efficient face detection and landmark tracking modules, our face swapper system shows stable and nearly real facial appearance. In the future, we will introduce more features to the face swapper, e.g. hair style.

5. REFERENCES

[1] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[2] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[3] S. Xiao, S. Yan, and A. Kassim. Facial landmark detection via progressive initialization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 33–40, 2015.